# The UMAP Journal

Vol. 28, No. 3

# Subscription Rates for 2007 Calendar Year: Volume 28

## Membership Plus

Individuals subscribe to *The UMAP Journal* through COMAP's Membership Plus. This subscription also includes a CD-ROM of our annual collection *UMAP Modules: Tools for Teaching*, our organizational newsletter *Consortium*, on-line membership that allows members to download and reproduce COMAP materials, and a 10% discount on all COMAP purchases.

| | | |
|---|---|---|
| **(Domestic)** | **#2720** | **$104** |
| **(Outside U.S.)** | **#2721** | **$117** |

## Institutional Plus Membership

Institutions can subscribe to the *Journal* through either Institutional Plus Membership, Regular Institutional Membership, or a Library Subscription. Institutional Plus Members receive two print copies of each of the quarterly issues of *The UMAP Journal*, our annual collection *UMAP Modules: Tools for Teaching*, our organizational newsletter *Consortium*, on-line membership that allows members to download and reproduce COMAP materials, and a 10% discount on all COMAP purchases.

| | | |
|---|---|---|
| **(Domestic)** | **#2770** | **$479** |
| **(Outside U.S.)** | **#2771** | **$503** |

## Institutional Membership

Regular Institutional members receive print copies of *The UMAP Journal*, our annual collection *UMAP Modules: Tools for Teaching*, our organizational newsletter *Consortium*, and a 10% discount on all COMAP purchases.

| | | |
|---|---|---|
| **(Domestic)** | **#2740** | **$208** |
| **(Outside U.S.)** | **#2741** | **$231** |

## Web Membership

Web membership does not provide print materials. Web members can download and reproduce COMAP materials, and receive a 10% discount on all COMAP purchases.

| | | |
|---|---|---|
| **(Domestic)** | **#2710** | **$41** |
| **(Outside U.S.)** | **#2710** | **$41** |

To order, send a check or money order to COMAP, or call toll-free
**1-800-77-COMAP** (1-800-772-6627).

*The UMAP Journal* is published quarterly by the Consortium for Mathematics and Its Applications (COMAP), Inc., Suite 3B, 175 Middlesex Tpke., Bedford, MA, 01730, in cooperation with the American Mathematical Association of Two-Year Colleges (AMATYC), the Mathematical Association of America (MAA), the National Council of Teachers of Mathematics (NCTM), the American Statistical Association (ASA), the Society for Industrial and Applied Mathematics (SIAM), and The Institute for Operations Research and the Management Sciences (INFORMS). The Journal acquaints readers with a wide variety of professional applications of the mathematical sciences and provides a forum for the discussion of new directions in mathematical education (ISSN 0197-3622).

Periodical rate postage paid at Boston, MA and at additional mailing offices.

**Send address changes to: info@comap.com**

COMAP, Inc., Suite 3B, 175 Middlesex Tpke., Bedford, MA, 01730

Vol. 28, No. 3 2007

# Table of Contents

## Publisher's Editorial



## Special Section on the MCM

# Publisher's Editorial

## Math Is More: Toward a National Consensus on Improving Mathematics Education

Solomon A. Garfunkel
Executive Director
COMAP, Inc.
175 Middlesex Turnpike, Suite 3B
Bedford , MA 01730–1459
s.garfunkel@mail.comap.com

Whether you are a parent or a politician, whether you work in business, industry, government or academia, the state of U.S. mathematics education is of fundamental importance to you and to those whom you care about. As the importance of mathematical and quantitative thinking increases, we must become more focused as a nation on providing our children a better mathematical education. This is not simply about economic competitiveness or getting higher scores on international comparisons. Rather, it is about equipping our children with the necessary tools to be effective citizens and skilled members of the workforce in the 21st century. Mathematics as a discipline and the applications of mathematics to the world around us have grown and changed significantly in the past 50 years. Our system of mathematics education must reflect that growth and change. Quite simply, *math is more*.

We want to do the best job possible with the most children possible. We are a group of mathematics educators, mathematicians, and concerned individuals committed to real and significant improvement in the performance of the complex system of mathematics education. To achieve this goal, however, we must be clear about what we mean. In this document, we specify ten planks that represent our beliefs and guide the direction of our efforts. It will take years of hard work by many people—teachers, administrators, policy makers, parents and students, mathematicians and mathematics educators, academics and practitioners across a wide spectrum—to achieve the goal of universal mathematical literacy and proficiency. The signers of this report (see list at end) commit ourselves to that effort.

---

*The UMAP Journal* 28 (3) (2007) 185–190. 

**Plank 1: Students need to see mathematics and the people who use mathematics in the broadest possible light.**

What do we mean by mathematical literacy? First, math is more than dividing decimals or solving equations. It is more than algebra or geometry as defined by a particular syllabus or set of textbooks. Math is the use of a graph to model a street network to solve traffic snarls; it is finding the "distance" between two strands of DNA to improve our understanding of the human species. It is about deduction, visualization, statistical and probabilistic reasoning, representation, and modeling. It is what enables our cell phones to work and our MRIs to function. It gives us insight into medicine, biology, economics, business, engineering, and the ways that we reason and make decisions. Mathematics education at all levels and in all courses must engage students with the practicality, the applicability, the power and the beauty of mathematics. This can be accomplished when students see mathematics as including skills, conceptual understandings, and a way of reasoning.

**Plank 2: Mathematics education must be viewed as a complex system requiring coherent coordination and a long-term investment in the quality of curriculum, instruction, and assessment.**

We do not believe that there are quick fixes or magic bullets that will lead to significant improvements in mathematics education. Rather, we believe that improvements in this complex system will be the result of a series of substantive changes that are informed by research and guided by experimentation with the proper and rigorous evaluation of the results. But change of this magnitude takes time. Among other things, both established and new teachers need to learn and experience mathematics as the rich discipline that we know it to be. Professional working conditions for teachers must allow time and opportunity for developing new understandings about mathematics, its applications, and the teaching of mathematics.

**Plank 3: Mathematics education at all levels, including advanced college programs, is a form of vocational and professional preparation.**

We must recognize that there is a compelling national (and local) interest in the state of mathematics education. While we do not see this as a zero-sum game, with our country (or state) vying to do better than another, our overall mathematical literacy and competence is important to our economic health. Industry, in addition to government, needs to be heavily involved. Employers are, after all, parents, and vice versa. Surely, having good high school math grades or SAT scores must be about more than getting into a good college. Being able to analyze and solve problems using quantitative reasoning is an increasingly necessary job skill. We believe that not enough emphasis has been placed on the needs of students. Their future will involve many different jobs. They will need to master current and emerging technologies. We know that they will need creativity, independence, imagination and problem-solving abilities in addition to skills proficiency. In other words, students will increasingly need advanced mathematical understanding and awareness of the tools mathematics provides to achieve their career goals.

**Plank 4: A coherent set of broad national curricular goals allowing for new results from educational research should be created.**

While we believe in accountability and we recognize the need for curricular coherence, we worry about the Babel of "Standards" being designed by individual states, districts, and more nationally-based organizations and think tanks. National standards in the spirit of curricular goals can serve a unifying purpose. Standards must, however, be generic enough to allow for the evolution of content and pedagogy. Although there must be room for trying new ideas, standards should increasingly be grounded in robust research demonstrating student learning of important mathematical ideas. Standards at the grain size of individual skills must be avoided. We also believe that the present multiplicity and specificity of standards is a barrier to innovation by both the authors and publishers of mathematics materials.

**Plank 5: The quality of instruction continues to be of critical importance to the improvement of student achievement.**

The mathematics classroom is more than where students encounter formal mathematics. It is where students decide if mathematics is "for them" and where the ideas must inspire and engage. Active learning produces life-long learning. There is no substitute for curiosity, engagement, pursuit of ideas, and use of prior knowledge, followed by exploration, experimentation, practice, and mastery. The use of applications, the design of rich interactions among students, and the creative use of technologies have produced promising results when accompanied by careful attention to students' progress through well-understood learning progressions. Accountability is hollow if it is not accompanied by robust efforts to improve instruction, by using exciting materials, and by including opportunities for teachers to be learners and to experience broader views of mathematics. Our task is to introduce students to the wonders of mathematics while providing the discipline to regulate their own learning and to ensure proficiency and mastery. Students should not be viewed simply as consumers of mathematics education but as active participants with the most to gain or lose. Their voices should be solicited and taken into serious consideration.

**Plank 6: Programs must be developed to help all students, recognizing their diverse needs, interests, talents, and levels of motivation.**

"Mathematics for All" is an important rallying cry. But to be meaningful, it requires that we recognize and act on the fact that different student populations need to be provided for differently. For a multitude of reasons, some students may be more motivated to learn than others. Some students have stronger background knowledge than others, and some learn more quickly. One size does not fit all. There is research that can be brought to bear on these issues—and we need to know and do more. We cannot afford a mathematics education system that works for the few and not the many.

**Plank 7: We must test what we value, both locally and nationally. Mathematical literacy is becoming a survival skill.**

We strongly believe in accountability to a rich set of mathematical goals. We want students to master core facts and procedures, but this is not enough. We want conceptual understanding, problem-solving, and flexible use of mathematics to solve both pure and applied problems. Like standards, assessments must reflect our goals—most importantly, the ability to apply mathematical reasoning to analyze and attack real-world problems. If mathematical literacy includes the ability to make use of mathematics, and we believe in the importance of mathematical literacy, then we must align our testing accordingly. Testing must not be about punishment for failure but about giving students and teachers a clearer understanding of what they do and do not know. Testing should inform instruction, not determine it.

**Plank 8: We must continue to develop and research new materials and pedagogies and translate that research into improved classroom practice.**

Education, as a scientific discipline, is a young field with an active community focused on R&D—*R*esearch on learning coupled with the *D*evelopment of new and better curriculum materials. In truth, however, much of the work is better described as D&R—informed and thoughtful *D*evelopment followed by careful analysis of *R*esults. It is in the nature of the enterprise that we cannot discover what works before we create the what. Curriculum development, in particular, is best conducted in analogy to an engineering paradigm. To test the efficacy of an approach, we must analyze needs, examine existing programs, build an improved model program, and test it—in the same way that we build scale models to design a better bridge or building. This kind of iterative D&R leads to new and more effective materials and new pedagogical approaches that better incorporate the growing body of knowledge of cognitive science. We understand that educational research has not yet provided all of the answers to how best to help children learn mathematics. However, there is a great deal that we do know about the motivational power of applications, the effectiveness of appropriate learning technologies, the use of collaborative learning with children, and the use of lesson- and case-study programs with teachers.

**Plank 9: Our country must make a major investment over the coming decade to sustain and rejuvenate the ranks of mathematics teachers in our nation's schools.**

Many mathematics classrooms are staffed with unqualified teachers. This is because school administrators can neither find enough qualified teachers nor provide adequate resources to upgrade staff qualifications. Mandates that every teacher be qualified won't improve the situation until there is a sufficient supply of mathematics teachers to meet the demand. To stave off this foreseeable crisis in our math classrooms, our nation needs to act to increase the numbers of young people entering mathematics and mathematics education disciplines in our universities and to improve significantly the continuing education of existing teachers. We must ensure that their education prepares them for current educational realities and that their working conditions as teachers permit them continuous mathematical and pedagogical improvements. We need to find more ways to support new teachers through the difficult induction years, especially young people who commit to teach in our least successful schools.

**Plank 10: We must build a sustainable system for monitoring and improving mathematics education.**

Perhaps the most important point is that our work must be sustainable. Just as with our students, we need to be there throughout the learning process—watching out for necessary course corrections and building with a long-range view. Too often in the past we have reacted to crises, whether it be Sputnik and fear of losing the space race, being overtaken economically by Japan, or out-sourcing our manufacturing jobs to China and India. Reports are written decrying the current state of affairs and funding is made available. But the need for excellent mathematics education will always be with us. We must build an infrastructure that recognizes this fact, and devotes consistent attention and resources to addressing the challenge of high quality mathematics for all, rather than a cycle of investment, neglect, investment, . . . .

---

The authors of this document share many beliefs—that mathematics is important as a discipline, as a field full of wonder and beauty, as a tool for modeling our world, as a prerequisite for knowledgeable citizenship in a participatory democracy, and as a means to better jobs and a better quality of life. We hold strong views on the importance of education in general and mathematics education in particular. We do not agree on all things, but we are, and intend to remain, inclusive. Clearly, there is much substance and detail that can be added to these planks. We need many voices and many hands and we call on you to join with us to ensure that every child receives the best mathematics education possible and recognizes that in their future, math is more. If you support these ideas and would like to work with us to make these planks a reality and/or receive regular updates on Math is More activities, please visit

`http://www.mathismore.net/forms/form.php`

**Jere Confrey** **North Carolina State University, Raleigh**
**Gary Froelich** **COMAP, Inc.**
**Sol Garfunkel** **COMAP,Inc.**
**Midge Cozzens** **Knowles Science Teaching Foundation**
**John Ewing** **American Mathematical Society**

# About This Issue

Paul J. Campbell
Editor

This issue of *The UMAP Journal* continues a practice inaugurated in Vol. 26. It runs longer than the usual size of 92 pp—in fact, almost 300 pp. However, not all of the articles in this issue are printed in the paper copy. Some articles appear only in the *Tools for Teaching* 2007 CD-ROM (and at `http://www.comap.com` for COMAP members), which will reach members and subscribers at a later time and will also contain the entire 2007 year of *Journal* issues.

However, all articles of this issue on the CD-ROM appear in the printed table of contents and are regarded as published in the *Journal*. In addition, the abstract of each Outstanding paper appears in the printed version. Pagination of the issue runs continuously, including in sequence articles that do not appear in printed form. *So, if you notice that, say, page 350 in the printed copy is followed by page 403, your copy is not necessarily defective!* The articles corresponding to the intervening pages will be on the CD-ROM.

We hope that you find this arrangement, if not entirely satisfying, at least satisfactory. It means that we do not have to procrusteanize the content of the *Journal* to fit a fixed number of allocated pages. For example, we might otherwise need to select only two or three Outstanding MCM papers to publish (a hard task indeed!). Instead, as in the past, we continue to bring you the full content.

# Modeling Forum

## Results of the 2007 Mathematical Contest in Modeling

Frank Giordano, MCM Director
Naval Postgraduate School
1 University Circle
Monterey, CA 93943-5000
`frgiorda@nps.navy.mil`

## Introduction

A total of 949 teams of undergraduates, from 313 institutions and 536 departments in 12 countries, spent the first weekend in February working on applied mathematics problems in the 23rd Mathematical Contest in Modeling (MCM).

The 2007 MCM began at 8:00 P.M. EST on Thursday, February 8 and ended at 8:00 P.M. EST on Monday, February 12. During that time, teams of up to three undergraduates were to research and submit an optimal solution for one of two open-ended modeling problems. Students registered, obtained contest materials, downloaded the problems at the appropriate time, and entered completion data through COMAP's MCM Website. After a weekend of hard work, solution papers were sent to COMAP on Monday. The top papers appear in this issue of *The UMAP Journal*.

Results and winning papers from the first 22 contests were published in special issues of *Mathematical Modeling* (1985–1987) and *The UMAP Journal* (1985–2006). The 1994 volume of *Tools for Teaching*, commemorating the tenth anniversary of the contest, contains all of the 20 problems used in the first 10 years of the contest and a winning paper for each year. Limited quantities of that volume and of the special MCM issues of the *Journal* for the last few years are available from COMAP. That 1994 volume is also available on COMAP's special Modeling Resource CD-ROM (`http://www.comap.com/product/?idx=613`). In addition, also available from COMAP is another CD, *The MCM at 21*, which contains

*The UMAP Journal* 28 (3) (2007) 191–230. 

all of the 20 problems from the second 10 years of the contest, a winning paper from each year, and advice from advisors of Outstanding teams.

This year's Problem A asked teams to draw congressional districts for a state so that the districts would have the "simplest" shapes; as an application, teams had to apply their method to New York State. Problem B asked teams to devise and compare procedures for boarding and deboarding airplanes of varying sizes. The 14 Outstanding solution papers are published in this issue of *The UMAP Journal*, along with commentary from problem authors, contest judges, and outside experts.

In addition to the MCM, COMAP also sponsors the Interdisciplinary Contest in Modeling (ICM) and the High School Mathematical Contest in Modeling (HiMCM). The ICM, which runs concurrently with MCM, offers a modeling problem involving concepts in operations research, information science, and interdisciplinary issues in security and safety. Results of this year's ICM are on the COMAP Website at `http://www.comap.com/undergraduate/contests`; results and Outstanding papers appeared in Vol. 28 (2007), No. 2. The HiMCM offers high school students a modeling opportunity similar to the MCM. Further details about the HiMCM are at `http://www.comap.com/highschool/contests`.

# Problem A: Gerrymandering

The United States Constitution provides that the House of Representatives shall be composed of some number (currently 435) of individuals who are elected from each state in proportion to the state's population relative to that of the country as a whole. While this provides a way of determining how many representatives each state will have, it says nothing about how the district represented by a particular representative shall be determined geographically. This oversight has led to egregious (at least some people think so, usually not the incumbent) district shapes that look "unnatural" by some standards.

Hence the following question: Suppose that you were given the opportunity to draw congressional districts for a state. How would you do so as a purely "baseline" exercise to create the "simplest" shapes for all the districts in a state? The rules include only that each district in the state must contain the same population. The definition of "simple" is up to you; but you need to make a convincing argument to voters in the state that your solution is fair. As an application of your method, draw geographically simple congressional districts for the state of New York.

# Problem B: The Airplane Seating Problem

Airlines are free to seat passengers waiting to board an aircraft in any order whatsoever. It has become customary to seat passengers with special needs first, followed by first-class passengers (who sit at the front of the plane). Then coach and business-class passengers are seated by groups of rows, beginning with the row at the back of the plane and proceeding forward.

Apart from consideration of the passengers' wait time, from the airline's point of view, time is money, and boarding time is best minimized. The plane makes money for the airline only when it is in motion, and long boarding times limit the number of trips that a plane can make in a day.

The development of larger planes, such as the Airbus A380 (800 passengers), accentuate the problem of minimizing boarding (and deboarding) time.

Devise and compare procedures for boarding and deboarding planes with varying numbers of passengers: small (85–210), midsize (210–330), and large (450–800). Prepare an executive summary, not to exceed two single-spaced pages, in which you set out your conclusions to an audience of airline executives, gate agents, and flight crews.

An article appeared in the *New York Times* (14 November 2006) addressing procedures currently being followed and the importance to the airline of finding better solutions. The article can be seen at: `http://travel2.nytimes.com/2006/11/14/business/14boarding.html`.

# The Results

The solution papers were coded at COMAP headquarters so that names and affiliations of the authors would be unknown to the judges. Each paper was then read preliminarily by two "triage" judges at either Appalachian State University (Gerrymandering Problem) or at the National Security Agency (Airplane Seating Problem). At the triage stage, the summary and overall organization are the basis for judging a paper. If the judges' scores diverged for a paper, the judges conferred; if they still did not agree on a score, a third judge evaluated the paper.

This year again, an additional Regional Judging site was created at the U.S. Military Academy to support the growing number of contest submissions.

Final judging took place at Asilomar Conference Center, Pacific Grove, CA. The judges classified the papers as follows:

| | Outstanding | Meritorious | Honorable Mention | Successful Participation | Total |
|---|---|---|---|---|---|
| Gerrymandering Problem | 5 | 53 | 91 | 202 | 351 |
| Airplane Seating Problem | 9 | 69 | 164 | 356 | 598 |
| | 14 | 122 | 255 | 558 | 949 |

The 14 papers that the judges designated as Outstanding appear in this special issue of *The UMAP Journal*, together with commentaries. We list those teams and the Meritorious teams (and advisors) below; the list of all participating schools, advisors, and results is in the **Appendix**.

## Outstanding Teams

| **Institution and Advisor** | **Team Members** |
|---|---|
| | |
| **Gerrymandering Papers** | |
| "When Topologists Are Politicians..." | |
| Duke University | Nikifor C. Bliznashki |
| Durham, NC | Aaron Pollack |
| David Kraines | Russell Posner |
| | |
| "What to Feed a Gerrymander" | |
| Harvard University | Ben Conlee |
| Cambridge, MA | Abe Othman |
| Clifford H. Taubes | Chris Yetter |
| | |
| "Electoral Redistricting with Moment of Inertia and Diminishing Halves Models" | |
| Massachusetts Institute of Technology | Andrew Spann |
| Cambridge, MA | Daniel Kane |
| Martin Z. Bazant | Dan Gulotta |
| | |
| "Applying Voronoi Diagrams to the Redistricting Problem" | |
| University of Washington | Lukas Svec |
| Seattle, WA | Sam Burden |
| James Allen Morrow | Aaron Dilley |
| | |
| "Why Weight? A Cluster-Theoretic Approach to Political Districting" | |
| University of Washington | Sam Whittle |
| Seattle, WA | Wesley Essig |
| Anne Greenbaum | Nathaniel S. Bottman |

**Airplane Seating Papers**

"Novel Approaches to Airline Boarding"
Duke University
Durham, NC
Anne Catlla

Qianwei Li
Arnav Mehta
Aaron Wise

"Boarding at the Speed of Flight"
Duke University
Durham, NC
Michael Brenner

Michael Bauer
Kshipra Bhawalkar
Matthew Edwards

"STAR: (Saving Time, Adding Revenues) Boarding/Deboarding Strategy"
National Univ. of Defense Technology
Changsha, China
Yi Wu

Bo Yuan
Jianfei Yin
Mafa Wang

"The Unique Best Boarding Plan? It Depends"
National University of Singapore
Singapore
Yannis Yatracos

Bolun Liu
Xuan Hou
Hao Wang

"Airliner Boarding and Deplaning Strategy"
Peking University
Beijing, China
Xufeng Liu

Linbo Zhao
Fan Zhou
Guozhen Wang

"The Best Boarding Uses Buffers"
Slippery Rock University
Slippery Rock, PA
Athula R. Herat

Kevin D. Sobczak
Eric J. Hardin
Bradley J. Kirkwood

"Modeling Airplane Boarding Procedures"
Truman State University
Kirksville, MO
Steven J. Smith

Bach Ha
Daniel Matheny
Spencer Tipping

"American Airlines' Next Top Model"
University of Puget Sound
Tacoma, WA
Michael Z. Spivey

Sara J. Beck
Spencer D. K'Burg
Alex B. Twist

"Boarding—Step by Step:
A Cellular Automaton Approach to
Optimising Aircraft Boarding Time"
University of Stellenbosch
Stellenbosch, South Africa
Jan H. van Vuuren

Chris Rohwer
Andreas Hafver
Louise Viljoen

## Meritorious Teams

**Gerrymandering Problem** (53 teams)
Beijing Forestry University, College of Biological Sciences and Biotechnology, Beijing, China (Gao Mengning)
Beijing Wuzi University, Informatics College, Beijing, China (Li Zhenping)
Bemidji State University, MN (Colleen Livingston)
Carroll College, Helena, MT (Holly S. Zullo)
Carroll College, Helena, MT (Kelly S. Cline)
Cornell University, Ithaca, NY (Alexander Vladimirsky)
Duke University, Dept. of Computer Science, Durham, NC (Owen Astrachan)
Harvard University, Cambridge, MA (Clifford H. Taubes)
Harvey Mudd College, Dept. of Mathematics, Claremont, CA (Jon Jacobsen)
Harvey Mudd College, Dept. of Computer Science, Claremont, CA (Ran Libeskind-Hadas)
Huazhong University of Science and Technology, Dept. of Industrial and Manufacturing System Engineering, Wuhan, Hubei, China (Gao Liang)
James Madison University, Harrisonburg, VA (David B. Walton)
Kansas State University, Manhattan, KS (Dave R. Auckly)
Korea Advanced Institute of Science & Technology, Daejeon, Korea (Kim Yong Jung)
MIT, Dept. of Mathematics, Cambridge, MA (Martin Z. Bazant)
MIT, Dept. of Physics, Cambridge, MA (Leonid Levitov—two teams)
National University of Defense Technology, Changsha, China (Wang Dan)
Ningbo Institute of Technology, Zhejiang University Fundamental Courses, Ningbo, Zhejiang, China (Wang Jufeng)
Northeastern University, School of Mechanical Engineering and Automation, Dept. of Modern Design and Analysis, Shenyang, Liaoning, China (He Xuehong)
Northwestern Polytechnical University Dept. of Applied Mathematics, Xi'an, Shaanxi, China (Nie Yufeng)
Northwestern Polytechnical University Dept. of Applied Physics, Xi'an, Shaanxi, China (Lei Youming)
Oregon State University, Corvallis, OR (Nathan L. Gibson)
Pacific University, Forest Grove, CA (John August)
Peking University School of Mathematical Sciences, Dept. of Business Statistics and Econometrics, Beijing, China (Zhang Junni)

Rose-Hulman Institute of Technology, Dept. of Mathematics, Terre Haute, IN (David J. Rader)
Rose-Hulman Institute of Technology, Dept. of Computer Science and Software Engineering, Terre Haute, IN (Cary Laxer)
Shandong University of Science and Technology, College of Information Science and Engineering, Qingdao, Shandong, China (Pang Shanchen)
Shanghai Hongkou Institute of Education, Shanghai, China (Hu Jun)
Shenyang Institute of Aeronautical Engineering, Shenyang, Liaoning, China (Zhu Limei)
Sichuan University Yangtze Center of Mathematics, Chengdu, Sichuan, China (Chen Bohui)
South China Normal University, Dept. of Information and Computational Science, Guangzhou, Guangdong, China (Yang Tan)
South China University of Technology, School of Computer Science and Engineering, Guangzhou, Guangdong, China (Tao Zhi-Sui)
Southern Connecticut State University, New Haven, CT (Ross B. Gingrich)
Tsinghua University Mathematical Science, Beijing, China (Ye Jun)
U.S. Military Academy, West Point, NY (Elisha Peterson)
University College Cork, Cork, Ireland (Dmitrii Rachinskii)
University of California–Davis, Davis, CA (Sarah A. Williams)
University of Colorado–Boulder, Dept. of Physics, Boulder, CO (Michael H. Ritzwoller)
University of Richmond, Richmond, VA (Kathy W. Hoke)
University of Science and Technology of China, Dept. of Geophysics, Hefei, Anhui, China (Zhan Zhongwen)
University of Science and Technology of China, Special Class for Gifted Young, Hefei, Anhui, China (Zhang Gaigong)
Wake Forest University, Winston Salem, NC (Miaohua Jiang)
Wesleyan College, Macon, GA (Joseph A. Iskra)
Western Washington University, Bellingham, WA (Tjalling Ypma)
Westminster College, New Wilmington, PA (Barbara T. Faires)
Wuhan University of Technology, Wuhan, Hubei, China (He Lang)
Xidian University, Xi'an, Shaanxi, China (Song Yue)
Xuzhou Institute of Technology, Xuzhou, Jiangsu, China (Jiang Yingzi)
Zhejiang University, Hangzhou, Zhejiang, China (Tan Zhiyi)
Zhejiang University City College, Dept. of Information and Computing Science, Hangzhou, Zhejiang, China (Wang Gui)
Zhejiang University of Technology, Jianxing College, Hangzhou, Zhejiang, China (Wang Shiming)
Zhuhai College of Jinan University, Zhuhai, Guangdong, China (Zhang Yuanbiao)

**Airplane Seating Problem** (69 teams)
Asbury College, Wilmore, KY (David L. Coulliette)
Beijing Institute of Technology, Beijing, China (Li Bing-Zhao)
Beijing Jiaotong University, Beijng, China (Fan Jixiang)
Beijing Jiaotong University, Dept. of Informaion Management, Beijing, China (Wang Bingtuan)
Beijing Normal University, Dept. of Physics, Beijing, China (Huang Haiyang)
Beijing Normal University, Dept. of Statistics, Beijing, China (Zhang Shumei)
Beijing University of Aeronautics and Astronautics, Dept. of Electronic Information Engineering, Beijing, China (Feng Wei)

Bemidji State University, Bemidji, MN (Colleen Livingston)
Carroll College, Helena, MT (Sam Alvey)
Central South University, Changsha, Hunan, China (Hou Muzhou)
Civil Aviation University of China, Dept. of Computer Science and Technology, Tianjin, China (Zhang Yuxiang)
Civil Aviation University of China, Science College, Tianjin, China (Mou Deyi)
Colgate University, Hamilton, NY (Warren Weckesser)
Cornell University, Ithaca, NY (Alexander Vladimirsky)
Dalian University of Technology, Institute of University Students' Innovation, Dalian, Liaoning, China (Tao Sun)
Hangzhou Dianzi University, Hangzhou, Zhejiang, China (Shen Hao)
Harbin Institute of Technology, Dept. of Management Science & Engineering, Harbin, Heilongjiang, China (Ge Hong)
Harvey Mudd College, Dept. of Computer Science, Claremont, CA (Ran Libeskind-Hadas)
Hefei University of Technology, Hefei, Anhui (Du Xueqiao)
Institute of Electronic Technology, Dept. of Information Engineering University Management, Zhengzhou, Henan, China (Jia Li Xin)
Korea Advanced Institute of Science and Technology, Dept. of Mechanical Engineering, Daejon, Korea (Joongmyeon Bae)
Lawrence Technological University, Southfield, MI (Ruth G. Favro)
Lawrence Technological University, Dept. of Physics, Southfield, MI (Valentina Tobos)
Loyola College in Maryland, Baltimore, MD (Jiyuan Tao)
McGill University, Montréal, Québec, Canada (Nilima Nigam)
Nanjing University, Dept. of Electronic Science and Engineering, Nanjing, Jiangsu, China (Wu Haodong)
Nanjing University, Dept of Life Science, Nanjing, Jiangsu, China (Wang Jin)
Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China (Kong Gaohua)
National University of Defense Technology, Changsha, Hunan, China (Wu Mengda)
National University of Ireland, Galway, Ireland (Niall Madden)
Northeastern University, Dept. of Information Science and Engineering, Shenyang, Liaoning, China (Zhao Shuying)
Northwestern Polytechnical University, Dept. of Applied Chemistry, Xi'an, Shaanxi, China (Sun Zhongkui)
Radford University, Radford, VA (Laura J. Spielman)
Renmin University of China, Beijing, China (Zhou Zemin)
Rice University, Dept. of Computational and Applied Mathematics, Houston, TX (Mark Patrick Embree)
Rice University, Dept. of Electrical and Computer Engineering, Houston, TX (Michael T. Orchard—two teams)
Rowan University, Glassboro, NJ (Hieu D. Nguyen)
Shandong University, Jinan, Shandong, China (Lu Tongchao)
Shandong University, Jinan, Shandong, China (Liu Baodong—two teams)
Shanghai University of Finance and Economics, Shanghai, China (Zhang Zhen Yu)
Shanghai University of Finance and Economics, Shanghai, China (Li Ming)
Shenyang University of Technology, Shenyang, Liaoning, China (Chen Yan)
Sichuan University, Chengdu, Sichuan, China (Hai Niu)
Sichuan University, Chengdu, Sichuan, China (Zhou Jie)

Slippery Rock University, Slippery Rock, PA (Richard J. Marchand)
South China University of Technology, Guangzhou, Guangdong, China (Pan Shaohua)
Southwest University, Chongqing, China (Deng Lei)
Southwestern University of Finance and Economics, Dept. of Economic Mathematics, Chengdu, Sichuan, China (Li Shaowen)
Sun Yat-sen University, Guangzhou, Guangdong, China (Li Yan Hui)
Trinity University, San Antonio, TX (Diane Saphire)
Tsinghua University, Beijing, China (Hu Zhiming)
University of Colorado–Boulder, Dept. of Physics, Boulder, CO (Michael H. Ritzwoller)
University of Colorado at Denver, Denver, CO (Gary A. Olson)
University of Colorado at Denver, Denver, CO (Lance W. Lana)
University of Delaware, Newark, DE (Louis F. Rossi—two teams)
University of Electrical Science and Technology of China, Dept. of Information and Computation Science, Chengdu, Sichuan, China (Qin Siyi)
University of Massachusetts Lowell, Lowell, MA (James Graham-Eagle)
University of Stellenbosch, Stellenbosch, Western Cape, South Africa (Jan H. van Vuuren)
University of Western Ontario, London, Ontario, Canada (Allan B. MacIsaac)
U.S. Military Academy, West Point, NY (Robert Burks)
U.S. Military Academy, Dept. of Systems Engineering, West Point, NY (John Willis)
Wake Forest University, Winston Salem, NC (Miaohua Jiang)
Westminster College, Dept. of Physics, New Wilmington, PA (Samuel Lightner)
Wuhan University, Wuhan, Hubei, China (Hu Xinqi)
Xavier University, Cincinnati, OH (Bernd E. Rossa)
Youngstown State University, Youngstown, OH (George T. Yates)

# Awards and Contributions

Each participating MCM advisor and team member received a certificate signed by the Contest Director and the appropriate Head Judge.

INFORMS, the Institute for Operations Research and the Management Sciences, recognized the teams from Harvard University (Gerrymandering Problem) and Duke University (Airplane Seating Problem) as INFORMS Outstanding teams and provided the following recognition:

- a letter of congratulations from the current president of INFORMS to each team member and to the faculty advisor;
- a check in the amount of $300 to each team member;
- a bronze plaque for display at the team's institution, commemorating their achievement;
- individual certificates for team members and faculty advisor as a personal commemoration of this achievement;

- a one-year student membership in INFORMS for each team member, which includes their choice of a professional journal plus the *OR/MS Today* periodical and the INFORMS society newsletter.

The Society for Industrial and Applied Mathematics (SIAM) designated one Outstanding team from each problem as a SIAM Winner. The teams were from MIT (Gerrymandering Problem) and Stellenbosch University (Airplane Seating Problem). Each of the team members was awarded a $300 cash prize and the teams received partial expenses to present their results in a special Minisymposium at the SIAM Annual Meeting in Boston, MA in July. Their schools were given a framed hand-lettered certificate in gold leaf.

The Mathematical Association of America (MAA) designated one Outstanding North American team from each problem as an MAA Winner. The teams were from University of Washington (Gerrymandering Problem) and Truman State University (Airplane Seating Problem). With partial travel support from the MAA, the Rice University team presented their solution at a special session of the MAA Mathfest in Knoxville, TN in August. Each team member was presented a certificate by Richard S. Neal of the MAA Committee on Undergraduate Student Activities and Chapters.

## Ben Fusaro Award

One Meritorious paper was selected for each problem for the Ben Fusaro Award, named for the Founding Director of the MCM and awarded for the second time this year. It recognizes an especially creative approach; details concerning the award, its judging, and Ben Fusaro are in Vol. 25 (3) (2004): 195–196. The Ben Fusaro Award winners were from University of Colorado–Boulder (Gerrymandering Problem) and Rowan University (Airplane Seating Problem).

# Judging

*Director*
Frank R. Giordano, Naval Postgraduate School, Monterey, CA

*Associate Directors*
Robert L. Borrelli, Mathematics Dept., Harvey Mudd College, Claremont, CA
Patrick J. Driscoll, Dept. of Systems Engineering, U.S. Military Academy, West Point, NY
William P. Fox, Dept. of Defense Analysis, Naval Postgraduate School, Monterey, CA

## Gerrymandering Problem

*Head Judge*
Marvin S. Keener, Executive Vice-President, Oklahoma State University, Stillwater, OK

*Associate Judges*
William C. Bauldry, Chair, Dept. of Mathematical Sciences, Appalachian State University, Boone, NC (Head Triage Judge)
Ben Fusaro, Dept. of Mathematics, Florida State University, Tallahassee, FL
Steve Horton, Dept. of Mathematical Sciences, U.S. Military Academy, West Point, NY (MAA Judge)
Mario Juncosa, RAND Corporation, Santa Monica, CA (retired)
Michael Moody, Olin College of Engineering, Needham, MA
David H. Olwell, Naval Postgraduate School, Monterey, CA (INFORMS Judge)
John L. Scharf, Mathematics Dept., Carroll College, Helena, MT (Fusaro Award Judge)
Michael Tortorella, Dept. of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ (Problem Author)

## Airplane Seating Problem

*Head Judge*
Maynard Thompson, Mathematics Dept., University of Indiana, Bloomington, IN

*Associate Judges*
Peter Anspach, National Security Agency, Ft. Meade, MD (Head Triage Judge)
Kelly Black, Mathematics Dept., Union College, Schenectady, NY
Karen D. Bolinger, Mathematics Dept., Clarion University of Pennsylvania, Clarion, PA
Jim Case (SIAM Judge)
J. Douglas Faires, Youngstown State University, Youngstown, OH
Jerry Griggs, Mathematics Dept., University of South Carolina, Columbia, SC
Veena Mendiratta, Lucent Technologies, Naperville, IL
Don Miller, Mathematics Dept., St. Mary's College, Notre Dame, IN
Kathleen M. Shannon, Dept. of Mathematics and Computer Science, Salisbury University, Salisbury, MD (MAA Judge)
Dan Solow, Mathematics Dept., Case Western Reserve University, Cleveland, OH (INFORMS Judge)
Marie Vanisko, Dept. of Mathematics, California State University—Stanislaus, Turlock, CA (from Fall 07 at Carroll College, Helena MT) (Fusaro Award Judge)
Richard Douglas West, Francis Marion University, Florence, SC

**Regional Judging Sessions**

*Head Judge*
Patrick J. Driscoll, Dept. of Systems Engineering, United States Military Academy (USMA), West Point, NY

*Associate Judges*
Merrill Blackman, Dept. of Systems Engineering, USMA
Tim Elkins, Dept. of Systems Engineering, USMA
Darrall Henderson, Dept. of Mathematical Sciences, USMA
Steve Henderson, Dept. of Systems Engineering, USMA
Steve Horton, Dept. of Mathematical Sciences, USMA
Michael Jaye, Dept. of Mathematical Sciences, USMA
Tom Meyer, Dept. of Mathematical Sciences, USMA

**Triage Sessions:**

**Gerrymandering Problem**

*Head Triage Judge*
William C. Bauldry, Chair, Dept. of Mathematical Sciences, Appalachian State University, Boone, NC

*Associate Judges*
Jeff Hirst, Rick Klima, Greg Rhoads, and René Salinas
—all from Dept. of Math'l Sciences, Appalachian State University, Boone, NC

**Airplane Seating Problem**

*Head Triage Judge*
Peter Anspach, National Security Agency (NSA), Ft. Meade, MD

*Associate Judges*
other judges from inside and outside NSA, who wish not to be named.

# Sources of the Problems

The Gerrymandering Problem was contributed by Michael Tortorella (Dept. of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ). The Airplane Seating Problem was contributed by Paul J. Campbell (Mathematics and Computer Science, Beloit College, Beloit, WI).

# Acknowledgments

Major funding for the MCM is provided by the National Security Agency and by COMAP. We thank Dr. Gene Berg of NSA for his coordinating efforts. Additional support is provided by the Institute for Operations Research and the Management Sciences (INFORMS), the Society for Industrial and Applied Mathematics (SIAM), and the Mathematical Association of America (MAA). We are indebted to these organizations for providing judges and prizes.

We also thank for their involvement and support:

- Two Sigma Investments. (This group of experienced, analytical, and technical financial professionals based in New York builds and operates sophisticated quantitative trading strategies for domestic and international markets. The firm is successfully managing several billion dollars using highly automated trading technologies. For more information about Two Sigma, please visit `http://www.twosigma.com`.)

We thank the MCM judges and MCM Board members for their valuable and unflagging efforts. Harvey Mudd College, its Mathematics Dept. staff, and Prof. Borrelli were gracious hosts to the judges.

# Cautions

*To the reader of research journals:*

Usually a published paper has been presented to an audience, shown to colleagues, rewritten, checked by referees, revised, and edited by a journal editor. Each of the student papers here is the result of undergraduates working on a problem over a weekend; allowing substantial revision by the authors could give a false impression of accomplishment. So these papers are essentially au naturel. Editing (and sometimes substantial cutting) has taken place: Minor errors have been corrected, wording has been altered for clarity or economy, and style has been adjusted to that of *The UMAP Journal*. Please peruse these student efforts in that context.

*To the potential MCM Advisor:*

It might be overpowering to encounter such output from a weekend of work by a small team of undergraduates, but these solution papers are highly atypical. A team that prepares and participates will have an enriching learning experience, independent of what any other team does.

---

COMAP's Mathematical Contest in Modeling and Interdisciplinary Contest in Modeling are the only international modeling contests in which students work in teams. Centering its educational philosophy on mathematical modeling, COMAP uses mathematical tools to explore real-world problems. It serves the educational community as well as the world of work by preparing students to become better-informed and better-prepared citizens.

# Appendix: Successful Participants

KEY:
P = Successful Participation
H = Honorable Mention
M = Meritorious
O = Outstanding (published in this special issue)

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| ALASKA | | | | | |
| University of Alaska Fairbanks | Computer Science | Fairbanks | Orion Sky Lawlor | H | |
| CALIFORNIA | | | | | |
| California Poly. State University | Mathematics | San Luis Obispo | Jonathan E. Shapiro | | P |
| Calif. State Polytechnic U. Pomona | Mathematics and Statistics | Pomona | Ioana Mihaila | | P |
| | | | Hubertus F. von Bremen | H | P |
| | Physics | | Nina Abramzon | | P |
| Calif. State Univ. at Monterey Bay | Mathematics and Statistics | Seaside | Hongde Hu | H | P |
| | Science and Env'l Policy | | Herbert Cortez | | H |
| Harvey Mudd College | Computer Science | Claremont | Ran Libeskind-Hadas | M | M |
| | Mathematics | | Jon Jacobsen | M P | |
| Humboldt State University | Env'l Resources Engineering | Arcata | Brad A. Finney | | H |
| San Diego State University | Mathematics and Statistics | San Diego | Kristin Duncan | | P |
| Sonoma State University | Mathematics | Rohnert Park | Sunil K. Tiwari | | H |
| University of California Davis | Mathematics | Davis | Sarah A. Williams | M P | |
| University of California Los Angeles | Mathematics | Los Angeles | Luminita Aura Vese | P | |
| University of San Diego | Mathematics | San Diego | Diane Hoffoss | | H P |
| | Physics | | Daniel Sheehan | P P | |
| COLORADO | | | | | |
| Colorado State University | Mathematics | Fort Collins | Michael J. Kirby | H | |
| Colorado State University – Pueblo | Mathematics | Pueblo | James Louisell | P | |
| University of Colorado – Boulder | Applied Mathematics | Boulder | Anne M. Dougherty | H | |
| | | | Bengt Fornberg | H | |
| | Physics | | Michael H. Ritzwoller | M | M |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| University of Colorado at Denver | Mathematics | Denver | Gary A. Olson | | M |
| | | | Lance W. Lana | | M |
| CONNECTICUT | | | | | |
| Connecticut College | Mathematics and CS | New London | Sanjeeva Balasuriya | | H |
| Sacred Heart University | Mathematics | Fairfield | Peter Loth | | H |
| Southern Connecticut State University | Mathematics | New Haven | Ross B. Gingrich | M | |
| DELAWARE | | | | | |
| Charter School of Wilmington | Mathematics | Wilmington | L. Charles Biehl | P | |
| University of Delaware | Mathematical Sciences | Newark | Louis F. Rossi | | M M |
| DISTRICT OF COLUMBIA | | | | | |
| George Washington University | Mathematics | Washington | Lowell Abrams | B | H P |
| FLORIDA | | | | | |
| Embry-Riddle Aeronautical University | Mathematics | Daytona Beach | Greg S. Spradlin | B | H |
| Jacksonville University | Mathematics | Jacksonville | Robert A. Hollister | B | P P |
| | Physics | | Paul R. Simony | P | |
| University of South Florida | Mathematics | Tampa | Brian Curtin | P | |
| GEORGIA | | | | | |
| Georgia Southern University | Mathematical Sciences | Statesboro | Goran Lesaja | P | H P |
| University of West Georgia | Mathematics | Carrollton | Scott Gordon | P | |
| Wesleyan College | Chemistry and Physics | Macon | Charles J. Benesh | | H |
| | | | Keith L. Peterson | | H |
| | Mathematics | | Joseph A. Iskra | M | H |
| ILLINOIS | | | | | |
| Northern Illinois University | Mathematical Sciences | DeKalb | Ying C. Kwong | | P |
| INDIANA | | | | | |
| Franklin College | Mathematics and Computing | Franklin | John P. Boardman | P | |
| Rose-Hulman Institute of Technology | CS and Software Engineering | Terre Haute | Cary Laxer | M | P |
| | Mathematics | | David J. Rader | M P | |
| Saint Mary's College | Mathematics | Notre Dame | Joanne R. Snow | P | H |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| IOWA | | | | | |
| Grand View Colege | Mathematics and CS | Des Moines | Michelle Ruse | | H |
| Grinnell College | Mathematics and Statistics | Grinnell | Karen L. Shuman | P | P |
| Luther College | Computer Science | Decorah | Steve Hubbard | | H |
| | Mathematics | | Reginald Laursen | H | P |
| Mt. Mercy College | Mathematics | Cedar Rapids | K. R. Knopp | | P |
| Simpson College | Chemistry and Physics | Indianola | Werner S. Kolln | P | |
| | Computer Science | | Paul Craven | | H |
| | Mathematics | | Martha E. Waggoner | H P | |
| KANSAS | | | | | |
| Benedictine College | Mathematics and CS | Atchison | Linda J. Herndon | P | |
| Emporia State University | Mathematics | Emporia | Brian D. Hollenbeck | P | |
| Kansas State University | Mathematics | Manhattan | Dave R. Auckly | M P | |
| KENTUCKY | | | | | |
| Asbury College | Mathematics and CS | Wilmore | David L. Coulliette | P | M |
| Morehead State University | Mathematics and CS | Morehead | Michael Dobranski | | P |
| Northern Kentucky University | Mathematics | Highland Heights | Gail Mackin | H | P |
| LOUISIANA | | | | | |
| Louisiana Tech University | Mathematics and Statistics | Ruston | Katie A. Evans | H | |
| MAINE | | | | | |
| Colby College | Computer Science | Waterville | John E. Augustine | | P P |
| | Mathematics | | Jan Holly | | P |
| MARYLAND | | | | | |
| Johns Hopkins University | Applied Mathematics and Statistics | Baltimore | Fred Torcaso | H | |
| Loyola College in Maryland | Mathematical Sciences | Baltimore | Jiyuan Tao | | M H |
| Salisbury University | Mathematics and CS | Salisbury | Steven M. Hetzler | | H |
| | | | Troy V. Banks | | H |
| Towson University | Mathematics | Towson | Mike O'Leary | P | |
| Villa Julie College | Economics | Stevenson | Eileen C. McGraw | | H P |
| | Mathematics | | Susan P. Slattery | P | H |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| MASSACHUSETTS | | | | | |
| College of the Holy Cross | Mathematics and CS | Worcester | Gareth E. Roberts | P | |
| Harvard University | Engineering and Applied Science | Cambridge | Michael Brenner | P P | |
| | Mathematics | | Clifford H. Taubes | O M | |
| Massachusetts Institute of Technology | Mathematics | Cambridge | Martin Z. Bazant | O M | |
| | Physics | | Leonid Levitov | M M | |
| Salem State College | Mathematics | Salem | Luis P. Poitevin | | P |
| Simon's Rock College | Mathematics | Great Barrington | Allen B. Altman | | H P |
| | Physics | | Mike Bergman | P | |
| University of Massachusetts Lowell | Mathematical Sciences | Lowell | James Graham-Eagle | | M |
| MICHIGAN | | | | | |
| Albion College | Mathematics and CS | Albion | Darren E. Mason | | P |
| Lawrence Technological University | Mathematics and CS | Southfield | Ruth G. Favro | P | M |
| | Physics | | Valentina Tobos | | M P |
| Siena Heights University | Mathematics | Adrian | Jeff Kallenbach | | P |
| | | | Timothy H. Husband | | P |
| MINNESOTA | | | | | |
| Bemidji State University | Mathematics and CS | Bemidji | Colleen Livingston | M | M |
| Bethel University | Mathematics | St. Paul | William M. Kinney | H | |
| Saint John's University | Mathematics | Collegeville | Robert J. Hesse | H | |
| University of Minnesota Duluth | Mathematics and Statistics | Duluth | Bruce B. Peckham | | H |
| MISSOURI | | | | | |
| Drury University | English | Springfield | Ken V. Egan Jr. | | P |
| | Mathematics and CS | | Keith J. Coates | | P P |
| | Physics | | Bruce Callen | | H |
| Northwest Missouri State University | Mathematics and Statistics | Maryville | Russell N. Euler | | P |
| Saint Louis University | Aerospace and Mechanical Eng'ing | Saint Louis | Sanjay Jayaram | | P P |
| Southeast Missouri State University | Mathematics | Cape Girardeau | Robert W. Sheets | | P |
| Truman State University | Mathematics | Kirksville | Steven J. Smith | H | O |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| MONTANA | | | | | |
| Carroll College | Math. Engineering and CS | Helena | Holly S. Zullo | M | |
| | | | Kelly S. Cline | M | |
| | Natural Sciences | | Sam Alvey | | M H |
| NEBRASKA | | | | | |
| Hastings College | Mathematics | Hastings | David B. Cooke | | P |
| Nebraska Wesleyan University | Mathematics and CS | Lincoln | Kristin A. Pfabe | | P |
| Wayne State College | Mathematics | Wayne | Timothy L. Hardy | P P | |
| NEW HAMPSHIRE | | | | | |
| Rivier College | Mathematics and CS | Nashua | William E. Bonnice | | P |
| NEW JERSEY | | | | | |
| Princeton University | Mathematics | Princeton | Robert Calderbank | H | |
| Richard Stockton College of New Jersey | Mathematics | Pomona | Wesley S. Cross | | P |
| Rowan University | Mathematics | Glassboro | Hieu D. Nguyen | | M |
| | | | F. Olcay Ilicasu | | P |
| | Physics and Astronomy | | Eduardo V. Flores | | P |
| NEW MEXICO | | | | | |
| New Mexico Inst. of Mining and Techology | Mathematics | Socorro | John D. Starrett | P | |
| New Mexico State University | Mathematical Sciences | Las Cruces | Mary M. Ballyk | P | |
| NEW YORK | | | | | |
| Clarkson University | Computer Science | Potsdam | Kathleen R. Fowler | | P P |
| | Mathematics | | Joseph D. Skufca | | H H |
| Colgate University | Mathematics | Hamilton | Warren Weckesser | | M P |
| Columbia University | Applied Physics and Applied Math. | New York | David E. Keyes | P | |
| Concordia College–New York | Mathematics | Bronxville | John F. Loase | | H P |
| Cornell University | Mathematics | Ithaca | Alexander Vladimirsky | M | M |
| | Op'ns Research & Ind'l Eng'ng | | Eric Friedman | P | P |
| Ithaca College | Computer Science | Ithaca | Ali Erkan | | H |
| | Mathematics | | John C. Maceli | P | |
| | Physics | | Bruce G. Thompson | H | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Nazareth College | Mathematics | Rochester | Daniel Birmajer | P | |
| Rensselaer Polytechnic Institute | Chemical and Biochem. Engineering | Troy | Shekhar Garde | H H | |
| | Mathematical Sciences | Troy | Peter R. Kramer | P | H |
| Union College | Mathematics | Schenectady | Paul D. Friedman | | P |
| U. S. Military Academy | Mathematics | West Point | Elisha Peterson | M | |
| | | | Robert Burks | | M |
| | Systems Engineering | | John Willis | | M P |
| Westchester Community College | Mathematics | Valhalla | Marvin L. Littman | | P |
| NORTH CAROLINA | | | | | |
| Appalachian State University | Mathematical Sciences | Boone | Katherine J. Mawhinney | P | P |
| Davidson College | Economics | Davidson | Fred H. Smith | | H |
| | Mathematics | | Timothy P. Chartier | | P P |
| | Physics | | Timothy H. Gfroerer | P | |
| Duke University | Computer Science | Durham | Owen Astrachan | M | O |
| | Mathematics | | Anne Catlla | | O |
| | | | David Kraines | O | |
| High Point University | Mathematics | High Point | Brian I. Fulton | | H |
| Meredith College | Mathematics and CS | Raleigh | Cammey E. Cole | | P |
| NC School of Science and Math. | Mathematics | Durham | Daniel J. Teague | H P | |
| Wake Forest University | Mathematics | Winston Salem | Miaohua Jiang | M | M |
| Western Carolina University | Geosciences & Natural Resource Mgmnt | Cullowhee | David Kinner | | P |
| | Mathematics and CS | | Erin McNelis | P | |
| | | | Jeffrey K. Lawson | | P |
| OHIO | | | | | |
| Bowling Green State University | Mathematics and Statistics | Bowling Green | Juan P. Bes | P | |
| Malone College | Mathematics | Canton | David W. Hahn | P | |
| Miami University | Mathematics and Statistics | Oxford | Doug E. Ward | | H |
| University of Dayton | Mathematics | Dayton | Youssef N. Raffoul | | P |
| Xavier University | Mathematics and CS | Cincinnati | Bernd E. Rossa | | M |
| Youngstown State University | Civil/Env'l and Chemical Engineering | Youngstown | Scott Martin | P | |
| | Mathematics and Statistics | | George T. Yates | H | M |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| OREGON | | | | | |
| Eastern Oregon University | Physics | La Grande | Anthony A. Tovar | | P |
| Lewis and Clark College | Mathematical Sciences | Portland | Elizabeth Stanhope | | P |
| Linfield College | Computer Science | McMinnville | Daniel K. Ford | P | H |
| | Mathematics | | Julia D. Fredericks | | P P |
| Oregon Institute of Technology | Mathematics | Klamath Falls | Jim Fischer | P | |
| Oregon State University | Mathematics | Corvallis | Nathan L. Gibson | M | |
| Pacific University | Mathematics | Forest Grove | John August | M | |
| | | | Michael Boardman | | P |
| Portland State University | Mathematics and Statistics | Portland | Gerardo Lafferriere | P | |
| Western Oregon University | Mathematics | Monmouth | Maria G. Fung | P | |
| PENNSYLVANIA | | | | | |
| Altoona Area High School | Mathematics | Altoona | Erin S. Wisor | P | P |
| Bloomsburg University | Mathematics, CS, and Statistics | Bloomsburg | Kevin K. Ferland | | P |
| Lafayette College | Mathematics | Easton | Thomas Hill | | P |
| Shippensburg University | Mathematics | Shippensburg | Paul T. Taylor | | P |
| Slippery Rock University | Mathematics | Slippery Rock | Richard J. Marchand | H | M |
| | Physics | | Athula R. Herat | P | O |
| University of Pittsburgh | Mathematics | Pittsburgh | Jonathan E. Rubin | P | |
| Westminster College | Mathematics and CS | New Wilmington | Barbara T. Faires | M | P |
| | Physics | | Samuel Lightner | | M P |
| SOUTH CAROLINA | | | | | |
| College of Charleston | Mathematics | Charleston | Amy N. Langville | P | P |
| Francis Marion University | Mathematics | Florence | David Szurley | P | |
| | Physics and Astronomy | | David Anderson | | P |
| Midlands Technical College | Mathematics | Columbia | John R. Long | H | P |
| University of South Carolina | Mathematics | Columbia | Lincoln Lu | | P |
| Mount Marty College | Computer Science | Yankton | Bonita Gacnik | | H |
| | Mathematics | | Stephanie A. Gruver | | P P |
| TENNESSEE | | | | | |
| Lipscomb University | Mathematics | Nashville | Mark A. Miller | | P |
| Tennessee Tech University | Mathematics | Cookeville | Andrew J. Hetzel | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| TEXAS | | | | | |
| Rice University | Computational and Applied Math. | Houston | Mark P. Embree | P | M |
| | Electrical & Computer Engineering | | Michael T. Orchard | | M M |
| Trinity University | Engineering Science | San Antonio | William Collins | | H |
| | Mathematics | | Brian Miceli | | P |
| | | | Diane Saphire | | M |
| VIRGINIA | | | | | |
| Eastern Mennonite University | Math and Sciences | Harrisonburg | Leah S. Boyer | | P P |
| Godwin High School | Science Math. and Tech. Ctr | Richmond | Ann W. Sebrell | | H |
| James Madison University | Mathematics and Statistics | Harrisonburg | Anthony Tongen | | H |
| | | | David B. Walton | M | |
| Longwood University | Mathematics and CS | Farmville | M. Leigh Lunsford | | P P |
| Maggie Walker Governor's School | Mathematics | Richmond | John A. Barnes | | H H |
| | Science | | Harold Houghton | | P |
| Radford University | Mathematics and Statistics | Radford | Laura J. Spielman | | M P |
| University of Richmond | Mathematics and CS | Richmond | Kathy W. Hoke | M | |
| Virginia Western | Mathematics | Roanoke | Steve T. Hammer | P | |
| Virginia Western Commmunity College | Mathematics | Roanoke | Ruth A. Sherman | | P |
| | Physics | | Gerald D. Benson | | P |
| WASHINGTON | | | | | |
| Central Washington University | Mathematics | Ellensburg | Stuart F. Boersma | | H |
| Heritage University | Mathematics | Toppenish | Richard Swearingen | | P |
| Pacific Lutheran University | Mathematics | Tacoma | Bryan C. Dorner | P | |
| | Mathematics | | Mei Zhu | P | |
| Seattle Pacific University | Computer Science | Seattle | Elaine Weltz | P | |
| | Electrical Engineering | | Melani Plett | | P |
| Seattle Pacific University | Mathematics | Seattle | Wai Lau | | H P |
| University of Puget Sound | Mathematics | Tacoma | Michael Z. Spivey | | O |
| University of Washington | Applied and Computational Math. | Seattle | Anne Greenbaum | O | H |
| | Mathematics | | James A. Morrow | O H | |
| Washington State University | Mathematics | Pullman | Mark Schumaker | | P |
| Western Washington University | Computer Science | Bellingham | Saim Ural | | P P |
| | Mathematics | | Tjalling Ypma | M | P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| WISCONSIN | | | | | |
| Beloit College | Mathematics and CS | Beloit | Paul J. Campbell | | P |
| Edgewood College | Mathematics | Madison | Steven Post | | P |
| University of Wisconsin – La Crosse | Mathematics | La Crosse | Barbara A. Bennie | P | |
| AUSTRALIA | | | | | |
| U. of Southern Queensland | Mathematics and Computing | Toowoomba | Sergey A. Suslov | | H |
| CANADA | | | | | |
| Brandon University | Mathematics and CS | Brandon | Doug A. Pickering | | H |
| Dalhousie University | Mathematics and Statistics | Halifax | Georg W. Hofmann | P | |
| Queen's University | Mathematics and Statistics | Kingston | Navin Kashyap | | P |
| York University | Mathematics and Statistics | Toronto | Hongmei Zhu | | H H |
| University of Western Ontario | Applied Mathematics | London | Allan B. MacIsaac | | M |
| McGill University | Mathematics and Statistics | Montreal | Nilima Nigam | P | M |
| | | | Stephen W. Drury | P | |
| University of Saskatchewan | Mathematics and Statistics | Saskatoon | James A. Brooke | P | |
| CHINA | | | | | |
| Anhui | | | | | |
| Anhui University | Applied Mathematics | Hefei | Yanhong Bao | | P |
| | Network and Information Engineering | Hefei | Quanbing Zhang | | P |
| | Statistics | | Huayou Chen | P | |
| Hefei University of Technology | Applied Mathematics | | Xueqiao Du | | M H |
| | Mathematics | | Youdu Huang | P | |
| | | | Yongwu Zhou | P | |
| Nonlinear Science Center | Gifted Young | Hefei | Chuanming Wei | | P |
| U. of Science and Technology of China | Automation | Hefei | Xuli Le | H | |
| | Electronic Engineering and Information Sci. | | Qiang Meng | | H |
| | Geophysics | | Zhongwen Zhan | M | |
| | Gifted Young | | Gaigong Zhang | M | |
| | | | Zijing Ding | | H |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Beijing | | | | | |
| Beihang University | Advanced Engineering | Beijing | Shangzhi Li | | H |
| | Automation Sci. and Electr. Eng'ng | | Haiyan Sun | | P H |
| Beijing Forestry University | Bio. Sciences and Biotech. | Beijing | Li HongJun | | P |
| | | | Mengning Gao | M | |
| | Information | | Mengning Gao | | P |
| | Mathematics | | Li HongJun | P | |
| | Science | | Sheng Liu | | P |
| | | | Xiaochun Wang | | P |
| Beijing High School No. 4 | Mathematics | Beijing | Wei Li | | P |
| | English | | Shi Wang | P | P |
| | Foreign Language | | Fang Fang | | P |
| Beijing Information Science and Technology University | Mathematics | Beijing | Xue Chunyan | P | P |
| Beijing Insitute of Technology | Mathematics | Beijing | Qun Ren | | H |
| | | | Gui-Feng Yan | | H |
| | | | Hong-Ying Man | | H |
| | | | Bing-Zhao Li | | M |
| Beijing Jiaotong University | Application Mathematics | Beijing | Jixiang Fan | | M P |
| | Computation Mathematics | | Yongguang Yu | | P |
| | Computer Science | | Xiaoxia Wang | P | H |
| | Electronics | | Chengchao You | P | |
| | Information Engineering | | Yongsheng Wei | | H P |
| | Information Management | | Bingtuan Wang | | M |
| | Mathematics | | Bingtuan Wang | | H |
| | Physics | | Xiaoming Huang | P | |
| | Software | | Wei Lu | P P | |
| | | | Gang Yuan | | P P |
| | Statistics | | Zhouhong Wang | | H |
| | Traffic Engineering | | Liang Wang | | H |
| | | | Jun Wang | | P |
| Beijing Language and Culture University | Computer Science | Beijing | Guilong Liu | | P P |
| | Information | | Xiaoxia Zhao | | P |
| | Mathematics | | Rou Song | | P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
| --- | --- | --- | --- | --- | --- |
| Beijing Normal University | Mathematics | | Qing He | | H |
| | | | Haiyang Huang | P | |
| | | | Hang Qiu | | P |
| | | | Qing He | H | P P |
| | Physics | | Haiyang Huang | | M |
| | Remote Sensing | | Dai Yongjiu | | H |
| | Statistics | | Hengjian Cui | P | |
| | | | Shumei Zhang | | M |
| | | | Yong Li | | P |
| | | | Fu Lai Liu | | H |
| | | | Yong Li | | H |
| | Statistics and Financial Mathematics | | Cui Hengjian | | P |
| | Water Science | | Honghui Wang | | P |
| Beijing University of Aeronautics and Astronautics | Advanced Engineering | Beijing | Linping Peng | | P |
| | Electronic Info. Engineering | | Feng Wei | | M |
| | Science | | Wu Sanxing | P | |
| Beijing University of Chemical Technology | Electric Science | Beijing | Guangfeng Jiang | H | |
| | | | Kaisu Wu | | P |
| | Mathematics | | Wenyan Yuan | | P |
| | | | Xinhua Jiang | | P |
| Beijing University of Posts and Telecomm. | Applied Mathematics | Beijing | Hongxiang Sun | | P |
| | | | Zuguo He | H | P |
| | Applied Physics | | Jinkou Ding | | H |
| | CS and Technology | | Zuguo He | | H |
| | Electronic Engineering | | Qing Zhou | | P |
| | Information Engineering | | Wenbo Zhang | H | |
| | Telecomm. Engineering | | Jianhua Yuan | H | |
| Beijing University of Technology | Applied Science | Beijing | Zhen Wang | P | |
| | | | Yi Xue | P | |
| Beijing Wuzi University | Informatics | Beijing | Zhenping Li | M | H |
| | | | Tian Deliang | H | P |
| | | | Cheng Xiaohong | | P P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Central University of Finance and Economics | Mathematics | Beijing | Xiuguo Wang | | P P |
| | Mathematics | | Xianjun Yin | H | |
| | Mathematics | | Zhaoxu Sun | P | |
| China Agricultural University | Honors Programme (Life Science) | Beijing | Xuewei Wang | | P |
| | Science | | Hui Zou | H | |
| | | | Jun Feng Liu | H | H |
| | | | Xu Hua Liu | | P |
| China University of Geosciences | Geosciences and Resources | Beijing | Deng Yan | H H | |
| | Water Resources and Environmental Science | | Zhaodou Chen | P P | |
| China University of Petroleum | Mathematics and Physics | Beijing | Xiaoguang Lu | H | |
| North China Electric Power University | Automation | Beijing | Wen Tan | | H |
| | Electrical Engineering | | Pan Zhi | | H |
| | Electric Power Engineering | | Li Guo Dong | | P |
| | Electrical Engineering | | Xirong Zhang | P | |
| | Mathematics | | Qiu QiRong | | P |
| | Mathematics and Physics | | Qi Rong Qiu | | P |
| Peking University | Applied Mathematics | Beijing | Minghua Deng | H | P |
| | Business Statistics and Economics | | Junni Zhang | M | |
| | Computer Science | | Xiaolin Wang | | H |
| | Finance | | Peng He | H H | |
| | Financial Mathematics | | Xin Yi | | P P |
| | Guiding Centre for Students' Extracurricular Activities | | Liang Lu | P | P |
| | Mathematics | | Xufeng Liu | | O H |
| | Quantum Electronics Institute | | Zhigang Zhang | | H |
| | Scientific and Engineering Computing | | Pingwen Zhang | | P |
| Renmin University of China | Applied Mathematics | Beijing | Litao Han | | H |
| | Computer Science | | Wang Wei | | H |
| | Information School | | Yunyan Yang | P | |
| | Management of Information Systems | | Deying Li | P | |
| | Mathematics | | Yunyan Yang | P | |
| | | | Zhou Zemin | | M |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Tsinghua University | Chemistry | Beijing | Xianqing Qiu | P | |
| | Mathematical Science | | Zhiming Hu | P | |
| | Mathematical Science | | Zhiming Hu | | M |
| | Mathematical Science | | Jun Ye | M | H |
| Chongqing | | | | | |
| Chongqing University | Chemistry | Chongqing | Li Zhiliang | H | P |
| | Computer Software Engineering | | Wu Kaigui | | P |
| | Mathematics | | Liu Qiongfang | | P |
| | Mathematics and Physics, Information and Computational Science | | Gong Qu | | P |
| Southwest University | Mathematics and Statistics | Chongqing | Xuegao Zheng | | P |
| | | | Chunlei Tang | | H |
| | | | Lei Deng | | M |
| | | | Wendi Wang | | H |
| Fujian | | | | | |
| Fujian Normal University | Computer Science | Fuzhou | Changfeng Ma | P | |
| | | | Huiling Lin | | P |
| | Mathematics | | Shenggui Zhang | | P |
| | | | Qinhua Chen | | P |
| Fujian University of Technology | Mathematics and Physics | Fuzhou | Lin Li | P | |
| Hua Qiao University | Mathematics | Quanzhou | Hai Zhou Song | P | |
| Xiamen University | Automation | Xiamen | Langcai Cao | | P |
| | Physics | | Guozhen Su | B | P P |
| Guangdone | | | | | |
| Guangzhou University | Mathematics | Guangzhou | Fu Ronglin | | P |
| | | | Shang Yadong | | P |
| | | | Wang Xiaofeng | | P |
| Jinan University | Computer Science | Guangzhou | Shi Zhuang Luo | P | |
| | | | ShiQi Ye | | P |
| | Mathematics | | Daiqiang Hu | | H |
| | | | Suohai Fan | | P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Jinan University, Zhuhai College | Computer Science | Zhuhai | Guangqing Lu | | H P |
| | | | Yuanbiao Zhang | M | |
| | Packaging Engineering | | Zhiwei Wang | | P |
| Shenzhen Polytechnic | Industrial Training Center | Shenzhen | Hongmei Tian | P | |
| | | | Dongping Wei | H | |
| | Mechanical and Electrical Engineering | | Kanzhen Chen | | H |
| South China Normal University | Information and Computationall Sci. | Guangzhou | Tan Yang | M | |
| | Mathematics | | ShaoHui Zhang | | H P |
| | Statistics | | QuanXin Zhu | | H |
| South China University of Technology | CS and Engineering | Guangzhou | Zhi-Sui Tao | M | |
| | Mathematical Sciences | | Shen-Quan Liu | P | |
| | | | Shaohua Pan | | M |
| | | | Ping Huang | | P |
| Sun Yat-Sen University | Computer Science | Guangzhou | XiaoMing Liu | H | |
| | Geography | | Zuo Jian Yuan | H | |
| | Mathematics | | Yan Hui Li | | M |
| | Statistics | | XiaoLing Yin | H | |
| Guangxi | | | | | |
| Guangxi Teacher Education University | Mathematics | Nanning | ChengDong Wei | P | |
| | Mathematics and CS | | JianWei Chen | | P |
| | | | XiongFa Mai | | P |
| Guangxi University of Finance and Economics | Mathematics and Statistics | Nanning | Yuanyuan Tan | | P |
| Guilin University of Electronic Techhology | Mathematics and CS | Guilin | Xingxing He | P | |
| | | | Yongxiang Mo | H | |
| | Information and Communication | | Sunyong Wu | | P |
| University of Guangxi | Applied Mathematics | Nanning | Xiaoceng Wu | P | P |
| | Computing Science | | Yuejin Lu | P | |
| | Information Science | | Zhongxing Wang | P | P |
| | Operational Management | | Ruxue Wu | P P | |
| Guizhou University for Nationalities | Mathematics and CS | Guiyang | Zhensheng Hong | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Hebei | | | | | |
| North China Electric Power University | Mathematics | Baoding | Gendai Gu | | P |
| | | | Peng Li | H | |
| | Mathematics and Physics | | Bo Xiong | | H |
| | | | Shenghua Wang | | P |
| | | | Po Zhang | | H |
| | | | Yagang Zhang | H | |
| Heilongjiang | | | | | |
| Daqing Petroleum Institute | Mathematics | Daqing | Yang Yunfeng | | H P |
| | | | Kong Lingbin | | P P |
| Harbin Engineering University | Info. and Computer Science | Harbin | Yu Fei | | P |
| | Mathematics | | Luo Yuesheng | H | |
| | | | Shen Jihong | H H | |
| Harbin Medical University | Bioinformatics | Harbin | Wang Qiang Hu | | H H H |
| Harbin University of Science and Technology | Mathematics | Harbin | Dongmei Li | | P |
| | | | Fengqiu Liu | | P |
| | | | Guangyue Tian | | H |
| | | | Zuobao Cao | | P |
| Heilongjiang University | Mathematical Sciences | Harbin | Weijun Ma | P | P |
| Jia Musi University | Mathematics | Jiamusi | Zhang Juhong | P | P |
| Northeast Agricultural University | Agricultural Engineering | Harbin | Fang Ge Li | P | |
| | CS and Technology | | Fang Ge Li | P | |
| | | | Qiufeng Wu | P | |
| | Information and Computing Science | | Ya Zhuo Zhang | | P |
| Harbin Institute of Technology | Environmental Scienceand Engineering | Harbin | Tong Zheng | | H |
| | Management Scienceand Engineering | | Hong Ge | | M H |
| | | | Wei Shang | H | P |
| | Mathematics | | Chiping Zhang | | H |
| | | | Guanghong Jiao | P | P |
| | | | Kean Liu | | P P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Harbin Institute of Tech. (cont'd) | Mathematics | | Shouting Shang | H | |
| | | | Guofeng Fan | P | |
| | | | Xianyu Meng | | P |
| | | | Chiping Zhang | P | |
| | | | Xilian Wang | H P | |
| | Science | | Yunfei Zhang | H | P |
| Heilongjang Inst. of Scienceand Tech. | Mathematics and Mechanics | Harbin | Hong Du | | P |
| Henan | | | | | |
| Information Engineering University | Information Security | Zhengzhou | Zhang Xiao Yong | P | |
| | Surveying and Mapping | | Zhang Guoliang | | P |
| | Management | | Jia Li Xin | | M |
| Hubei | | | | | |
| Huazhong University of Science and Tech. | Electronics and Information Engineering | Wuhan | Yan Dong | | P |
| | Industrial and Manufacturing Sys. Eng'ng | | Liang Gao | M | P |
| | Mathematics | | Yizhi Wang | P | |
| Wuhan University | Electronic Information | Wuhan | Yuanming Hu | | P |
| | Mathematics | Wuhan | Hu Xinqi | | M |
| | | | Luo Zhuangchu | | H |
| | Mathematics and Statistics | | Yuanming Hu | | H |
| | | | Shihua Cheng | | P |
| | | | Xinqi Hu | | P |
| | | | Aijiao Deng | | P P |
| | | | Liuyi Zhong | | P P |
| Wuhan University of Technology | Automobile Engineering | Wuhan | Gao Fei | P | |
| | Control Engineering | | Huang Xiaowei | P | |
| | | | Yang Wenxia | | P |
| | | | Zhou Jun | | H |
| | | | Zhu Huaping | | P |
| | Mathematics | | Chen Jianye | | P |
| | | | Chu Yangjie | P | |
| | | | He Lang | M | |
| | | | Zhu Huiying | H | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Wuhan University of Technology (cont'd) | Statistics | | Tian Wufeng | H | |
| | | | Mao Shuhua | | P |
| HUNAN | | | | | |
| Central South University | Applied Mathematics | Changsha | Zheng Zhoushun | P | |
| | | | Qin Xuanyun | P | |
| | Geographical Information Systems | | Zhang Hongyan | | P |
| | Information and CS | | Zhang HongYan | H | |
| | Math'l Scienceand Computing Tech. | | Hou Muzhou | | M |
| | | | Zhu Shihua | P | |
| Hunan University | Applied Mathematics | Changsha | Xiaopei Li | P | |
| | CS and Technology | | Hao Wu | P | |
| | Probability and Statistics | | Han Luo | | H |
| | | | Liping Wang | | P |
| National University of Defense Technology | Applied Mathematics | Changsha | Dan Wang | M | P |
| | Mathematics | | Ziyang Mao | P | |
| | | | Yi Wu | | O P |
| | | | Mengda Wu | | M P |
| | Physics | | Meihua Xie | | P |
| | | | Xiaojun Duan | P | H |
| Inner Mongolia | | | | | |
| Inner Mongolia University | Automation | Huhehaote | Zhuang Ma | P | |
| | Mathematics | | Haitao Han | | P |
| Jiangsu | | | | | |
| China University of Mining and Technology | Applied Methematics | Xuzhou | Wu Zongxiang | P | |
| | Information and Electrical Engineering | | Gong Dunwei | | P |
| | Mathematics | | Zhou Shengwu | | P P |
| Institute of System Engineering | Mathematics | Zhenjiang | Honglin Yang | P | P |
| Jiangsu University | Mathematics | Zhenjiang | Guilong Gui | | P P |
| | Science | | Yimin Li | P | P |
| Nanjing Univ. of Scienceand Technology | Applied Mathematics | Nanjing | Peibiao Zhao | | P |
| | | | Zhipeng Qiu | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Nanjing University of Scienceand Tech. (cont'd) | Statistics | | Liwei Liu | | P |
| | | | Zhang Zhengjun | | P |
| Nanjing University | Electronic Science and Engineering | Nanjing | Haodong Wu | | M P |
| | | | Qian Chen | | H |
| | Finance | | Dixin Zhang | | P |
| | Life Science | | Jin Wang | | M |
| | Mathematics | | Weihua Huang | | P P |
| | Physics | | Huimin Shao | P | |
| | | | Fa Liu | H | |
| Nanjing University of Posts and Telecomm. | Mathematics and Physics | Nanjing | Xu LiWei | | P |
| | | | Kong Gaohua | | M |
| | | | Qiu Zhonghua | | H |
| | | | Shi Aiju | | H P |
| Nantong University | Electrical Engineering | Nantong | Guoping Lu | | H |
| | Science | | Yuehua Guo | | P |
| | | | Zhao Min Group | | P |
| Nonlinear Scientifc Research Center | Mathematics | Zhenjiang | Gang Xu | P | H |
| PLA (People's Liberation Army) University of Science and Technology | Applied Mathematics and Physics | Nanjing | Shi Hansheng | H | |
| | Communication Engineering | | Liu Shousheng | | P |
| | Science | | Shen Jinren | | P |
| | | | Teng Jiajun | P | |
| Southeast University | Mathematics | Nanjing | Enshui Chen | P | P |
| | | | Feng Wang | | P P |
| | | | Dan He | H | P |
| | | | Liyan Wang | H | H |
| Xuzhou Institute of Technology | Mathematics | Xuzhou | Jiang Yingzi | M P | |
| | | | Li Subei | | H P |
| Jianxi | | | | | |
| Gannan Normal University | Computer | Ganzhou | Yan Shenhai | P | |
| | Mathematics | | Xie Xianhua | | H |
| | | | Xu Jingfei | | H |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Nanchang University | Mathematics | Nanchang | Chen Tao | H | |
| | | | Chen Yuju | P | |
| | | | Chuanrong Liao | | P |
| | | | Xinsheng Ma | H | |
| Jilin | | | | | |
| Beihua University | Basic Courses | Jilin City | Hongwei Zhao | | P |
| | | | Yuncai Wei | | H |
| | | | Zhaojun Chen | | P |
| | | | Ming Zhao | | P |
| Jilin Teachers' Institute of Engineering and Technology | Science | Changchun | Changchun Li | | P |
| Jilin University | Communication Engineering | Changchun | Chunling Cao | H | |
| | | | XiuLing Yao | | P |
| | Mathematics | | Peichen Fang | P | P |
| | | | Jinying Liu | P | |
| | | | Lu Xianrui | | H |
| | | | Qingdao Huang | P | |
| | | | Shaoyun Shi | | P P |
| | | | Yang Cao | | P |
| | | | Zhao Shishun | | P |
| Liaoning | | | | | |
| Dalian Fisheries University | Science | Dalian | Zhang Lifeng | | P |
| Dalian Maritime University | Mathematics | Dalian | Y. J. Zhang | | H P |
| Dalian Nationalities Innovation College | Innovation Education Center | Dalian | Chen Xing Wen | | P P |
| | | | Tian Yun | | H P |
| Dalian Nationalities University | CS and Engineering | Dalian | Yan De Jun | P | P |
| | | | Liu Rui | P | |
| | | | Liu Xiang Dong | | P |
| | Dean's Office | | Ge Ren Dong | P | |
| | | | Guo Qiang | | P |
| | | | Ma Yu Mei | P | |
| | | | Xue Ye | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Dalian Nationalities University (cont'd) | Dean's Office | | Li Xiao Niu | | H |
| | | | Liu Jian Guo | P | |
| | | | Zhang Heng Bo | P | |
| | | | Fu Jie | | P |
| Dalian Navy Academy | Computer Science | Dalian | Yin Cheng yi | | P P |
| | Operation Research | | Feng Jie | | H P |
| Dalian Neusoft Institute of Information | InformationTechnologyand Business Management | Dalian | Yuxin Zhao | | P |
| Dalian University | Information Engineering | Dalian | Liu Guangzhi | | P |
| | | | Dong Xiangyu | | P P |
| | Mathematics | | Gang Jiatai | | H |
| | | | Liu Zixin | | H |
| | | | Tan Xinxin | H | |
| Dalian University of Technology | Applied Mathematics | Dalian | Mingfeng He | P | |
| | | | Cai Yu | | P |
| | | | Geng Xinghua | | P |
| | | | He Mingfeng | H | |
| | | | Wang Yi | H | |
| | | | Xiao Di | P | |
| | City Institute | | Hongzeng Wang | | P |
| | | | Lianfu Li | H | |
| | | | Meng Lin | | P |
| | | | Xubin Gao | | P |
| | Institute of University Students' Innovation | | Feng Lin | | P |
| | | | He Mingfeng | | P |
| | | | Sun Tao | | M P |
| | | | Wu Zhenyu | | P P |
| | | | Xi Wanwu | | P |
| | Software School | | Jiang He | | P |
| | | | Li Zhe | | P P |
| | | | Xu Shengjun | P | |
| | | | Zhe Li | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Liaoning Province Shiyan High School | Year 08 (Grade 2) | Shenyang | Chunzhi Zhang | | P |
| Northeastern University | AI and Robotics | Shenyang | Feng Pan | P | |
| | | | Yunzhou Zhang | | P |
| | Computer System | | Huilin Liu | | H |
| | Control and Simulation | | Peifeng Hao | H | |
| | | | Jianjiang Cui | | P |
| | Information Scienceand Engineering | | Chengdong Wu | H | |
| | | | Shuying Zhao | | M |
| | Modern Design and Analysis | | Xuehong He | M | |
| Shenyang Inst. of Aero. Engineering | Information and CS | Shenyang | Shiyun Wang | | P |
| | Science | Shenyang | Weifang Liu | | P |
| | | | Limei Zhu | M | |
| | | | Feng Shan | H | H |
| | | | Feng Shan | | P |
| Shenyang Normal University | Mathematics and System Science | Shenyang | Li Xiaoyi | P | |
| | | | Liu Yuzhong | | P |
| | | | Meng Xianji | P | |
| Shenyang Pharmaceutical University | Mathematics Teaching and Research | Shenyang | Xiang Rong Wu | | P P |
| Shenyang University of Technology | Mathematics | Shenyang | Yan Chen | | M P |
| Shenyang University | Mathematics | Shenyang | Guiyan Mu | | P |
| Shaanxi | | | | | |
| North University of China | Mathematics | Taiyuan | Yang Ming | P | |
| | | | Wang Peng | | P |
| | Science | | Yang Xiaofeng | | P |
| North University of China Fenxiao | Electrical Engineering | Taiyuan | Xiaoren Fan | | H P |
| | Foreign Language | | Yongjian Ren | | P |
| Northwest A&F University | Science | Yangling | Wang Jingmin | P P | |
| Northwest University | Physics | Xi'an | Yingjie Du | P | |
| | | | Qingyan Dong | H | |
| Northwestern Polytechnical University | Applied Chemistry | Xi'an | Genjiu Xu | P | |
| | | | Sun Zhongkui | | M |
| | Applied Mathematics | | Peng Guohua | | H |
| | | | Nie Yufeng | M | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Northwestern Polytech. University (cont'd) | Applied Physics | | Youming Lei | M | |
| | Natural and Applied Science | | Sun Hao | | H |
| | | | Xiao Huayong | | H P |
| | | | Yong Xu | | H |
| Taiyuan University of Technology | Mathematics | Taiyuan | Yiqiang Wei | | P |
| Xi'an Communication Institute | Computer Science | Xi'an | Hong Wang | | P |
| | Electronic Engineering | | Xinshe Qi | P | |
| | Mathematics | | Guo Li | | P P |
| | Physics | | Dongsheng Yang | | P |
| Xi'an Jiaotong University | Applied Mathematics | Xi'an | Xiaoliang He | | P |
| | | | Zhuosheng Zhang | H | |
| | Mathematics | | Feng Liu | P | |
| | | | Lizhou Wang | | H |
| Xidian University | Applied Mathematics | Xi'an | Yue Song | M | |
| | Computational Mathematics | | Hongyun Meng | | P |
| | Industrial and Applied Mathematics | | Feng Ye | | H |
| | | | Guoping Yang | | H |
| Shandong | | | | | |
| Shandong University | Computer Science and Technology | Jinan | Xinshun Xu | H | |
| | | | Xinshun Xu | | P |
| | Mathematics and System Science | | Baodong Liu | | M |
| | | | Shuxiang Huang | | P P P |
| | | | Tongchao Lu | | M P |
| | | | Baodong Liu | P | M P |
| Shandong University (Weihai) | Applied Mathematics | Weihai | Zhulou Cao | | P P P |
| Shandong University of Science and Tech. | Information Science and Engineering | Qingdao | Shanchen Pang | M H | |
| Shanghai | | | | | |
| China Textile University | Mathematics | Shanghai | Chen Gang | | H |
| Donghua University | InformationScience and Technology | Shanghai | Jie Qi | | H |
| | | | Yongsheng Ding | | P |
| | Science | | Yong Ge | | P |
| | | | Junmei Hou | | P |
| | | | Mengyu Hu | | P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| East China Normal University | Statistics | Shanghai | Yiming Cheng | | H |
| East China University of Science and Technology | Mathematics | Shanghai | Liu Zhaohui | | P |
| | | | Qin Yan | | P P |
| | Physics | | Su Chunjie | | H |
| Fudan University | CS and Engineering | Shanghai | Haibin Kan | | H |
| | International Finance | | Hongzhong Liu | H | H |
| | Mathematics | | Yuan Cao | | P |
| | | | Zhijie Cai | | P |
| | Mechanics and Engineering Science | | Sheng Cui | | P |
| Fudan University Research Ctr for Nonlinear Science | Applied Mathematics | Shanghai | Wei LIN | | P |
| Shanghai Finance University | Applied Mathematics | Shanghai | Yumei Liang | | P |
| | Finance | | Rui Li | | P P |
| Shanghai Financial and Economic U. | Economics | Shanghai | Zheng Xu | P | |
| Shanghai Foreign Language School | Administration | Shanghai | Liqun Pan | | H P |
| | | | Yu Sun | | H P |
| | Mathematics | | Feng Xu | P | P |
| | | | Weiping Wang | | P P |
| | Principal's Office | | Liang Tao | H | P |
| Shanghai Hongkou Inst. of Educ. | Mathematics | Shanghai | Jun Hu | M | P |
| | | | Shengyang Ye | | H P |
| Shanghai Hongkou Youth Center | Mathematics | Shanghai | Jian Tian | | P |
| Shanghai Jiading No. 1 Senior HS | | Shanghai | Xie Xilin and Fang Yunping | | P. P |
| Shanghai Jiaotong University | Mathematics | Shanghai | Baorui Song | H | P |
| | | | Jianguo Huang | | P P |
| Shanghai Normal University | Applied Mathematics | Shanghai | Jizhou Zhang | | P |
| | Applied Mathematicsand Statistics | | Yongbing Shi | | P |
| | | | Haiyan Xu | P | |
| | Computational Mathematics | | Qian Guo | P | |
| Shanghai University | Mathematics | Shanghai | Dong Hua Wu | P | H |
| | | Shanghai | Youhua He | | P |
| | Science | | YuanDi Wang | P | |
| | | | Wei Huang | P | P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Shanghai University of Finance and Economics | Applied Mathematics | Shanghai | Zanzan Li | P | |
| | | | Dawu Yu | | P |
| | | | Li Ming | | M |
| | | | Zhen Yu Zhang | | M |
| | | | Chengdong Dong | | H |
| | | | Yu Zhu | H | |
| | Finance | Shanghai | Shuyuan Pan | | P |
| Shanghai Youth Center of Science and Technology Education | Mathematics | Shanghai | Gan Chen | | P |
| | Scientific Training | | Gan Chen | | H |
| Sino European School of Technology | Fundamental Science and Technology | Shanghai | Yang Yongjian | H | P |
| Tongji University | Environmental Science and Eng'ng | Shanghai | Hailong Yin | P | P |
| | Mathematics | | Hualong Zhang | P | P |
| | | | Jin Liang | | P |
| | Software | | Jialiang Xiang | H | H |
| | | | Xiongda Chen | | P |
| University of Shanghai for Science and Technology | Science | Shanghai | Jia Gao | P | H |
| Yucai Senior High School | Mathematics | Shanghai | Zhenwei Yang | P | H |
| Sichuan | | | | | |
| Chengdu University of Technology | Information Management | Chengdu | Chen Guodong | | P |
| | | | Huang Guangxin | | P |
| Sichuan University | Applied Mathematics | Chengdu | Hai Niu | | M |
| | Mathematics | | Yang Weng | P | |
| | | | Zhou Jie | | M |
| | | | Li Xiao bin | H | |
| | | | Shuchao Zou | | H |
| | Polymer Science and Engineering | | Ming Shu Fan | P | |
| | Yangtze Center of Mathematics | | Bohui Chen | M | |
| Southwestern University of Finance and Econ. | Business Administration | Chengdu | Sun Yun Long | | P |
| | Economic Mathematics | | Dai Dai | | P |
| | | | Shaowen Li | | M |
| | | | Yunlong Sun | P | |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Univ. of Elec. Scienceand Technologyof China | Applied Mathematics | Chengdu | Gao Qing | | H H |
| | Biomedical Engineering | | Wang Zhu | H | |
| | Informationand Computation Science | | Qin Siyi | | M |
| | | | Xu Quanzi | | H |
| Tianjin | | | | | |
| Civil Aviation University of China | Air Traffic Control | Tianjin | Nie Runtu | | P |
| | | | Zhaoning Zhang | | H |
| | CS and Technology | | Yuxiang Zhang | | M |
| | | | Liu Shan | | P |
| | Science | | Chen Shang Di | | P |
| | | | Yongxin Gao | | P |
| | | | Deyi Mou | | M |
| | | | Tian Ming | | H |
| Nankai University | Economic Technology | Tianjin | Tan Xu | | H |
| | Informatics and Probability | Tianjin | Xingwei Zhou | | H |
| | | | Chunsheng Zhang | H | |
| | | | Jishou Ruan | | P |
| Yunnan | | | | | |
| Yunnan University | Computer Science | Kunming | Wang Rui | | H |
| | Statistics | | Jie Meng | P | |
| | Telecomm. Engineering | | Guanghui Cai | | P |
| Zhejiang | | | | | |
| Hangzhou Dianzi University | Applied Physics | Hangzhou | Chengjia Li | | H |
| | | | Zhifeng Zhang | | P |
| | Information and Mathematics | | Hao Shen | | M |
| | | | Zheyong Qiu | | H |
| Shaoxing University | Mathematics | Shaoxing | Hu Jinjie | P | P |
| | | | Sun Jianxin | P | |
| Zhejiang Gongshang University | Informationand Computing Science | Hangzhou | Hua Jiukun | | H H |
| | Mathematics | | Zhu Ling | H | P |
| Zhejiang Normal University | CS and Technology | Jinhua | Wenqing Bao | | H |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| Zhejiang Normal University (cont'd) | Mathematics | | Xinzhong Lu | | H |
| | | | Qiusheng Qiu | P | |
| Zhejiang Sci-Tech University | Applied Physics | Hangzhou | Hu Jueliang | H | |
| | Mathematics | | Luo Hua | | P |
| | | | Shi Guosheng | | P |
| Zhejiang University | Mathematics | Hangzhou | Qifan Yang | P | |
| | | | Zhiyi Tan | M | M |
| | Ningbo Institute of Technology | Ningbo | Jufeng Wang | M | H |
| | | | Zhening Li | H | H |
| | Science | Hangzhou | Jiaer Fei | | H H |
| Zhejiang University City College | CS and Technology | Hangzhou | Huizeng Zhang | H | |
| | Information and Computing Science | | Gui Wang | M | H |
| | | | Xusheng Kang | | H |
| Zhejiang University of Finance and Econ. | Mathematics and Statistics | Hangzhou | Fulai Wang | | P |
| | | | Ji Luo | | H |
| Zhejiang University of Science and Technology | Science | Hangzhou | Mingjun Wei | | P |
| | | | Yongzhen Zhu | | P |
| Zhejiang University of Technology | Jianxing College | Hangzhou | Wenxin Zhuo | | H |
| | | | Shiming Wang | M | M |
| | Mathematics | | Minghua Zhou | | P |
| | | | Xuejun Wu | | P |
| FINLAND | | | | | |
| Helsingin Matematiikkalukio | Mathematics | Helsinki | Esa I. Lappi | | P P |
| Päivölä College of Mathematics | Computer Sciences | Tarttila | Jukka Ilmonen | P | |
| | Mathematics | | Janne J. E. Puustelli | P | |
| University of Helsinki | Mathematics and Statistics | Helsinki | Petri Ola | P | |
| INDONESIA | | | | | |
| Institut Teknologi Bandung | Mathematics | Bandung | Rieske Hadianti | | H P |
| IRELAND | | | | | |
| National University of Ireland | Mathematical Physics | Galway | Petri T. Piiroinen | P | H |
| | Mathematics | | Niall Madden | | M P |

| INSTITUTION | DEPARTMENT | CITY | ADVISOR | A | B |
|---|---|---|---|---|---|
| University College Cork | Applied Mathematics | Cork | Dmitrii Rachinskii | M | |
| | | | Liya A. Zhornitskaya | P | |
| | Mathematics | | Donal J. Hurley | H | |
| | | | Benjamin W. McKay | P | |
| University College Dublin | Mathematical Sciences | Dublin | Maria G Meehan | P | |
| | | | Ted Cox | | P |
| JAMAICA | | | | | |
| University of Technology | Chemical Engineering | Kingston | Nilza G. Justiz-Smith | | H |
| KOREA | | | | | |
| Korea Adv. Inst. of Science and Technology | Mathematical Sciences | Daejeon | Yong Jung Kim | M | H |
| | Mechanical Engineering | | Joongmyeon Bae | | M |
| | Physics | | Hawoong Jeong | | P |
| NEW ZEALAND | | | | | |
| Victoria University of Wellington | Mathematics, Statistics, and CS | Wellington | Mark J. McGuinness | | H |
| SINGAPORE | | | | | |
| National University of Singapore | Mathematics | Singapore | Gongyun Zhao | P | P |
| | Statistics and Applied Probability | | Yannis Yatracos | | O |
| SOUTH AFRICA | | | | | |
| University of Pretoria | Mathematics and Applied Mathematics | Pretoria | Ansie F. Harding | | H |
| University of Stellenbosch | Applied Mathematics | Stellenbosch | Jan H. van Vuuren | | O M |

**Editor's Note:** For team advisors from China, I formerly followed the *New York Times* in listing family name first. That convention is probably because Chinese call each other by family name first. In most Chinese names, the family name is a single syllable and the given name is two syllables; so in most cases it is easy to identify family name. However, if each name has one syllable, it is not easy—even for native speakers—to distinguish them from the English transliteration. Since I have always doubted the wisdom of the *Times* convention, since Chinese Outstanding teams have requested that family names be listed last, and because even with expert help I cannot be completely correct, this year I have left the names in the order given on the registration form.

# When Topologists Are Politicians ...

Nikifor C. Bliznashki
Aaron Pollack
Russell Posner
Duke University
Durham, NC

Advisor: David Kraines

## Summary

Former Supreme Court Justice Sandra Day O'Connor once noted that any politician who did not do everything to secure power for the party "ought to be impeached." Though Congress argues that wild election districts such as "a pair of earmuffs" are inherently fair and reasonable, they are so counterintuitive that such claims can be exceedingly difficult to believe.

Defining the big picture of what is "fair" can be left to philosophers—or computers. Using a novel method, we divide states into districts of equal population, with each district as compact and elementary as possible, where compactness is defined as the moment of inertia of the district with respect to the population density. By not examining any other demographic data in the grouping, we avoid many of the biases that people may impose. We obtain districts considerably more compact than current congressional districts for Ohio and New York.

Since it is constitutionally sound to group people into congressional districts by "shared interests," we extend the problem by allowing other demographic data to be considered in the formation of such districts. to form revised districts that seek to preserve uniformity of these qualities. We identify how suitable these solutions are and determine their advantages and disadvantages.

Finally, we propose alternative districting techniques that take into account county boundaries and natural boundaries.

The text of this paper appears on pp. 249–260.

# What to Feed a Gerrymander

Ben Conlee
Abe Othman
Chris Yetter
Harvard University
Cambridge, MA

Advisor: Clifford H. Taubes

## Summary

Gerrymandering, the practice of dividing political districts into winding and unfair geometries, has a deleterious effect on democratic accountability and participation. Incumbent politicians have an incentive to create districts to their advantage (California in 2000, Texas in 2003); so one proposed remedy for gerrymandering is to adopt an objective, possibly computerized, methodology for districting.

We present two efficient algorithms for solving the districting problem by modeling it as a Markov decision process that rewards traditional measures of district "goodness": equality of population, continuity, preservation of county lines, and compactness of shape. Our Multi-Seeded Growth Model simulates the creation of a fixed number of districts for an arbitrary geography by "planting seeds" for districts and specifying particular growth rules. The result of this process is refined in our Partition Optimization Model, which uses stochastic domain hill-climbing to make small changes in district lines to improve goodness. We include as an extension an optimization to minimize projected inequality in district populations between redistrictings.

As a case study, we implement our models to create an unbiased, geographically simple districting of New York using tract-level data from the 2000 Census.

The text of this paper appears on pp. 261–280.

# Electoral Redistricting with Moment of Inertia and Diminishing Halves Models

Andrew Spann
Daniel Kane
Dan Gulotta
Massachusetts Institute of Technology
Cambridge, MA

Advisor: Martin Z. Bazant

## Summary

We propose and evaluate two methods for determining congressional districts. The models contain explicit criteria only for population equality and compactness, but we show that other fairness criteria such as contiguity and city integrity are present, too.

The Moment of Inertia Method creates districts whose populations are within 2% of the mean district size, minimizing the sum of the squares of distances between the district's centroid and each census tract (weighted by population size). We prove that this model gives convex districts.

In the Diminishing Halves Method, the state is recursively halved by lines perpendicular to best-fit lines through the centers of census tracts.

From U.S. Census 2000 data, we extract the latitude, longitude, and population count of each census tract. By parsing data at the tract level instead of the county level, we model with high precision. We run our algorithms on data from New York as well as Arizona (small), Illinois (medium), and Texas (large).

We compare the results to current districts. Our algorithms return districts that are not only contiguous but also convex, aside from borders where the state itself is nonconvex. We superimpose city locations on district maps to check for community integrity. We evaluate our proposed districts with various quantitative measures of compactness.

The initial conditions do not greatly affect the Moment of Inertia Method. We run variants of the Diminishing Halves Method and find that they do not improve over the original. Based on our results, district shapes should be convex, and city boundaries and contiguity can be emergent properties, not explicit considerations. We recommend our Moment of Inertia Method, as it consistently performed the best.

The text of this paper appears on pp. 281–299.

# A Voronoi Model for Districting

Benjamin O. Barrow
Andrew F. Glugla
John B. Shelton
University of Colorado
Boulder, CO

Advisor: Michael H. Ritzwoller

## Summary

The U.S. Constitution allots each state a number of seats in the House of Representatives proportional to the state's population. However, it says nothing about how the districts associated with seats should be defined. This silence allows politicians to modify district borders to their advantage in future elections, a practice known as "gerrymandering."

We offer a model for establishing congressional districts in a fair and unbiased manner. We provide rigorous definitions of "fairness" and "simplicity." For simplicity and to remove any political bias, we ignore all census statistics except population density. We also ignore transportation infrastructure, since political bias can often affect construction of roads, railways, and airports.

Our model uses a Voronoi diagram to divide a state into districts. We place Voronoi points on a state map and divide the state into regions that enclose each point separately. All territory within a given region is closer to that region's Voronoi point than to any other Voronoi point, resulting in the formation of simple, convex polygonal areas. The model then uses a logically derived law to move the Voronoi points and thus trade territory between neighboring districts. Lower-population districts gain population, and greater-population districts lose population, until all districts arrive at population equilibrium.

Our Voronoi model reliably produces workable results, though the final district borders are moderately sensitive to the initial positions of the Voronoi points (particularly for areas with lower population density gradients). The Voronoi model's greatest strengths and greatest weaknesses both revolve around stability. The model is not guaranteed to converge to a stable district configuration. However, in every real-world situation for which we tested it, it produced stable district borders. We believe that its great testing success easily outweighs any uncertainty associated with its use.

[Editor's Note: This Meritorious paper won the Ben Fusaro Award for the Gerrymandering Problem. The full text of the paper does not appear in this issue of the *Journal*.]

# Why Weight? Moment A Cluster-Theoretic Approach to Political Districting

Sam Whittle
Wesley Essig
Nathaniel S. Bottman
University of Washington
Seattle, WA

Advisor: Anne Greenbaum

## Summary

Political districting has been a contentious issues in American politics over the last two centuries. Since the landmark case of *Baker v. Carr* (1962), in which the U.S. Supreme Court ruled that the constitutionality of a state's legislated districting is within the jurisdiction of a federal court, academics have attempted to produce a rigorous system for districting a state.

We propose both a modified form of classical K-means clustering and the shortest-splitline algorithm to accomplish impartial redistricting. We apply our methods to redistricting New York, and, as further examples, Texas and Colorado. Both methods use only population-density data and state boundaries as inputs and run in a feasible amount of time.

Our criteria for successful redistricting include contiguity, compactness, and sufficiently uniform population.

The K-means method produces districts similar to convex polygons, and the splitline method guarantees that the resulting districts have piecewise linear boundaries. The K-means method has the advantage of allowing seeding of the district centers. The centers of the generated districts then roughly correlate to the existing districts, by proper seeding, but the resulting boundaries are vastly simpler.

The text of this paper appears on pp. 301–313.

# Applying Voronoi Diagrams to the Redistricting Problem

Lukas Svec
Sam Burden
Aaron Dilley
University of Washington
Seattle, WA

Advisor: James Allen Morrow

## Summary

Gerrymandering is an issue plaguing legislative redistricting. We present a novel approach to redistricting that uses *Voronoi* and population-weighted *Voronoiesque* diagrams. Voronoi regions are contiguous, compact, and simple to generate. Regions drawn with Voronoiesque diagrams attain small population variance and relative geometric simplicity.

As a concrete example, we apply our methods to partition New York State. Since New York must be divided into 29 legislative districts, each receives roughly 3.44% of the population. Our Voronoiesque diagram method generates districts with an average population of $(3.34 \pm 0.74)\%$.

We discuss possible refinements that might result in smaller population variation while maintaining the simplicity of the regions and objectivity of the method.

The text of this paper appears on pp. 315–332.

# Boarding at the Speed of Flight

Michael Bauer
Kshipra Bhawalkar
Matthew Edwards
Duke University
Durham, NC

Advisor: Anne Catlla

## Summary

We seek an efficient method for boarding a commercial airplane that accommodates unpredictable human behavior, with a framework that allows us to compare and contrast different procedures. Passenger dependencies, bottlenecks, and the rate of interferences are critical factors for airplane boarding time.

Boarding without seating assignments is fastest, since each person is in the correct order for their flexible seat choice; it removes all interferences and makes the boarding time depend solely on the entrance rate of passengers into the plane. Hoping to emulate the performance of this method, which we call "random greedy," we design a new algorithm to model its average seating order: the parabola boarding scheme, which breaks the plane into parabola-shaped zones.

We use a discrete-time simulation engine to model current boarding schemes as well as the parabola and random greedy algorithms. The zone-boarding schemes outside-in, pyramid, and parabola are almost identical in performance; back-to-front and alternating rows are significantly worse.

We examine the effects of scheme-independent parameters on boarding time. Ensuring a fast rate of people entering the plane and fast luggage stowage are both critical; an airline could reduce boarding time by improving either of these regardless of boarding scheme.

By varying both the rate of people entering the plane and time to stow luggage, we find a correlation between average boarding time and the difference between best and worst times. The random greedy algorithm has the smallest difference; outside-in, pyramid, and parabola have equal differences. Faster boarding algorithms are also more reliable and allow for tighter scheduling.

The best boarding algorithms do not have assigned seating. If, however, an airline feels that assigned seating is mandatory for customer satisfaction, then any of outside-in, pyramid, or parabola will result in a consistently fast boarding time with minimum deviation from average times and will be a marked improvement over the traditional back-to-front boarding method.

The text of this paper appears on pp. 333–352.

# Novel Approaches to Airplane Boarding

Qianwei Li
Arnav Mehta
Aaron Wise
Duke University
Durham, NC

Advisor: Owen Astrachan

## Summary

Prolonged boarding not only degrades customers' perceptions of quality but also affects total airplane turnaround time and therefore airline efficiency [Van Landeghem 2002].

The typical airline uses a zone system, where passengers board the plane from back to front in several groups. The efficiency of the zone system has come into question with the introduction and success of the open-seating policy of Southwest Airlines.

We use a stochastic agent-based simulation of boarding to explore novel boarding techniques. Our model organizes the aircraft into discrete units called "processors." Each processor corresponds to a physical row of the aircraft. Passengers enter the plane and are moved through the aircraft based on the functionality of these processors. During each cycle of the simulation, each row (processor) can execute a single operation. Operations accomplish functions such as moving passengers to the next row, stowing luggage, or seating passengers. The processor model tells us, from an initial ordering of passengers in a queue, how long the plane will take to board, and produces a grid detailing the chronology of passenger seating.

We extend our processor model with a genetic algorithm to search the space of passenger configurations for innovative and effective patterns. This algorithm employs the biological techniques of mutation and crossover to seek locally optimal solutions.

We also integrate a Markov-chain model of passenger preference into our processor model, to produce a simulation of Southwest-style boarding, where seats are not assigned but are chosen by individuals based on environmental constraints (such as seat availability).

We validate our model using tests for rigor in both robustness and sensitivity. Our model makes predictions that correlate well with empirical evidence.

We simulate many different a priori configurations, such as back-to-front, window-to-aisle, and alternate half-rows. When normalized to a random

boarding sequence, window-to-aisle—the best-performing pattern—improves efficiency by 36% on average. Even more surprising, the most common technique, zone boarding, performs even worse than random.

We recommend a hybrid boarding process: a combination of window-to-aisle and alternate half-rows. This technique is a three-zone process, like window-to-aisle, but it allows family units to board first, simultaneously with window-seat passengers.

The text of this paper appears on pp. 353–370.

# STAR: (Saving Time, Adding Revenues) Boarding/Deboarding Strategy

Bo Yuan
Jianfei Yin
Mafa Wang
National University of Defense Technology
Changsha, China

Advisor: Yi Wu

## Summary

Our goal is a strategy to minimize boarding / deboarding time.

- We develop a theoretical model to give a rough estimate of airplane boarding time considering the main factors that may cause boarding delay.
- We formulate a simulation model based on cellular automata and apply it to different sizes of aircraft. We conclude that outside-in is optimal among current boarding strategies in both minimizing boarding time (23–27 min) and simplicity to operate. Our simulation results agree well with theoretical estimates.
- We design a luggage distribution control strategy that assigns row numbers to passengers according to the amount of luggage that they carry onto the plane. Our simulation results show that the strategy can save about 3 min.
- We build a flexible deboarding simulation model and fashion a new inside-out deboarding strategy.
- A 95% confidence interval for boarding time under our strategy has a half-width of 1 min.

We also do sensitivity analyses of the occupancy of the plane and of passengers taking the wrong seats, which show that our model is robust.

The text of this paper appears on pp. 371–384.

# The Unique Best Boarding Plan? It Depends...

Bolun Liu
Xuan Hou
Hao Wang
National University of Singapore

Advisor: Yannis Yatracos

## Summary

We devise and compare strategies for boarding and deboarding planes of varying capacity. We clarify what properties a good strategy should have. We apply the same assumptions regarding basic boarding procedure, inner structure of planes, and behavior of passengers to all the cases.

For boarding, we study prevailing strategies and a seemingly excellent strategy, seat–by–seat, proposed in past literature, and categorize them into two types, assigned–seating and open–seating. We develop a model and a simulation for each type . Our criteria identify two good candidates, reverse-pyramid and open-seating. We develop our own comprehensive strategy, simulate it, and compare it with those two. However, the optimal boarding strategy is not the same for different planes. Some values of parameters, such as the passengers' luggage size and weight, greatly influence the final result. Based on these discoveries, we suggest how to modify a boarding procedure in practice to make it optimal.

For deboarding, a simple strategy beats a complicated one; but we still give a theoretically optimal model, then modify it to achieve a concise strategy applicable in practice.

The text of this paper appears on pp. 385–404.

# Airliner Boarding and Deplaning Strategy

Linbo Zhao
Fan Zhou
Guozhen Wang
Peking University
Beijing, China

Advisor: Xufeng Liu

## Summary

To reduce airliner boarding and deplaning time, we partition passengers into groups that board in an arranged sequence. We assume that first-class and business-class passengers board first; our model treats only economy class. Since deplaning is the converse process of boarding, a strategy for boarding gives a strategy for deplaning.

We develop a model of interferences among passengers, which determine boarding time. We try to find a strategy with the least interferences. By running Lingo, we tackle the resulting nonlinear integer programming problem and obtain near-optimal strategies for fixed numbers of groups. This model supports the outside-in and reverse-pyramid strategies.

We develop another model to give a global lower bound for interferences. We also prove that individual boarding sequence, which boards passengers one by one in a particular order, attains that lower bound.

We develop code in C++ to simulate boarding strategies and test various strategies for three airliners: Canadair CRJ-200 (small), Airbus A320 (midsize) and Airbus A380 (large). Individual boarding sequence, reverse-pyramid, and outside-in are the best three strategies in terms of both average boarding time and its standard deviation.

We test strategies under various luggage loads and levels of occupancy, with and without late passengers and those with special needs. Outside-in and reverse-pyramid are stable under variation of parameters, whereas individual boarding sequence is extremely sensitive, though not to luggage.

Our conclusions discredit traditional back-to-front strategies and support individual boarding sequence, reverse-pyramid, and outside-in. The more groups, the worse the situation with back-to-front. Taking cost into consideration, random sequencing should also be recommended.

Finally, we analyze deplaning and see how its time can be minimized.

The text of this paper appears on pp. 405–420.

# Loading and Unloading Passenger Airliners: A Simulation Approach

Walter Jacob
Joshua Dunn
Matthew Oster
Rowan University
Glassboro, NJ

Advisor: Hieu D. Nguyen

## Summary

Grounded planes cost airlines money. A major factor in determining the grounding time of an airliner is the time that it takes to board passengers. An optimal boarding method would therefore reduce costs to the airlines and maximize profits by reducing the time the plane has to be on the ground and also enabling the airlines to offer more flights.

Our assumptions were made with the real-world situation in mind. The result is a model that behaves well and parallels the results of other contemporary research efforts. The considerations upon which our constraints are based reflect the deterministic nature of the model.

We performed a series of empirical tests to obtain acceptable ranges for parameters such as passenger walking speed and time required to stow carry-on luggage in an overhead compartment. Four different seating methods were tested: open seating, back to front seating, Wilma, and our own modified reverse pyramid seating.

Although each method has its own benefits, we concluded that Wilma outperformed the competing methods for the widest number of configurations. In our simulations Wilma offered an average decrease of 1.8 min in small planes, 5.1 min in medium planes, and 2.6 min in large planes. Our model performed very well with the tested scenarios and scales easily to cover other situations.

[Editor's Note: This Meritorious paper won the Ben Fusaro Award for the Airplane Seating Problem. The full text of the paper does not appear in this issue of the *Journal*.]

# Best Boarding uses Buffers

Kevin D. Sobczak
Eric J. Hardin
Bradley J. Kirkwood
Slippery Rock University
Slippery Rock, PA

Advisor: Athula R. Herat

## Summary

By constructing a mathematical model of human behavior, we find:

- **Back-to-front block loading is the least efficient boarding method.** As passengers enter the aircraft in groups, aisle congestion becomes greatest at the front of the plane, consequently increasing the time required for the next group to enter and take their seats. Aisle congestion in this case is primarily attributed to the time for a passenger to navigate the aisle and reach the assigned seat if obstructed by another passenger sitting in the same row.
- **Small planes and large planes exhibit minimal turnaround times.** Small planes have a single aisle but few passengers, hence little congestion. In large planes, multiple aisles and decks offset the congestion found in single-aisle midsize planes; a large plane can be modeled as several small planes.
- **Boarding strategies are optimized when 10% of the passengers are late.** Fewer passengers enter initially, so there is less congestion. When passengers enter late, congestion that would otherwise have occurred is averted.

Our first observation concurs with researchers who suggest abandoning back-to-front boarding in favor of more-elaborate schemes [Finney 2006; van den Briel et al. 2004; Ferrari and Nagel 2005]; however, these new models make erroneous assumptions about human behavior. A comprehensive scheme must include the time to navigate a congested aisle, stow luggage, and maneuver through a filled row if necessary. We recommend the following:

- **Abandon back-to-front block boarding and consider alternatives.** We suggest a hybrid group-boarding method utilizing a rotating seating arrangement that incorporates back-to-front and window-to-aisle seating.
- **Incorporate a second aisle into midsize aircraft.**
- **Reduce carry-on luggage.**
- **Queue passengers into lines prior to gangway entry.**

The text of this paper appears on pp. 421–434.

# Modeling Airplane Boarding Procedures

Bach Ha
Daniel Matheny
Spencer Tipping
Truman State University
Kirksville, MO

Advisor: Steven J. Smith

## Summary

We describe two models that simulate the process of passengers boarding an an aircraft and taking their seats. Using these models, we simulate common boarding procedures on popular aircraft to analyze efficiency. The second model is more ambitious and tries to model the situation more accurately, but even the first one addresses the major problems involved in boarding an airplane.

From running the simulations and analyzing the data, we find that the fastest and most consistent procedures are outside-in and reverse-pyramid. Both allow those closest to the windows to be seated first and proceed inward (though reverse-pyramid is slightly more complex). Reverse-pyramid is slightly faster.

The text of this paper appears on pp. 435–450.

# American Airlines' Next Top Model

Sara J. Beck
Spencer D. K'Burg
Alex B. Twist
University of Puget Sound
Tacoma, WA

Advisor: Michael Z. Spivey

## Summary

We design a simulation that replicates the behavior of passengers boarding airplanes of different sizes according to procedures currently implemented, as well as a plan not currently in use. Variables in our model are deterministic or stochastic and include walking time, stowage time, and seating time. Boarding delays are measured as the sum of these variables. We physically model and observe common interactions to accurately reflect boarding time.

We run 500 simulations for various combinations of airplane sizes and boarding plans. We analyze the sensitivity of each boarding algorithm, as well as the passenger movement algorithm, for a wide range of plane sizes and configurations. We use the simulation results to compare the effectiveness of the boarding plans. We find that for all plane sizes, the novel boarding plan Roller Coaster is the most efficient. The Roller Coaster algorithm essentially modifies the outside-in boarding method. The passengers line up before they board the plane and then board the plane by letter group. This allows most interferences to be avoided. It loads a small plane 67% faster than the next best option, a midsize plane 37% faster than the next best option, and a large plane 35% faster than the next best option.

The text of this paper appears on pp. 451–462.

# Boarding—Step by Step: A Cellular Automaton Approach to Optimising Aircraft Boarding Time

Chris Rohwer
Andreas Hafver
Louise Viljoen
University of Stellenbosch
Stellenbosch, South Africa

Advisor: Jan H. van Vuuren

## Summary

We model the boarding time for the aircraft using a cellular automaton. We investigate possible solutions and present recommendations about effective implementation.

The cellular automaton model is implemented in three stages:

- Initialisation of the seating layout for a chosen aircraft type and assignment of seats to passengers
- The sorting of passengers according to various proposed boarding methods
- "Propagating" the passengers through the aisle(s) of the aircraft and seating them at their assigned places.

The rules governing the automaton take into account various factors. Among these are the load factor (percentage filled) of the craft, different walking speeds of passengers walking through the aisle, and time delays from stowing luggage and obstructions by other passengers during the seating process. The algorithm accommodates predefined aircraft layouts of common aircraft and also user-defined aircraft layouts.

We modeled and tested various boarding strategies for efficiency with regard to total boarding time and average boarding time per passenger. Thus, our approach focuses not only on optimisation of the process in favour of the airlines, but also yields information regarding convenience to passengers. Random boarding (where passengers with assigned seat numbers enter the plane in a random sequence) was used as a point of reference. Among other strategies tested were boarding the plane in groups from either end, boarding from seats farthest from the aisles toward the aisles, and combinations of these approaches.

We conclude that boarding strategies starting farthest away from the entrance or farthest away from the aisles yield shorter boarding times than random boarding. The most successful methods are combinations of these strategies, their detailed implementation depending on the exact layout/size of the aircraft. The method yielding the shortest total boarding time is not necessarily the one with shortest average boarding time per passenger. By considering standard deviations of total and individual boarding times over many iterations of the simulation, we can derive conclusions regarding the stability/consistency of the specific boarding strategies and how evenly the waiting time is distributed amongst the passengers.

By selecting appropriate strategies, time savings of 2–3 min for small and medium aircraft could be achieved. For a custom 800-seat aircraft with two aisles, more than 6 min could be saved compared to random boarding. Having compared our results to actual turnaround times quoted by airlines, we believe them to be realistic.

The text of this paper appears on pp. 463–478.

# When Topologists Are Politicians...

Nikifor C. Bliznashki
Aaron Pollack
Russell Posner
Duke University
Durham, NC

Advisor: David Kraines

## Summary

According to Supreme Court Justice William Brennan, former Supreme Court Justice Sandra Day O'Connor once noted that any politician who did not do everything to secure power for the party "ought to be impeached" [Toobin 2003]. Though Congress argues that wild election districts such as "a pair of earmuffs" are inherently fair and reasonable, they are so counterintuitive that such claims can be exceedingly difficult to believe.

Defining the big picture of what is "fair" can be left to philosophers—or computers. Using a novel method, we divide states into districts of equal population, with each district as compact and elementary as possible, where compactness is defined as the moment of inertia of the district with respect to the population density. By not examining any other demographic data in the grouping, we avoid many of the biases that people may impose. We obtain districts considerably more compact than current congressional districts for Ohio and New York.

Since it is constitutionally sound to group people into congressional districts by "shared interests," we extend the problem by allowing other demographic data to be considered in the formation of such districts. to form revised districts that seek to preserve uniformity of these qualities. We identify how suitable these solutions are and determine their advantages and disadvantages.

Finally, we propose alternative districting techniques that take into account county boundaries and natural boundaries.

---

*The UMAP Journal* 28 (3) (2007) 249–259. 

# Introduction

When topologists are politicians . . . the dimension of New York's 13th congressional district might not even be an integer. Districting is usually handled by a political and partisan body, such as the state legislature or the governor. Unsurprisingly, partisans district to maximize the number of representatives of their party will in Congress. This process is called *gerrymandering*, after Elbridge Gerry, governor of Massachusetts in 1812, who famously approved a congressional district that resembled a salamander.

Gerry's salamander of 1812 could not even hold a candle to such well-known districts of today such as Louisiana's "the 'Z' with drips," or Pennsylvania's "supine seahorse" and "upside-down Chinese dragon" districts. Such awkward and complex districts can lose sight of the primary goal of the House of Representatives as outlined in the U.S. Constitution: to provide regional representation to the people. We seek a "fair" and "simple" districting that maximizes accessibility of all people to regional representation, while providing a partitioning insensitive to partisan motives.

We define an objective function $F$ whose minimum gives what we define to be the best districting. We apply this method to New York and to Ohio.

# Definitions

- **Block:** A unit of area that corresponds to a fixed number of people. Since population densities vary, block sizes vary. A block is marked in the plane by a pair of $(x, y)$ coordinates.

- **District:** A collection of a fixed number of blocks (thus having a constant population).

- **Capitol:** The average of the coordinates of each block in the district, an approximation of the center of its population.

- **Fairness:** In *Shaw v. Reno* [1993], the U.S. Supreme Court mentioned that acceptable ways of districting a state include "compactness, contiguity, [and] respect for political subdivisions or communities defined by actual shared interests." By compactness, the justices were alluding to a vague notion that congressional districts should be more like squares or circles than "spitting amoebas" (read: Maryland's Third District).

- **Compactness:** Suppose that a district $D$ contains $n$ blocks, $z_1, \ldots, z_n$, with capitol $c$. Compactness $C$ is the variance of the spatial distribution of the population:

$$C = \sum_{i=1}^{n} \|z_i - c\|^2.$$

When $C$ is small, we conclude that the district's constituents live within a relatively small area.

- **Shared Interests** ($S$)**:** We assign a vector to each block, giving one component to each interest, and minimize the sum of the variances of the components over the district. That is, given vectors $v_i$ associated with blocks $z_i$ and mean interest vector $\vec{\mu} = \frac{1}{n}\sum_{i=1}^{n} \vec{v}_i$, the shared interest is

$$S = \sum_{i=1}^{n} \|\vec{v}_i - \vec{\mu}\|^2.$$

## A Note on Shared Interest

Though race, gender, age, and religion are important issues for many, legal ambiguities exist in the use of such measures. Because the benefit is yet unproven of either grouping together or dispersing such groups among congressional districts, we do not use these data. Though measuring political affiliation is an entirely legal and often implemented districting tool, we remain nonpartisan to avoid inadvertently favoring one party over another.

# Specific Formulation of the Problem

We measure fairness of a partition of a state into congressional districts. Let $P = \{D_1, \dots, D_k\}$ be a partition of blocks into congressional districts, such that each district is contiguous and has the same number of blocks. We define

$$f(D_i) = w_1 C_i + w_2 S_i,$$

where $w_1$ and $w_2$ are positive weights that are the same for all districts. We define globally

$$F(P) = \sum_{i}^{k} f(D_j).$$

We seek a partition that minimizes $F$.

# Assumptions

- We have accurate data about a state's population, geographical layout, and other relevant factors.
- The population represented by one block is small enough to ensure that districts have negligibly different numbers of people.
- The initial assignment of blocks to districts is random enough to assure that the districting to which our algorithm converges is near the global minimum.

# Background and Goals

Weaver and Hess [1963] set the standard for computerized nonpartisan districting. Using integer programming methods, a set of capitols (called "LDs" in their paper) were matched with blocks ("EDs") so as to minimize moment of inertia, i.e., our $C$. Repeatedly, the LDs were relocated to the appropriate centers of mass and then the EDs were redistributed to each LD until the moment of inertia hit a local minimum. By repeating many times with a large number of initial conditions, they hoped to approximate the global minimum and thus derive the most compact districting. Though precise, integer programming algorithms on large sets of data are extremely time-consuming. Though Weaver and Hess's methods found applicability at the county and small state level at best, their landmark work paved the way for the development of a variety of approaches.

We create a model that expands on theirs with the following goals:

- Find the ideal partition $P^*$, i.e., the one that minimizes $F$ globally.
- The method to find $P^*$ should be versatile—able to find the ideal partition for a wide variety of shared interest functions $S$.
- The method should be scalable—able to handle large quantities of data quickly.

# Friendly Trader Method

Our method starts with an initial arrangement of blocks into districts and moves blocks between districts to decrease $F$. Let the blocks be arranged into $n$ districts. By our method, district $D$ attempts to trade blocks to reduce its $f(D)$. However, our districts are "friendly traders"—they conduct only trades that make the districts as a whole better off (i.e., reduces $F$). Our districts are so friendly, in fact, that they will execute trades that raise $f(D)$ so long as $F$ decreases. Since the composition of our districts changes after each trade, capitols must be recalculated at each step. The problem of finding a minimum then reduces to finding and executing all trades that reduce $F$ until no more exist.

## How Are Blocks Determined?

We obtained demographic data are obtained from the 2000 U.S. Census [U.S. Census Bureau n.d.]. New York State is partitioned into roughly 5,000 tracts, each with a specific population and coordinates in latitude and longitude. For each minor civil division, we assign one block per 250 people, rounding population to the nearest 250. We spread these blocks evenly within their minor civil division. Thus, each block has the same population, and population density corresponds to block density.

## How Are Districts Initialized?

We try to devise an initial partition with low $F$. First, we arbitrarily choose 29 blocks to be district capitols. Each capitol's position is assumed to be the center of population and its interests the mean interests of the district. One by one, each capitol picks districts that "fit well" with the capitol's location and interests. The process is like a professional sports draft, where teams take turns picking players who suit each team well. After all the blocks are assigned to capitols, trading of blocks begins.

## How Do We Maximize Compactness?

Let $D_1$ and $D_2$ be two districts and $b_k$ a block in $D_1$. By moving $b_k$ from $D_1$ to $D_2$, we form districts $D_1'$ and $D_2'$. Define

$$\Delta F(D_1, D_2, b_k) = f(D_1') + f(D_2') - f(D_1) - f(D_2).$$

To determine which trades to make, we first find a block $b*_{12}$ in $D_1$ such that $\Delta F(D_1, D_2, b*_{12}) \leq \Delta F(D_1, D_2, b_k)$ for all blocks $b_k$ in $D_1$. Let us call $b*_{12}$ the *best block* from $D_1$ to $D_2$.

Now we can define a fully connected directed graph $G$, with the vertices of $G$ the districts $D_j$ and the edge $v_i \rightarrow v_j$ having length $\Delta F(D_i, D_j, b*_{ij})$, where $b*_{ij}$ is the best block from $D_i$ to $D_j$, where the length is negative if the trade decreases $F$. To reduce $F$ by trades, we search for cycles of negative length in $G$. (The length of a cycle is the sum of the lengths of the edges composing the cycle, counting multiplicity if an edge appears more than once.) A cycle with negative length corresponds to a group of trades that reduce $F$. Hence, finding good trades of blocks between districts reduces to finding cycles of negative length in the digraph.

That problem in turn reduces to the simpler problem of finding a shortest path between any two vertices—that is, a path of minimum length in which no edge is used more than once. The Bellman-Ford-Moore algorithm modifies a standard shortest-path algorithm to find any negative cycle [Cherkassy and Goldberg 1999]. (This algorithm finds only the *first* negative cycle encountered and thus gives no choice of cycle.) Once a trade is found, it is completed and capitols are recalculated. We reject any trade that, after recalculation of capitols, actually increases $F$. Since we are making only trades that strictly reduce $F$, when $F$ can no longer decrease through trading, we have achieved a local minimum.

# Results

We implemented the algorithm in a computer program and simulated the process for New York and for Ohio. No matter the starting configuration for the state, we always ended up with the same shape for the districts, making us believe that these data sets are big and diverse enough to converge always to a unique global minimum.

**Figure 1** shows our apportionment of New York, calculated to make the regions most compact ($w_2 = 0$). We then redid New York using compactness as a guide but with the objective function $F$ weighted toward preserving population density. The results are shown in **Figure 2**.

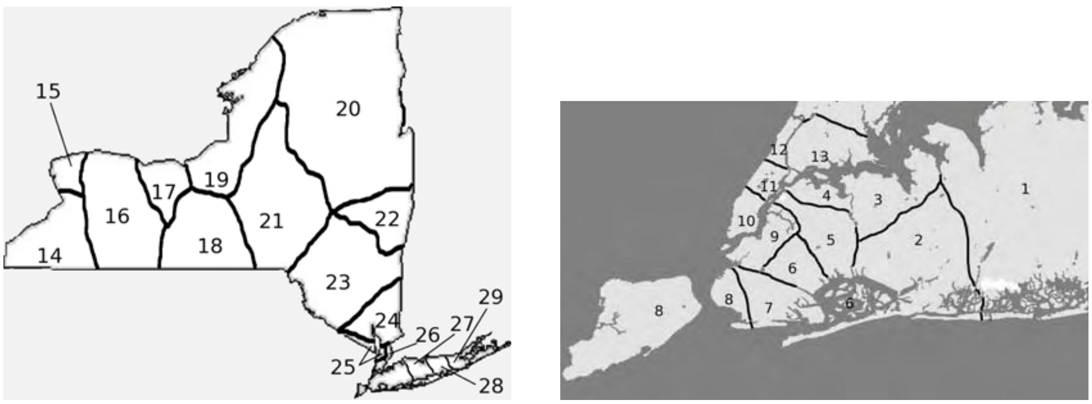



**Figure 1.** The most compact apportionment of New York, with close-up of New York City on right.

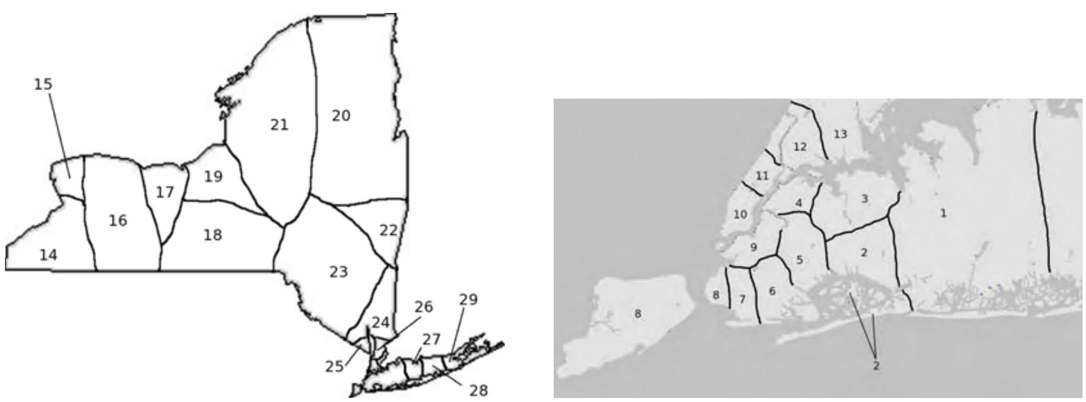



**Figure 2.** Apportionment of New York based both on compactness and population density.

We also tested our algorithm on Ohio to check that our method is applicable in other circumstances. In **Figure 3**, Ohio is partitioned using only compactness as a guide; **Figure 4** use the same function $F$ as **Figure 2**.

In Ohio, congressional districts are designed with preserving county lines in mind. In **Figure 6**, we attempt not to split counties between congressional districts. We thus add a term in $f(x)$ to take into account county separation. This idea can easily be extended to natural boundaries such as rivers and highways.

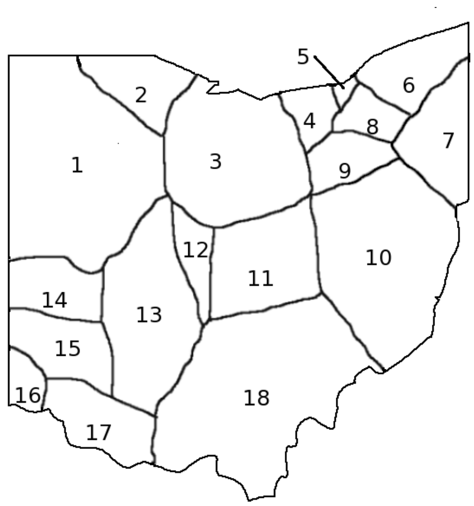



**Figure 3.** Districting of Ohio, based solely on compactness.

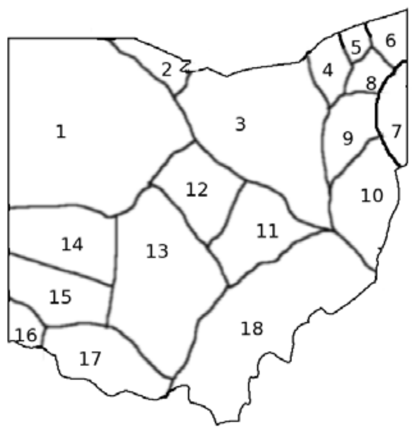



**Figure 4.** Districting of Ohio, based on both compactness and population density.


**Figure 6.** Districting of Ohio, based solely on preserving county boundaries.

# Analysis of Results

Our results are primarily visual and not numerical. One weakness of our method is the lack of quantifiable data for comparison. Since $F$ itself can be varied, normalizing it for use from one application to another is far from trivial. In addition, the remarkable reproducibility of our results given a wide variety of initial conditions almost entirely eliminates the need for separate numerical results.

In **Figures 2** and **4**, after the sorting of blocks, we connected a line surrounding all boundary blocks in a district and softened the line to make it smooth. Compared to current congressional districts, those produced by our algorithm are a vast improvement in simplicity, corresponding to a reduction in $C$ by a factor of 7 for Ohio and by a factor of about 22 for New York. Since our model evaluates districting through distance, it promotes star-shaped districts (in which the capitol is connected to every point in the district by a straight line) rather than fully concave districts, improving accessibility and reducing complexity of the districts. Our model handles the problems of districting New York and Ohio wonderfully, achieving very simple, reasonable results.

However, interest-weighted models are where our model really shines (**Figures 3** and **5**). Since we chose not to gauge common interest by controversial factors such as race or age, we chose the tamest quantity possible: population density at each block. We felt that this quantity would be useful to group districts by, since urban issues tend to differ from rural issues, and thus both city slickers and farmers alike could obtain representation for their grievances.

In some ways, our model already favors uniform population density across districts. Since blocks in urban areas are more densely packed, they naturally migrate to the same district. Our adjustment then merely increases the population density component, producing a noticeable change for Ohio, with tightened districts around the major cities of Columbus in the center, Cincinnati in the southwest, and Cleveland in the northeast, as well as a tightening of the districts around the densely populated Bronx and Queens in New York City. These solutions improve compactness and uniformity of population density compared to current congressional districting (quite unsurprisingly); they also demonstrate the existence of a range of reasonable solutions that satisfy the goals of compactness, simplicity, and fairness.

Since many states have independent county governments (including New York and Ohio), we ran an alternative solution set for Ohio in which we tried to preserve county lines. Our mean interest vector $\vec{\mu}$ assigned a direction for each county tabulating the number of blocks in each. We then weighted our solution with the goal of preserving county lines. **Figure 6** shows the great success of this solution. In most districts, boundaries coincide almost perfectly with county lines. The advantage of the county-based districting solution is clear: Since citizens pay taxes to and receive services from county governments, allowing counties to have exclusive congressional representation allows for easier handling of issues on the local level. Surprisingly, such lines can be taken into consideration with little consequence on compactness or simplicity.

# Why Our Model is Fair

The question of fairness is much more difficult. To give an example of this challenge, we consider the Fourteenth Amendment to the U.S. Constitution, which states that all races are strictly equal under the law. However, the Voting Rights Act of 1965 states that the government will assist in facilitating the voting of minority areas. Thus, even the government itself has trouble deciding whether "fair" involves helping the often-disadvantaged to realize their rights or involves giving every person exactly the same treatment. We argue that our model is fair because it remains passive and uninvolved. It only takes a set of directives (i.e., the function $F$) and produces a solution that divides the region into relatively uniform districts with respect to $F$. If nonpartisan goals are incorporated, such as uniform population density or compactness, a nonpartisan solution arises.

However, any component of data can be (and likely has been) misused. For example, African Americans tend to be affiliated with the Democratic Party, while those in rural areas tend to be Republican. Thus, race and population density can be delicately used to achieve partisan aims. Our model is fair because it assigns no judgment to any of these considerations. Rather, it is designed to improve the citizen's accessibility to attentive and diligent representation in order to maximize every individual's rights and powers.

# Strengths and Weaknesses of Model

Our model achieves all of the initial goals. It is fast, and could handle large quantities of data, but also has flexibility. Though we did not test all possibilities, we showed that our model optimizes state districts for any of a number of variables. If we had input income, poverty, crime, or education data into our interest function, we could have produced high-quality results with virtually no added difficulty. As well, our method is robust. Moreover, we can divide areas into fairly simple, contiguous, and uniform regions. Our model also consistently leads to useful minima.

The primary weakness of our model is the absence of good nicknames for our districts—somehow districts such as "egg" and "sort of diamond-shaped thing" don't raise any excitement.

Though we achieved solid equilibria, our model in no way guarantees that it will ever find a global solution. To see this, consider a rectangle with different sides, and assume that we have 10 points at each of its vertices. Moreover, let the districts initially be the long sides. It is easy to check that no trade will occur, and thus this configuration is a local, but not a global, minimum.

The other primary weakness of our model is our lack of metrics for comparison. Though compactness and shared interest levels are appropriate measures for comparison of two models within a state, we lack invariant metrics for assessing the quality of one districting versus another.

# Food for Thought

Given the proper data, our model can do much more than merely political districting. At heart, it simply attempts to group regions into smaller parts, unified by whatever characteristics desired. For example, if a governing body wanted to determine where to build police stations or hospitals, it could input weighted crime, health, poverty, and/or age statistics into the model. The model could then partition the area into small regions united not only by spatial relations but also by needs and desires. Thus, our model could help politicians and authorities most effectively deploy public resources and services. Ironically, our nonpartisan partitioning method could be a politician's best friend! Of course, by inputting political affiliation data, politicians could identify partisan strongholds in order to plan campaigns.

# Conclusion

We set forth an algorithm to determine congressional districts, given data on location, population, and any other factors desired. The algorithm is intended to be fair, or nonpartisan, in stark contrast to the political process of gerrymandering. Characteristics that we consider to be fundamental in the division of a state into congressional districts include contiguousness, compactness, and shared interests or concerns among a district's citizens.

We assume that a state can be divided into blocks of small constant population and interpret the problem of congressional district apportionment as the distribution of these blocks to the districts so that each district contains a fixed number of blocks (and therefore all districts have the same population). Furthermore, we define an objective function $F$ that measures the quality of a distribution. Finding good partitions is equivalent to finding distributions with low values of $F$. Our goal, therefore, was to find a partition that minimizes $F$. This is a useful formulation of the problem because, if all agree to use this method beforehand, the existence of a global minimum of $F$ (our problem is finite) guarantees that if this minimum is found, there can (should) be no partisan squabbling as to the legitimacy of the solution obtained.

Admittedly, our algorithm is only guaranteed to find local minima of $F$. However, simulations with random initial starting values seem to converge to the same final apportionment, suggesting that the local minima that our algorithm finds are close to the global minima. Additionally, while we have implemented certain particulars to quantify shared interests of citizens in a district, our procedure for determining congressional districts is flexible; with a simple change in the particulars, it can partition blocks into districts under other criteria.

# References

Cherkassy, B.V., and A.V. Goldberg. 1999. Negative-cycle detection algorithms. *Mathematical Programming* 85: 277–311.

Toobin, Jeffrey. 2003. The great election grab. *New Yorker* 79 (38) (8 December 2003): 63–80.

*Shaw v. Reno*, 509 U.S. 630 (1993).

U.S. Census Bureau. n.d. FactFinder. `http://factfinder.census.gov/home/saff/main.html?_lang=en` .

Weaver, James B., and Sidney W. Hess. 1963. A procedure for nonpartisan districting: Development of computer techniques. *Yale Law Journal* 73: 288–308.

Pp. 261–450 can be found on the *Tools for Teaching* 2007 CD-ROM.

# What to Feed a Gerrymander

Ben Conlee
Abe Othman
Chris Yetter
Harvard University
Cambridge, MA

Advisor: Clifford H. Taubes

## Summary

Gerrymandering, the practice of dividing political districts into winding and unfair geometries, has a deleterious effect on democratic accountability and participation. Incumbent politicians have an incentive to create districts to their advantage (California in 2000, Texas in 2003); so one proposed remedy for gerrymandering is to adopt an objective, possibly computerized, methodology for districting.

We present two efficient algorithms for solving the districting problem by modeling it as a Markov decision process that rewards traditional measures of district "goodness": equality of population, continuity, preservation of county lines, and compactness of shape. Our Multi-Seeded Growth Model simulates the creation of a fixed number of districts for an arbitrary geography by "planting seeds" for districts and specifying particular growth rules. The result of this process is refined in our Partition Optimization Model, which uses stochastic domain hill-climbing to make small changes in district lines to improve goodness. We include as an extension an optimization to minimize projected inequality in district populations between redistrictings.

As a case study, we implement our models to create an unbiased, geographically simple districting of New York using tract-level data from the 2000 Census.

## What is Gerrymandering?

*Gerrymandering* is division of an area into political districts that give advantage to one group. Frequently, this involves concentrating "unfavorable"

*The UMAP Journal* 28 (3) (2007) 261–280. 

voters in a few districts to ensure that "favorable" voters will win in many more districts. To squeeze unfavorable voters into a few districts, gerrymandering creates snaky and odd shaped regions. The eponymous label was created when politician Elbridge Gerry pioneered this technique in early 19th century and his opponents claimed that the districts resembled salamanders (**Figure 1**).


**Figure 1.** The original "Gerry-mander" from the *Boston Centinel* (1812). Source: Wikipedia [2007], which in turn was cropped from U.S. Department of the Interior [2007].

## Basic Terminology

- **Packing:** Forcing a disproportionately high concentration of a particular group into one district to lessen their impact in nearby districts.
- **Cracking:** Spreading members of a group into several districts to reduce their impact in each of these districts.
- **Forfeit district:** A district where group $A$ packs the members of group $B$ so that group $B$ wins this district but loses several surrounding districts that $B$ might win with a different districting scheme.
- **Wasted Vote:** A vote cast by a member of group $A$ in a district where $A$ is already assured victory, so voting has no bearing on the result. In general, the group with more wasted votes is made worse off by a districting plan.

## Why Is It So Bad?

Gerrymandering reduces electoral competition within districts, since cracking/packing makes elections uncompetitive. Further, incumbent representatives are in no real danger of losing elections, so they do not campaign vigorously, which can lead to lower voter turnout. Exacerbating the problem, incumbents' increased advantage means that they have less incentive to govern based on their constituents' interests, so democratic accountability and engagement mutually deteriorate.

Gerrymandering also presents the practical problem that it is difficult to explain to voters why district shapes are so labyrinthine. Some districts connect demographically-similar but geographically-distant regions using thin filaments (**Figure 2**). "Niceness" of district shape almost always takes a back seat to political and racial concerns. Example: In the 2000 California realignment, Democrats and Republicans united to design incumbent-favoring districts, which resulted in the re-election of all of the 153 incumbents in 2004. How can one argue that this is in voters' best interests?

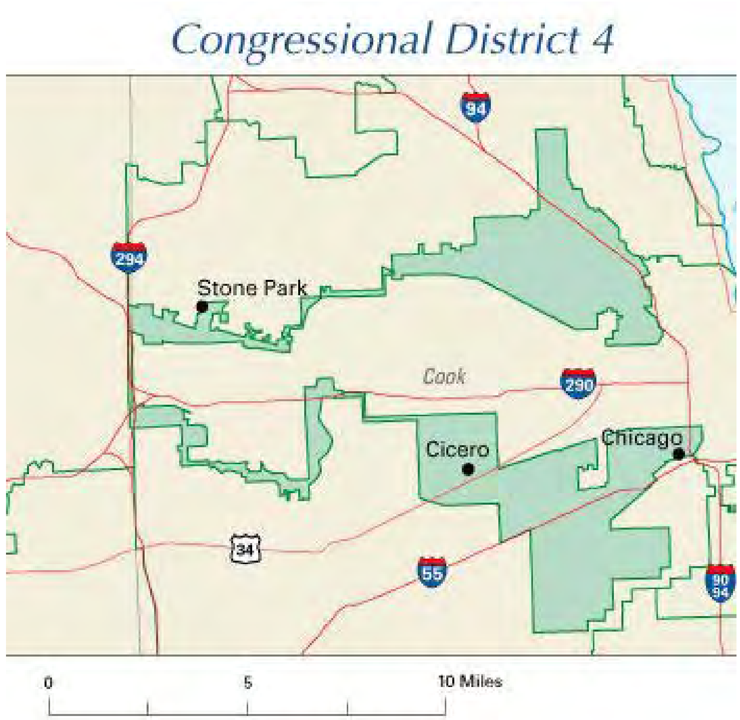



**Figure 2.** A present-day gerrymander, the Illinois 4th congressional district. The two "earmuffs" are connected by a narrow band along Interstate 294. Source: Wikipedia [2007], in turn cropped from U.S. Department of the Interior [2007].

However, gerrymandering can be considered appropriate in specific situations. For instance, the Arizona Legislature gerrymandered a division between the historically hostile Hopi and Navajo tribes even though the Hopi reservation is entirely surrounded by the Navajo reservation.

## The Legality of Gerrymandering

Though gerrymandering is objectionable to many, it is legal. The Voting Rights Act of 1965, which eliminated poll taxes and other discriminatory voting policies, may have inadvertently increased gerrymandering. One interpretation of the Act is that it mandates nondiscriminatory election *results*, which has led to a strange reversal of vocabulary in which creating "majority-minority" districts is considered beneficial. These gerrymandered districts are packed with minorities to guarantee minority representation in Congress.

However, in *Shaw v. Reno* (1993), and later in *Miller v. Johnson* (1995), the Supreme Court ruled that racial/ethnic gerrymanders are unconstitutional. Nevertheless, *Hunt v. Cromartie* (1999) approved of a seemingly racial gerrymandering because the motivation was mostly partisan rather than racial. The recent case *League of United Latin American Citizens v. Perry* (June 2006) held that states are free to redistrict as often as they like so long as the redistrictings are not purely racially motivated.

# Assumptions and Notation

## What Can We Consider When Districting?

1. Population equality between districts (legally mandated)
2. Contiguity of districts (legally mandated, excepting islands)
3. Respect for legal boundaries (counties, city limits, townships)
4. Respect for natural geographic boundaries
5. Compactness of district shapes
6. Respect for human-made boundaries (highways, parks, etc.)
7. Respect for socioeconomic similarity of constituents
8. Similarity to past district boundaries
9. Partisan political concerns
10. Desire to make districts (un)competitive
11. Racial/ethnic concerns
12. Desire to protect (or unseat) incumbent politicians

In our model, we consider only the top seven factors. The case *SC State Conference of Branches v. Riley* (1982) ruled that past districts (factor 8) are a legitimate tool for creating new districts, but we ignore past districtings, since they are heavily biased by factors 9–12, all related to political or racial concerns.

## Geography and Similar Characteristics

The U.S. Census Bureau provides data on legal, natural, and human-made boundaries as well as socioeconomic similarity of regions. In each census, the United States population is divided at several levels of accuracy, the smallest of which are: *blocks* (40 people on average), *block groups* (1,500 people), and *tracts*

(4,500 people). We follow the practice in Young [1988] by districting based on tracts.

> Census tract boundaries normally follow visible features, but may follow governmental unit boundaries and other non-visible features, and they always nest within counties. Census tracts are designed to be relatively homogenous units with respect to population characteristics, economic status, and living conditions at the time the users established them.
> [Caliper Corporation n.d.]

For these reasons, we believe that tracts are acceptably small and homogenous to use as a base unit. Further, a tract is completely contained a county, so we can easily check whether or not a district breaks county integrity.

# Notation

Let $n$ be the number of districts and $m$ the number of census tracts. We denote districts by $D_i$, tracts by $T_l$ and the set of all tracts by $\Gamma = \{T_l\}_{1 \leq l \leq m}$, which we call a *state*. Denote the set of all districts at a particular time by $\Delta = \{D_i\}_{1 \leq j \leq n}$; we call this a *partition* of the state.

## Adjacency

Define the symmetric relation $T_p \sim T_q$ for tract pairs $(T_p, T_q)$ that are adjacent. Let $d(T_l)$ be the district to which $T_l$ belongs. We also naturally extend the definition of $d$ to sets of tracts.

Define the *neighbor set* of tract $T_l$ by $a_T(T_l) = \{T_p \in \Gamma | T_l \sim T_p\}$, all tracts neighboring $T_l$; and define $a_D(T_l) = d(a_T(T_l))$ to be the set of all districts containing neighbors of $T_l$. Every tract borders at least one other tract, so all $a_T(T_l)$ and $a_D(T_l)$ are nonempty.

## Borders

Let the *border* of district $D_i$ be $\partial D_i = \{T_l \in D_i | a_D(T_l) \neq \{D_i\}\}$, the set of tracts in $D_i$ adjacent to at least one district other than $D_i$. The *interior* of district $D_i$ is $I_i = D_i \setminus \partial D_i$, the set of tracts in $D_i$ whose neighbors are all in $D_i$. Let $m_i = |D_i|$ be the number of tracts in $D_i$ and $b_i = |\partial D_i|$ the number of tracts bordering $D_i$.

The *frontier* of $D_i$ is $F_i = \left(\cup_{T_l \in D_i} a_T(T_l)\right) \setminus D_i$, the set of tracts outside of $D_i$ that border the boundary tracts of $D_i$.

## Counties

We denote a county by $C_j$ and the set of all counties by $\Lambda$. Districts can (and often do) break county boundaries, but tracts are contained entirely within counties, so a county is a set of tracts. Districts are also sets of tracts, so we interpret $D_i \cap C_j$ as the set of tracts in both district $D_i$ and county $C_j$.

## Population

Let the population of the state be $P$ and let $\bar{p} = P/n$ be the optimal district size. We use the function $p(\cdot)$ to denote the population of an object; for instance, $p(T_l)$ and $p(C_j)$ are the populations of tract $T_l$ and county $C_j$, respectively. We use the shorthand $p_i = p(D_i)$ for the population of districts.

**Table 1** is a useful reference of these numerous definitions.

**Table 1.**
Variables and their meanings.

| Variable | Definition |
|---|---|
| $n$ | Number of congressional districts |
| $D_i$ | The $i$th district $(1 \le i \le n)$ |
| $\Delta$ | Set of all districts in a state, a *partition* |
| $m$ | Number of census tracts |
| $T_l$ | The $l$th tract in $(1 \le l \le m)$ |
| $\Gamma$ | Set of all tracts in a state |
| $d(T_l)$ | District to which tract $T_l$ belongs |
| $T_p \sim T_q$ | Tracts $T_p$ and $T_q$ are adjacent |
| $a_T(T_l)$ | Set of tracts adjacent to tract $T_l$ |
| $a_D(T_l)$ | Set of districts containing tracts neighboring $T_l$ |
| $\partial D_i$ | Border of $D_i$, tracts that neighbor another district |
| $I_i$ | Interior of $D_i$, tracts that do not neighbor another district |
| $m_i$ | Number of tracts in $D_i$ |
| $b_i$ | Number of tracts in $\partial D_i$ |
| $F_i$ | Set of all tracts outside of $D_i$ that border $\partial D_i$ |
| $C_j$ | The $j$th county |
| $c(T_l)$ | The county to which tract $T_l$ belongs |
| $c(D_i)$ | The set of counties containing district $D_i$ |
| $P$ | Total population of the state |
| $\bar{p}$ | Average population of a district |
| $p(\cdot)$ | Population of an arbitrary object |
| $p_i$ | Population of district $D_i$ |

# Past Models

Cirincione et al. [2000] judge the quality of a districting plan based on equal population, preservation of county integrity, and district area compactness. They require that district populations differ by no more than 1% from exact equality of number of constituents and point contiguity of a district. They construct districts by picking a random block group, then adding additional block groups to the new district until the population reaches $\bar{p}$. At this point, they repeat the process starting with a new random block group. Compactness is based on minimum bounding rectangles, and county integrity is encouraged by "randomly" selecting new block groups with a preference for block groups in counties already in the emerging district.

Mehrotra et al. [1998] and Garfinkel and Nemhauser [1970] implement a "branch-and-price" method in the optimization step. They first obtain a districting and then optimize over constraints such that population sizes are allowed to vary. In a final step, they split up population units to ensure population equality. They define compactness in a graph-theoretical manner, where connected nodes are adjacent tracts. They define the "center" of a district to be the tract with the smallest maximum distance to another other tract. They consider a district compact when sum of distances from each node to the center is small.

We do not use their measure, since it does not uniquely define the center of a graph, and (contrary to their claims) does allow for oddly-shaped districts, such as a district whose graph is a star-shaped tree with one tract in the center and many noncontiguous paths emanating from it. Such a tree structure is one a salient feature of gerrymandering.

We also do not use a "branch-and-price" method of optimization. Following suggestions of Nagel [1965] and Kaiser [1966], we employ a local search algorithm in which tracts are swapped between existing districts to maximize the objective function.

## Measuring Compactness

The notion of compactness of a planar region has no uniformly accepted definition. Young [1988] suggests that any reasonable measure of compactness should consider population units (census tracts in our case study) as indivisible but laments that no one measure seems to work well for all geographic configurations.

Young's measures include the maximum total perimeter, the relative height and width, and the moment of inertia of the district. All these fail to consider both perimeter and area simultaneously.

The Isoperimetric Theorem states that the quantity $A/P^2$, the ratio of the area $A$ of a planar region (not necessarily contiguous) to the square of its perimeter, is maximized at $1/4\pi$ when the region is circular. We define *compactness* of a region as the ratio $4\pi A/P^2$. This ratio is 1 for the circle, with higher values indicating greater compactness. The compactness of a square is $4\pi/16 \approx 0.785$, an upper bound for compactness of any rectangle.

This ratio is a good measure of "regularity" of a region. Specifically, any shear of factor $s$ applied to a circle decreases the compactness by a factor of $s$, and any concave region has lower compactness than its convex hull. In fact, the convex hull of a concave region has greater area *and* smaller perimeter.

# The Multi-Seeded Growth Model

We take a two-stage approach to finding the best districting. In the *Multi-Seeded Growth Model (MSGM)*, we find an initial allocation of $n$ districts so that the partition has modest levels of population equality and county preservation.

Our *Partition Optimization Model (POM)* edits and improves the rough sketch from MSGM.

The reason that we use two phases is speed. Our initial inclination was to allocate tracts randomly to the $n$ districts and then optimize by swapping tracts to improve some objective function. However, a random initial configuration is so far from the global maximum that the search might take years.

The MSGM generates a very crude districting that ensures district contiguity and tries for population equality and county preservation. Its districts are unacceptable for an actual plan but save enormous amounts of computing time.

## How It Works

We grow the $n$ districts simultaneously until they cover the state.

We start by allocating the entire state to a dummy district $D_0$, and then allocate $n$ tracts that serve as the initial "seeds" for the final districts, such that each $D_i$ begins as only a single tract. While $|D_0| > 0$, we consider the set $S$ of all possible moves that involve taking a district from $D_0$ while preserving contiguity. That is:

$$S(D_0; D_1, \ldots, D_n) = \bigcup_{i=1}^{n} \bigcup_{T_l \in F_i} M(T_l, D_0, D_i),$$

where $M(T_l, D_i, D_j)$ represents a *move of tract* $T_l$ from $D_i$ to $D_j$ and $F_i$ is the set of tracts that border $T_i$. Each move is scored by desirability of the prospective partition according to the score if we were to accept only that move. We perform the top 3% of moves. This method preserves contiguity, because by definition any $T_l \in F_i$ must be contiguous with $D_i$, and thus the $D_i$ are contiguous at each step.

Even though in the MSGM we do not consider moves between two "true" districts (rather, we consider only moves between a true district and the dummy district), the score of a move does not exist in isolation. Consider two adjacent districts $D_i$ and $D_j$, a shared frontier tract $T_l \in F_i \cap F_j$, and an unshared frontier tract $T_k \in F_i \cap F_j^c$. The acceptance of $M(T_k, D_0, D_i)$ alters the heuristic value of every move associated with $F_i$, which could potentially affect the optimality of further moves with $D_i$, such as the acceptance of $M(T_l, D_0, D_i)$ rather than $M(T_l, D_0, D_j)$. Furthermore, the acceptance of $M(T_l, D_0, D_i)$ likely expands the size of $F_i$. Perhaps there is an optimal move opened up in this new frontier that we do not even consider, because we have not even calculated its value.

It would be better to perform only the best move, but such a strategy is too computationally intensive. We compromise by taking in each step an elite fraction of the moves before recalculating $S$ and the values of its associated moves. In this respect, our approach is analogous to the strategy of *modified policy iteration* for solving a Markov decision problem, in which a fixed number of rounds of value iteration are made between policy iterations. The tradeoff of possible inefficiency is more than compensated for by speed gain, especially considering that the solution obtained by MSGM will be further refined by POM.

The MSGM scheme uses a variable number of moves between recalculating the value of the frontier. Our scheme causes us to be delicate in our selections of tract allocations, making moves virtually one at a time at the beginning and end of the MSGM. By focusing on the beginning and end of the problem, we attempt to avoid having a single district grow too large through inefficient allocation.

Unlike Cirincione et al. [2000], we use random initial seeds weighted by population rather than seeds equally spaced around the state. The process works as follows: While there are still random seeds to be selected, we find a candidate initial seed tract $T_l$ in $D_0$. We accept $T_l$ as an initial seed with probability $p(T_l)/\hat{p}$ so that tract selection is proportional to population. The MSGM algorithm produces the best initial results when all the districts have the same population rather than the same number of tracts. The geographically optimal placement of five (or fewer) starting seeds in the NYC Metropolitan area and Long Island evinces the fallibility of the equidistant initial-seed method.

The heuristic by which we rank candidate moves has two components: a population score and a county score.

## Population Score

We want to minimize egregious disparities in population between districts. The population component of our heuristic should give the highest score to a district when $p_i = \bar{p}$. Additionally, we want to penalize large deviations from the optimal population, so our function should be concave down.

Let $f(p_i)$ be the population heuristic score for a district with population $D_i$. We use a piecewise definition for $f$:

$$f(p_i) = \begin{cases} M\sqrt{\dfrac{p_i}{\bar{p}}}, & \text{if } p_i \leq \bar{p}; \\[2ex] M - \dfrac{4M}{p_i^2}\,(p_i - \bar{p})^2, & \text{if } p_i > \bar{p}. \end{cases}$$

Notice that $f$ is steeper for values $p_i > \bar{p}$ because we do not want growing districts to engulf too much population; we penalize deviations above $\bar{p}$ worse than deviations below $\bar{p}$. **Figure 3** shows the function $f$.

## County Preservation Score

We measure a district's county preservation score in terms of the percentage of counties that it completes on a population basis. To encourage growing districts to add remaining tracts in nearly complete counties, the marginal value of

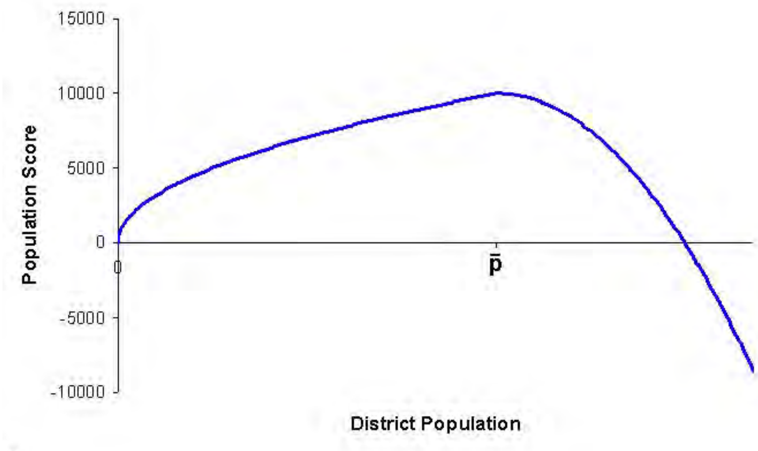



**Figure 3.** MSGM heuristic for population.

adding these should increase with the fraction of the population already contained in that district. To accomplish this, we use the square of the proportion contained in a county. The county score $g$ for a district $D_i$ is:

$$g(D_i) = \sum_{C_j \in \Lambda} \left( \frac{\sum_{T_l \in D_i \cap C_j} p(T_l)}{p(C_j)} \right)^2 . \qquad \textbf{(1)}$$

For instance, if a district completely contains one county and contains 30% of each of two other counties' populations, its score would be $(1^2 + 0.3^2 + 0.3^2) = 1.18$. **Figure 4** shows a plot of the county score that a district receives based on what percentage of a county's population said district contains.

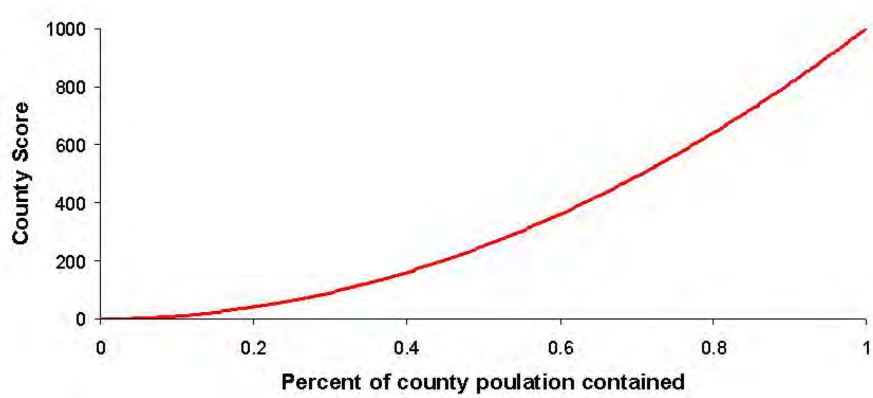



**Figure 4.** MSGM heuristic for county completeness.

# Partition Optimization Model

We refine the MSGM solution through local search.

## The Objective Function

The only characteristics of the district and the county that we use are the populations $p(P) = \{p_i\}_{1 \le i \le n}$, the compactness measures $c(P) = \{c_i\}_{1 \le i \le n}$, and the fractions $\rho(P) = \{\rho_{i,r} | 1 \le i \le n, 1 \le r \le c\}$ of the population of county $r$ that is contained in district $i$. We would like our score function $s(P) = s\big(p(P), c(P), \rho(P)\big)$ to have the following properties:

1. The score function should be unimodal as a function of $p_i$, with mode at $p_i = \bar{p}$.
2. The score should increase more by adding tracts that lie in $\chi(D_i)$, so that we prefer having as few districts as possible in a given county.
3. The score should increase more by adding tracts that increase the sum of all compactness measures by the greatest amount.

These desiderata suggest that we consider the three vectors $p(P), c(P), \rho(P)$ independently of one another in the score function. In other words, we would like our score function to be a separable function of these three vectors, that is, have the form

$$s(P) = f\big(p(P)\big) + g\big(c(P)\big) + h\big(\rho(P)\big),$$

where $f, g, h$ are functions.

### One (Wo)man, One Vote

The state has total population $P$ and average population of $\bar{p} = P/n$ per district. Letting $p_i$ be the population in district $i$, we consider three potential metrics for the population variance between districts:

1. Variance: $\mathrm{Var}(p_1, p_2, \ldots, p_n)$
2. Maximum deviation: $\max\{|p_i - \bar{p}|\}$
3. Maximum difference: $\max\{p_i\} - \min\{p_i\}$

For each measure, lower values are preferable and the minimum is 0. We submit that variance is the best alternative. To see why, consider two possible population distributions between districts:

- *Situation A:* One district has a population of $1.05\bar{p}$, one has $0.95\bar{p}$, and all of the others have $\bar{p}$.
- *Situation B:* Half of the districts have population $1.05\bar{p}$ and half have $0.95\bar{p}$ (any left-over odd district has $\bar{p}$).

In Situation A, only two districts are different from the ideal population level $\bar{p}$; but in Situation B, very few districts have population $\bar{p}$. So a good metric should rank $B$ worse than $A$. Clearly, the variance of populations is higher in $B$ than in $A$, so variance passes this test. The maximum deviation test gives $0.05\bar{p}$ for both $A$ and $B$, and the maximum difference gives $0.1\bar{p}$ for both.

We see that variance is the best measure of similarity, since it factors in the pairwise difference in all district populations.

By penalizing extreme variation away from $\bar{p}$, MSGM creates districts with approximate population equality. However, in one typical run, the final populations of districts vary from 600,000 to 700,000, an unacceptable difference.

## Compactness

To measure the compactness of a district, we would ideally use our compactness measure:

$$c_i = \frac{\text{Area}(D_i)}{[\text{Perimeter}(D_i)]^2},$$

such that:

$$g\big(c(P)\big) = \beta \sum_{i=1}^{n} c_i,$$

where $\beta$ is some constant.

Unfortunately, we could not calculate the perimeter of an arbitrary tract (the C++ library that we used to interact with our census data shapefiles featured massive memory leaks for large-scale union operations, questionable accuracy for pairwise unions, and seemingly arbitrary calculations of intersection length).

Yet it is a poor craftsman who blames the tools, so we adopt a different measure of compactness. The *clustering coefficient* provides a rough approximation for compactness:

$$cc(D_i) = \frac{\sum_{T_l \in D_i} |\{T_k \in D_i | T_k \sim T_j\}|}{\binom{m_i}{2}},$$

such that

$$g\big(c(P)\big) = \beta \sum_{i=1}^{n} cc(D_i),$$

where $\beta$ is some constant. The clustering coefficient provides a ratio of the total number of interdistrict boundaries to the maximum possible number of interdistrict boundaries. If all tracts were uniformly shaped, this measure would prize square- and circle-shaped districts, while winding single-tract-width districts would be penalized. However, given the asymmetry of tract shapes, this measure does little to reflect negatively upon district shapes such as the dumbbell (two circular clusters of tracts connected by a narrow band of tracts). In general however, the clustering coefficient values adding to districts tracts that are "close" and removing from districts those tracts that are auxiliary.

### County Preservation

We adopt the same county preservation measure **(1)** used in the MSGM, with the option of adding a scaling factor to the entire function to refine empirical performance.

## Search Method and Neighborhood Function

To refine our solution from MSGM, we must move tracts between districts. Yet the space of all possible contiguous moves is too large. We consider a range of possible moves with respect to only one district and perform the best move on this dramatically reduced state space.

By selecting our target district at random in each iteration, our strategy is best described as *stochastic domain hill climbing*, a method that combines the best aspects of both random and deterministic local search methods. We perform optimal moves while avoiding getting stuck trying to increase the score of only a single district. Simple first-order moves on the district level, that is, adding or removing individual tracts, cannot reduce the variance metric to the extremely low standard that we demand, so we include second-order moves, that is, "swaps"—both an add and a remove within a single operation.

If the maximum connectedness of any tract on the graph is $k$, checking for all adds and removes separately for district $D_i$ involves considering

$$\mathcal{O}(k \cdot |\partial D_i| + |F_i|) = \mathcal{O}(km_i)$$

moves, while looking at all swaps involves considering $\mathcal{O}(k \cdot |\partial D_i| \cdot |F_i|) = \mathcal{O}(km_i^2)$ moves. One might contend, then, that the operation of checking *every* district for first-order moves might be a better algorithm, since it would take $\mathcal{O}(\sum_{i=1}^{n} km_i) = \mathcal{O}(nkm_i)$ heuristic evaluations. One could even supplement such an algorithm with a degree of randomness, to avoid being caught in a loop of futility, by employing simulated annealing, stochastic hill climbing, or tabu search on the resulting list of possible future states. We found, however, that checking for second-order moves provides far better empirical results with acceptable time performance, while an algorithm enumerating all the possible second-order states, requiring $\mathcal{O}(\sum_{i=1}^{n} km_i^2) = \mathcal{O}(nkm_i^2)$ heuristic evaluations, was too slow to be effective.

The heart of POM is **Algorithm 1**. For simplicity and readability:

- $M_{\text{add}}(D_i)$ is the set of all moves in which we add a frontier tract to $D_i$,

- $M_{\text{remove}}(D_i)$ is the set of all moves in which we remove a border tract from $D_i$, and
- $M^{-1}$ is the move inverse to $M$, such that applying both $M$ and $M^{-1}$ in turn has no effect.

```
Input: Iteration count iter, initial partition P.
Output: Final partition P.
count ← 0
while count < iter do
    curscore ← s(P)
    D ← randomDistrict()
    bestscore ← curscore
    foreach M_a ∈ {∅ ∪ M_add(D)} do
        foreach M_r ∈ {∅ ∪ M_remove(D)} do
            performMove(M_a)
            performMove(M_r)
            if isContiguous(P) then
                tmpscore ← s(P)
                if tmpscore > bestscore then
                    bestscore ← tmpscore
                    bestadd ← M_a
                    bestremove ← M_r
                end
            end
            performMove(M_a^{-1})
            performMove(M_r^{-1})
        end
    end
    if bestscore > curscore then
        performMove(bestadd)
        performMove(bestremove)
        count ← count + 1
    end
    return P
end
```

**Algorithm 1**: Stochastic domain hill-climbing algorithm for districting.

We guarantee that the solution will be contiguous by not even considering moves that would break contiguity, and that we perform a move only if it increases the score of our current state.

## Achieving Absolute Equality

U.S. law mandates that the populations of each district in a state be equal *to within one person* according to the census data [*Karcher v. Daggett* (1983)]. We deal with entire census tracts, so our algorithm cannot meet that standard. This last step must be implemented by splitting tracts between districts.

To our knowledge, this problem beyond population unit level (no smaller than block groups) has not been addressed in the literature. Clearly, the simplest way to do this is to split one of the border tracts. While we do not implement this, we describe a methodology for it.

Let $G$ be a graph whose vertices are the districts and whose edges are the pairs of bordering districts. If we can find a pair of districts such that splitting a border tract between them gives both districts populations within one person of the mean population, then we would optimally do so and ignore those two districts for the remainder of the algorithm. However, to guarantee that the algorithm finishes, we require that the graph $G$ remain connected (otherwise, $G$ may divide into two or more connected components, such that the constituent districts cannot attain populations equal to the overall mean). Taking out two districts at a time by splitting only a single tract splits the fewest possible tracts.

We search for an edge of $G$ such that removal of its two vertices and all edges emanating from them leaves a new graph $G_1 \subset G$ that is connected. We call the deletion of a single vertex from a graph that leaves the graph connected a *paring*. If these two vertices have some special properties, we perform the double paring and then perform the algorithm on $G_1$, and continue until all districts have equal population. If no such pair of districts exists, we then perform a single paring and ensure that the removed district has population $\bar{p}$ before removing it. Define *tract splitting* to be the process of splitting a border tract into two disjoint areas and two disjoint populations allocated between two bordering districts.

There always exists an edge on a connected graph $G$ that permits a double paring of $G$, except for a very specialized set of connected graphs. However:

**Theorem.** *Every connected graph permits a paring.*

A proof of this theorem is given in the **Appendix**.

We recursively update the districts to get population equality. We iteratively pare the graph $G$ of districts such that each time we pare a district or pair of districts, those districts have populations which equal the population mean. By the theorem, this process always ends with all districts having equal population.

The algorithm removes at least one vertex from $G$ at each step, and the whole algorithm can therefore be performed with $(n - d)$ tract splittings, where $n$ is the number of districts and $d$ is the number of double parings performed.

# Case Study: New York

## The Data

The 2000 census for New York State contains 4,907 tracts, some with no population [Empire State Development 2007]. These empty districts are the "holes" on our maps. Trimming these tracts leaves 4,827 tracts to examine.

## Results

Running the MSGM on our initial allocation gives 29 haggard districts, varying from 281,000 to 970,000 population. We use this solution as a starting point. Though our algorithmic process of refinement is stochastic, generally more than 90% of the moves in any run involve swaps; this is particularly true at the very end of a run, where population differences between districts are minute. As a result, swapping provides a way to adjust population smoothly and also "cleans up" tattered fringes of districts, increasing their compactness even with vigorous population changes. After refinement, district populations ranged from 652,561 to 655,760.

The results in **Figures 5–7** demonstrate a partitioning into contiguous, compact, and reasonable districts. Furthermore, the simulations that produced these visually pleasing results also achieve extremely high degrees of population equality and county preservation.

# Analysis of the Models

## Solving the Problem

By combining the Multi-Seeded Growth Model with the Partition Optimization Model, we create fair and geometrically compact districts. The districts conform to well-accepted measures of goodness: population equality, contiguity, preservation of county boundaries, and compactness of shape.

The districts produced are both simple and fair. Geometric simplicity is measured by compactness, as determined by how close the members of a districts live relative to one other. Additionally, our method penalizes splitting counties between several districts, so that neighboring citizens, who have similar concerns, have the same representative. Fairness of our methodology is evident in its indifference to partisan politics, incumbent protection, and race/ethnicity.

## Strengths

The model successfully generates district partitions that simultaneously excel against the standard metrics of county integrity, compactness, and popu-

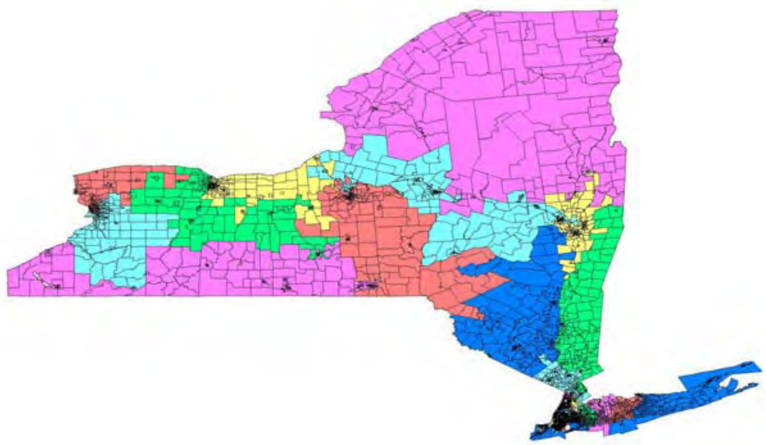


**Figure 5.** New York congressional districts from the POM.

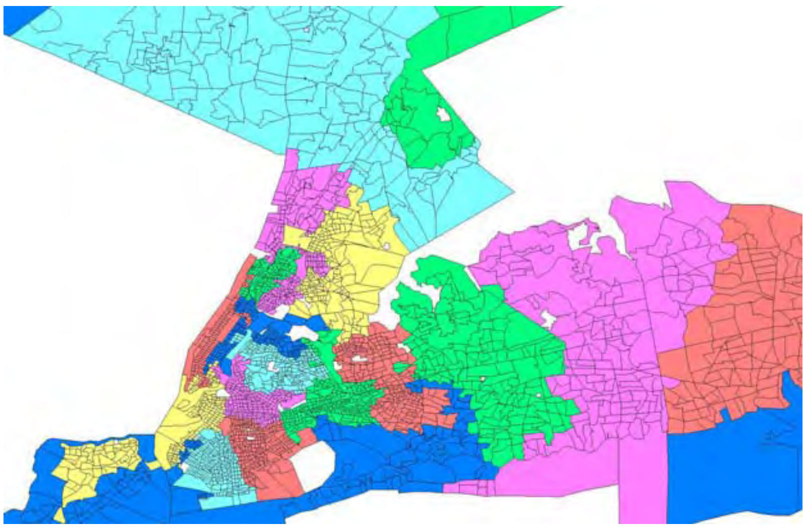


**Figure 6.** NYC metro-area POM.

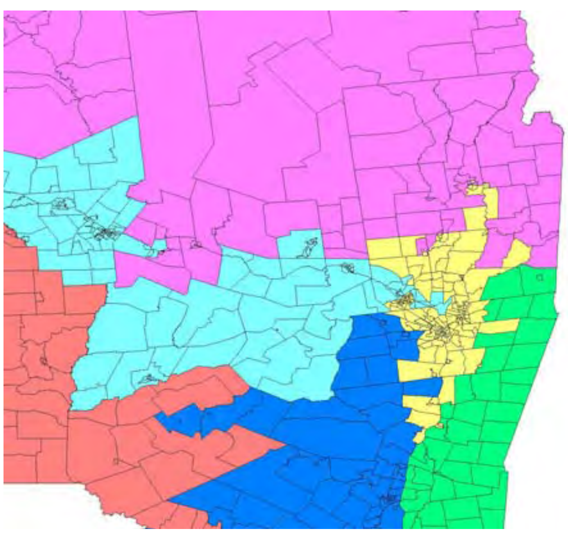


**Figure 7.** Close-up of the Albany area POM.

lation equality. Unlike other models in the literature, we provide an algorithm for reducing population differences to at most one person by breaking up a minimal number of tracts.

The model runs independently of the distribution of population, and works well both in low- and high-density locales, and with regular and oddly-shaped census tracts. This is evidenced by the successful districtings that our model produces in rural, small city, and large metropolitan areas (**Figures 5–7**).

The algorithm can generate districts for a large state in less than an hour.

## Weaknesses

The model assumes contiguity of the entire state; so in cases where contiguity cannot be forced, such as Hawaii or Michigan, we must change the algorithm slightly. One solution could be to divide the state into several regions and run the model separately on each region, allocating the proportionally correct number of representatives to each region based on population.

The model appears to tend toward creating districts that are either very low- or high-density, instead of splitting smaller population centers into a number of districts. Since political affiliation and race are likely correlated with population density, the algorithm may inadvertently generate districts that separate various demographic groups into separate districts, which could be viewed as gerrymandering. Yet, another camp would argue that it is appropriate to divide urban, suburban, and rural areas into separate districts, since their residents have different concerns.

# Conclusion

Since the 19th century, Elbridge Gerry's lizard has grown into a terrible, twisting serpent, eating away at our democracy. It is time to put Gerrymanders on a healthier diet.

# References

Barkan, J.D., P.J. Densham, and G. Rushton. 2006. Space matters: Designing better electoral systems for emerging democracies. *American Journal of Political Science* 50 (4): 926–939.

Bong, C., and Y. Wang. 2006. A multi-objective hybrid metaheuristic for zone definition procedure. *International Journal of Services Operations and Informatics* 1 (1/2): 146–164.

Caliper Corporation. n.d. About census summary levels. `http://www.caliper.com/Maptitude/Census2000Data/SummaryLevels.htm`.

Cirincione, C., T.A. Darling, and T.G. O'Rourke. 2000. Assessing South Carolina's 1990s Congressional redistricting. *Political Geography* 19: 189–211.

Empire State Development. 2007. New York State Data Center. Census 2000. `http://www.empire.state.ny.us/nysdc/popandhous/Census2000.asp` .

ePodunk Inc. 2007. New York: Population change, 2000 to 2003. `http://www.epodunk.com/top10/countyPop/coPop33.html` .

Garfinkel, R.S., and G.L. Nemhauser. 1970. Optimal political districting by implicit enumeration techniques. *Management Science* 16 (4): B495–B508.

*Hunt v. Cromartie*, 526 U.S. 541 (1999).

*Karcher v. Daggett*, 462 U.S. 725 (1983).

Kaiser, H. 1966. An objective method for establishing legislative districts. *Midwest Journal of Political Science* 10 (2) (May 1966): 210–213.

*League of United Latin American Citizens v. Perry*, 548 U.S. (2006).

Luttinger, J.M. 1973. Generalized isoperimetric inequalities. *Proceedings of the National Academy of Sciences of the United States of America* 70: 1005–1006.

Macmillan, W. 2001. Redistricting in a GIS environment: An optimisation algorithm using switching-points. *Journal of Geographical Systems* 3 (2): 167–180.

————, and T. Pierce. 1994. Optimization modeling in a GIS framework: The problem of political districting. In *Spatial Analysis and GIS*, edited by S. Fotheringham and P. Rogerson. Bristol, UK: Taylor and Francis.

Mehrotra, A., E.L. Johnson, and G.L. Nemhauser. 1998. An optimization based heuristic for political districting. *Management Science* 44 (8): 1100–1114.

*Miller v. Johnson*, 515 U.S. 900 (1995).

Nagel, S. 1965. Simplified bipartisan computer redistricting. *Stanford Law Review* 17: 863–899.

*Shaw v. Reno*, 509 U.S. 630 (1993).

*SC State Conference of Branches, etc. v. Riley*, (1982). 533 F. Supp. 1178 (DSC). Affirmed 459 US 1025.

U.S. Department of the Interior. 2007. Printable maps. `http://nationalatlas.gov/printable/congress.html#list` .

Weaver, J.B., and S.W. Hess. 1963. A procedure for nonpartisan districting: Development of computer techniques. *Yale Law Journal* 73 (1): 287–308.

Wikipedia. 2007. Gerrymandering. `http://en.wikipedia.org/wiki/Gerrymandering`.

Young, H.P. 1988. Measuring the compactness of legislative districts. *Legislative Studies Quarterly* 13: 105–115.

# Appendix: Proof of Theorem

**Theorem.** *Every connected graph permits a paring.*

*Proof:* We proceed by induction on the number $y$ of vertices. We prove a stronger statement, namely that for any connected graph $G$ with at least two vertices, there exist at least two parings. The claim clearly holds for $y = 2$.

Suppose that the claim holds for $y = k$, where $k \geq 2$, and that it does not hold for $y = k + 1$. Then, since $y \geq 3$, take any vertex $v$ of $G$ such that removal of $v$ leaves $G$ unconnected, and consider the two disjoint subgraphs $G_1, G_2$ into which $G$ is divided upon removal of this vertex. By the induction hypothesis, there exist vertices $v_1, v_2$ of $G_1$ such that removal of either one leaves $G_1$ connected.

We claim that removal of one of $v_1, v_2$ from the original graph $G$ leaves $G$ connected. To see this, note that neither $v_1$ nor $v_2$ is adjacent to any vertex in $G_2$, as $G_1$, $G_2$ have no common edges. If both $v_1, v_2$ are adjacent to $v$, then removal of $v_1$ leaves $G$ connected. This is because if we let $G' = G - \{v_1\}$ and $G'_1 = G_1 - \{v_1\}$, then $G'$ consists of $G'_1 \cup \{v\}$ and $G_2$, which are both connected and connected to each other, as $v$ is necessarily connected to $G_2$. This means that $G - \{v_1\}$ is connected.

If one of $v_1, v_2$ is not adjacent to $v$, without loss of generality assume that it is $v_1$. Then removing $v_1$ from $G$ leaves the graph connected, as $G'_1 \cup \{v\}$ is connected, as is $G_2$, and they are connected to each other. Some such vertex which admits a paring also exists in $G_2$, yielding two vertices which permit a paring. This proves the result by induction.

Abe Othman, Chris Yetter, and Ben Conlee.

# Electoral Redistricting with Moment of Inertia and Diminishing Halves Models

Andrew Spann
Daniel Kane
Dan Gulotta
Massachusetts Institute of Technology
Cambridge, MA

Advisor: Martin Z. Bazant

## Summary

We propose and evaluate two methods for determining congressional districts. The models contain explicit criteria only for population equality and compactness, but we show that other fairness criteria such as contiguity and city integrity are present, too.

The Moment of Inertia Method creates districts whose populations are within 2% of the mean district size, minimizing the sum of the squares of distances between the district's centroid and each census tract (weighted by population size). We prove that this model gives convex districts.

In the Diminishing Halves Method, the state is recursively halved by lines perpendicular to best-fit lines through the centers of census tracts.

From U.S. Census 2000 data, we extract the latitude, longitude, and population count of each census tract. By parsing data at the tract level instead of the county level, we model with high precision. We run our algorithms on data from New York as well as Arizona (small), Illinois (medium), and Texas (large).

We compare the results to current districts. Our algorithms return districts that are not only contiguous but also convex, aside from borders where the state itself is nonconvex. We superimpose city locations on district maps to check for community integrity. We evaluate our proposed districts with various quantitative measures of compactness.

The initial conditions do not greatly affect the Moment of Inertia Method. We run variants of the Diminishing Halves Method and find that they do not

*The UMAP Journal* 28 (3) (2007) 281–299. 

improve over the original. Based on our results, district shapes should be convex, and city boundaries and contiguity can be emergent properties, not explicit considerations. We recommend our Moment of Inertia Method, as it consistently performed the best.

# Assumptions and Justifications

## About States

- **The Earth's geometry is Euclidean**. No state is so big that the spherical shape of the earth significantly distorts distance calculations obtained from Euclidean geometry.

- **County lines are not inherently more significant than other boundaries**. Some states attempt to not split counties when determining districts, and other states give only slight consideration to county lines. Since several of New York's counties are too big to use as discrete units for dividing representatives, and representing county boundaries in the model is difficult, we instead use the census tract as our base unit of population.

- **Deviation from the current district division is not a major factor**. There are no inherent transitional problems with switching to a completely new division if it can be shown to have a higher degree of fairness.

- **District populations may vary by as much as 2% from the average value**. We use the 2% allowance to get around problems with our data on populations not being fine enough. The error could be made smaller if census blocks were used instead of census tracts.

## About Census Data

- **Census data are always accurate**. There are no other reasonable data.

- **Census tracts individually satisfy fair apportionment criteria**. No U.S. census tract is gerrymandered; there is no political benefit to doing so.

- **All population in a census tract can be approximated as located at a single point**. For data input to our program, we read in latitude and longitude for each census tract. We assume that the entire population of a census tract is located at this point. Since we have 6,398 census tracts for New York State, none of which have more than 4% of the population for a congressional district (and most of which are considerably smaller), this does not provide a very severe discretization problem.

# Literature Review

Gerrymandering has attracted scholarly attention for decades.

Attempts to assign districts with computers began in the 1960s and 1970s with models by Hess et al. [1965], Nagel [1965], and Garfinkel and Nemhauser [1970]. These methods typically represent population as a series of weighted $(x, y)$ coordinates and attempt to draw equal-population districts based on compactness and contiguity. The methods criteria for compactness vary, and a collection of compactness metrics is reviewed in Young [1988]. Computer resources were limited in this era, and Garfinkel and Nemhauser even report being unable to compute a 55-county state. For a more detailed review of early papers, see Williams [1995].

Many versions of the redistricting problem are NP-hard [Altman 1997]. Recent papers have tried graph theory [Mehrotra et al. 1988], genetic algorithms [Bacao et al. 2005], statistical physics [Chou and Li 2006], and Voronoi diagrams [Galvao et al. 2006]. There are also papers such as Cirincione et al. [2000], which are intellectual and stylistic descendants of the old papers but using modern computer resources that enable finer population blocks and tighter convergence criteria.

We use a moment-of-inertia model similar in formulation to Hess et al. [1965] but with differences in optimization.

# Criteria for Fair Districting

We list factors considered in districting and explain which ones we choose. Later, we describe the specific expressions of these criteria in our two models and their mathematical consequences. Core criteria are:

- *Equality of population*. The population difference between two districts can vary only by at most a certain number of people, usually on the order of 5%.
- *Contiguity*. Each district must be topologically simply connected.
- *Compactness*. There are differing opinions on how to quantitatively define compact, but all agree that small wandering branches are bad.

A criterion not emphasized in the literature is *convexity*, a stronger form of contiguity: Any two points in the region can be connected by a straight line segment contained within the region. This disallows holes or extraneous arms that contribute to most poorly-shaped districts. The worst case for a convex region is a district containing sharp angles or that is very elongated.

Other criteria [Nagel 1965; Williams 1995] serve one of two purposes:

- *Targeted homogeneity or heterogeneity*. Nagel explicitly expresses a desire to use predicted voting data to create "safe districts" and "swing districts," where the outcomes of elections are more predictable or less predictable, respectively. The stated reasons for this involve balancing the state's districts so that some parts of a state have experienced candidates who are stable to long term change and other parts more responsive. Other papers discuss clustering groups based on race, economic status, age, or other demographic data into a district where statewide minorities have a local majority.

- *Similarity to boundaries or precedent*. Whenever possible, people in the same city should have the same representative. Likewise, it can be viewed as unfair to a representative if the people represented change too quickly. It also makes sense for districts to follow rivers, lakes, mountains, and other natural boundaries where appropriate. Usually, boundary of precedent objectives are accomplished by keeping county boundaries intact whenever possible.

These optional criteria conflict with the earlier core criteria.

The explicit criteria should be as minimalist as possible, so that more-complicated measures of good districting emerge rather than be forced. Additionally, with complicated objectives, politicians could gerrymander by tweaking parameters of the objective function.

We explain why we do not include the two optional criteria listed above:

- We do not consider targeted homogeneity or heterogeneity criteria because we consider it highly unethical to write a computer program to draw districts that benefit a particular candidate or party, even if the stated reasons appear well-intentioned. The goal of computer assignment of districts is to eliminate all manipulations of this form, so including criteria of this form in the objective function is unacceptable.
- Although the use of existing county or natural boundaries might work well for small states with a high ratio of counties to congressional districts, the county borders of New York are ill-suited for this purpose. New York has only 62 counties but 29 representatives. Following county borders whenever possible but splitting counties where reasonable involves much more work in preprocessing data to incorporate county information and still places pressure on creating noncompact districts.

We formulate a methodology that involves only *equality of population* and *compactness*.

# Moment of Inertia Method

## Description

By equality of population, we mean that no district's population should differ by more than 2% from the mean population per district in the state. There does not appear to be any clear court-mandated tolerance for population difference [Williams 1995], so we simply pick a reasonable number that is within the feasibility of computation based on the discretized units of census tracts. We could tighten the bounds further if we were willing to tolerate an increase in computational time and use smaller divisions, such as census blocks.

Young [1988] lists eight different measures of compactness, none of which is perfect. The most intuitive definition is to minimize the expected squared distance between all pairs of two people in a district. This has the nice physical interpretation of being analogous to the moment of inertia (if the distance is Euclidean). Papers such as Galvao et al. [2006] minimize inertia based on travel-time distance (adjusted for roads, lakes, etc) rather than absolute distance; but we consider only absolute distance, which is easier to find. Also, if district borders are affected by travel time, then it is possible to gerrymander by constructing strategic roads or bridges.

# Response to Prior Literature Commentary

Young [1988] finds two problems with moment of inertia as a measure of compactness:

- It gives good ratings to "misshapen districts so long as they meander within a confined area."
- There is a significant bias based on the area of the district (the moment of inertia is uses squared distances).

In response to the first objection, we get districts that are not only contiguous but also convex (except where they meet nonconvex state lines). We draw districts where it must be possible to travel between any two points in a district in a straight line without leaving the district. This eliminates the first of Young's concerns, since the cited examples of misshapen districts, such as spirals, that cause moment of inertia to predict poorly all have the property of being nonconvex.

The concerns about bias toward large-area districts is perhaps more serious. If the complaint is true, then the moment of inertia compactness criterion could lead to stretched or awkward urban districts so as to smooth out larger neighboring districts. In our experimental runs, this problem was not severe.

# Mathematical Interpretation

We describe the mathematics of the moment-of-inertia criterion and its objective function. We derive an important result: Any local minimum of our objective function should consist of a collection of convex districts (except where the state border is nonconvex).

We use the average squared distance between two people in the same district as a measure of the misshapenness of that district. We assume a Euclidean metric. Let $\mathrm{E}[x]$ and $\mathrm{Var}\, x$ represent the expectation and variance of a random variable $x$. Let the coordinates of two randomly chosen people in the district be $(x_1, y_1)$ and $(x_2, y_2)$, and let the coordinates of an arbitrary randomly chosen person be $(x, y)$. Then our measure is

$$\mathrm{E}\left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right] = 2\,\mathrm{Var}\,x + 2\,\mathrm{Var}\,y = 2\,\mathrm{E}\left[|(x, y) - (\overline{x}, \overline{y})|^2\right],$$

where $(\overline{x}, \overline{y})$ is the center of mass of people in the district. Furthermore, this quantity is increased if $(\overline{x}, \overline{y})$ is replaced by another point.

Let there be $N$ people in the state to be divided into $k$ districts. Our objective is equivalent to partitioning the people into $k$ sets $S_1, \ldots, S_k$ of equal size, and picking points $p_1, \ldots, p_k$ to minimize

$$\sum_{i=1}^{k} \sum_{x \in S_i} d(x, p_i)^2,$$

where $d$ is Euclidean distance. Taking the points $p_i$ to be fixed, we find that even if we allow ourselves to split a person between districts (which we do not do in the actual program), we can recast this as a linear programming problem. We let $m_{x,i}$ be the proportion of $x$ that is in district $i$. We then have

$$m_{x,i} \geq 0; \qquad \textbf{(1)}$$

and for any $x$,

$$\sum_i m_{x,i} = 1. \qquad \textbf{(2)}$$

The restriction of district sizes says that for any $i$, we must have

$$\sum_x m_{x,i} = \frac{N}{k}, \qquad \textbf{(3)}$$

where $N$ is the total population of the state. The objective function is

$$\sum_{x,i} m_{x,i} d(x, p_i)^2.$$

A global minimum exists since $0 \leq m_{x,i} \leq 1$, implying that our domain is compact. By linear programming duality, at the point that minimizes the objective, the objective function can be written as a positive linear combination of the tightly satisfied constraints in the solution. For this linear combination, let $C_i$ be the coefficient of **(3)**, $D_x$ the coefficient of **(2)**, and $E_{x,i}$ the coefficient of **(1)**. We have that $C_i$ and $D_x$ are arbitrary, but $E_{x,i} \geq 0$ with equality unless $m_{x,i} = 0$. Comparing the $m_{x,i}$ coefficients of our objective and this linear combination of constraints, we get that

$$d(x, p_i)^2 = C_i + D_x + E_{x,i}.$$

If $m_{x,i} \geq 0$, then $E_{x,i} = 0$, hence that $E_{x,i} \leq E_{x,j}$. In particular, person $x$ can be only in the district $i$ for which $E_{x,i} = d(x,p_i)^2 - C_i - D_x$ is minimal. Equivalently, they are in the district $i$ for which $d(x,p_i)^2 - C_i$ is minimal. Therefore, for the optimal solution, there are numbers $C_i$ and the $i$th district is the set of people $\{x : d(x,p_j)^2 - C_j \text{ is minimized for } j = i\}$. Furthermore, these regions are uniquely defined up to exchanging people at the boundaries.

The next thing to note is that the $i$th district is defined by the equations

$$d(x,p_i)^2 - C_i \leq d(x,p_j)^2 - C_j. \tag{4}$$

Rotating and translating the problem so that $p_i = (0,0)$ and $p_j = (a,0)$, and letting $x = (x,y)$, **(4)** reduces to

$$x^2 + y^2 - C_i \leq (x-a)^2 + y^2 - C_j,$$

or

$$2ax \leq a^2 + C_i - C_j.$$

Therefore, each district is defined by a number of linear inequalities. Hence, we have shown that our measure has the nice property that *the optimal districts with fixed $p_i$ are convex*, so any local minimum of our objective function should consist of a partition into convex regions.

## Computational Complexity

It would be nice to compute the global optimum, but we probably cannot do so in general. Adapting the linear program above, we wish to minimize

$$\sum_i \text{Var } X_i$$

where $X_i$ is a randomly chosen person in district $i$. This is equal to

$$\sum_i \left( \frac{k}{N} \sum_x m_{x,i} |\vec{x}|^2 - \frac{k^2}{N^2} \left| \sum_x m_{x,i} \vec{x} \right|^2 \right).$$

Notice that the term

$$\sum_i \sum_x m_{x,i} |\vec{x}|^2 = \sum_x |\vec{x}|^2 \sum_i m_{x,i} = \sum_x |\vec{x}|^2$$

is a constant. Hence, we wish to maximize the sum of the squares of the magnitudes of the centers of mass of the districts. This is an instance of quadratic programming where we try to maximize a positive semidefinite objective function. Since general quadratic programming is NP-hard, it seems likely that it is not easy to find a global maximum for our problem. On the other hand, we have shown that even local maxima have many properties that we want, e.g., convexity. Furthermore, these local maxima are significantly easier to find—e.g., from the quadratic programming formulation using the simplex method.

Unfortunately, the quadratic programming approach leads to an optimization involving $kN$ variables, which can be quite large. Instead, we consider the formulation where to have a local maximum we need to pick $p_i$ and $C_i$ (thus defining our districts by "$x$ goes in the district $i$ for which $d(x, p_i)^2 - C_i$ is minimal") in such a way that the districts have the correct size and so that $p_i$ is the center of mass of the $i$th district. This will imply that we have a local maximum of the quadratic program, since near our solution (up to first order) our objective function is

$$C - \sum_{x,i} m_{x,i} d(x, p_i)^2, \tag{5}$$

for some constant $C$. Since we have a global maximum of **(5)**, moving a small amount in any direction within our constraint does not decrease our objective, up to first order. Furthermore, since our objective is positive semidefinite, we are at a local maximum. This formulation is much better, since we are now left with only $3k$ degrees of freedom for $k$ districts.

## Comparison to Hess et al.

This procedure is very similar to that of Hess et al. [1965]. They too were attempting the minimize the summed moments of inertia of districts. They also converged on their solution via an iterative technique that alternates between finding the best districts for given centers and finding the best centers for given districts. Our approach differs from theirs in two main points: the method of finding new districts for given centers, and the general philosophy toward achieving exact population equality. Both of these differences stem from our having finer data (Hess et al. used 650 enumeration districts for dividing Delaware into 35 state House and 17 state Senate seats, whereas we have 10 times as many census tracts) and more computational power. We cannot determine exactly what algorithm Hess et al. used to determine optimal districts with given centers other than a "transportation algorithm," possibly the linear programming formulation from earlier (possibly using a min-cost-matching formulation). We have many more census tracts to work with and use an algorithm better adjusted to this problem. We also have different perspectives about what to do to even out population. Our fundamental units are sufficiently small that we can just run our algorithm, adjusting district sizes in a natural way until all districts are within 2% of the desired population. Hess et al. used a solution method that divided fundamental units of population between districts and later had to perform post-iteration checks and alterations so that units were no longer split and population equality still worked out. This readjustment has the potential to increase moments of inertia and could theoretically lead to a failure to converge.

# Diminishing Halves Method

As an alternative against which to compare our moment of inertia algorithm, we use the Method of Diminishing Halves proposed by Forrest [1964].

## Definition

The Diminishing Halves Method splits the state into two nearly-equal-sized districts and recurses on each of the two halves. The idea is to split into relatively compact halves. Forrest does not specify exactly how the state must be split into two halves at each step, but rather argues that the method for splitting the state in two could be adjusted based on preferences for keeping counties intact or other goals.

Suppose that we run a least-squares regression on the latitude and longitude coordinates of the state's census tracts. We would expect that dividing along this best-fit line would be a bad idea, since it would probably cut major cities in half or cover a long distance across the state. If we take a line perpendicular to the best-fit line, then hopefully we get the opposite properties. Therefore, we divide the state at each stage with a line whose slope is perpendicular to the best-fit line of the census tracts. We are not aware of this specific criterion being used in previous literature.

## Mathematical Interpretation

The best-fit line is an approximation of the shape of the state is of the form

$$(X-\overline{X})\sin\theta+(Y-\overline{Y})\cos\theta=0.$$

The left-hand side is the distance of a point $(X,Y)$ from the line. We minimize

$$\begin{aligned}\mathrm{E}\left((X-\overline{X})\sin\theta+(Y-\overline{Y})\cos\theta\right)^2 &=\\ &\sin^2\theta\,\mathrm{Var}\,X+2\sin\theta\cos\theta\,\mathrm{Cov}\,XY+\cos^2\theta\,\mathrm{Var}\,Y.\end{aligned}$$

This value is minimized when

$$\sin\theta\cos\theta(\mathrm{Var}\,X-\mathrm{Var}\,Y)+(\cos^2\theta-\sin^2\theta)\,\mathrm{Cov}\,XY=0,$$

or when

$$\tan(2\theta)=\frac{-2\,\mathrm{Cov}\,XY}{\mathrm{Var}\,X-\mathrm{Var}\,Y}.$$

To divide the population into $k$ districts, we divide the state by a line perpendicular to the best fit that splits the population in ratio $\left\lfloor\frac{k}{2}\right\rfloor:\left\lceil\frac{k}{2}\right\rceil$. When we need to divide into an odd half and an even half, the ceiling half goes to the southern side.

# Experimental Setup

## Extraction of U.S. Census Data

We used a Perl script to extract data at the census tract level from the 2000 U.S. Census [U.S. Census Bureau 2001]. For New York, there are 6,661 tracts in the database, 6,398 of which have nonzero population. We extract the population along with the latitude and longitude of a point from each district. The districts have populations from 0 to 24,523 with a median of 2,518. We model the population density by assuming that the entire population of a tract is located at the coordinates given. We adjust for the fact that one degree of latitude and one degree of longitude represent different lengths on the Earth's surface by having our program internally multiply all longitudes by the cosine of the average latitude. We also extracted data for Arizona (small—8 congressional representatives), Illinois (medium—19 representatives), and Texas (large—32 representatives).

## Implementation in C++

We use a C++ program to compute an approximate local minimum of our Moment of Inertia objective function. We do so without splitting census tracts between districts, and this discretization requires us to allow a little lenience about the exact sizes of our districts (we allow them to vary from the mean by as much as 2%).

We attempt to converge to a local optimum via two steps. First we pick guesses for the points $p_i$. We then numerically solve for the $C_i$ that make the district sizes correct, giving us some potential districts. We allow a variation of 2% from the mean, beginning with a 20% allowable deviation in the first few iterations and tightening the constraint on subsequent iterations. We then pick the center of mass of the new districts as new values of $p_i$, and repeat for as long as necessary. Each step of this procedure decreases the quantity in **()**, because our two steps consist of finding the optimal districts for given $p_i$ and finding the optimal $p_i$ for given districts. We find the correct values of $C_i$ by alternately increasing the smallest district and decreasing the largest one. When this adjustment overshoots the necessary value, we halve the step size for that district, and when it overshoots by too much, we reverse the change. For New York, convergence to the final districts took a couple of minutes.

After determining our districts, we output them to a PostScript file that displays the census tracts color-coded by district, so that one can visually determine compactness. Finally, we computed some of the compactness measures discussed in [Young 1988].

We also created a C++ program to implement the Diminishing Halves Method.

# Measures of Compactness

We need an objective method for determining how successful our program is at creating compact districts. Young [1988] gives several measures for the compactness of a region. We use some of these to compare our districts with those produced by other methods. Since our algorithms generate convex districts except where the state border is nonconvex, we perform all of these results on the convex hull of our districts, so that the test results are not unfairly affected by awkwardly-shaped state borders.

## Definitions

**Inverse Roeck test**. Let $C$ be the smallest circle containing the region $R$. We measure Area$(C)$/Area$(R)$, a number larger than 1, with smaller numbers corresponding to more compact regions. This is the reciprocal of the Roeck test as phrased in Young. We have altered it so that all of our measures in this section have smaller numbers corresponding to more compact regions.

**Length-Width test**. Inscribe the region in the rectangle with largest length-to-width ratio. This ratio is greater than 1, with numbers closer to 1 corresponding to more compact regions.

**Schwartzberg test**. We compute the perimeter of the region divided by the square root of $4\pi$ times its area. By the isoperimetric inequality, this is at least 1 with a value of 1 if and only if the region is a disk. This test considers a region compact if the value is close to 1.

## Calculation in Mathematica

We used the tests above to check compactness of the proposed districts. We implemented the tests in Mathematica with aid of the Convex Hull and Polygon Area notebooks [Weisstein 2004; 2006].

For the Roeck test, we compute the area of the polygon by triangulating it. We find the circumradius by noting that if every triple of vertices can be inscribed in a disk of radius $R$, then the entire polygon fits into the disk. This is because a set of points all fit in a disk of radius $R$ centered at $p$ if and only if the disks of radius $R$ about these points intersect at $p$. Let $D_i$ be the disk of radius $R$ centered around the $i$th point. If every triple of points can be covered by the same disk, then any three of the $D_i$s intersect. Therefore, by Helly's theorem [Weisstein 1999] all the disks intersect at some point, and hence the disk of radius $R$ at this point covers the entire polygon. Hence, we need for any three points the radius of the disk needed to contain them all. This is either half the length of the longest side if the triangle formed is obtuse, or the circumradius otherwise.

For the Length-Width test, we pick potential orientations for our rectangle in increments of $\pi/100$ radians. At each increment, we project our points parallel and perpendicular to a line with that orientation. The extremal projections determine the bounding sides of our rectangle. We choose the value from the orientation that yields the largest length-to-width ratio.

Calculating the Schwartzberg test is straightforward.

# Results for New York

**Figure 1** presents maps of the Moment of Inertia Method districts, the districts from the Diminishing Halves Method, and the actual current congressional districts of New York.

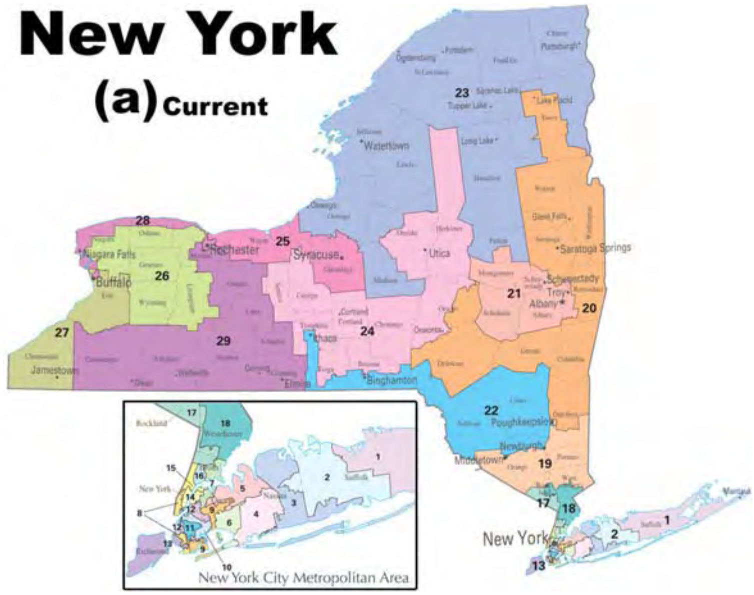
Our program's raw output plots the latitude and longitude coordinates of each census tract using a different color and symbol for each district. The state border and black division lines are added separately. There appears to be a slight color bleed across the borderlines near crowded cities, but this is due to the plotting symbols having nonzero width. Zooming in on our plot while the data are still in vector form (before rasterization) shows that our districts are indeed convex.

## Discussion of Districts

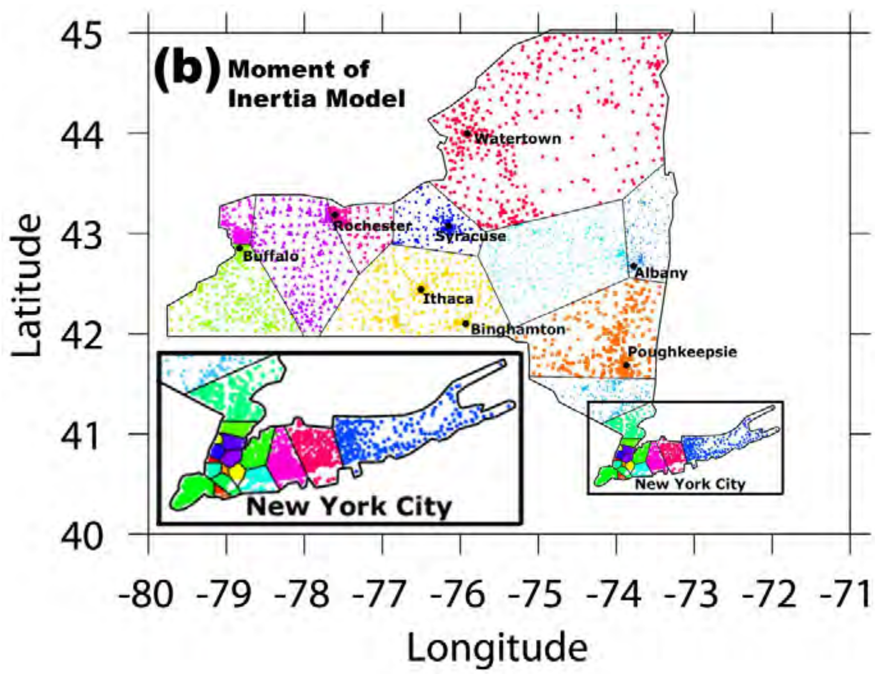
Both methods produce more compact-looking results than the current districts. Some current New York districts legitimately try to respect county lines, but there are a few egregious offenders, such as Districts 2, 22, and 28, where the boundaries conform to neither county lines nor good compactness. The current District 22 has a long arm that connects Binghamton and Ithaca, and the current District 28 hugs the border of Lake Ontario to connect Rochester with Niagara Falls and the northern part of Buffalo. Both of our methods allow Ithaca and Binghamton to be in the same district, but without stretching the district to the land west of Poughkeepsie. Buffalo and Rochester are kept separate in both of our models.

Our data do not contain information about county lines. However, both of our methods do a good job at keeping the major cities of New York intact. Buffalo and Rochester are divided into at most two districts in our methods, instead of three under the current districting. The Diminishing Halves Method has a cleaner division for Rochester, but the Moment of Inertia Method handles Syracuse much better.

Both methods produce districts with linear boundaries. The Diminishing Halves Method has a tendency to create more sharp corners and elongated districts, whereas the Moment of Inertia Method produces rounder districts. The Diminishing Halves Method tends to regions that are almost all triangles and quadrilaterals. Where three districts meet with the Diminishing Halves Method, the odds are that one of the angles is a 180° angle. The Moment of Inertia Method does a better job of spreading out the angles of three intersecting regions more evenly and thus results in more pleasant district shapes.

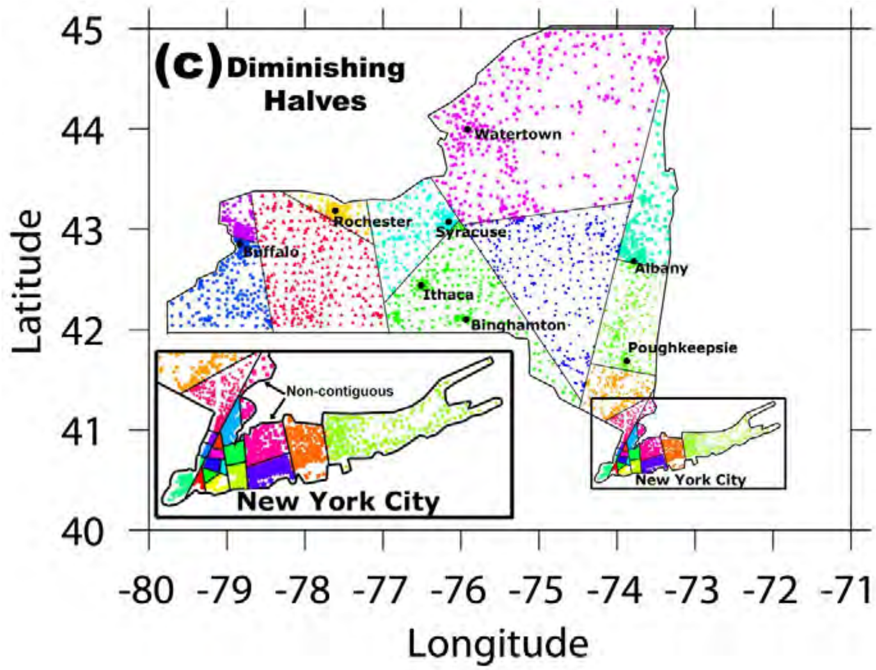



**a.** Current (adapted from U.S. Department of the Interior [2007]).




**b.** Moment of Inertia Method.

**Figure 1.** New York districts.




**c.** Diminishing Halves Method.

**Figure 1 (continued).** New York districts.

That Greater New York City contains roughly one-half of the state's population is convenient for the Diminishing Halves Method. However, that algorithm does not deal very well with bodies of water. This feature leads to the creation of one noncontiguous district (marked as "Non-contiguous" in Long Island Sound in **Figure 1c**. Overall, the shapes given by the Moment of Inertia Method look rounder and more appealing.

## Compactness Measures

**Table 1** lists results of the compactness tests. Smaller numbers correspond to more compact regions.

**Table 1.**

Mean and standard deviation for compactness measures of districts; smaller is better.

| Districts | Inverse Roeck Test | Schwartzberg Test | Length-Width Test |
|---|---|---|---|
| NY (Moment of Inertia) | $2.29 \pm 0.66$ | $1.64 \pm 0.62$ | $1.91 \pm 0.61$ |
| NY (Diminishing Halves) | $2.50 \pm 0.87$ | $1.74 \pm 0.69$ | $1.91 \pm 0.77$ |

According to these measures, the Moment of Inertia Method does marginally better than the Diminishing Halves Method. The diminishing halves numbers appear to be larger by about one-seventh of a standard deviation. This probably is caused by a few of the more misshapen districts.

All three measures are calibrated so that the circle gives the perfect measurement of 1. Roughly speaking, the Roeck test measures area density, the Length-Width test measures skew in the most egregious direction, and the Schwartzberg test measures overall skewness. Each measure tells us approximately the same thing: the Moment of Inertia Method performs a little bit better than the Diminishing Halves Method.

It would be desirable to compare the numbers in **Table 1** to the current districts, but there are two reasons why we cannot do this:

- The data that we used do not offer congressional district identification at the census-tract level. To compute compactness, we would need to choose a finer population unit, hence the numbers would not be directly comparable to those in **Table 1**.
- All of our districts in both methods are convex except for where the state border is nonconvex. This is not true for the current districts, and it is unclear how useful the compactness numbers are at comparing convex districts to nonconvex districts.

## Results for Other States

To test how well our algorithms perform on states with different sizes, we also computed districts for Arizona (small—8 districts), Illinois (medium—19 districts), and Texas (large—32 districts). [EDITOR'S NOTE: We omit the corresponding figures and specific analysis.]

### Compactness Measures

The Diminishing Halves Method produces consistently worse results by all three measures. This fact suggests (and the maps seem to confirm) that this fault is due largely to producing a small number of very elongated districts.

Given the evidence, we recommend the Moment of Inertia Method over the Diminishing Halves Method.

## Sensitivity to Parameters

To test for robustness, we tweak some of the parameters to the Moment of Inertia Model and test variants of the Diminishing Halves Method.

### Initial Condition

We ran the Moment of Inertia Model on each of the states with three different random seeds. The results were almost identical each time.

# Population Equality Criterion

We ran the New York case of the Moment of Inertia Model using a 5% allowable deviation from the mean in district population instead of a 2% allowable deviation. We observed no significant change in the results.

## Variants of the Diminishing Halves Method

We modified our criterion for determining the dividing line in the Diminishing Halves Method to use a mass-weighted best-fit line, weighted to account for different census tracts containing different numbers of people. We ran this modified method on New York, Arizona, and Illinois. We also tried a modification of the Diminishing Halves Method on the New York case that draws vertical and horizontal (longitude and latitude) lines.

In all these modified cases, results were visibly much worse. The modified methods tended to split cities into more districts than the original method.

# Strengths and Weaknesses

## Strengths:

- **Emergent behavior from simple criteria**. We specify criteria only for population equality and compactness. We satisfy contiguity and city integrity without explicitly trying to do so.
- **Simple, intuitive measure of complexity of districts**. In the Moment of Inertia Method, our measure of the noncompactness of a district gives a model that is easy to understand and does not use arbitrary constants that could be tuned to gerrymander districts.
- **Results in convex districts**. Both models produce districts guaranteed to be convex, aside from where the state border is nonconvex. This provides a fairly strong argument for the compactness of the resulting districts.
- **Easily computable**. Our final districting can be computed in a few minutes.
- **Nice-looking final districts**. The districts that we get appear very nice.

## Weaknesses:

- **No theoretical bounds on convergence time**. We could not prove that our algorithm converges in reasonable time, although it has done so in practice.

- **Potential for elongated smaller districts**. Some of the smaller districts produced by the Moment of Inertia Method may be stretched to accommodate larger districts. The Diminishing Halves Method may not correctly divide regions such as discs or squares that are not described well by a best-fit line.

- **Does not respect natural or cultural boundaries**. Our algorithms do not take natural or cultural boundaries into account. Doing so would have the advantage of not having district boundaries cross rivers but could place pressure on making districts noncompact and allow for loopholes that could be exploited by malicious politicians.

- **Does not necessarily find the global optimum**. Our Moment of Inertia algorithm finds only a local minimum. This leads potentially to some nondeterminism in the resulting districts, which could allow gerrymandering; but the amount is small.

- **Can only draw new districts, not determine if existing districts are gerrymandered**. Cirincione et al. [2000] give a pseudoconfidence interval analysis to assess whether South Carolina's 1990 redistricting had been gerrymandered. We do not perform such analysis here.

# Conclusion

We formulated and tested two methods for assigning congressional districts with a computer.

The Moment of Inertia Method searches for the answer that satisfies the intuitive criterion that people within the same district should live as close to each other as possible. We implemented this method and obtain results that would not have been computationally feasible in the 1960s and 1970s.

The Diminishing Halves Method recursively divides the population in half, which is very simple to explain to voters. To avoid elongated districts and to cut along sparsely populated areas rather than densely populated regions, our implementation chooses a dividing line perpendicular to the statistical best-fit line through the latitude and longitude coordinates of the census tracts.

We have some concrete recommendations for state officials:

- **Processing data at the census tract level or finer is computationally feasible**. It would not be unreasonable to process at the block group level if the extra resolution would be beneficial.

- **Districts should be convex**. Most models in the literature check only for contiguity. However, even severely gerrymandered districts such as Arizona District 2 satisfy contiguity. Requiring all districts to be convex greatly reduces the potential for political abuse.

- **City boundaries and contiguity of districts should be emergent properties, not explicit considerations**. Neither of our methods explicitly requires districts to be contiguous, yet the districts they generate are not only contiguous but convex. Neither of our methods attempts to preserve city or county boundaries, yet the Moment of Inertia Method does a good job of keeping cities together whenever reasonable. It is probably sensible for smaller states with a high ratio of counties to congressional representatives to be concerned with county boundaries; but for New York, where there are comparatively few counties, looking at city integrity instead of county integrity is more reasonable.

- **A good algorithm can handle states of different sizes**. Algorithms that perform well on large states might not yield good results for a small state with only one or two large cities. We tested our algorithms on states of different sizes; the Moment of Inertia Method behaves well in all cases.

- **We recommend a moment of inertia compactness criterion**. The Moment of Inertia Method, compared to the Diminishing Halves Method,
  - consistently produces more visually-appealing districts,
  - has better results on the compactness tests, and
  - does a better job of respecting city boundaries.

# References

Altman, M. 1997. The computational complexity of automated redistricting: Is automation the answer? *Rutgers Computer and Technology Law Journal* 23: 81–136.

Bacao, F., V. Lobo, and M. Painho. 2005. Applying genetic algorithms to zone design. *Soft Computing* 9: 341–348.

Chou, C.I., and S.P. Li. 2006. Taming the gerrymander—Statistical physics approach to political districting problem. *Physica A* 369: 799–808.

Cirincione, C., T.A. Darling, and T.G. O'Rourke. 2000. Assessing South Carolina's 1990s Congressional redistricting. *Political Geography* 19: 189–211.

Forrest, E. 1964. Apportionment by computer. *American Behavioral Scientist* 8 (4) (December 1964): 23, 35.

Galvao, L.C., A.G.N. Novaes, J.E.S. de Cursi, and J.C. Souza. 2006. A multiplicatively-weighted Voronoi diagram approach to logistics districting. *Computers and Operations Research* 33: 93–114.

Garfinkel, R.S., and G.L. Nemhauser. 1970. Optimal political districting by implicit enumeration techniques. *Management Science* 16 (4): B495–B508.

Hess, S.W., J.B. Weaver, H.J. Siegfeldt, J.N. Whelan, and P.A. Zitlau. 1965. Nonpartisan political redistricting by computer. *Operations Research* 13 (6): 998–1006.

Mehrotra, Anuj, Ellis L. Johnson, and George L. Nemhauser. 1998. An optimization based heuristic for political districting. *Management Science* 44 (8): 1100–1114.

Nagel, S.S. 1965. Simplified bipartisan computer redistricting. *Stanford Law Review* 17: 863–899.

U.S. Census Bureau. 2001. Census 2000 Summary File 1 [New York]. `http://www2.census.gov/census_2000/datasets/Summary_File_1/New_York/` .

U.S. Department of the Interior. 2007. National Atlas of the United States. Printable Maps. Congressional Districts—110th Congress. `http://nationalatlas.gov/printable/congress.html` .

Weisstein, Eric W. 1999. Helly's theorem. From *MathWorld*—A Wolfram Web Resource. `http://mathworld.wolfram.com/PolygonArea.html` .

________. 2004. Polygon area. From *MathWorld*—A Wolfram Web Resource. `http://mathworld.wolfram.com/PolygonArea.html` .

________. 2006. Convex hull. From *MathWorld*—A Wolfram Web Resource. `http://mathworld.wolfram.com/ConvexHull.html` .

Williams, J.C. 1995. Political redistricting—A review. *Papers in Regional Science* 74: 13–39.

Young, H.P. 1988. Measuring the compactness of legislative districts. *Legislative Studies Quarterly* 13: 105–115.

# Why Weight? A Cluster-Theoretic Approach to Political Districting

Sam Whittle
Wesley Essig
Nathaniel S. Bottman
University of Washington
Seattle, WA

Advisor: Anne Greenbaum

## Summary

Political districting has been a contentious issues in American politics over the last two centuries. Since the landmark case of *Baker v. Carr* (1962), in which the U.S. Supreme Court ruled that the constitutionality of a state's legislated districting is within the jurisdiction of a federal court, academics have attempted to produce a rigorous system for districting a state.

We propose both a modified form of classical K-means clustering and the shortest-splitline algorithm to accomplish impartial redistricting. We apply our methods to redistricting New York, and, as further examples, Texas and Colorado. Both methods use only population-density data and state boundaries as inputs and run in a feasible amount of time.

Our criteria for successful redistricting include contiguity, compactness, and sufficiently uniform population.

The K-means method produces districts similar to convex polygons, and the splitline method guarantees that the resulting districts have piecewise linear boundaries. The K-means method has the advantage of allowing seeding of the district centers. The centers of the generated districts then roughly correlate to the existing districts, by proper seeding, but the resulting boundaries are vastly simpler.

*The UMAP Journal* 28 (3) (2007) 301–313. 

# Introduction

The writers of the Constitution created the House of Representatives to be the branch of government most responsive to the people. The reality is just the opposite. Though representatives are elected every two years, almost 400 of the 435 seats are not effectively contestable, because of gerrymandering. With the immensely detailed amount of data and unlimited computing power available to politicians today, gerrymandering has been elevated to an art. With only the requirements that districts be connected and all have equal population, it is possible to pinpoint candidates and place them in a different district than their neighbors [Toobin 2003].

Though undemocratic, gerrymandering is nearly always legal (see, for instance, Backstrom [1986]) and has been used to obtain striking results. In 2002, only four incumbent representatives lost their bid for re-election—the lowest total ever [Toobin 2003]. We will argue that it is certainly true that any attempt to restructure legislative districts fairly needs to ignore the human factors that overwhelmingly determine current redistricting.

Defining a measure of compactness is essential to ensure fair districts. Both methods that we offer produce districts that at first glance are clearly simpler than the existing ones. We use the centers of the existing districts as seeds for a clustering algorithm. Thus, the new districts have some correlation to the existing districts, but their boundaries are determined in a fair manner. The core of many districts will be roughly the same, while the boundaries will be dramatically simpler. This effectively counteracts the effects of gerrymandering, without being overly difficult to implement.

## Plan of Attack

Our goal is an algorithm to divide a region into $k$ districts that satisfy some heuristic definition of *fairness*. To accomplish this, we must do the following:

- Define fairness and simplicity.
- State assumptions and constraints.
- Define metrics for comparing algorithms.

## Defining Simplicity

We say that district $A$ is *simpler* than district $B$ if $A$ is contiguous and more compact than $B$.

- **Contiguity.** A district is *contiguous* if it is arcwise-connected; that is, if one can travel from any point $a$ to any other point $b$ in $A$ while remaining entirely within $A$. If $A$ contains regions separated by bodies of water, $A$ is contiguous if all regions are connected by water and each region is arcwise-connected.

- **Compactness.** Intuitively, a district is compact if it does not meander excessively. This is a hard concept to formalize; many authors give only a hasty definition, and some even argue that compactness is ambiguous to the point of being irrelevant. Nonetheless, we attempt a suitable definition.

## Towards a Suitable Definition of Compactness

Young [1988] gives compelling reasons for abandoning all previous definitions of compactness (see the **Appendix**). But Young does not consider the following adjusted version of the Schwartzberg Test, which is alluded to in Garfinkel and Nemhauser [1970]:

**Definition 1** *District $A$ is more compact than district $B$ if*

$$\frac{4\pi\ \text{Area}_A}{\left(\text{Perimeter}_A\right)^2} > \frac{4\pi\ \text{Area}_B}{\left(\text{Perimeter}_B\right)^2}.$$

*We call the quantity* $4\pi$ Area / Perimeter$^2$ *the* compactness quotient.

For a circle of radius $r$, this ratio is equal to 1. It is well-known that the shape with the largest ratio of area to squared perimeter is the circle (see, for instance, Folland [2002]) so the compactness quotient is between 0 and 1.

As seen in **Figure 1**, a compactness quotient of $0.13$ is visually quite bad. Using the fact given in Bourke [1988] that the area of a non-self-intersecting closed $N$-gon (with the $k$th vertex in counterclockwise order equal to $(x_k, y_k)$) is equal to

$$\tfrac{1}{2}\sum_{i=1}^{N-1}\left(x_i y_{i+1} - x_{i+1} y_i\right),$$

we calculated the compactness quotients of several actual districts by approximating their boundaries by piecewise linear segments. Two of New York's more sprawling districts have compactness quotients 0.097 and 0.101 (**Figure 2**)—even worse than the gerrymander in **Figure 1**! The two most compact districts in New York, the 26th and 21st, have compactness quotients 0.406 and 0.498.

We decide that the mean for any state should be at least 0.6, so that the average district would be better than the best current districts in New York. Furthermore, we insist that 0.25 should be more than 2 standard deviations from the mean. It is not possible to require that all districts be greater than 0.25, since several districts inevitably have most of their border coincide with the state border.

## Defining Fairness

Almost all unfairness occurs when political and social measures factor into redistricting decisions. Concentrating supporting voters in a single district,

**Figure 1.** The compactness quotients of the circle, square, and gerrymander are $1$, $\pi/4 \approx 0.79$, and $23\pi/576 \approx 0.13$, respectively.


**Figure 2.** Current New York districts 8 and 28 (in dark shade), with compactness quotients of 0.097 and 0.101. Source: U.S. Department of the Interior [2007].

diluting opposing voters over several districts, placing two incumbents in the same district and forcing them to run against each other, and isolating minorities (see Toobin [2003] and Hayes [1996]) are all the result of districting being controlled by those who attempt to skew voting patterns. In general:

- *Unfair districting stems from either human biases or poorly designed algorithms.*

Our computer simulations use only population density and the boundary of the state, so the determination of districts is completely unbiased. While a district may be unfair on a local scale, in that it divides up a community with a common interest—for instance, a community of apple-growers may be split between two districts—on the national scale, such imbalances will even out. Because of this, there will be no pathological examples of disproportionate representation.

# Applying the Theory of Data Clustering

Data clustering is classifies observations (or objects) into groups. The main benefits of a cluster-theoretic algorithm are:

- *Data clustering often reveals an internal structure that may not have been initially apparent.*
- *It is easier to work with a small number of clusters than with a large number of raw data points.*

The philosophy of data clustering is to divide data into a (not necessarily fixed) number of clusters, with the elements in a cluster somehow *similar*. Data clustering is often applied to problems that deal with a large number of variables, and it is usually very difficult to determine the "proper" way to cluster data [Afifi and Clark 1984]. We apply data clustering in the following way:

- *Split the state into small, discrete units.* Our units correspond to geographic locations of interpolated census population measurements [Center for International Earth Science Information Network 2007].
- *Determine some partition of these units into clusters.* Note that the only variables present are the location and population of each unit.

After defining a method for ordering the preference of cluster arrays, we might suppose we are done with the problem: All that is left is to look at all possible cluster arrays and choose the best one. However, this turns out not to be feasible. Abramowitz and Stegun [1968] prove that the number of ways of sorting $n$ observations into $m$ groups is a Stirling number of the second kind:

$$S_m^{(n)} = \frac{1}{m!} \sum_{k=0}^{m} (-1)^{m-k} \binom{m}{k} k^n.$$

For instance, there are more than $10^{15}$ ways to sort 25 objects into 5 groups. We need an algorithmic process to determine an appropriate array of clusters.

# Cluster-Theoretic Districting

# The K-means Algorithm

## Standard Algorithm

The K-means algorithm is an iterative method for data clustering [Shapiro and Stockman 2001]. Let $D = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{R}^n$ be the data to be clustered, and let $S = \{\mathbf{s}_j\}_{j=1}^K$ be a set of seeds. Suppose that we desire to partition $D$ into $K$ clusters; let the $i$th cluster be denoted by $C_i$. Associate to the $i$th cluster a *geographical center*, denoted by $\mathbf{c}_i$. Given a distance function $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, the K-means algorithm proceeds as follows.

- **Initialization:** For all $C_i$, let $\mathbf{c}_i = \mathbf{s}_i$

- **Iteration:**
  - **Assign points to clusters:** For all $\mathbf{x} \in D$, associate $\mathbf{x}$ to a cluster $C_i$ whose center $\mathbf{c}_i$ minimizes $f(\mathbf{x}, \mathbf{c}_i)$.
  - **Update cluster centers:** Redefine $\mathbf{c}_i = \left(\sum_{\mathbf{x}\in C_i} \mathbf{x}\right) / \left(\sum_{\mathbf{x}\in C_i} 1\right)$.
  - **Repetition:** If updating cluster centers changes at least one cluster center, repeat the iteration step. Otherwise, stop.

## Weighted Algorithm

To generate districts of appropriate population, we add a weighting system to the standard algorithm. Let each cluster correspond to a legislative district. Let $D = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{R}^2$ be the set of census coordinates. Thus, $x \in D$ corresponds to the position of a population measurement. Define a population function $p : D \to \mathbb{R}$ such that $p_i$ is the population at the coordinates specified by $\mathbf{x}_i$. A cluster $C_j$ is defined by its points $\mathbf{x} \subset \mathbb{R}^2$, its center $\mathbf{c}_j \in \mathbb{R}^2$, and some *weight* $\alpha_j$. Define $f$ to be the Euclidean distance function in $\mathbb{R}^2$. Our weighted K-means algorithm proceeds as follows:

- **Initialization:** Using the *standard* K-means algorithm, assign points to clusters and centers to appropriate positions.
- **Iteration:**
  - **Assign points to clusters:** For all $\mathbf{x} \in D$, associate $\mathbf{x}$ to a cluster $C_i$ whose center $\mathbf{c}_i$ minimizes $f(\mathbf{x}, \mathbf{c}_i)$.
  - **Update cluster centers:** Redefine $\mathbf{c}_i = \left(\sum_{\mathbf{x}\in C_i} p_i\mathbf{x}\right) / \left(\sum_{\mathbf{x}\in C_i} p_i\right)$.
  - **Update cluster weights:** Redefine

$$\alpha_j = g\left(\sum_{\mathbf{x}_i \in C_j} p_i \alpha_j\right),$$

    where $g$ is defined below.
- **Repetition:** If the properties of the clusters are within tolerance, stop. Otherwise, repeat the iteration step.

By adjusting the weights, we control the growth or decay of the clusters. If the weight of a cluster increases, data points are more likely to be grouped in other clusters. Similarly, decreasing the weight helps to increase the population of a cluster. Thus, the weight function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is crucial in the performance of the algorithm. We define:

$$g(p, w) = w\sqrt{\frac{i}{i_0}} + w \cdot \frac{p}{p_0} \cdot \sqrt{1 - \frac{i}{i_0}},$$

where $i$ is the current iteration, $i_0$ is the maximum number of iterations, and $p_0$ is the desired population for each cluster. Towards the beginning of the algorithm, $i/i_0$ is small, causing the second term to dominate the weight function. As $i$ increases, the weight fluctuates less because the first term begins to dominate. This formula enables the weights to change rapidly at the beginning of the iterative process, causing the clusters to vary greatly between iterations. However, by the end of the algorithm, the weights do not change as readily, allowing stabilization over an optimal clustering. This is somewhat similar to simulated annealing, where initial negative actions allow the algorithm to escape local optima and the probability that a negative action is taken decreases over time.

# Splitline Algorithm [Smith 2007]

## Method

The idea behind the shortest splitline algorithm is quite simple:

- Start with the number of districts for the state. Divide that number in two as evenly as possible, using integers.
- Find the *shortest* line that divides the state into two parts such the ratio of their populations is the same as the ratio determined in the previous step.
- Repeat this process recursively on the subdivided parts until the number of parts is the same as the number of districts. At every step, the division is just a line, so the resulting districts have piecewise linear boundaries. Using the *shortest* line ensures that the districts will have a good compactness quotient.

## Demonstration

**Figure 3** is a demonstration of the splitline algorithm creating 5 districts from 15 people; there need to be 3 people in each district. The ratio 3:2 is the most balanced integer ratio that 5 can be divided into. At step 1, the algorithm divides the state into two regions with 9 and 6 people, the correct ratios for 3 and 2 districts. At step 2, it acts recursively on the 2 subdivisions. Thus, the region with 6 people is divided into regions with 3 each, with no more subdivision needed. The other region is divided into regions with 6 and 3 people. At the third and final step, the last region is split in two and the process is complete. By using the shortest line at each step, none of the shapes ends up with an unsatisfactory compactness quotient.

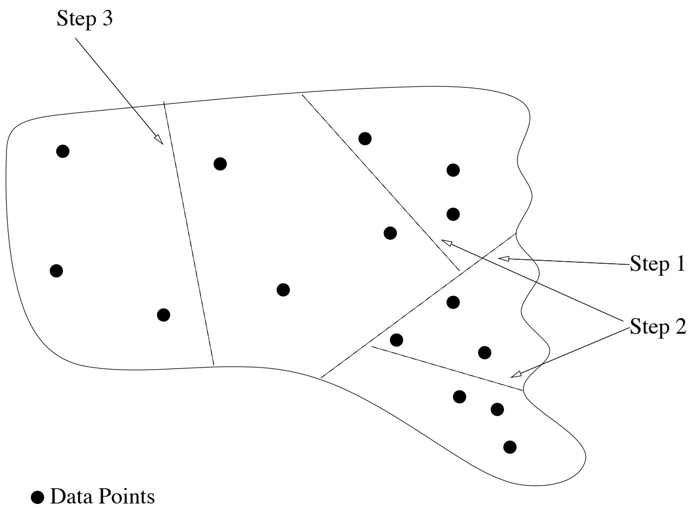



**Figure 3.** An illustration of the splitline method.

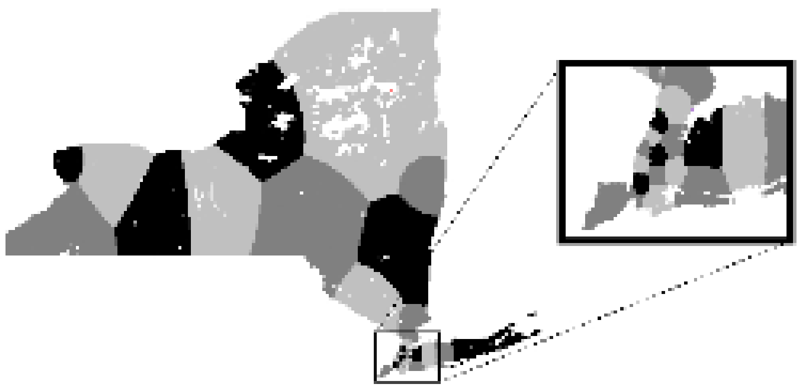


**Figure 4.** A proposed redistricting of New York, using the K-means algorithm.

# Districting of New York State

## K-means Algorithm

The results given by the K-means algorithm in **Figure 4** are generally quite good. Traditionally, when applying cluster-theoretic algorithms to redistricting, it is common practice to split off any regions with particularly high population density and apply the algorithm to those regions separately (see, for instance, Garfinkel and Nemhauser [1970]). This was not needed for the K-means algorithm: Even though the maximum population density of New York City is roughly 2,000 times the mean population density of the state of New York, the K-means algorithm produced results within our tolerance levels.

To confirm that the weighted K-means algorithm is an effective aid for determining districts, we also used the algorithm to redistrict Texas too. Texas is a good choice because it is large and contains a variety of population densities.

The K-means algorithm worked overall well with only a few districts outside our target tolerance.

## Splitline Algorithm

To obtain results within our desired tolerance, it was necessary to calculate the districts of New York City separately from the remainder of the state. A limitation of the splitline algorithm is that it does not guarantee contiguity of districts (see **Figure 5**). However it produces contiguous (and, furthermore, convex) districts for a convex state.

# Conclusions

We conclude that both the K-means algorithm and the splitline algorithm are viable methods for fair and simple redistricting. K-means produced much better results for New York: The greatest value of

$$\max_{\text{alldistricts}} \left( \frac{1 - \text{cluster population}}{\text{target population}} \right),$$

is no more than 2.5%. As an interesting note, while the unweighted K-means method clusters data into regions with piecewise-linear boundaries, inclusion of the weight function effectively rounds the boundaries of the produced districts. These rounder districts have superb compactness coefficients. K-means also has a visually appealing output and meets all other criteria.

The splitline algorithm results are not quite as satisfactory; however, we believe that this is a result of our implementation and not of the algorithm itself. Even our flawed version of splitline produced districts simpler than the current districts in New York. Our implementation could achieve districts with either even population or high compactness coefficients, but not both simultaneously. It is also difficult to enforce the contiguity requirement in regions with a highly irregular border. When the splitline algorithm is applied to states with convex boundaries, there are no discontiguities; furthermore, every district is convex. In the case of simple states, the splitline algorithm works well, perhaps even better than the K-means algorithm. Its intuitive simplicity is also likely to make shortest-splitline more appealing to the public.

Both K-means and splitline are deterministic: that is, when each algorithm is applied to a fixed problem, and all parameters are constant, the final result is unique. Some authors have expressed the opinion that any good districting algorithm is deterministic [Hayes 1996]. There is one human element involved in the K-means algorithm: The choice of seeds is made, in some sense, subjectively, by the person implementing the algorithm. This factor could be completely eliminated by randomly picking the seeds, but this is not the most desirable solution. Random seeds can produce solutions far from the global

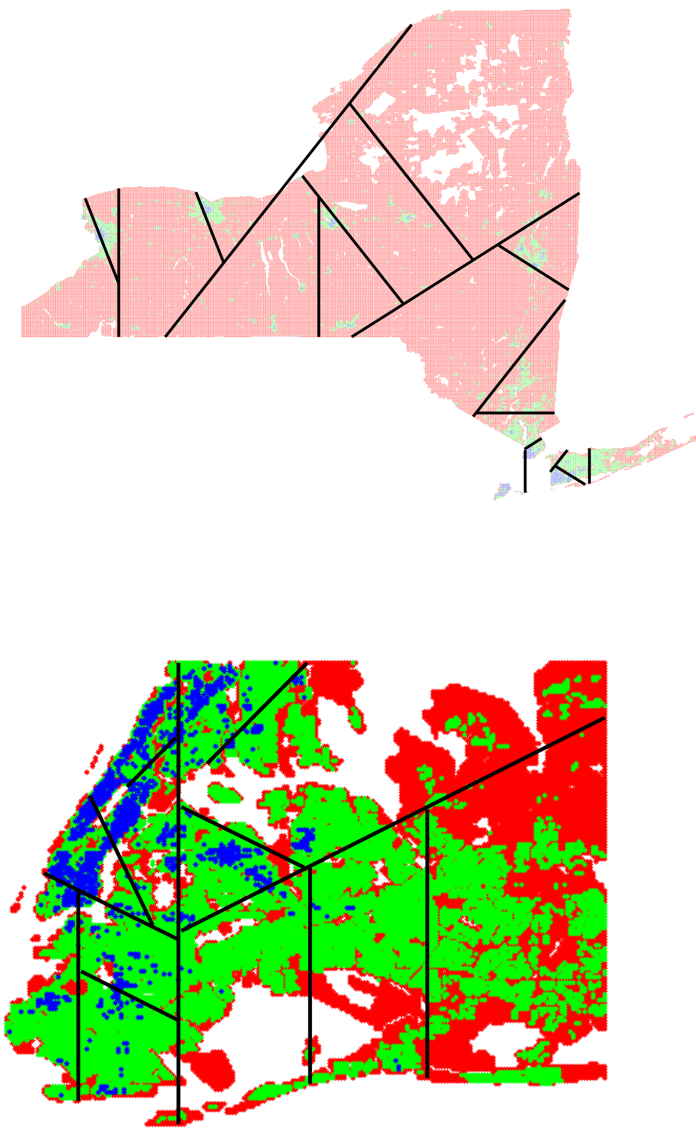


**Figure 5.** A proposed redistricting of New York, using the splitline algorithm and calculating the districts within New York City separately from the remaining districts.

optimum of the optimization function and can require many more iterations to get an answer within a given tolerance level. The natural choice is to use the approximate centers of existing districts as seeds. At first, this may seem contrary to our goal of reversing the effects of gerrymandering. A closer analysis of gerrymandering shows that this is not true. Gerrymandering relies on intricately carving districts based on data that are invisible to our algorithm—say, ethnicity, income level, or political affiliation.

The K-means algorithm clearly performs better on more-complex data sets. The splitline algorithm should not be abandoned, but our final recommendation is that

> *The K-means algorithm quickly and deterministically produces districts that are simple and fair, and applying this algorithm would produce a drastic improvement over current districts in any state.*

# References

Afifi, A.A., and Virginia Clark. 1984. *Computer-aided Multivariate Analysis*. Belmont, CA: Lifetime Learning Publications.

Anderberg, Michael R. 1973. *Cluster Analysis for Applications*. New York: Academic Press.

Abramowitz, M., and I.A. Stegun. 1968. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Washington, DC: U.S. Government Printing Office.

*Baker v. Carr*, 369 U.S. 186 (1962).

Backstrom, Charles H. 1986. The Supreme Court prohibits gerrymandering: A gain or a loss for the states? *The State of American Federalism* 17 (3): 1–33.

Bourke, Paul. 1988. Calculating the area and centroid of a polygon. `http://local.wasp.uwa.edu.au/~pbourke/geometry/polyarea/` . Accessed 11 February 2007.

Center for International Earth Science Information Network. 2007. Socioeconomic data and applications center. `http://sedac.cisesin.columbia.edu/` . Accessed 8 February 2007.

Folland, Gerald B. 2002. *Advanced Calculus*. Upper Saddle River, NJ: Prentice Hall.

Garfinkel, R.S., and G.L. Nemhauser. 1970. Optimal political districting by implicit enumeration techniques. *Management Science* 16 (8): B495–B508.

Hayes, Brian. 1996. Machine politics. *American Scientist* 84 (6): 522–526.

Shapiro, Linda G., and George C. Stockman. 2001. *Computer Vision.* Upper Saddle River, NJ: Prentice Hall.

Smith, Warren B. 2007. Examples of our unbiased district-drawing algorithm in action. `http://rangevoting.org/RangeVoting.html` . Accessed 12 February 2007.

Toobin, Jeffrey. 2003. The great election grab. *New Yorker* 79 (38) (8 December 2003): 63–80.

U.S. Department of the Interior. 2007. National Atlas of the United States. Printable Maps. Congressional Districts—110th Congress. `http://nationalatlas.gov/printable/congress.html` .

Young, H.P. 1988. Measuring the compactness of legislative districts. *Legislative Studies Quarterly* 13 (1): 105–115.

# Appendix: Definitions of Compactness

The following definitions of compactness are said in Young [1988] to be representative of those definitions favored in past and present scholarship.

As discussed by Young [1988], the following definitions of compactness are often used or cited in the literature.

- **Visual test.** A district is more compact if it appears to be more compact.
- **Roeck test.** Find the smallest circle containing the district and take the ratio of the district's area to that of the circle. This ratio is always between 0 and 1; the closer to 1, the more compact the district.
- **Schwartzberg test.** Construct the adjusted perimeter of the district by connecting with straight lines points on the district boundary where three or more constituent units (*i.e.*, census tracts) from any district meet. Divide the length of the adjusted perimeter by the perimeter of a circle with area equal to that of the district.
- **Length-width test.** Find a rectangle enclosing the district and touching it on all four sides, such that the ratio of length to width is a maximum. The closer the ratio to 1, the more compact the district.
- **Taylor's test.** Construct the adjusted perimeter of the district by connecting by straight lines those points on the district boundary where three or more constituent units (*i.e.*, census tracts) from any district meet. At each such point the angle formed is "reflexive" if it bends away from the district and "non-reflexive" otherwise. Subtract the number of reflexive from the number of non-reflexive angles and divide by the total number of angles. The resulting number is always between 0 and 1; the closer to 1, the more compact the district.

- **Moment of Inertia test.** Locate the geographical center $\mathbf{c}_i$ of each census tract $i$ in the district. Select an arbitrary point $\mathbf{x}$ and calculate the square of the distance from $\mathbf{x}$ to $\mathbf{c}_i$, multiplied by the population of tract $i$. The sum of these numbers is the district's moment of inertia about $\mathbf{x}$. The point that gives the minimum moment of inertia is the center of gravity of the district. The smaller the moment of inertia about the center of gravity, the more compact the district.

- **Boyce-Clark test.** Determine the center of gravity of the district and measure the distance from the center to the outside edges of the district along equally-spaced radial lines. Compare the percentages by which each radial distance differs from the average radial distance, and find the average of the percentage deviations over all radials. The closer the result is to $0$, the more compact is the district.

- **Perimeter test.** Find the sum of the perimeters of all the districts. The shorter the total perimeter, the more compact the districting plan.

Members of both Outstanding teams from the University of Washington in the Gerrymander Problem. Top row, from left: Sam Whittle, Aaron Dilley, Sam Burden, advisor Jim Morrow; bottom row: Lukas Svec, Wesley Essig, Nate Bottman. Not shown: Advisor Anne Greenbaum.

# Applying Voronoi Diagrams to the Redistricting Problem

Lukas Svec
Sam Burden
Aaron Dilley
University of Washington
Seattle, WA

Advisor: James Allen Morrow

## Summary

Gerrymandering is an issue plaguing legislative redistricting. We present a novel approach to redistricting that uses *Voronoi* and population-weighted *Voronoiesque* diagrams. Voronoi regions are contiguous, compact, and simple to generate. Regions drawn with Voronoiesque diagrams attain small population variance and relative geometric simplicity.

As a concrete example, we apply our methods to partition New York State. Since New York must be divided into 29 legislative districts, each receives roughly 3.44% of the population. Our Voronoiesque diagram method generates districts with an average population of $(3.34 \pm 0.74)\%$.

We discuss possible refinements that might result in smaller population variation while maintaining the simplicity of the regions and objectivity of the method.

## Introduction

Defining Congressional districts has long been a source of controversy in the U.S. Since the district-drawers are chosen by those in power, the boundaries are often created to influence future elections by grouping an unfavorable minority demographic with a favorable majority; this process is called *gerrymandering*. It is common for districts to take on bizarre shapes, spanning slim sections of multiple cities and crisscrossing the countryside in a haphazard fashion. The only lawful restrictions on legislative boundaries stipulate that districts must

*The UMAP Journal* 28 (3) (2007) 315–332. 

contain equal populations, but the makeup of the districts is left entirely to the district-drawers.

In the United Kingdom and Canada, districts are more compact and intuitive. The success of these countries in mitigating gerrymandering is attributed to turning over boundary-drawing to nonpartisan advisory panels. However, these independent commissions can take two to three years to finalize a new division plan, calling their effectiveness into question. It seems clear that the U.S. should establish similar unbiased commissions yet make some effort to increase the efficiency of these groups. Accordingly, our goal is to develop a small toolbox to aid in the redistricting process. Specifically, we create a model that draws legislative boundaries using simple geometric constructs.

# Current Models

The majority of methods for creating districts fall into two categories: ones that depend on a current division arrangement (most commonly counties) and ones that do not. Most fall into the former category. By using current divisions, the problem is reduced to grouping these divisions in a desirable way using a multitude of mathematical procedures. Mehrotra et al. [1998] uses graph partitioning theory to cluster counties to total population variation of around 2% of the average district size. Hess et al. [1965] use an iterative process to define population centroids, use integer programming to group counties into equally populated districts, and then reiterate the process until the centroids reach a limit. Garfinkel and Nemhauser [1970] use iterative matrix operations to search for district combinations that are contiguous and compact. Kaiser begins with the current districts and systematically swaps populations with adjacent districts [Hamilton 1966]. All of these methods use counties as their divisions since counties partition the state into a relatively small number of sections. This is necessary because most of the mathematical tools they use become slow and imprecise with many divisions. (This is the same as saying they become unusable in the limit when the state is divided into more continuous sections.) Thus using small divisions, such as zipcodes (which on average are one-fifth the size of a county in New York), becomes impractical.

The other category of methods is less common. Forrest's method continually divides a state into halves while maintaining population equality until the required number of districts is satisfied [Hamilton 1966; Hess et al. 1965]. Hale et al. create pie-shaped wedges about population centers; this creates homogeneous districts that all contain portions of a large city, suburbs, and less populated areas [Hamilton 1966]. These approaches are noted for being the least biased, since their only consideration is population equality and they do not use preexisting divisions. Also, they are straightforward to apply. However, they do not consider any other possibly important considerations for districts, such as geographic features of the state or how well they encompass cities.

## Developing Our Approach

Since our goal is to create new methods, we focus on creating district boundaries independently of current divisions. It is not obvious why counties are a good beginning point for a model: Counties are created in the same arbitrary way as districts, so they may also contain biases. Many of the division-dependent models end up relaxing their boundaries from county lines so as to maintain equal populations, which makes the initial assumption of using county divisions all the more unnecessary and also allows for gerrymandering if the relaxation method is not well regulated.

Treating the state as continuous (i.e., without considering pre-existing divisions) does not lead to any specific approach. If the Forrest and Hale et al. methods are any indication, we should focus on keeping cities within districts and introduce geographical considerations. (These conditions do not have to be considered if we treat the problem discretely, because current divisions, like counties, are probably dependent on prominent geographical features.)

> **Goal:** *Create a method for redistricting by treating the state continuously. We require the final districts to contain equal populations and to be contiguous. Additionally, the districts should be as simple as possible and optimally take into account important geographical features.*

# Notation and Definitions

- **Compactness:** One way to look at compactness is the ratio of the area of a bounded region to the square of its perimeter:

$$C_R = \frac{A_R}{p_R^2} = \frac{1}{4\pi}\,\mathcal{Q},$$

  where $C_R$ is the compactness of region $R$, $A_R$ is the area, $p_R$ is the perimeter and $\mathcal{Q}$ is the isoperimetric quotient. We do not explicitly use this equation, but we keep this idea in mind when we evaluate our model.

- **Contiguous:** A set $R$ is contiguous if it is pathwise-connected.
- **Decomposition:** Process in which a state is divided into regions using Voronoi and Voronoiesque diagrams.
- **Degeneracy:** The number of districts represented by one generator point.
- **Generator point:** A node of a Voronoi diagram.
- **Population center:** A region of high population density.
- **Simple:** Simple regions are compact and convex.

- **Voronoi diagram:** A partition of the plane with respect to $n$ nodes such that points are in the same region with a node if they are closer to that node than to any other point.
- **Voronoiesque diagram:** A variation of the Voronoi diagram based on equal masses of the regions.

# Theoretical Evaluation of our Model

How we analyze our model's results is tricky, since there is disagreement in the literature on key issues. *Population equality* is well-defined. By law, the populations within districts have to be the same to within a few percent of the average population per district.

Creating a successful redistricting model also requires *contiguity*. In accordance with state law, districts need to be pathwise connected. This requirement is meant to maintain locality and community within districts. It does not, however, restrict districts from including islands if the island's population is below the required population level for a district.

Finally, there is a desire for the districts to be *simple*. This is the most ambiguous criterion. Most commonly, simplicity of districts is gauged by *compactness* (whcih by no means leads to a unique definition of simple). Taylor [1973] defines simple as a measure of divergence from compactness due to indentation of the boundary and gives an equation relating the nonreflexive and reflexive interior angles of a region's boundary. Young [1988] provides seven more measures of compactness. The *Roeck* test is a ratio of the area of the largest inscribable circle in a region to the area of that region. The *Schwartzberg* test takes ratio between the adjusted perimeter of a region to a the perimeter of a circle whose area is the same as the area of the region. The *moment of inertia* test measures relative compactness by comparing "moments of inertia" of different district arrangements. The *Boyce-Clark* test compares the difference between points on a district's boundary and the center of mass of that district, where zero deviation of these differences is most desirable. The *perimeter test* compares different district arrangements by computing the total perimeter of each; a smaller perimeter means more compact. Finally, there is the *visual* test, which decides simplicity based on intuition [Young 1988].

Young [1988] notes that "a measure [of compactness] only indicates when a plan is more compact than another." Thus, not only is there no consensus on how to analyze redistricting proposals, it is also difficult to compare them.

Finally, we remark that the above list only constrains the shape of generated districts without regard to any other potentially relevant features—for example, how well populations are distributed or how well the new district boundaries conform with other boundaries, like counties or zipcodes. Even with this short list, we are not in a position to define simplicity rigorously. What we can do, however, is argue for which features of proposed districts are simple and which are not. This is in line with our goal, since this list can be provided to

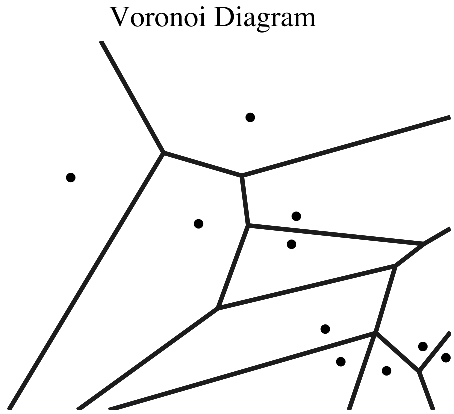



**Figure 1.** Illustration of Voronoi diagram generated with Euclidean metric. Note the compactness and simplicity of the regions.

a districting commission who decide how relevant those simple features are. *We do not explicitly define simple, we loosely evaluate simplicity based on overall contiguity, compactness, convexity, and intuitiveness of the model's districts.*

# Method Description

Our approach depends heavily on using Voronoi diagrams. We begin with a definition, its features, and motivate its application to redistricting.

## Voronoi Diagrams

A Voronoi diagram is a set of polygons, called *Voronoi polygons*, formed with respect to $n$ generator points in the plane. Each generator $p_i$ is contained within a Voronoi polygon $V(p_i)$ with the following property:

$$V(p_i) = \{q \mid d(p_i, q) \leq d(p_j, q),\ i \neq j\},$$

where $d(x, y)$ is the distance from point $x$ to point $y$. That is, $V(p_i)$ is the set of all points $q$ that are closer to $p_i$ than to any other $p_j$. The Voronoi diagram is the set $\mathbf{V} = \{V(p_1), \ldots, V(p_n)\}$ of all Voronoi polygons (see **Figure 1**).

Of the many possible metrics, we use the three most common:

- Euclidean metric: $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$
- Manhattan metric: $d(p, q) = |x_p - x_q| + |y_p - y_q|$
- Uniform metric: $d(p, q) = \max\{|x_p - x_q|, |y_p - y_q|\}$

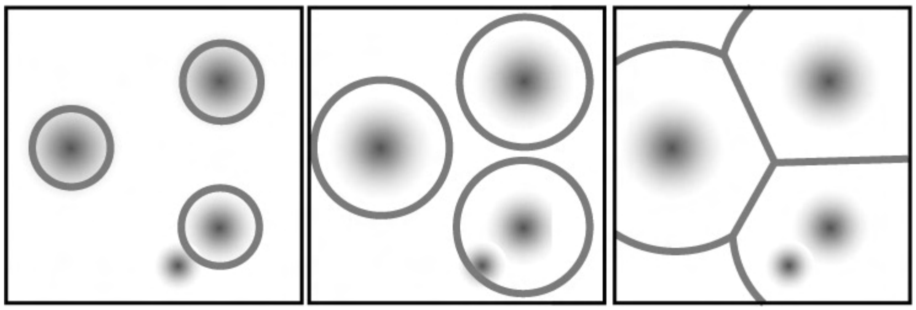


**Figure 2.** The process of "growing" a Voronoiesque diagram with respect to a population density. Only three three generator points are used. Figures from left to right iterate with time.

### Useful Features of Voronoi Diagrams

- The Voronoi diagram for a set of generator points is unique and produces contiguous polygons.
- The nearest generator point to $p_i$ determines an edge of $V(p_i)$
- The polygonal lines of a Voronoi polygon do not intersect the generator points.
- When working in the Euclidean metric, all regions are convex.

The first property tells us that regardless of how we choose the generator points, we generate unique regions. The second property implies that each region is defined in terms of the surrounding generator points, while in turn each region is relatively compact. These features of Voronoi diagrams effectively satisfy two of the three criteria for partitioning a region: contiguity and simplicity.

## Voronoiesque Diagrams

The second method that we use to create regions is a modification of Voronoi diagrams; we call them *Voronoiesque diagrams*. One way to visualize the construction of Voronoi diagrams is to imagine shapes (determined by the metric) that grow radially outward at the same constant rate from each generator point. In the Euclidean metric, these shapes are circles; in the Manhattan metric, they are diamonds; in the Uniform metric, they are squares. As the regions intersect, they form the boundary lines for the regions.We define Voronoiesque diagrams to be the boundaries defined by the intersections of these growing shapes. The fundamental difference between Voronoi and Voronoiesque diagrams is that Voronoiesque diagrams grow the shapes radially outward at a constant rate, even though their radial growth is defined with respect to a real function on a subset of $\mathbb{R}^2$ (representing the space on which the diagram is being generated) (see **Figure 2**).

More rigorously, we define a Voronoi diagram to be the intersections of the $\mathcal{V}_i^{(t)}$s, where $\mathcal{V}_i^{(t)}$ is the Voronoiesque region generated by the generator point

$p_i$ at iteration $t$. With every iteration,

$$\mathcal{V}_i^{(t)} \subset \mathcal{V}_i^{(t+1)}$$

and

$$\int_{\mathcal{V}_i} f(x,y)\, dA = \int_{\mathcal{V}_j} f(x,y)\, dA$$

for all $\mathcal{V}_i$, $\mathcal{V}_j$ representing different regions. The manner in which the $\mathcal{V}_i^{(t)}$s are grown radially from one iteration to the next is determined by the metric used.

What's useful about Voronoiesque diagrams is that their growth can be controlled by requiring that the area under the function $f$ for each region is the same at every iteration. In our model, we take $f$ to be the population distribution of the state. Thus, the above equation is a statement of population equality. Also, when $f$ is constant, the regions grow at a constant rate until they intersect, so the resulting diagram is Voronoi.

The final consideration for using Voronoiesque diagrams is determining the location for generator points.

## Determining Generator Points

Our first approach is to place generator points at the $m$ largest set of peaks that are well distributed throughout the state, where $m$ is the required number of districts. Doing this keeps larger cities within the boundaries that we will generate with Voronoi or Vornoiesque diagrams and makes sure that the generator points are well dispersed throughout the state. One problem that arises is when a city is so large that for districts to hold the same number of people, a city must be divided into districts. A perfect example is New York City, which contains enough people for 13 districts.

Our second approach is to choose the largest peaks in the population distribution and assign each peak a weight. The weight for each generator point is the number of districts into which the population surrounding that peak needs to be divided. We call this weight the *degeneracy* of the generator point. We begin assigning generator points to the highest populated cities with their corresponding degeneracies until the sum of all the generator points and their respective degeneracies equals $m$. As we will see with New York, this method works well.

## Creating Regions

Once we have our generator points, we generate our diagrams:

- First, generate the diagram using the given generator points.
- Second, assign each generated region, called a *subdivision*, with some degeneracy $r$, create $r$ new generator points within that subdivision by finding the $r$ largest population density peaks.

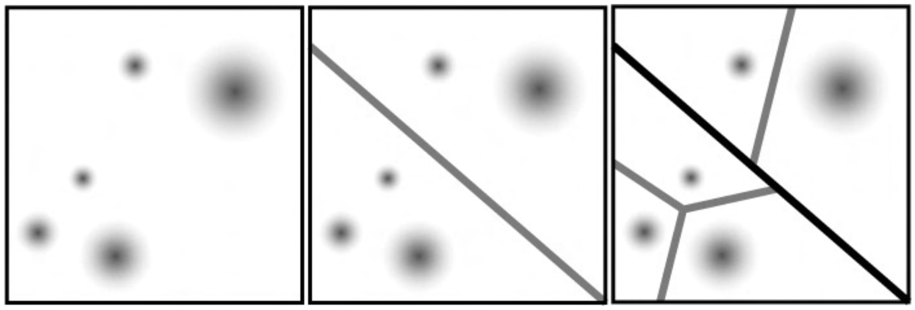


**Figure 3.** Creating divisions by first subdividing the map. Left: Population density distribution of hypothetical map with five desired districts. Middle: A subdivision of the map into two regions generated from two unshown generator points. Right: Final division of each subregion from the middle figure into desired final divisions.

- Third, create another diagram within that subdivision using the $r$ generator points.

We call this three-step process a *decomposition* (**Figure 3**).

# Redistricting in New York State

We begin by explaining our choice of generator points at population centers. Then we describe several methods for generating Voronoi and Voronoiesque diagrams from these points and present the corresponding results. Finally, we discuss how to use these diagrams to create political districts for New York.

## Population Density Map

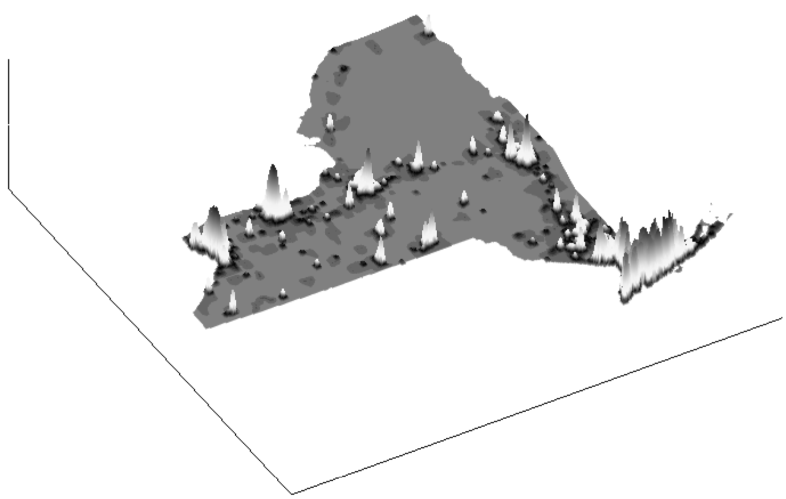
The U.S. Census Bureau maintains a database of census tract-level population statistics; when combined with boundary data for each tract, it is possible to generate a density map with a resolution no coarser than 8,000 people per region. Unfortunately, our limited experience with the Census Bureau's data format prevented us from accessing these data directly, and we contented ourselves with a 792-by-660 pixel approximation to the population density map [Irwin 2006].

We loaded this raster image into Matlab and generated a surface plot where height represents population density. To remove artifacts introduced by using a coarse lattice representation for finely-distributed data, we applied a 6-pixel moving average filter to the density map. The resulting population density is shown in **Figure 6**.

### Limitations of the Image-Based Density Map

The population density image that we used yields a density value, for every one-third of a square mile, from the following set (measured in people per

**a.** White areas represent high population density over New York.



**b.** Angled View: Clearer view of population distribution over New York.

**Figure 4.** New York State population density map. Data from 792-by-660 pixel raster image at Irwin [2006]; color and height indicate the relative population density at each point.

square mile):

$$\{0,\ 10,\ 25,\ 50,\ 100,\ 250,\ 500,\ 1000,\ 2500,\ 5000\}\,.$$

This provides a decent approximation for regions with a density smaller than 5,000 people/sq. mi. However, the approximation will break down at large population centers; New York City's average population density is 26,403 people/sq. mi. [Wikipedia 2007].

## Selecting Generator Points

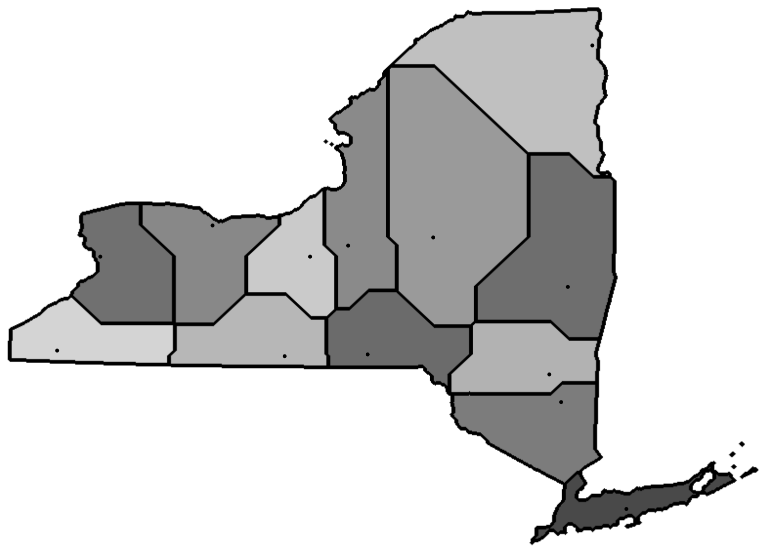
Our criteria for redistricting stipulate that the regions must contain equal populations. New York State must be divided into 29 congressional districts, so each region must contain 3.44% of the state's population. Since a state's population is concentrated primarily in a small number of cities, we use local maxima of the population density map as candidates for generator points.

If we were to simply choose the highest 29 peaks from the population density map as generator points, they would be contained entirely in the largest population centers and would not be distributed evenly over state. For the largest population centers, we assign a single generator point with a degeneracy. After all the generator points have been assigned, we generate a Voronoi diagram for the state. Then, we return to the regions with degenerate generator points and repeat the process of finding generator points for that region and generate a Voronoi diagram from them. See **Figure 3** for an illustration of the decomposition before and after subdivision.

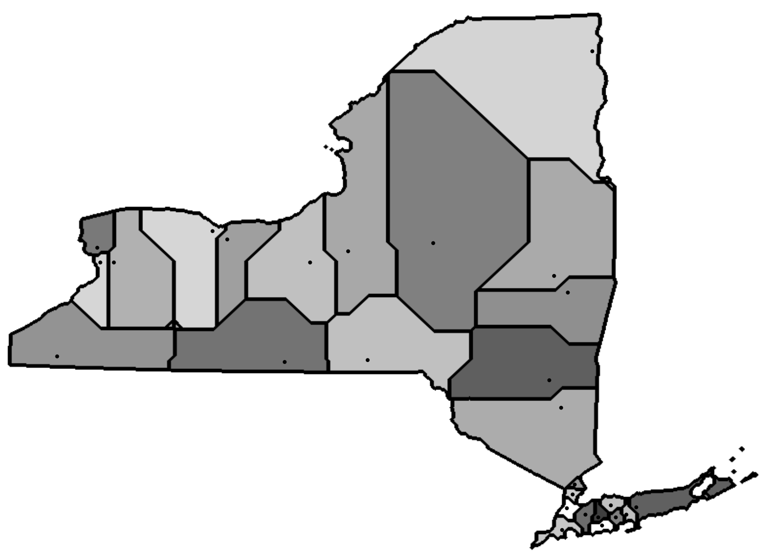
We subdivide the region around New York City into 12 subregions, Buffalo into 3 subregions, and Rochester and Albany into 2 subregions. This roughly corresponds to the current allocation, where New York City receives 14 districts, Buffalo gets 3, and Rochester and Albany both get roughly 2. New York City's population is underestimated, since the average density there far exceeds our data's density range. With a more detailed data set, our method would call for more-accurate degeneracy values for each subdivision.

## Applying Voronoi Diagrams to NY

The simplest method that we consider for generating congressional districts is to generate the Voronoi diagram from a set of generator points. We achieve this by iteratively "growing" regions outward with the function $f$ constant. That way, the regions grow at a constant rate, and hence the resulting diagram is Voronoi. A region's growth is limited at each step by its radius in a certain metric; we considered the Euclidean, Manhattan, and Uniform metrics. Once the initial diagram has been created, a new set of generator points for subdivisions is chosen and those regions are subdivided using the same method. The decomposition of New York is seen in **Figures 5–6**.

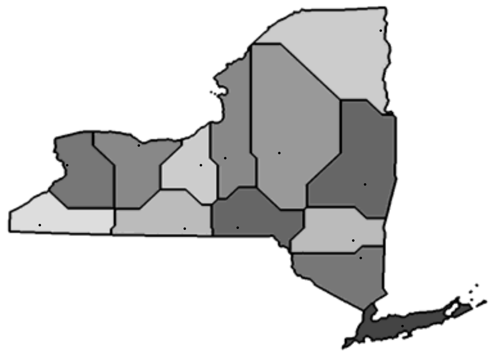


**a.** Regions created using the Manhattan metric before subdivisions are implemented.

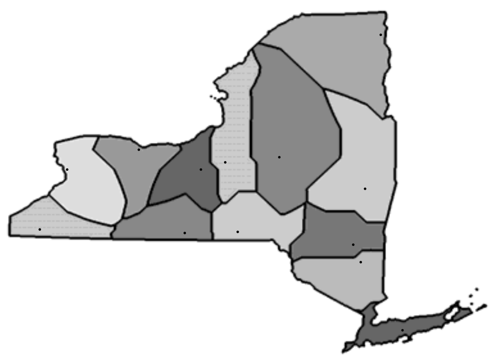


**b.** Regions created using the Manhattan metric after subdivisions are implemented. Subdivisions are created in New York City, Buffalo, Rochester, and Albany.

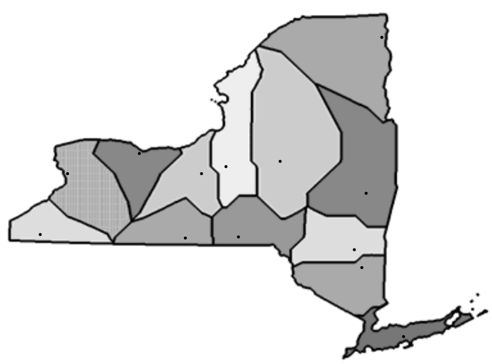
**Figure 5.** Implementation of Voronoi diagrams with the Manhattan metric. in three steps: assigning degeneracies to generator points, using the points to generate regions, and creating subregions generated by degenerate points. Only the last two steps are depicted. (Dots in each region represent generator point locations.)



**a.** Regions created using the Manhattan metric before subdivisions. Average Population $= (3.5 \pm 2.2)\%$.



**b.** Regions created using the Euclidean metric before subdivisions. Average Population $= (3.7 \pm 2.6)\%$.



**c.** Regions created using the Uniform metric before subdivisions. Average Population $= (3.7 \pm 2.6)\%$.

**Figure 6.** Voronoi diagrams generated with three distance metrics before subdivision of densely populated regions. (Dots in each region represent generator point locations.)

Each metric produces relatively simple districts, though the Manhattan metric has simpler boundaries and yields a slightly smaller population variance between regions.

## Applying Voronoiesque Diagrams to NY

Though our simple Voronoi diagrams produced simple regions with a population mean near the desired value, the population variance between regions is enormous. In this sense, the simple Voronoi decomposition doesn't meet one of the main goals. However, the Voronoi regions are so simple that we prefer to augment this method with population weights rather than abandon it entirely.

**Figure 7** shows the result of this decomposition, along with exploded views of the two regions which were subdivided more than twice in the refinement stage of the diagram generation. The population in each region varies from 2.44% to 6.15%.

## Precisely Defining Boundary Lines

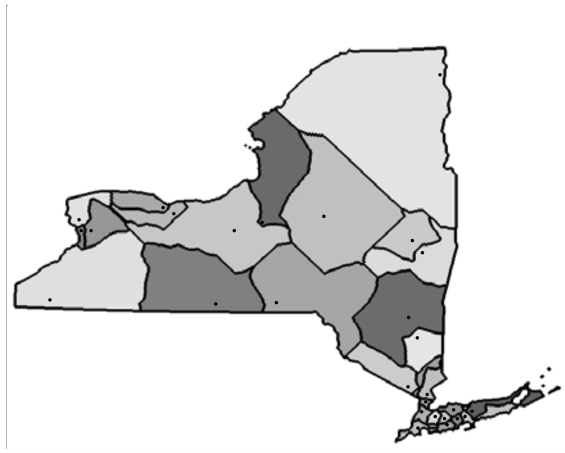
It is not satisfactory to say that the regions created by our models should define the final boundary locations. At least, boundaries should be tweaked so that they don't accidentally divide houses into two districts. However, given the scale at which the Voronoi and Voronoiesque diagrams were drawn, it seems reasonable to assume that their boundaries could be modified to trace existing boundaries—like county lines, zipcodes, or city streets—without changing their general shape or average population appreciably. As an example, the average area of a zipcode in New York state is 10 sq. mi. and roughly 200 city blocks per square mile in Manhattan, while the minimum size of one of our Voronoi regions is 73 sq. mi. and the average size is 2,000 sq. mi. Therefore, it seems reasonable that we could approximate the boundaries of our Voronoi and/or Voronoiesque diagrams by pre-existing boundaries.

# Analysis

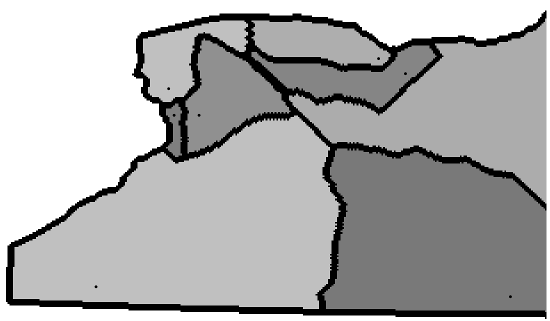
## New York State Results

In terms of simplicity of generated districts, our Voronoi-diagram method is superior, particularly when applied with the Manhattan metric: The generated regions are contiguous and compact, while their boundaries—being polygonal—are about the simplest that could be expected. However, this method falls short in achieving equal population distribution among the regions.

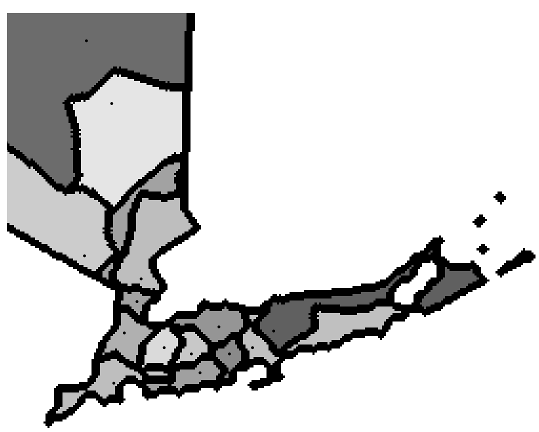
When we modify the Voronoi diagram method to generate population-weighted Voronoiesque regions, we cut the population variance by a factor



**a.** Overall New York Voronoiesque regions.



**b.** Exploded view of regions around Buffalo.



**c.** Exploded view of regions around Long Island.

**Figure 7.** Districts created by the Voronoiesque diagram for New York. Average population per region $= (3.34 \pm 0.74)\%$. (Dots in each region represent generator point locations.)

of four—from $\pm 2.8\%$ to $\pm 0.7\%$—while suffering a small loss in the simplicity of the resulting regions. In particular, regions in the Voronoiesque diagrams appear to be less compact, and their boundaries are more complicated, than their Voronoi diagram counterparts, though contiguity is still maintained.

Any implementation of a diagram generated from either method would have to make small localized modifications to ensure that the district boundaries make sense from a practical perspective. Though this would appear to open the door to politically-biased manipulation, the size of the necessary deviations (on the order of miles) is small enough when compared to the size of a Voronoi or Voronoiesque region (on the order of tens or hundreds of miles) to make the net effect of these variations insignificant. Therefore, though we have provided only a first-order approximation to the congressional districts, we have left little room for gerrymandering.

# General Results

## Population Equality

The largest problem with this requirement occurs when we try to make regions too simple. Typically, our Voronoi method has the most room for error here. If a state has high population density peaks with a relatively uniform decrease in population density extending away from each peak, then the regions will differ quite a bit. This is because in this situation, ratios of populations are then roughly equal to the ratios of areas between regions. However, our final method focuses primarily on population, so equality is much easier to regulate.

## Contiguity

Contiguity problems arise if the state itself has little compactness, like Florida, or has some sort of oceanic sound, like Washington. The first two methods focus on population density without acknowledging the boundaries of the state. So it's possible for a district to be divided by a geographic obstruction, such as a body of water or a mountain range. Our final method fixes this by growing in increments, which allows for regions to grow not over but around specified obstacles.

## Compactness

The Voronoi diagram method creates convex regions. Though the Voronoiesque method cannot guarantee convexity, the form of a region is similar in shape and size to the Voronoi region. A nice property of Voronoi regions is that we can make slight adjustments to the boundaries while still maintaining convexity (see below). This is good for taking population shifts across districts into account between redistrictings.

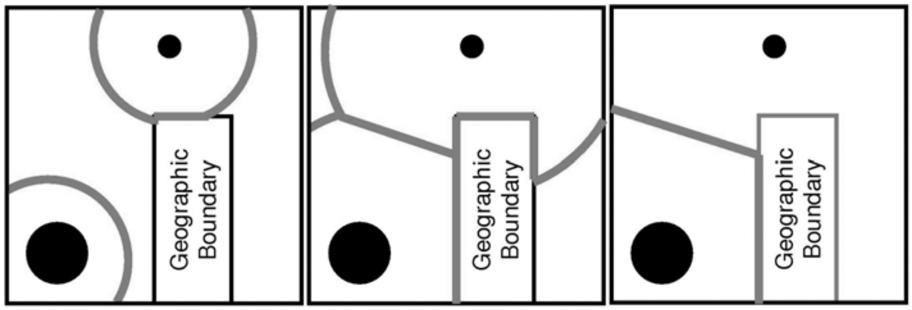



**Figure 8.** Illustration of Voronoiesque diagram generation that takes geographic obstacles into account.

# Improving the Method

## Boundary Refinement

The Voronoi approach is good at generating polygonal districts but not as successful at maintaining population equality. One improvement is vertex repositioning. Adjacent districts generated by this method all share a vertex common to at least three boundaries. From this vertex extends a finite number of line segments that partially define the boundaries of these adjacent regions. Connecting the endpoints of these segments yields a polygon. Now we are free to move the common vertex anywhere in the interior of this polygon while still maintaining convexity; we can redraw boundaries between regions to equalize population size.

In the Voronoiesque method, too, there are ways to adjust population inequality: Looking at the region with the lowest population, systematically increase the area of the low-population regions while decreasing the area of the neighboring high-population regions.

## Geographic Obstacles

Our methods don't consider geographic features such as rivers, mountains, canyons, etc. The Voronoiesque method, however, has the potential to implement these features. The same algorithm that detects intersections between Vornoiesque regions can detect a defined geographic boundary and stop growing in that direction. An illustration of this idea is shown in **Figure 8**.

# Conclusion

Our model requires the use of only a state's population distribution but as an option can incorporate county, property, and geographic considerations.

Our Voronoiesque model satisfies our proposed goal. We supply a model for a redistricting committee to generate district boundaries that are simple, contiguous, and produce districts with equal populations. In particular, Voronoi-esque diagrams redistrict New York very well. What is particularly attractive about our methods is that generating the districts is intuitive and accessible to the general public. The computer generation process takes less than 10 seconds.

# References


Garfinkel, R.S., and G.L. Nemhauser. 1970. Optimal political districting by implicit enumeration techniques. *Management Science* 16 (8): B495–B508.

Hamilton, Howard D. 1966. *Reapportioning Legislatures*. Columbus, OH: Charles E. Merril Books, Inc.

Hess, S.W., J.B. Weaver, H.J. Siegfeldt, J.N. Whelan, and P.A. Zitlau. 1965. Nonpartisan political redistricting by computer. *Operations Research* 13 (6): 998–1006.

Irwin, Jim. 2006. `http://upload.wikimedia.org/wikipedia/en/e/e2/New_York_Population_Map.png` .

Mehrotra, Anuj, Ellis L. Johnson, and George L. Nemhauser. 1998. An optimization based heuristic for political districting. *Management Science* 44 (8): 1100–1114.

Taylor, Peter J. 1998. A new shape measure for evaluating electoral district patterns. *American Political Science Review* 67 (3): 947–950.

________. 2007. Census 2000 Summary File 1 [New York]. `http://www2.census.gov/census_2000/datasets/Summary_File_1/New_York/` .

U.S. Census Bureau. 2005. 2005 first edition tiger/line data, Feb. 2007.

Wikipedia. 2007. New York City. `http://en.wikipedia.org/wiki/New_York_City` .

Young, H.P. 1988. Measuring the compactness of legislative districts. *Legislative Studies Quarterly* 13 (1): 105–115.

Aaron Dilley.

Lukas Svec, advisor Jim Morrow, and Sam Burden.

# Boarding at the Speed of Flight

Michael Bauer
Kshipra Bhawalkar
Matthew Edwards
Duke University
Durham, NC

Advisor: Anne Catlla

## Summary

We seek an efficient method for boarding a commercial airplane that accommodates unpredictable human behavior, with a framework that allows us to compare and contrast different procedures. Passenger dependencies, bottlenecks, and the rate of interferences are critical factors for airplane boarding time.

Boarding without seating assignments is fastest, since each person is in the correct order for their flexible seat choice; it removes all interferences and makes the boarding time depend solely on the entrance rate of passengers into the plane. Hoping to emulate the performance of this method, which we call "random greedy," we design a new algorithm to model its average seating order: the parabola boarding scheme, which breaks the plane into parabola-shaped zones.

We use a discrete-time simulation engine to model current boarding schemes as well as the parabola and random greedy algorithms. The zone-boarding schemes outside-in, pyramid, and parabola are almost identical in performance; back-to-front and alternating rows are significantly worse.

We examine the effects of scheme-independent parameters on boarding time. Ensuring a fast rate of people entering the plane and fast luggage stowage are both critical; an airline could reduce boarding time by improving either of these regardless of boarding scheme.

By varying both the rate of people entering the plane and time to stow luggage, we find a correlation between average boarding time and the difference between best and worst times. The random greedy algorithm has the smallest difference; outside-in, pyramid, and parabola have equal differences. Faster boarding algorithms are also more reliable and allow for tighter scheduling.

*The UMAP Journal* 28 (3) (2007) 333–352. 

The best boarding algorithms do not have assigned seating. If, however, an airline feels that assigned seating is mandatory for customer satisfaction, then any of outside-in, pyramid, or parabola will result in a consistently fast boarding time with minimum deviation from average times and will be a marked improvement over the traditional back-to-front boarding method.

# Introduction

Short of a single minor detail, the airplane boarding problem would be easily solved using a very simple algorithm. Given his performance in the the film *Snakes on a Plane* [2007], we know Samuel L. Jackson is an optimal de-boarder of snakes from planes. Assuming that he maintains equal effectiveness with people, simply invert his role and you have an optimal passenger boarding algorithm. We model people as snakes, play the film in reverse, and determine the boarding time!

# Conventions

## Terminology

- **Passenger**: A person traveling on the plane who is not part of the crew.
- **Boarding Scheme**: An assignments of zones or groups according to which passengers board the plane.
- **Interference**: An event in which a passenger cannot progress towards their seat because of another passenger blocking the way.

## Variables

- $C$: the number of columns in the plane, which is also the number of seats in a row.
- $R$: the number of rows in the plane. For the most part, we ignore or treat in a different manner distinctions between classes of seating.
- $B$: the time for a person to stow luggage, assumed to be constant in our preliminary analysis but allowed to vary in the simulation.
- $v$: the walking speed of passengers, constant.
- $s$: the time for an already-seated passenger to get up and get out into the aisle to let another passenger pass, constant for our preliminary analysis but variable in the simulation.

- $\lambda$: the rate at which people enter the plane through the main door, constant since any variability among passengers is mitigated by walking down the jet-bridge to the plane.

# Assumptions

- Passengers with physical limitations, families with infants, and passengers advanced in years board the plane before other passengers. The time for this "pre-boarding" is a constant overhead that airlines cannot avoid.
- First-class passengers board separately.
- All passengers during general boarding walk at the same speed, limited more by the environment (aisle size, people in the way) than by physical capacity. Passengers board and walk independently, that is, no groups wait for one another. Family members are assigned seats next to one another.
- We confine our analysis to the interior of the plane, ignoring terminal effects beyond requiring that gate agents supply passengers at a certain rate. If the plane cannot "process" passengers quickly enough, they queue in the jet-bridge. The interior of the plane is regular and symmetric, with all rows of equal size.
- All planes fly at maximum capacity and all passengers are present when their zone is called, which they follow obediently. Empty seats only speed up the process. Late or noncompliant passengers can be accounted for by adding a time overhead.
- We confine our recommendations and analysis to methods that do not overly alter the status quo. We analyze ticketless methods for comparison but seek the best boarding method for ticketed contexts. We further consider only zone-based boarding calls, assuming that is logistically impossible to require passengers to line up in any verifiable order.

# Motivation and Subproblems

What if all variables involving passenger boarding could be controlled? How would we schedule the boarding optimally? We would use a modified version of the outside-to-inside method. We first order the passengers into groups of equal size $R$ by the following set of criteria in descending order of priority:

- Individual location in row: Window has highest priority, aisle has least.
- Side of plane: left side of plane has priority over right side.

- Row number: Rows in back have priority over those in front.

The following algorithm then boards the plane optimally:

> Each group proceeds down the aisle until each person reaches their row (since people are in order, they all reach their rows simultaneously). They step into the first seat in their row and begin stowing their luggage. During this time, the next group commences down the aisle. The only time when a group might stall in the aisle is if $B > 2R/v$, in which case every other group must wait in the aisle for $B - 2R/v$ seconds. (This accounts for the additional term in the second part of **(1)** below.)

The ideal boarding algorithm places a lower bound on the time to board an airplane:

$$\text{Ideal boarding time} = \begin{cases} C\,\dfrac{R}{v} + B, & B \leq 2\,\dfrac{R}{v}; \\[2ex] C\,\dfrac{R}{v} + B + \left(d\frac{C}{2} - 1\right)(B - 2R), & B \geq 2\,\dfrac{R}{v}. \end{cases} \qquad \textbf{(1)}$$

Key points about the operation of the algorithm are:

- The main aisle is continuously busy unless passengers have to wait for people in their row to finish stowing luggage.
- Passengers are "pipelined" to minimize the blocking effect of stowing luggage.

Imperfect ordering forces us to consider the following:

- **Random orderings**: How out of order are people and how does this impact other dependencies in boarding?
- **Flow rates**: How long does it take people to enter the plane and walk down the aisle without blocking it?
- **Luggage**: How large is the luggage and how long does it take to stow?

All of these introduce dependencies into the system. Randomness prevents us from determining the occurrence or duration of these dependencies and therefore forces us to design boarding schemes capable of tolerating their effects.

One way to remove dependencies is to force people to continue moving as far back in the plane and over in a row as long as they don't get blocked. We return to this random greedy approach later, since it represents the intuitive motivation for our best airplane boarding scheme.

## Predicting Bottlenecks with Queuing Theory

One model for airplane boarding is a stochastic process, a collection of random variables that must take on a value at every state, with states indexed by a parameter (in our case, time) [Trivedi 2002]. Queuing theory deals with analyzing how the random variables in stochastic processes interact. Traditionally, queuing theory is used to determine the average throughput of a system. While the plane boarding problem does not possess a quantity directly corresponding to throughput, we gain a better understanding of bottlenecks and their effects by using this approach.

We partition the plane into a series of queues. We place a "processor" at each row. This processor corresponds to a passenger making a decision at this point either to keep walking or to stop and enter their row. Each processor has a queue that stores passengers. A queue has size 1 and if full will stop the processor feeding it; this would represent people backing up if someone stops in the aisle. A diagram for this layout is in **Figure 1**.

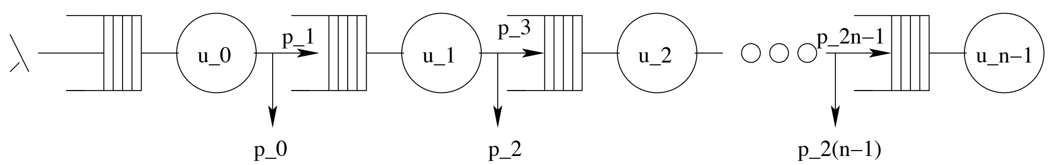



**Figure 1.** A queuing-theory model of airplane boarding.

In **Figure 1**, $u_k$ is the processing rate, the average walking speed of passengers. Each $p_k$ represents some probability with which passengers divert into the their rows or continue walking in the aisle. In some cases, people will take longer to get into their rows, depending upon how long it takes to stow their luggage. The processor associated with that row then takes longer to process that job, causing the flow of people through the aisle to stall. Downfalls of this model are that all passengers are eventually supposed to leave the system (i.e., get into their seats) and it doesn't accurately reflect that each row should only ever hold $C$ passengers; so we do not use the queuing theory model as our main model. However, it gives us useful knowledge concerning bottlenecks in the aisle.

To convert the open system shown of **Figure 1** to a closed-form system that can be solved by queuing theory, we use Jackson's Theorem (no, not Samuel L. Jackson again!) [1957]:

> *An open system can be represented by a feedback loop if the rate of processing at each processor is augmented proportional to the rate of flow prior to that processor.*

In our application, all rates depend on $p_0$ because the rate into the next queue depends on the output of the previous processor, so terms cancel under our assumptions for the value of any $p_j$. Using this theorem, we can redraw our airplane model as in **Figure 2**. The closed form allows us to determine the

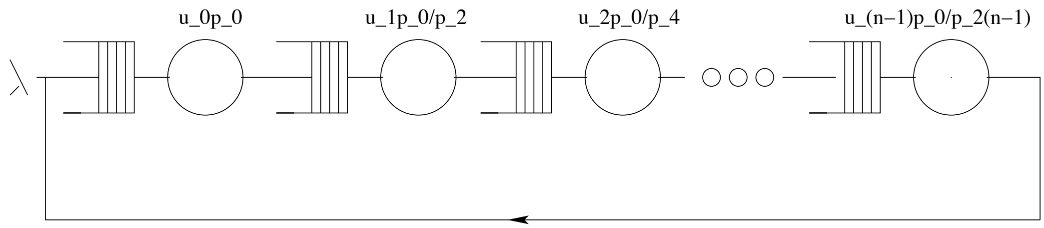



**Figure 2.** A closed-form queuing model of airplane boarding.

probability of having a given number of passengers at a specific node at a given time. We let $\rho(k_0, k_1, \ldots, k_{n-1})$ be the probability of $k_i$ people in position $i$ in the aisle. Conceptually, this implies that we have an $n$-dimensional state space, since the number of passengers at each node is potentially different.

We now write down conditions that $\rho$ must satisfy and use these to find an equation for $\rho$.

The first condition is that $\rho$ must "conserve" passengers by maintaining flow of passengers into and out of each state in the state-space. This ensures that passengers are never "lost" in the system:

$$\left(\lambda + \sum_{j=0}^{n-1} \mu_j\right) \rho(k_0, k_1, \ldots, k_{n-1}) = \lambda \rho(k_0 - 1, k_1, \ldots, k_{n-1}) + \mu_{n-1} \rho(k_0, \ldots, k_{n-2}, k_{n-1} + 1) + \sum_{j=0}^{n-2} \mu_j \rho(\ldots, k_j + 1, k_{j+1} - 1, \ldots).$$

We also need to define the boundary states of the state space, which must ensure that no state can have a negative number of passengers at any processor:

$$\begin{aligned}
(\mu_0 + \lambda)\rho(k_0, 0, 0, \ldots, 0) &= \mu_1 \rho(k_0, 1, 0, \ldots, 0) + \\
&\quad \lambda \rho(k_0 - 1, 0, 0, \ldots, 0), \qquad k_0 > 0; \\
(\mu_{n-1} + \lambda)\rho(0, 0, \ldots, k_{n-1}) &= \mu_{n-2} \rho(0, 0, \ldots, 1, k_{n-1} - 1) + \\
&\quad \mu_{n-1} \rho(0, 0, \ldots, k_{n-1} + 1), \qquad k_{n-1} > 0; \\
\lambda \rho(0, 0, \ldots, 0) &= \mu_0 \rho(1, 0, \ldots, 0).
\end{aligned}$$

Lastly, all probabilities must sum to 1:

$$\sum_{k_{n-1} \geq 0} \sum_{k_{n-2} \geq 0} \cdots \sum_{k_0 \geq 0} \rho(k_0, k_1, \ldots, k_{n-1}) = 1.$$

We can then extend the solution presented in Trivedi [2002] from a two-processor chain and see that that $\rho$ has the form

$$\rho(k_0, k_1, \ldots, k_{n-1}) = \prod_{j=0}^{n-1} (1 - \rho_j) \rho_j^{k_j},$$

where each $\rho_j$ takes the form

$$\rho_j = \frac{\lambda}{\mu_j}$$

and $\mu_j$ is the rate of processing of the $j$th processor. From Trivedi, we know that the bottleneck of the system occurs at the processor with the largest $\rho_j$.

We now consider a random ordering of people entering the plane, in which a passenger turns into a given row with probability $1/n$ or continues walking with probability $(n-1)/n$. We assume that in the original system $u_0 = u_1 = \ldots = u_{n-1}$ and therefore all nodes in the closed system must have a rate of $\rho_j = (n-j)\lambda/u_j$ for all $j$. This implies that $\rho_0$ is the largest in the system and is therefore the bottleneck. If we recursively apply this for an airplane with $(n-1)$ rows, we see that the bottleneck will always be the first processor. We can then recognize three important properties of airline boarding:

- The critical bottleneck for boarding is always the first row in the plane.
- The criticality of the main bottleneck is linearly proportional to the number of rows in the plane.
- The farther back a row, the less it contributes to bottlenecking.

## Effects of Row and Column Interferences

Boarding gets more complicated when people board out of order, which leads to row interference and column interferences that hold up traffic. Here we use probabilistic estimation to assess zone configurations that are affected least by shuffling passengers in a given zone. For the sake of simplicity, we analyze a plane with 6 seats per row, but the analysis generalizes. First, we develop some lemmas, based on assuming that passengers board in random order.

*Row interferences* occur when a passenger sitting in an aisle or middle seat has to get up to let in the person who has the window seat or the middle seat. We calculate the expected number of times that a passenger has to get up if the passengers sitting in a row of $k$ seats board in random order.

**Lemma 1** *The expected number of interferences in a row of $k$ people is $k(k-1)/4$.*

In particular, the expected number of interferences for 3 seats is $3/2$.

When a passenger stands in the aisle to stow luggage, the passengers behind must wait. We assume that a passenger can proceed to the right row and stow luggage as long as the passenger is not blocked by another passenger stowing luggage. The lemma below finds the longest sequence of passengers who can be stowing their luggage at once. If the rows are numbered in increasing order from the back of the plane to the front, the problem can be reduced to finding a largest increasing subsequence of row assignments among the passengers, since these passengers then can proceed to their seats and stow their bags.

**Lemma 2 (Kiwi 2006)** *The expected length of longest increasing subsequence in a permutation of* $\{1, 2, \ldots, k\}$ *is (asymptotically) of size* $2\sqrt{k}$.

The proof is quite involved and we do not discuss it. The lemma tells us that if $k$ passengers sitting in different rows board the plane at once, then $2\sqrt{k}$ of them can proceed to their seats and stow luggage without encountering an interference. If we have $m$ people spread over $k$ rows, then it will take them $\lfloor m/(2\sqrt{k})\rfloor B$ time to stow their luggage.

We use these lemmas to estimate the boarding time for a group of passengers to be seated in different configurations.

## Configuration 1: Dense Distribution over Rows

The zone is composed of $m$ passengers spread densely over $k$ rows. For 6 passengers in a row, dense means that all 6 are in the same zone. The expected number of row interferences for this configuration is $\frac{3}{2} \cdot 2k$. The boarding time for this zone is approximately

$$T = \left\lfloor \frac{m}{2\sqrt{k}} \right\rfloor B + 3ks,$$

where $B$ is bag stowage time and $s$ is the time for a passenger to get out of their seat to allow a fellow passenger to pass and then sit down again. The time for people to walk down the aisle can be ignored, since in this case it is overshadowed by bag stowage and reseating.

## Configuration 2: Sparse Distribution over Rows

The zone is composed of $m$ passengers sparsely distributed over $k$ rows, meaning at most two passengers in a row, mostly on different sides of the aisle. Having a sparse distribution totally eliminates the effect of reseating time but results in walking time becoming the critical factor. The walking time for this configuration is roughly $kv$, where $v$ is the time to walk from one row to next. Thus, total time for boarding this group is

$$T = \left\lfloor \frac{m}{2\sqrt{k}} \right\rfloor B + kv.$$

# Boarding Schemes

Currently-used boarding systems include:

**Back-to-front:** (Air Canada, Alaska, American, British Airways, Continental, Frontier, Midwest, Spirit, Virgin Atlantic [Finney 2006]) The most widely-used boarding scheme. Passengers are divided into zones and board at the front door in a back-to-front order.

**Outside-in:** (Delta and United Airlines) Passengers are boarded windows first, followed by middle seats, with aisle seats boarding last.

**Reverse-pyramid:** (US Airways on some routes) This scheme boards people in a V-like manner, with rear middle and windows boarding first, followed by rear aisles and front aisle.

**No assigned seats:** (Southwest Airlines) Ostensibly the fastest boarding scheme. Passengers are not assigned seats and may sit anywhere. This scheme has not been widely copied, since it does not lead to high customer satisfaction and is often likened to a cattle car.

**Figure 3** offers a visual comparison. We color seats according to the order in which they fill, with red (dark) earlier and green (light) later. The entry door is at the top and the bottom is the back row. We include an ordering named "Parabolas" that we introduce later.

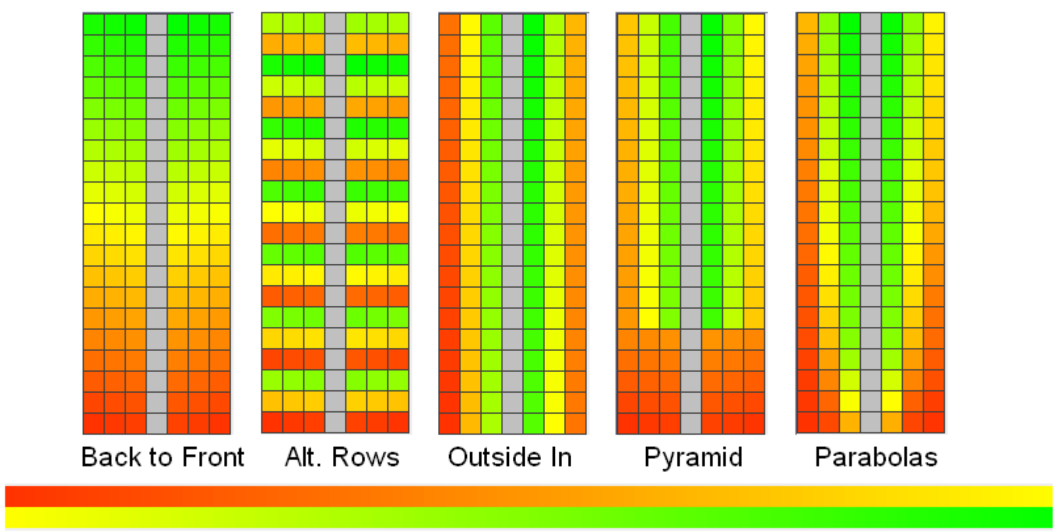



**Figure 3.** Seat ordering schemes.

# Simulation Design and Details

We produced a comprehensive flexible boarding simulator that we use to compare boarding algorithms and the effects of various situations. Our simulation techniques were inspired by stochastic Petri nets, finite time-step simulations, and cellular automata [Marelli et al. 1998].

## Process

Our simulation model runs through time in small intervals. At each interval, it moves each participant in the simulation according to rules defined by the input parameters. Certain events take extra time and create blocks for other participants in the model. For example, a passenger stowing luggage blocks the aisle for a certain amount of time.

## Plane

The plane is a variably-sized rectangular grid of seats with a single aisle for passenger movement in the center of the columns and a single door for entry at the beginning of the aisle. The space between rows (*pitch* in industry terms) and between columns is adjustable.

## Behavior Modeling

Passengers can board with either assigned or unassigned seats.

- **Assigned seats:** Passengers move to their seats as fast as walking speed allows, waiting as necessary for obstacles to clear. They make no mistakes in moving to their assigned seats.
- **Unassigned seats:** Passengers walk as far back as possible before sitting. If the aisle is blocked, they sit in the current row to avoid waiting standing up. When they sit, they are generous and move all the way toward the window to save future passengers' time.

A passenger has an associated seating delay time for moving into their row, which corresponds to the time to stow luggage, wait for already-seated passengers to move out of the way, move in, and get settled. The seating delay rises as more people sit in the row, reflecting decreasing space in the overhead compartment and accompanying longer time to find space for a bag.

When a person stowing luggage blocks the aisle and someone else comes up behind them, there is a certain *pass percentage* representing the chance that the blocked person can pass by and proceed to a seat farther along.

## Parameters

The simulation is run with a *passenger input rate*, affected by the gate check-in speed, that is, how fast passengers are processed in the terminal. Passengers have a constant *walking speed* when not blocked.

With assigned seats, passengers are typically called in groups, with each group some approximately contiguous segment of seats. The *group size* is variable, and passengers within each group are randomly ordered. Groups themselves satisfy certain *seating assignment schemes*, for example, ordering groups back to front.

## Parameter Estimation

For our simulation trials, we use the following default values and distributions. Estimated values are based on critical thinking; parameters dependent on the plane size will be specified later. Times are in seconds.

- **Walking Speed** = 140 cm / s. This varies based (at least) on the age and gender distribution of the passengers. We used the FAA evacuation simulation requirements that call for a simulated plane's population to be at least 40% female, at least 35% over age 50, and at least 15% both. Our average distribution is balanced male and female with 40% over age 50. The average comfortable walking speeds based on age and gender are from Bohannon [1997].
  *Affected by:* Passenger demographics, aisle width, ceiling height, number and size of bags per person.

- **Seating Delay** = $U[10, 20] + P_c + P_r$. The seating delay is uniformly distributed and includes the compartment-filling penalty $P_c$ and the row-out-of-order penalty $P_r$.
  *Affected by:* Other penalties, plus row spacing, luggage size and number, compartment size and layout, and passenger demographics.

- **Compartment Penalty** = $3p$, where $p$ is the number of people already seated in your row.
  *Affected by:* Size and layout of overhead compartment, luggage size and number.

- **Pass Rate** = 0.05 (an estimate).
  *Affected by:* Aisle width, passenger demographics, luggage size and number.

- **Row Out-of-Order Penalty** = $15p$, where $p$ is how many people have to move to let you into your seat.
  *Affected by:* Row spacing, passenger demographics, aisle width.

- **Entry Delay** = 5.0 (estimate). The time between successive passengers entering the plane.
  *Affected by:* Check-in procedure, flight attendant behavior, luggage size and number, out-of-plane characteristics.

## Summary

The simulation model is configurable. We use it to test different strategies and measure the effects of certain changes on the process. We can model

- different types and sizes of planes with varying interior configurations (aisle width, seat spacing, overhead compartment size)

- passengers with and without assigned seats in many arrangements and zone groupings;

- the effects of luggage count, luggage size, compartment size, and stowing speed; and
- the effects of the gate check-in process speed.

# Deriving a New Scheme

Random boarding with unassigned seats tends to be fastest [Finney 2006]. Despite this fact, many airlines do not adopt it because it often leads to low customer satisfaction. We derive a new seating method inspired by the seating patterns of passengers in a random assignment-less environment.

From our earlier analysis, the best strategy would move passengers as far back as possible and also ensure that passengers boarding within a block are spread out over several rows. We use this intuition to develop heuristics.

Seats that are always the first ones to be filled are assigned to the first zone. The next group of seats to get filled are assigned to the second zone, and so on (**Figure 4**). This zone assignment gives a boarding scheme for passengers with assigned seats. Since in the simulation these passengers had minimal interference with one another, we hope that similar results would occur even with shuffling within zones.

We observe that the zones returned by our learning algorithm resemble parabolas, hence we define the zones as seats highlighted by different parabolas centered near the far end of plane and the center of the rows, superimposed on the seating chart. The parabolas get steeper for higher-numbered zones, since we are boarding aisles at that time.

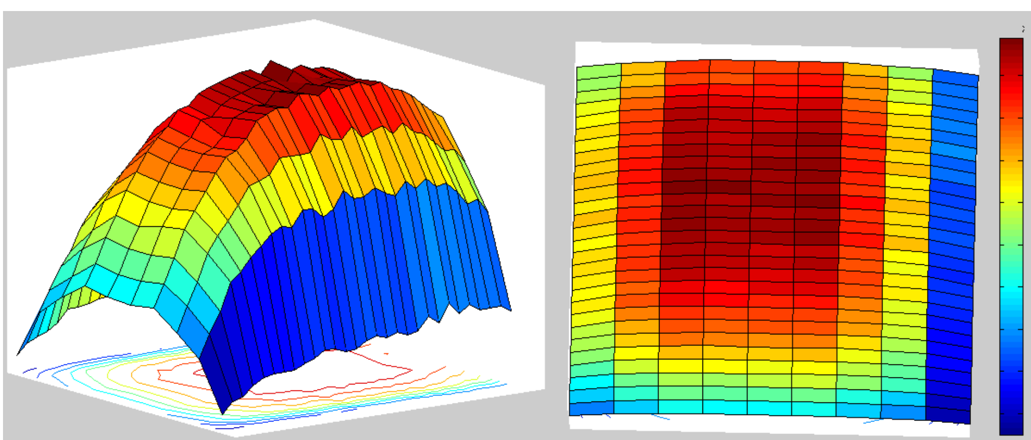


**Figure 4.** Results of no-assignment seating simulation. Blue-green (light) are passengers seated first and orange-red (dark) are passengers seated last. Traces of equal height take the shape of parabolas.

We wrote a computer program to compute these parabolas for planes of arbitrary size. We refer to this method of assigned seat grouping as the *Parabola boarding method*.

# Relative Effect of Parameters

We vary input parameter values to determine their impact. We perform these simulations using the default parameters from above and the plane layout of a Boeing 757-200 (39 rows, 6 columns).

## Walking Speed

We analyze the effect of passenger walking speed in **Figure 5**. We vary it from the approximate comfortable walking speed of a 70-year-old female to the approximate maximum walking speed of a 70-year-old male [Bohannon 1997]. Boarding time is not always lowered by increasing walking speed (except in the back-to-front scheme). This fact reflects our key insight from queuing theory analysis that the entry rate is a more critical bottleneck. *Ensuring high walking speed is not critical*.

## Luggage Stowage Time

We analyze the effect of changing the luggage stowage time in **Figure 6**. Specifically, we change the average value of the uniform distribution from which we select stowage time. This value has a large effect on the boarding time, following our insight that keeping the aisle full or "pipelined" is important: If we slow the process at this pipeline, performance suffers. *Ensuring quick luggage stowage is critical*.

## Plane Entrance Rate

We analyze the effect of changing the plane entrance rate in **Figure 7**. Increasing the delay between successive entries (that is, lowering the rate of incoming passengers) increases the time to board. At a certain large value, all seat assignment methods become equivalent, presumably because no bottlenecks form—since passengers enter so slowly (effectively, each passenger enters independently, one at a time, without conflicts), queueing and overflow effects do not emerge. *Ensuring adequate plane entrance speed is critical*.

## Intra-Row Movement Time

In **Figure 8**, we look at the effects of changing the time to shuffle in and out of a row to let in a fellow passenger. Increasing the row movement time raises the boarding time marginally for back-to-front and alternating rows but not for the other algorithms. However, this is to be expected: The other methods are designed to avoid row conflicts, with passengers almost always arriving in outside-in order. *So decreasing row movement time is not critical*, particularly because we can avoid its effects with certain algorithms.

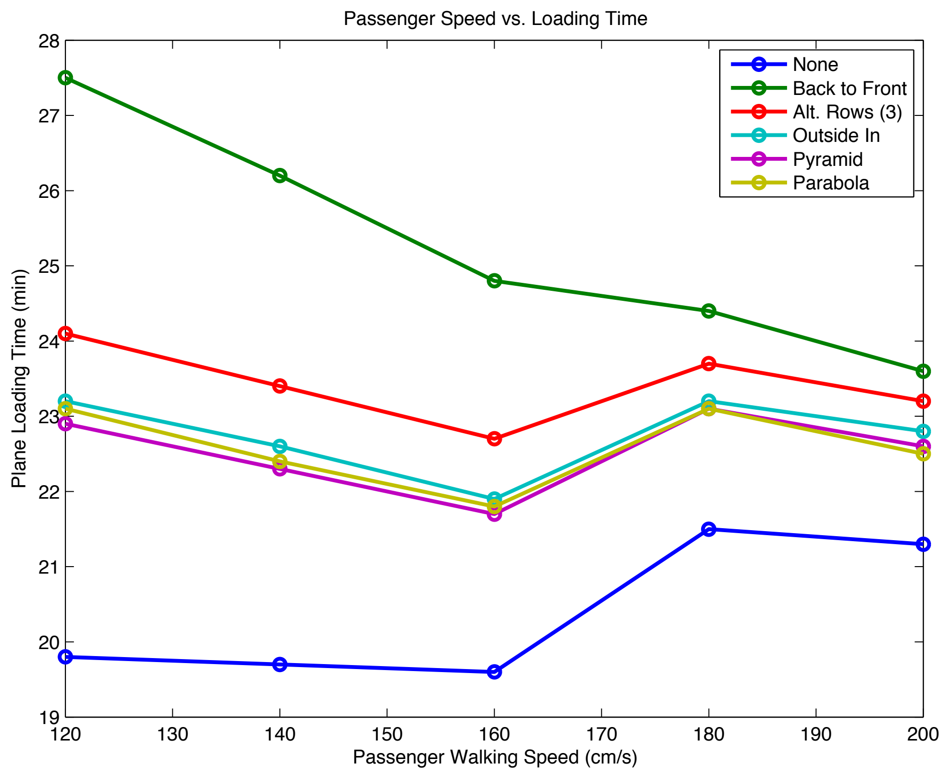



**Figure 5.** Boarding time as a function of walking speed $v$.

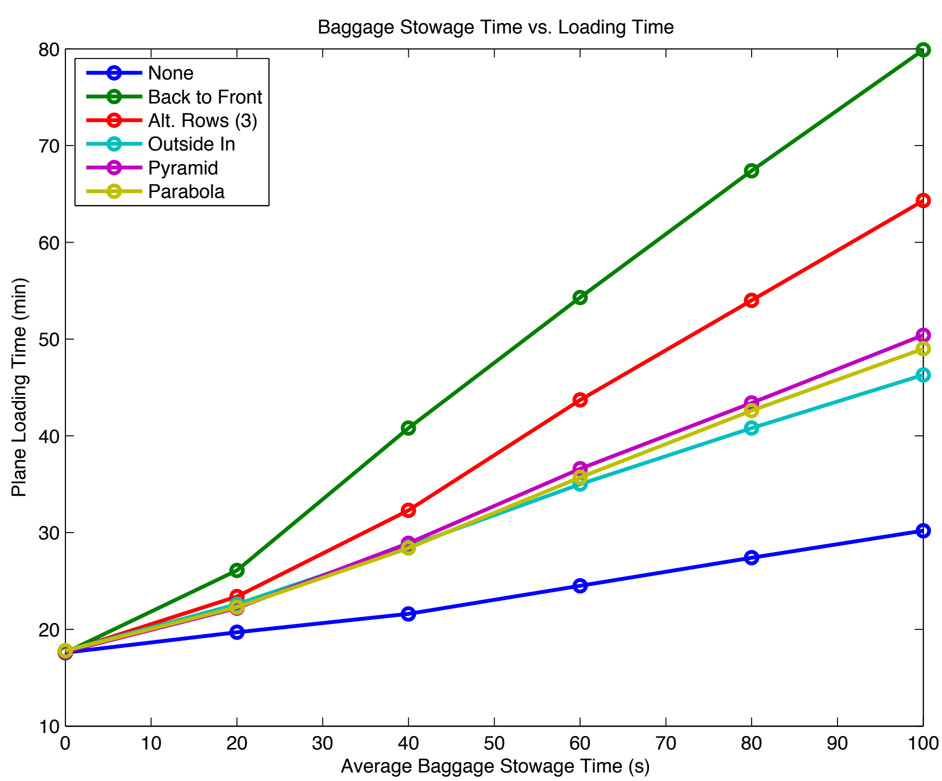



**Figure 6.** Boarding time as a function of luggage stowage time $B$.

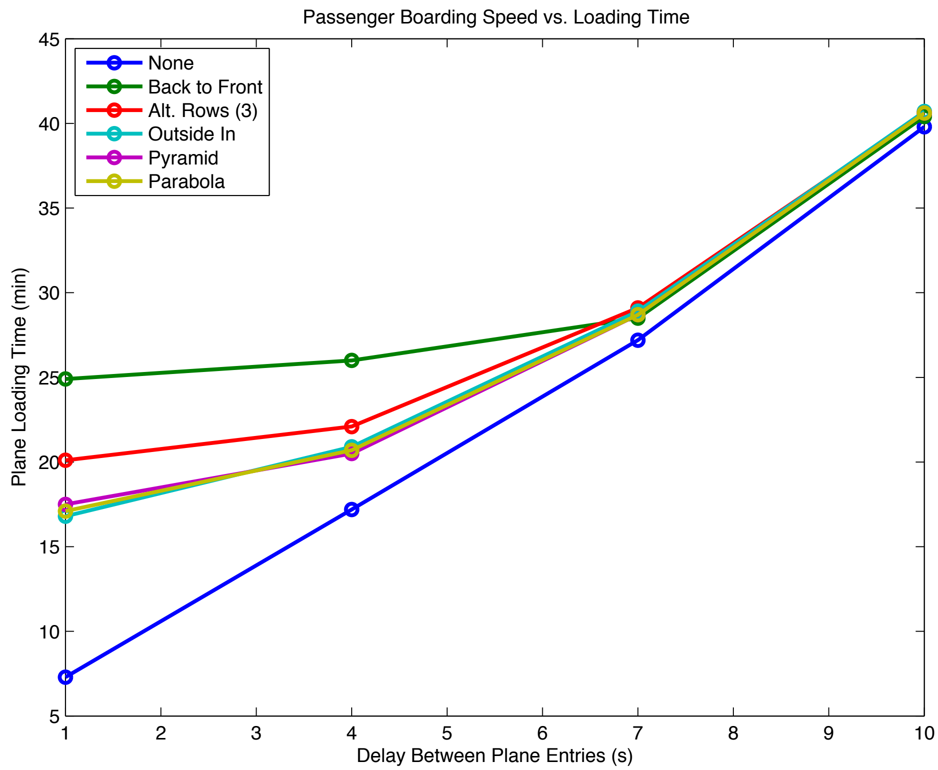



**Figure 7.** Boarding time as a function of plane entrance rate $\lambda$.

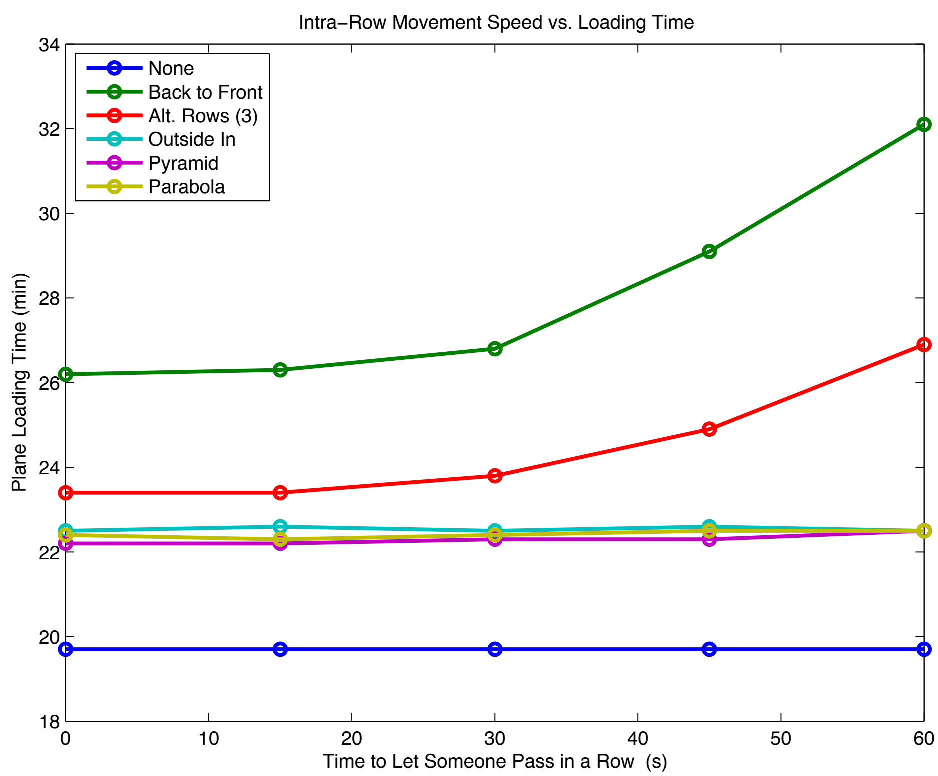



**Figure 8.** Boarding time as a function of intra-row conflicts.

## Summary

We have analyzed the relative impact of the parameters of our model for a representative airplane. Two factors are of key importance: average luggage stowage time and plane entrance rate. Ranking the various strategies in the light of their performance on these factors produces the ordering:

- Outstanding: No assigned seats
- Meritorious: Outside-in, reverse-pyramid, parabola
- Honorable Mention: Alternating rows
- Limited Success: Back-to-front

Some insight into this ordering comes from comparing average seating order after mixing within groups vs. without seat assignments (**Figure 9**). Pyramid and parabola most closely approximate the order achieved by the fast no-assignments method. We conclude that outside-in captures most of the key benefits, since in general it is as fast as the other two while being a less-close approximation of the random greedy model.

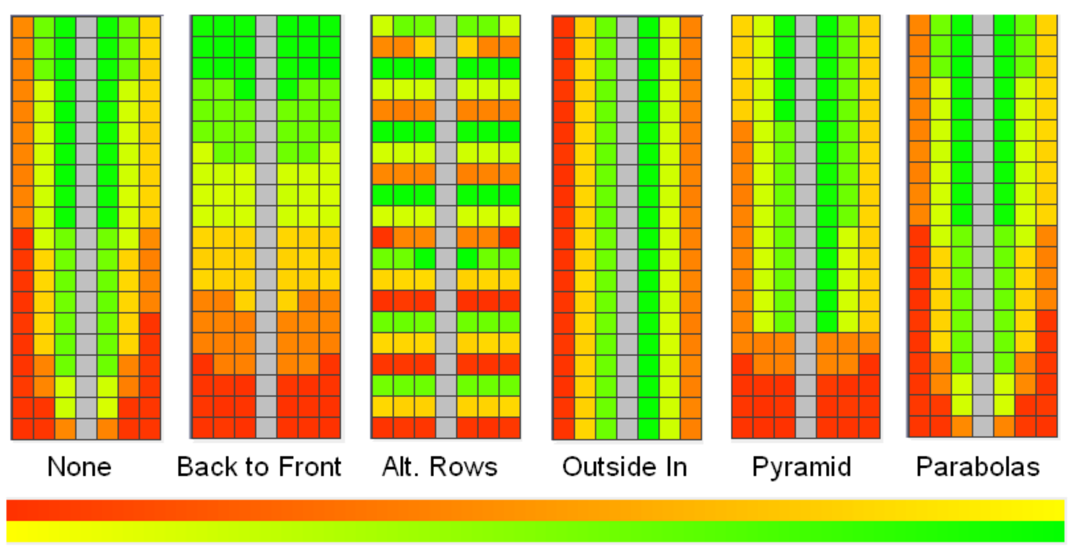



**Figure 9.** Average seating order with group mixing.

# Strategy Robustness and Dependability

Average boarding speed is not the only measure of success. A fast boarding method does no good if once a week it takes twice as long as its average; airlines need to depend on a consistent time to produce achievable and reliable schedules. Therefore, we prefer boarding methods that vary little between

**Table 1.**
Time range for each boarding method.

| Algorithm | Time Range (min) |
|---|---|
| No assigned seats | 0.7 |
| Outside-in | 2.6 |
| Parabola | 2.8 |
| Reverse-pyramid | 3.1 |
| Alternating rows | 4.6 |
| Back-to-front | 6.2 |

worst and best cases. We simulated plot boarding times for our various schemes over 500 trials, with results in **Table 1**.

The smallest spread between longest and shortest load times is for no assigned seats. Interestingly, there is a direct correlation between time to board and variability in boarding time. The outside-in, reverse-pyramid, and parabola methods have similar boarding times and distributions. Similarly, Back-to-Front and Alternating Rows take the longest and have the largest spread. To some extent, these results suggest that a faster boarding algorithm is also more dependable; however, this may not be true for all cases.

# Model Generalization

Our simulation assumes that the plane is boarded from one end of the seating area with passengers walking down aisles at the center of the rows. But some planes have several aisles or passengers boarding on different levels. Our model can be easily generalized to accommodate for different plane designs and layouts. We divide the problem into sections; each subsection is modeled as its own plane with entry rates changed appropriately. Depending on whether boarding of subsections occurs in serially or in parallel, the times are added or compared (and the maximum taken).

# Specific Results

We apply our model to various real-world planes of different sizes to compare the speed of the boarding processes. We let Outside-In serve represent outside-in, reverse-pyramid, and parabola, which are similar in timing. We apply our generalization techniques to model multi-aisle, -class, and -level planes with given configurations [Airbus . . . 2007, Boeing . . . 2007] (**Table 2**).

The results support our previous conclusions. In several cases, back-to-front is quite close to outside-in, perhaps because the plane entry rate was not high enough.

**Table 2.**
Simulation boarding times for multi-aisle, multi-class, and multi-level planes.

| Plane | Passengers | Unassigned | Back-to-Front | Outside-In |
|---|---|---|---|---|
| DC 9-40 | 125 | 12 | 17 | 14 |
| Airbus A320 | 164 | 14 | 21 | 17 |
| Boeing 757-200 | 234 | 20 | 26 | 23 |
| Boeing 747-400 | 313 | 31 | 31 | 32 |
| Airbus A380 | 555 | 35 | 35 | 36 |

# Conclusion

While our approaches and models are effective and produce results, there remain several model weaknesses:

- We assume independent, perfect-knowledge, infallible passengers who always put their luggage directly above themselves, as well as other perfect scenarios (planes of equally-sized rows, jet-bridges of constant flow instead of stairs or buses that bring passengers to the plane).
- There are several areas of the problem that we left untested because they seemed to be of secondary importance, such as varying the number of zones.
- Our comparison of boarding algorithms is simulation-based and therefore by nature not exhaustive. There may be better algorithms that we did not test, such as single-zone random boarding or rotating row-group zones.
- We stay within the current boarding paradigm so as not to produce too much uncomfortable change for passengers. However, greater improvements might obtain if a wider range of choices were available; simple examples might be assigning passengers only to a row and letting them choose a seat there, or hiding money under one seat to encourage speedy boarding.

Overall, we believe the strengths inherent in our approach overcome many of the weaknesses and allow us to make useful recommendations:

- Our multilayered approach produced key insights.
- Our simulation model can be extended to test new algorithms and situations with minimal changes.
- We provide a relative ranking of factors affecting boarding speed, not just a ranked list of algorithms. An airline can still make improvements if they don't want to switch their process, or if they already have a fast process.

## Summary

Our key observations are:

- The aisle is the main bottleneck, especially near the entrance, and it is necessary to "pipeline" passengers to maintain a high throughput.
- The rate of passengers entering is also critical, since it determines the maximum rate at which passengers can proceed down the aisle and be seated.
- Sending in passengers with closely-situated seating assignments in short time intervals results in numerous interferences and increases boarding time. Instead, passengers should be enter by zones that distribute seats over several rows.

Our simulations confirm these insights and show that boarding schedules that follow these rules perform better in terms of both speed and reliability.

We offer the following recommendations to airlines to improve their boarding time, turnaround time, and ultimately their bottom line:

- **Passenger entry speed:** The faster passengers enter the plane, the faster it boards. This means that ticket-checking should be as quick as possible, hence with an optimal number of gate agents. Flight attendants should be stationed at critical junctions (such as entrances to aisles in a multi-aisle plane) to direct passengers to the correct rows.
- **Luggage stowage time:** The faster passengers stow their bags and sit down, the faster the plane boards. Stowage time can be reduced by changing or enforcing carry-on luggage limits and by having flight attendants assist passengers with large or heavy bags.
- **Switch from back-to-front to another boarding method.** Outside-in boarding provides a 10%–30% advantage over back-to-front. Foregoing assigned seats results in a further 10%–30% advantage over outside-in. Faster methods are also considerably more reliable: Outside-in has a time range 50% smaller than back-to-front.

# References

Airbus aircraft families. 2007. `http://www.airbus.com/en/aircraftfamilies.` Accessed 11 February 2007.

Boeing: Commercial airplanes, products. 2007. `http://www.boeing.com/commercial/products.html` . Accessed 11 February 2007.

Bohannon, R.W. 1997. Comfortable and maximum walking speed of adults aged 20–79 years; reference values and determinants. *Age and Aging* 26: 15–19.

Trivedi, K.S. 2002. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. 2nd ed. New York: Wiley-Interscience.

Finney, Paul Burnham. 2006. Loading an airliner is rocket science. *New York Times* (14 November 2006). `http://travel2.nytimes.com/2006/11/14/business/14boarding.html`. Accessed 10 February 2007.

Jackson, J.R. 1957. Networks of waiting lines. *Operations Research* 5: 518–527.

Kiwi, M. 2006. A concentration bound for the longest increasing subsequence of a randomly chosen involution. *Discrete Applied Mathematics* 154 (13): 1816–1823.

Marelli, Scott, Gregory Mattocks, and Remick Merry. 1998. The role of computer simulation in reducing airplane turn time. *Aero Magazine* (4th Quarter 1998). `http://www.boeing.com/commercial/aeromagazine/aero_01/textonly/t01txt.html`.

*Snakes on a Plane*. 2007. `http://www.imdb.com/title/tt0417148/`. Accessed 11 February 2007.

van den Briel, Menkes H.L., J. René Villalobos, Gary L. Hogg, Tim Lindemann, and Anthony V. Mulé. 2005. America West Airlines develops efficient boarding strategies. *Interfaces* 35 (3) (May-June 2005): 191–200.

Kshipra Bhawalkar, Matthew Edwards, Michael Bauer, and advisor Anne Catlla.

# Novel Approaches to Airplane Boarding

Qianwei Li
Arnav Mehta
Aaron Wise
Duke University
Durham, NC

Advisor: Owen Astrachan

## Summary

Prolonged boarding not only degrades customers' perceptions of quality but also affects total airplane turnaround time and therefore airline efficiency [Van Landeghem 2002].

The typical airline uses a zone system, where passengers board the plane from back to front in several groups. The efficiency of the zone system has come into question with the introduction and success of the open-seating policy of Southwest Airlines.

We use a stochastic agent-based simulation of boarding to explore novel boarding techniques. Our model organizes the aircraft into discrete units called "processors." Each processor corresponds to a physical row of the aircraft. Passengers enter the plane and are moved through the aircraft based on the functionality of these processors. During each cycle of the simulation, each row (processor) can execute a single operation. Operations accomplish functions such as moving passengers to the next row, stowing luggage, or seating passengers. The processor model tells us, from an initial ordering of passengers in a queue, how long the plane will take to board, and produces a grid detailing the chronology of passenger seating.

We extend our processor model with a genetic algorithm to search the space of passenger configurations for innovative and effective patterns. This algorithm employs the biological techniques of mutation and crossover to seek locally optimal solutions.

We also integrate a Markov-chain model of passenger preference into our processor model, to produce a simulation of Southwest-style boarding, where

*The UMAP Journal* 28 (3) (2007) 353–370. 

seats are not assigned but are chosen by individuals based on environmental constraints (such as seat availability).

We validate our model using tests for rigor in both robustness and sensitivity. Our model makes predictions that correlate well with empirical evidence.

We simulate many different a priori configurations, such as back-to-front, window-to-aisle, and alternate half-rows. When normalized to a random boarding sequence, window-to-aisle—the best-performing pattern—improves efficiency by 36% on average. Even more surprising, the most common technique, zone boarding, performs even worse than random.

We recommend a hybrid boarding process: a combination of window-to-aisle and alternate half-rows. This technique is a three-zone process, like window-to-aisle, but it allows family units to board first, simultaneously with window-seat passengers.

# Survey of Previous Research

## Discrete Random Process

Bachmat et al. [2006] propose a discrete boarding process in which passengers are assigned seats before boarding. The inputs to the process are an index for the position of each passenger in the queue and a seat assignment for each passenger. Additionally, the researchers define the aisle space that each passenger occupies, the time it takes to clear the aisle once the designated row is reached, and the distance between consecutive rows. The first two parameters are sampled from distributions defined by the researchers.

The model considers the travel path of each passenger. The passenger moves down the aisle until reaching an obstacle, which is either the back of a queue or a person who is preparing to sit in their row. Passengers who arrive at their row clear the aisle after a delay time; after that, the passengers behind continue down the aisle.

The researchers define an ordering relation between passengers. Each passenger is assigned a pointer to the last passenger who blocked their path. By following the trail of passengers, the longest chain in the ordering ending at any particular passenger can be identified. This chain specifies the number of rounds needed for the simulation.

## Other Simulation Studies

Van Landeghem [2002] simulates different patterns of boarding sequences, based on a plane with 132 seats divided into 23 rows, with Row 1 and 23 having 3 seats and the others having 6. The first objective is to reduce total boarding time; the second objective is to augment the quality perception of the passengers by evaluating the average and maximum individual boarding times as seen by the passengers. Van Landeghem simulates calling passengers to board at random or by block (contiguous full rows), half-block (contiguous rows, port or starboard halves only), row, half-row, or individual seat. The shortest boarding time is by seat (in a particular order).

# Model Overview

We present a simulation model that can be considered a stochastic agent-based approach.

We treat the plane as a line, with destinations (seats) at regular distances along the line. Each passenger, modeled as an agent, moves along the line until reaching the assigned seat. Each agent has a speed constrained by the slowest person in front.

This model takes into account the topology of the airplane. Each row is a discrete unit. We call these units *processors*, since they determine the rate that an individual moves through the system. Each processor has a queue, a list of people waiting to be processed by it (and hence moved to the next node of the system). Each agent has a destination processor, the row of the assigned seat.

We consider two major collision parameters. A scenario where a passenger is waiting for another passenger to stow baggage is a *baggage collision*. We also model *seat collisions*: when a passenger is sitting between another passenger and that passenger's seat (for example, the passenger with an assigned window seat must move around a passenger in an aisle seat).

We attempt to optimize boarding time based on the order in which passengers enter the plane, via a genetic algorithm over the search space of all possible orderings. Crossovers and mutations are defined so that no seats are "lost."

Our final model includes a Markov chain to model passenger preferences in an open seating environment. This model simulates a boarding process like the one used by Southwest Airlines.

# Details of the Model

## Basic Model

We use a compartmental model, calling the compartments "processors," each physically analogous to the space of one row. Differing layouts of processors can model varied plane topologies.

Each passenger is randomly assigned a seat. These seats are not necessarily unique; they are uniformly drawn from all seats on the plane. A seat is represented as a coordinate pair $(c, r)$, where $r$ is a row and $c$ is a seat number.

Passengers move based on the function of the processors. The processors are in series, with each processor having the next as an output (**Figure 1**).

Since movement is performed by processors pushing passengers from one row to the next, each passenger stores only that passenger's destination. A passenger who reaches a processor waits in a first-in/first-out queue to be

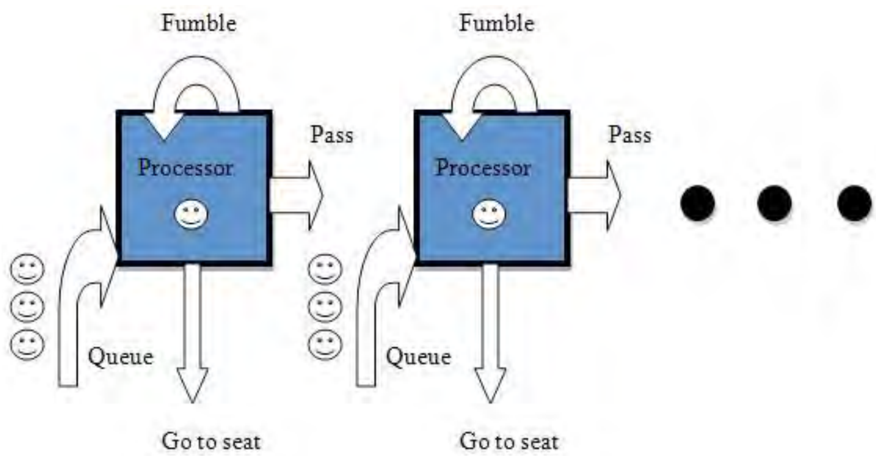



**Figure 1.** Processor-based model.

processed (people cannot move around one another while in the aisle). The initial state of the plane is that all passengers are queued at the first processor.

In each iteration, each processor looks at the destination of the passenger and performs one of the functions:

- **Pass.** A passenger who passes moves from the current processor to the end of the queue of the next processor.
- **Fumble.** With a certain small probability, the processor will do nothing this cycle (a bag gets caught in the aisle, a passenger trips, or some other time-wasting random event occurs). A fumble is not equivalent to time spent stowing baggage or rearranging passengers; our basic model accounts only for random time-wasting events.
- **Sit Down.** If this row contains the assigned seat for the passenger currently in the processor, the passenger leaves the aisle and is seated.
- **Idle.** If there is no passenger in the processor (and the queue is empty), the processor does nothing.

The processors run sequentially from back to front until every passenger is seated.

## Assumptions

- The initial configuration is that all passengers are queued at the first row. In actuality, all passengers are initially queued at the ticket counter, where their boarding passes are scanned and they walk a short distance to the plane. Hence, a more realistic alternative would be a Poisson arrival process from the ticket counter to the queue for the first row. However, this additional process is unnecessary because of the high speed at which boarding passes are scanned, which approximates the speed of normal walking. Hence, passengers reach the queue at a much higher rate than they are moved forward through the plane; the queue at the first processor forms instantly when the first passengers walk into the plane.

- There is no idle time between the first passenger entering the first queue and the last passenger doing so. The airline could wait until there is no queue left before calling additional passengers to board. However, doing this is never to the airline's advantage.
- Special-needs and business-class passengers have already boarded; airlines have an obligation to these customers for early boarding. We start our simulation clock after these special classes of passenger have already boarded and deal only with the bulk passenger class.
- Every passenger functions individually. We expect improved efficiency when passengers travel in groups, since groups are self-organizing (the individuals in a group do not collide with one another).

# Extended Model

## Seat assignment

The initial model assigns seats randomly and without uniqueness. We remedy this to a one-to-one correspondence between passengers and seats.

## Assumptions

- The plane is fully booked and every seat is occupied. This assumption allows us to optimize over the worst-case scenario.

## Seat collisions

A common occurrence is a passenger needing to cross over a seated passenger. To account for such a *seat collision*, we implement a new processor function:

- **Rearrange.** This cycle is spent waiting for the aisle to clear after the seat collision. This operation reduces the seat collision counter by 1.

A seat collision has an associated time penalty that depends on the type of collision. When there is a seat collision, the processor for that row spends a number of cycles equal to the time-cost sorting out the collision. During that time, no other passengers can enter the processor (though they can enter the processor queue).

We determined values for the seat collision costs by physical experimentation involving multiple trials over a simulated plane row. All seat collisions have the same time cost, except that the penalty for crossing over two passengers is about 50% more than for crossing over a single passenger. We expect the variation among passengers to be small.

## Luggage

A major factor in boarding times is stowing hand luggage. As the overhead bins fill, it takes longer to stow a bag. Hence, we developed a statistical model of luggage stowing. Luggage stowing is performed by the processor at a given row using the command:

- **Stow.** This cycle is spent by a passenger stowing a bag in the overhead bin. The stowing counter is decreased by 1.

For luggage stowage times, we use a Weibull distribution because of its flexibility in shape and scale. The probability density function is

$$f(x, \kappa, \lambda) = \frac{\kappa}{\lambda}\left(\frac{x}{\lambda}\right)^{\kappa-1} e^{-\left(\frac{x}{\lambda}\right)^{\kappa}},$$

where $\lambda$ is a scaling parameter, $\kappa$ is a shape parameter, and $x$ is the number of people who have entered the plane. Its cumulative distribution function

$$F\left(x, \kappa, \lambda\right) = 1 - e^{-\left(\frac{x}{\lambda}\right)^{\kappa}}.$$

is a measure of the additional time to stow hand luggage as the plane fills up.

The waiting time of passenger $x$ is

$$\lceil cF\left(x, \kappa, \lambda\right) + N\rceil,$$

where $c$ is a measure of the additional time to store baggage when the plane is full and $N$ is a Gaussian noise parameter that accounts for the nonuniformity of the boarding process.

## Queue Size

The initial model assumes that each processor has an unlimited queue. This makes sense for the initial processor, whose queue consists of all passengers lined up along the loading ramp. However, for a processor inside the aircraft, the queue actually takes up physical space. We cap all processor queues but the first at a length of 2, which corresponds well with the ratio of aisle length to passenger size.

# Multiple Aisles

We model multi-aisle planes as processor sets with multiple pipelines. Using this technique, planes of arbitrary sizes, topologies, and entrance points can be modeled. We describe here the technique for the modeling of a double-aisle plane, such as the Boeing 777.

As in the single-aisle model, all passengers are initially queued at a single processor. For a double-aisle plane, this processor represents the junction point at the entry of the plane. No passengers are assigned seats in this row. From the first processor, a passed passenger may move to either of two different processors. Each of these two processors begins a serial chain of processors akin to a single-aisle plane. Each passenger chooses an aisle based on seat assignment. As in real aircraft, certain rows of the plane are widened so that a passenger can move from one aisle to the other.

Some passengers (for example, those in the middle of a row) have seats equidistant to two aisles; they take the first available aisle and can switch aisles at junction points.

This procedure can be generalized to four-aisle aircraft as well, such as the forthcoming Airbus A380. In that aircraft, not all aisles connect: A passenger cannot move across from an upstairs aisle to a downstairs one.

We can also simulate a plane with the gate in the middle, or with two gates or more, by changing the configuration of processors. Thus, our procedure can be used to simulate any plane.

### Assumptions

- All passengers choose the correct aisle, as usually happens, since a steward is positioned at the junction point (i.e., the first processor) to direct traffic. To make this choice easier, the airline could have color-coded boarding passes.
- Only passengers with middle-seat assignments switch aisles.

## Deplaning

Our processor-based model can handle deplaning. The processors are reversed: They push passengers from the back of the plane towards the front. Time spent retrieving baggage follows an opposite distribution from the base model: The first passengers must spend more time retrieving luggage than later passengers. Furthermore, there are no seat collisions; everybody clears out of the plane in order. The destination of all passengers during deplaning is the front of the plane.

### Assumptions

- Deplaning is uncoordinated. Though some variant of aisle-to-window deplaning is likely the fastest, we believe that any coordinated deplaning method would greatly decrease customer satisfaction. For example, an aisle-to-window deplaning process would cause window seat passengers near the front of the plane to have to wait for virtually the entire plane to disembark. In any case, it is impossible to control the movement of the passengers.

# Genetic Algorithm

We used our model above to find the average times taken by various boarding techniques, including back-to-front and window-to-aisle. However, such known orderings may not be optimal.

Since the set of all possible orderings is vast, we need a heuristic to explore parts of the space that interest us the most. This heuristic, if it converges, gives an optimum that—while unlikely to be a global optimum—will be a strong local optimum.

To perform this search, we implement a genetic algorithm, a type of search algorithm that derives the principles of its functioning from evolutionary biology. A genetic algorithm models a solution as a set of "organisms." In our case, an organism is one possible arrangement of passengers in line waiting to board the plane. The algorithm begins with a set of organisms called the *population*. Each organism in the population is run through our processor model, and, based on the time that it takes for all passengers to be seated, given a "fitness" score.

Based on the scores, some organisms are selected to survive, while others die. Organisms with the highest score have the highest survival probability. Organisms that survive are kept in the population, and the others are deleted. The population is replenished by the addition of new organisms. New organisms are either offspring of two surviving organisms from the previous round or randomly generated. The algorithm runs for a set number of generations, at which point it returns the best organism remaining in the pool.

The core of a genetic algorithm is the evolution of the population over time. Over a significant number of generations (for our model, around 60), the algorithm converges. The convergence is a local maximum; the point of convergence is dependent on the initial random population of individuals. The point of convergence is reached using the properties of mutation and crossover.

## Mutation and Crossover

Mutation is the process by which an organism changes from one generation to the next. A crossover is the genetic offspring of two individual organisms. We account for both types of evolution in our model.

We first must consider what the genome or "DNA" of our organisms looks like. An organism is a listing of passengers and seats in order (see **Table 1**).

Mutations are relatively simple. During a mutation, a random, sequential section of the DNA is chosen and moved to a different location. A mutation of the initial DNA could look like the bottom row of **Table 1**, which permutes the seat assignments among the passengers.

Crossovers are more complicated. A special property of our solution space is the one-to-one correspondence between passengers and seats. This means that the order of seat numbers in the DNA can be switched, but the seat numbers must stay the same. In normal DNA, a sequential piece of one organism's DNA is exchanged with the corresponding sequence of the other organism. Due to

**Table 1.**
Example of mutation.

Original organism

| | | | | | |
|---|---|---|---|---|---|
| Passenger | 1 | 2 | 3 | 4 | 5 |
| Seat number | 22A | 23C | 7A | 30F | 2B |

Mutated organism

| | | | | | |
|---|---|---|---|---|---|
| Passenger | 1 | 2 | 3 | 4 | 5 |
| Seat number | 7A | 30F | 22A | 23C | 2B |

the one-to-one correspondence property of our data, we cannot use this type of crossover: If the two sequences chosen did not have the same set of seats, our offspring would not have a valid genetic code.

Hence, we formulate a new form of crossover that preserves the elements of a DNA code but changes its order (**Figure 2**).

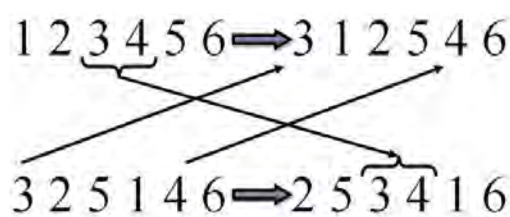



**Figure 2.** Processor-based model.

The crossover algorithm first chooses a sequence of seats from the genome of the first organism. It then identifies the indices of these seats in the second organism. The genomes of the two organisms are rearranged such that the ordering of the selected seats is switched between the two organisms, while all other seat assignments remain the same. In the example in the figure, the seat sequence (3. . . 4) is selected as the crossover. The indices of (3. . . 4) are 3 and 4 in the first organism and 1 and 5 in the second. After the crossover, the indices of (3. . . 4) are 1 and 5 in the first organism and 3 and 4 in the second. The order of all the other seats remains the same, but their indices are shifted due to the change in location of 3 and 4.

## Population Seeding

We ran the genetic algorithm in two configurations, in the first determining the initial population randomly and in the second "seeding" it (adding nonrandom organisms). For seeding, we added two examples of each of the tested types of boarding configuration (e.g., window-to-aisle and back-to-front). Seeding helps the algorithm approach the global maximum, since the beginning population then contains individuals that have high fitness.

# The Southwest Model

## Model Overview

In the Southwest system, passengers board in order of arrival with no assigned seats.

In our model, seat preferences are encapsulated in a matrix

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & \dots & b_6 \end{bmatrix},$$

which represents the spatial arrangement of seats in each row: Elements $b_1$ and $b_6$ represent the relative preferences for window seats while elements $b_2$ and $b_5$ represent those for middle seats.

The passenger's desire to sit at a given row, to move forward, or to move backward is encapsulated in a transition matrix,

$$\mathbf{P} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,6} \\ a_{2,1} & a_{2,2} & \dots & a_{2,6} \\ \vdots & \vdots & \ddots & \vdots \\ a_{30,1} & a_{30,2} & \dots & a_{30,6} \end{bmatrix},$$

where $\mathbf{P}$ satisfies

$$0 \leq \mathbf{P}_{i,j} \leq 1, \qquad 1 \leq i, j \leq N;$$

$$\sum_{j=1}^{N} \mathbf{P}_{i,j} = 1, \qquad 1 \leq i \leq N.$$

Element $a_{i,i}$ represents the passenger's desire to sit in row $i$. Element $a_{i,i+1}$ represents the passenger's desire to move forward to row $(i+1)$. Element $a_{i,i-1}$ represents the passenger's desire to move back to row $(i-1)$. Although a passenger may prefer to move back a row, that is not possible in our model.

Since $\mathbf{P}$ is an irreducible aperiodic transition matrix, Markov-chain theory tells us that there is a stationary distribution $\bar{\pi}$, which gives the probability that a passenger will end up at a particular row.

The model incorporates each passenger's decision-making, the passenger's location within the plane, and environmental constraints. In deciding whether to move forward or to sit at the current row, a passenger first considers the current location. If at the end of the plane, there is no option but to sit in the last row. If the number of available seats in front of the passenger's current position exceeds the number of people ahead of the passenger, then the passenger can move forward to the next row; if not, the passenger has to sit in the current row. A passenger cannot move backwards.

As the passenger progresses, preferences need to be adjusted:

- As the passenger moves forward, there are fewer rows to consider.

- As the plane fills, certain rows no longer have available seats to consider.

In both cases, preferences are redistributed so that the relative preferences between all remaining available rows remain the same. Similarly, when seats in a particular row are occupied, the passenger's preference for a particular seat in that row is readjusted so that the relative preferences for available seats remain the same and the sum of seat preferences across the row is 1. Therefore, the preferences for each standing passenger are recomputed each time a passenger finds a seat.

When a passenger gets to a row, the decision of *whether* to sit is governed by a random process that favors the row according to the relative preference that passenger has for that particular row.

After a passenger decides to sit at a given row, if the row contains more than one available seat, his choice of *where* to sit is governed by a random process that favors each seat according to the passenger's relative preference.

From a macro perspective, each passenger makes the decision of where to sit autonomously. This decision is driven, however, by certain preferences and their corresponding probabilities that lend order to the seating sequence in the plane. In each cycle, the model recomputes the preferences of each passenger for each particular row and each seat.

## Assumptions

- The movement of passengers along the aisle of the plane is unidirectional. Additionally, passengers are aware of the number of people and available seats in front of them. They will not move forward unless the number of available seats exceeds the number of people in front of them.

- All passengers share a common propensity to sit at any given row or to move forward along the aisle. Because passengers prefer seats closer to the front of the aircraft, the desire to sit at any given row is greater than the desire to move forward.

- All passengers share a common preference for seats, favoring window over aisle and aisle over middle. Having a wall or empty space on one side does not seem terribly unappealing; the window is most preferable because it offers a view and the benefit of resting your head.

- The decision to sit in a particular row is independent of the decision to sit at a particular seat in that row. In most cases, passengers first decide on row preference and then decide which seat they prefer.

- When a row of seats is filled, the probability that a passenger sits in that particular row becomes zero. The probability previously attributed to that row is then redistributed proportionally among the unfilled rows according to the preference probability already attributed to them. This process ensures that the relative preferences of the unfilled rows remain the same.

# Boarding Patterns

Although our algorithm may be used to model planes of any size, we focus on a standard 180-person plane with 30 rows and 6 seats in every row.

## Random Boarding

This boarding process is used as a baseline for comparison to other models. The process involves the random assignment of seats to passengers in the boarding queue followed by the boarding simulation.

## Window-to-Aisle Boarding

Window-to-aisle boarding involves filling up all the window seats, followed by the middle seats, and then the aisle seats. In **Figure 3**, black tiles represent the earliest passengers to enter the plane and white tiles represent later passengers. The darkness of each tile decreases with increasing passenger numbers in the boarding queue.

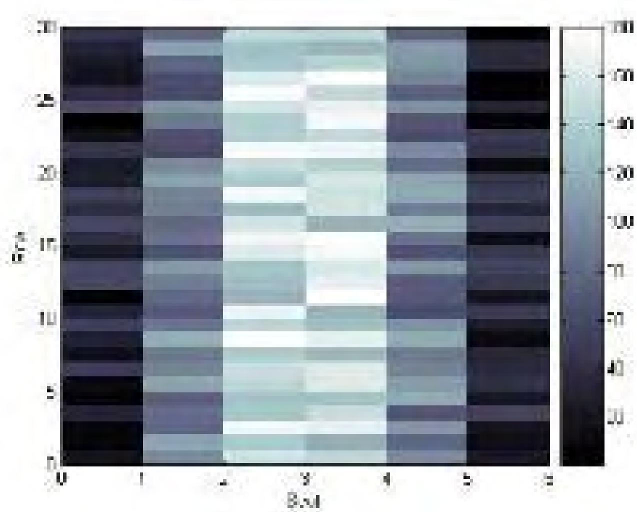


**Figure 3.** Window-to-aisle boarding.

This "outside-in" method eliminates all seat collisions. The sequence of window seat passengers is random; likewise, the orders of passengers with middle and aisle seats are each independently random. Thus, this boarding pattern still demonstrates significant baggage collisions from passengers interfering with one another's passage to their seat row.

## Alternating Half-Rows Boarding

The plane is split into two halves along the aisle and one half is filled before the other half starts boarding. Each half is filled by loading every third row starting from the back. Once we reach the front, the process is repeated from the second-to-last row followed by the third-to-last row (**Figure 4**). The rows are filled in a random order, so there may be seat collisions. Each row must be filled before the next row can start loading. Once passengers in one half of the plane have all boarded, the second half begins boarding with the same process.

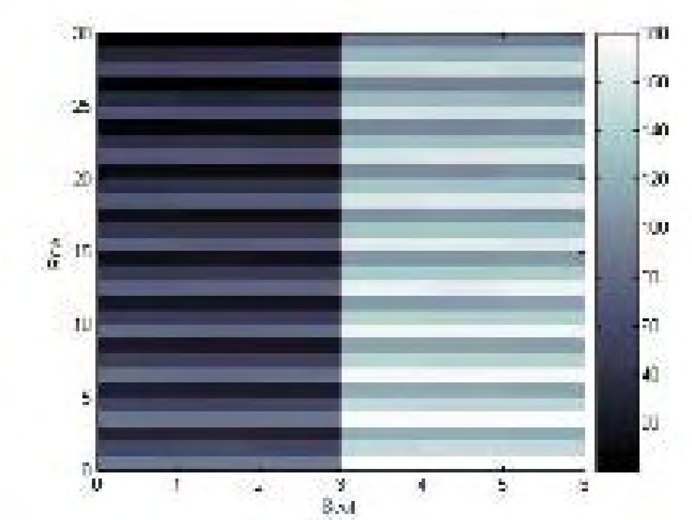


**Figure 4.** Alternating half-rows boarding.

## Zone Boarding

In this boarding pattern, the plane is split into contiguous and evenly divided zones based on row number. The passengers in each zone are then randomly assigned to a seat in each zone. The zone farthest back in the plane boards first, followed by the next furthest, and so on till we reach the front of the plane (**Figure 5**). Passengers in a particular zone must board the plane before passengers in the next zone can begin boarding.

## Rotating-Zone Boarding

Rows are filled from back to front in an alternating fashion. Thus, the seats in the back row are filled first, followed by the seats in then front row, the seats in the second to last row, and so on till we reach the middle rows of the plane (**Figure 6**). The seats in each row are assigned randomly and the passengers of a row must board before the passengers for the next row board.

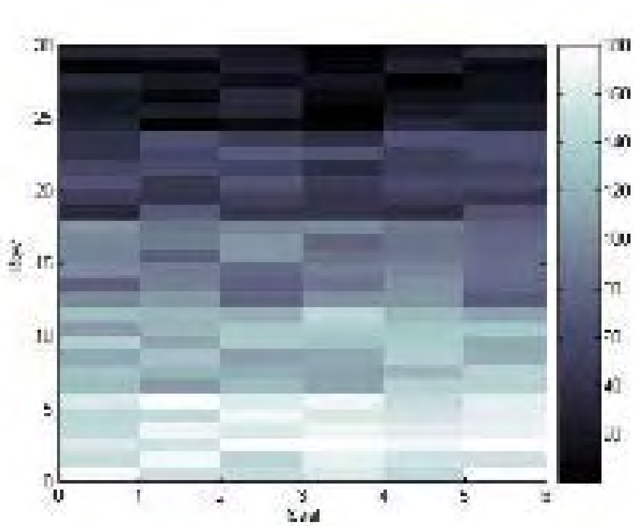


**Figure 5.** Zone boarding.

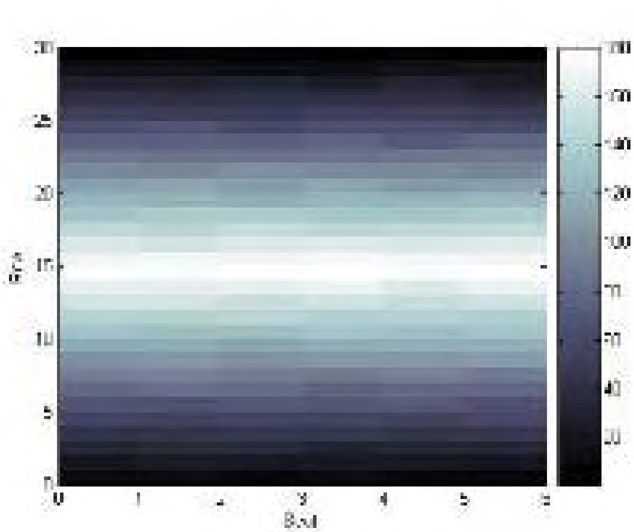


**Figure 6.** Rotating-zone boarding.

# Results

We evaluated the efficiency of each seating pattern by averaging 30 runs of each simulation with 35 trials per simulation on a 180-person plane (30 by 6). Each run used a randomly-generated seating arrangement within the constraints of the pattern. The waiting times were normalized to the average waiting time of a randomly-generated seating arrangement. The normalization value was derived from analysis of 50 different random patterns.

Results are shown in **Table 2**.

**Window-to-aisle** is the most efficient; it eliminates seat collisions but its randomized column arrangement allows baggage collisions.

**Table 2.**
Average normalized times for boarding patterns.

| Boarding pattern | Time |
|---|---|
| Deplaning | 0.48 |
| Window-to-aisle | 0.64 |
| Seeded genetic algorithm | 0.67 |
| Alternate half-rows | 0.73 |
| Genetic algorithm | 0.81 |
| Random | 1 |
| Southwest model | 1.09 |
| Back-to-front | 1.10 |
| Rotating-zone | 1.71 |

**Alternate half-rows** minimizes spatial overlap between alternating groups of 3 passengers; any seat collision is not large enough (in a spatial sense) to extend to the half-row following. However, this localized congestion also explains why alternate half-rows is slower than window-to-aisle. It is possible that the time for this scenario is overstated, since three passengers walking to a half-row may self organize.

**Back-to-front,** the most common boarding technique, performs surprisingly poorly—worse than random, due to local congestion propagating to waiting passengers.

**Rotating-zone** presents collisions of the same sort as alternate half-rows. However, while in half-row boarding there are potentially 6 collisions among 3 passengers, rotating-zone boarding allows for 15 collisions among 6 people.

**Southwest boarding** suffers because passengers share a preference for seats closest to the exit, which can increase queuing early in the plane, and for aisle seats over middle seats, which also increases seat collisions.

**The genetic algorithm applied to a random seating arrangement** reached a steady-state solution that is most likely a local minimum for that problem instance. We ran the simulation multiple times, and the results displayed similar properties.

**The seeded genetic algorithm** resulted in a hybrid between window-to-aisle and alternating half-row boarding. Window seats fill first, followed by the middle and then aisle seats. However, on one side of **Figure 7** there are distinct alternating bands every third row. The algorithm shows a distinct window-to-aisle and alternating half-row hybrid boarding process (**Figure 7**), demonstrating that this hybrid forms a strong local optimum. We observe that the minimum obtained by this hybrid is quasistable. We do not notice any influence from the rotating-zone or back-to-front boarding patterns, indicating that these populations were not as "fit" as the former two and were eliminated from the gene pool. This boarding pattern allows for small families to board together.

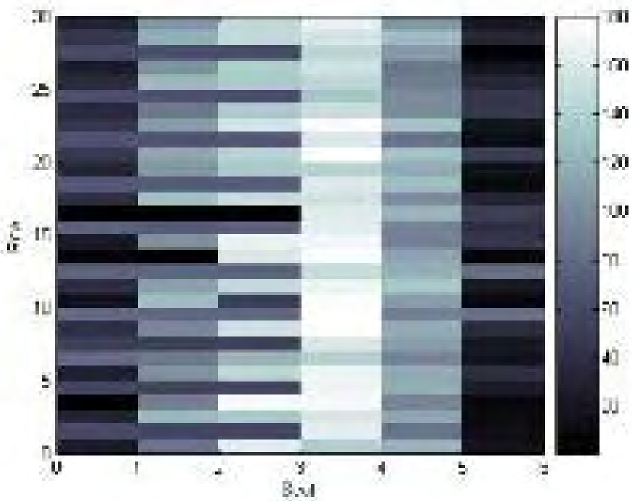


**Figure 7.** Seeded genetic algorithm.

**Deplaning** is 25% faster than window-to-aisle boarding and is consequently less useful to optimize.

# Sensitivity and Robustness Testing

The robustness of our model is a measure of how it performs in extreme cases; a robust model is one that does not break down in such cases. The sensitivity of our model is a measure of the effect of small parameter changes; a good model should show small changes in response to small parameter changes.

Our model is well behaved; that is, it does not exhibit chaotic behavior. Small changes in parameters demonstrate small changes in results, demonstrating good sensitivity.

## Baggage

We eliminated the delay from stowing luggage, a key factor responsible for aisle collisions. We expected that the window-to-aisle boarding would benefit more than alternate half-rows boarding. We observe a 26% decrease in time for window-to-aisle and a 16% decrease for alternating half-rows, consistent with our prediction.

## Seat collisions

We eliminated time delays due to seat collisions. We expected that random boarding would perform as well as window-to-aisle, since the primary contribution to delay time will be aisle collisions. Our simulation performs as expected, with only a 2.3% difference in times.

### Queuing

We allowed an unlimited queue for each processor. We expected elimination of local congestion and increases in efficiency for zone boarding. Our simulation confirms that zone boarding is 25% more efficient than random boarding.

## Discussion and Conclusions

### Results

To identify better boarding techniques, we employed a simulation model based on a stochastic agent-based approach. We simulated boarding sequences with embedded stochastic variability, including aisle and row congestion.

We find through simulation that window-to-aisle boarding is the most efficient. However, aisle congestion remains significant due to the random sequencing of passengers within the same boarding group. This in turn contributes to substantial delays due to the stowing of luggage.

Alternate half-row boarding is the next most efficient. Its speed derives from minimization of aisle congestion, despite seat collisions in each half-row.

We could both eliminate seat collisions and minimize aisle congestion by specifying the sequence of each passenger in the boarding queue; but such a method would not be practical, since it would require all passengers to arrive at the gate punctually and gate agents to spend time organizing passengers.

In general, seat collisions have relatively less impact near the end of boarding, because the time to stow luggage increases.

### Optimal Recommendation

We found a hybrid between alternate half-rows and window-to-aisle to be a local optimal solution. We recommend hybrid boarding because it offers the versatility of both group and individual boarding. In this solution, the first boarding call is for families and window passengers. Since families self-organize, minimizing collisions, we expect hybrid boarding to be more efficient in practice than predicted by simulation.

## Strengths and Weaknesses

#### Strengths

- Processor-based model has few input parameters, leading to good robustness and sensitivity.
- Genetic algorithm explores and optimizes known configurations.
- Variety of boarding patterns explored, including planned layouts, genetic optimization, and passenger preference

- Accounts for all major factors involved in plane boarding.
- Simulates both boarding and deplaning processes.
- Uses a variety of modeling techniques in an integrated holistic model.

**Weaknesses**

- Parameters have to be derived from physical occurrences.
- Genetic algorithm has high computational requirements and cannot identify global optimum.
- Does not account for non-uniform preferences among passengers.

## Future Work

- Identify at which rows bottlenecks occur for any given time point.
- Investigate efficient deplaning algorithms.
- Better quantify passenger seating preferences

# References

Bachmat, Eitan, Daniel Berend, Luba Sapir, Steven Skiena, and Natan Stolyarov. 2005. Analysis of airplane boarding times. `http://www.cs.bgu.ac.il/~ebachmat/ORrevisionfinal.pdf` .

________. 2006. Analysis of airplane boarding via space-time geometry and random matrix theory. *Journal of Physics A: Math and General* 39 (29): L453–L459. `http://www.cs.bgu.ac.il/~ebachmat/jpafinal-new.pdf` .

Disney, R., and P. Kiessler, P. 1987. *Traffic Processes in Queueing Networks: A Markov Renewal Approach*. Baltimore, MD: John Hopkins University Press.

Finney, Paul Burnham. 2006. Loading an airliner is rocket science. *New York Times* (14 November 2006). `http://travel2.nytimes.com/2006/11/14/business/14boarding.html` .

Lawler, Gregory F. 2006. *Introduction to Stochastic Processes*. 2nd ed. Boca Raton, FL: Chapman and Hall / CRC.

van dan Briel, Menkes H.L., J. Rene Villalobos, and Gary L. Hogg. 2003. The aircraft boarding problem. *Proceedings of the 12th Industrial Engineering Research Conference (IERC)*, article number 2153. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf` .

Van Landeghem, H. 2002. A simulation study of passenger boarding times in airplanes. `http://citeseer.ist.psu.edu/535105.html` .

# STAR: (Saving Time, Adding Revenues) Boarding/Deboarding Strategy

Bo Yuan
Jianfei Yin
Mafa Wang
National University of Defense Technology
Changsha, China

Advisor: Yi Wu

## Summary

Our goal is a strategy to minimize boarding / deboarding time.

- We develop a theoretical model to give a rough estimate of airplane boarding time considering the main factors that may cause boarding delay.
- We formulate a simulation model based on cellular automata and apply it to different sizes of aircraft. We conclude that outside-in is optimal among current boarding strategies in both minimizing boarding time (23–27 min) and simplicity to operate. Our simulation results agree well with theoretical estimates.
- We design a luggage distribution control strategy that assigns row numbers to passengers according to the amount of luggage that they carry onto the plane. Our simulation results show that the strategy can save about 3 min.
- We build a flexible deboarding simulation model and fashion a new inside-out deboarding strategy.
- A 95% confidence interval for boarding time under our strategy has a half-width of 1 min.

We also do sensitivity analyses of the occupancy of the plane and of passengers taking the wrong seats, which show that our model is robust.

*The UMAP Journal* 28 (3) (2007) 371–384. 

# Introduction

Airline boarding and deboarding has been studied extensively in operations research literature. U.S. domestic carriers lose $220 million per year in revenue for take-off delays [Funk 2003].

We examine strategies for boarding and deboarding planes with varying numbers of passengers, trying to minimize the boarding and deboarding time.

# Literature Review

Marelli et al. [1998] designed a computer program called PEDS (Passenger Enplaning/Deplaning Simulation) that used a probabilistic discrete-event simulation to simulate boarding methods. PEDS predicted that it would take 22 min to board a Boeing 747-200. However, the paper did not lay out the boarding procedure.

Van Landeghem [2000] stated that the fastest boarding strategy is individually boarding by seat and row number, and the second fastest is a back-to-front "alternate half-row" boarding system, which was cited to take 15.8 min. He also proposed strategies with small numbers of boarding groups that are both faster and more robust against disturbances. A problem with the data is that only five replications were done for each boarding procedure tested [Pan 2006].

Later, van den Briel et al. [2003] showed that a reverse-pyramid boarding strategy could reduce airplane's turn time by 3-5 min compared to a traditional back-to-front boarding approach. The boarding time depends on events called "interferences."

Unfortunately, all of these researchers used simulation based on small or mid-size airplanes that do not extend to the much larger aircraft under development today. Our approach and results can be applied in all sizes of airplanes.

# Basic Assumptions

- **First-class passengers board first.** Hence, our simulation considers only economy-class passengers.
- **Passengers do not try to pass other passengers in the aisle.** The aisles are narrow, so passengers have to wait to move until there are no "obstacles" in front of them.
- **A "call-off" system is used.** Passengers board in ordered groups; gate agents announce which group is to board.
- **A passenger does not take the wrong seat and does not walk past the row of the right seat.** Such mistakes inevitably delay boarding.

# Reasons for Boarding Delay

## Normal Delay

"Interference" is the main reason for boarding delay. Van den Briel et al. [2003; 2005] divide boarding interferences into two types:

- **Aisle interference:** Since the aisle is narrow enough to allow only one passenger to proceed forward, aisle interference occurs when a passenger stows luggage. To do this, the passenger must stand in the aisle for a moment, thereby acting as an "obstacle" for passengers behind.

- **Seat interference:** This kind of interference occur when a passenger is stalled by another one or two passengers sitting in the same half-row. Because of the limited space between contiguous rows, this passenger must ask these passenger already sitting in their seats to stand up and move into the aisle.

## Abnormal Delay

Passengers take the wrong seats, or are late. These behaviors can hardly be avoided. Because of their complexity and variety, we don't take them into consideration. Our main objective is to reduce seat and aisle interference.

# Theoretical Estimate Model

We consider boarding time as made up of two parts:

- Free boarding time $t_{\text{free}}$, the total time if all passengers board without any interference or delay.

- Interference time $t_{\text{inter}}$, the total interference time including aisle interference and seat interference.

So the total boarding time is

$$T_{\text{total}} = t_{\text{free}} + t_{\text{inter}}, \tag{1}$$

## Free Boarding Time

We consider the passengers as a steady flow that pours into the plane at a rate of $v_{\text{flow}}$ passengers per minute. So the free boarding time is

$$t_{\text{free}} = \frac{n}{v_{\text{flow}}}, \tag{2}$$

where $n$ is the number of passengers.

# Interference Time

## Seat Interference

We assume that the times to get from the seat to the aisle and get back are the same, both denoted by $t_S$. Suppose that three passengers on the same side of a row are assigned to the same boarding group, passengers sitting in positions A (window), B (middle), and C (aisle). There are six equally likely kinds of seat interferences, corresponding to the boarding orders ABC, ACB, BAC, BCA, CAB, CBA. We calculate the interference time for each case. Take ACB as an example: The window-seat passenger boards first, followed by the aisle seat passenger; then the middle-seat passenger needs the aisle-seat passenger to get up and move to the aisle, the middle-seat passenger moves from the aisle to the seat, and the aisle-seat passenger and sits back down again. So the interference time is $3t_S$. The results are shown in **Table 1**.

**Table 1.**
Seat interference time by boarding order.

| Boarding order | ABC | ACB | BAC | BCA | CAB | CBA |
|---|---|---|---|---|---|---|
| Seat interference time | 0 | $3t_S$ | $3t_S$ | $5t_S$ | $6t_S$ | $8t_S$ |

The average seat interference time for 3 passengers in the same half-row is

$$\bar{t}_S = \frac{25}{6}\,t_S.$$

With $n$ passengers boarding, the total seat interference time is

$$t_{S:\text{inter}} = \bar{t}_S \cdot \frac{n}{3} = \frac{25}{6}\,t_S \frac{n}{3}. \qquad \textbf{(3)}$$

## Aisle Interference

Let $P_1, \ldots, P_n$ be the passengers in order in the queue, with corresponding row numbers $r_1, \ldots, r_n$. We say $P_i$ *blocks* $P_j$ if $r_i < r_j$. We use the number of blocking times as the number of aisle interference times, that is, when we calculate total interference times, we don't consider the situation that two or more blockings happen together. For example, for passengers $P_1, \ldots, P_5$ in rows 1, 4, 5, 2 and 3, $P_1$ blocks $P_2$, $P_2$ blocks $P_3$, and $P_4$ blocks $P_5$. But actually, after $P_1$ is seated, $P_2$ and $P_4$ can stow luggage simultaneously, and only $P_3$ and $P_5$ need to wait (two intervals of interference) to stow luggage. To simplify the calculations, we think of this as a total of three intervals of interference.

As a result, to calculate the aisle interference times, we need calculate only the number of instances of $r_i < r_{i+1}$. Since the order of passengers is random, the number $i$ of aisle interference times is a random variable. We assume that every permutation is equally likely, so the average aisle interference time is

$$I = \frac{1}{n!} \sum i(r_1, \ldots, r_n),$$

where we sum over all permutations. The permutations can be divided into $n!/2$ pairs, each of which is the reverse of the other; together, each pair will have $(n-1)$ instances of $r_i < r_{i+1}$. Hence

$$I = \frac{n-1}{2}.$$

With $t_L$ for the average time to stow luggage, the total aisle interference time is

$$t_{A\text{:inter}} = \frac{n-1}{2} \cdot t_L. \tag{4}$$

From **(1)–(4)**, we get the total boarding time as

$$T = \frac{n}{v_{\text{flow}}} + \frac{25}{6}\, t_s \, \frac{n}{3} + \frac{n-1}{2}\, t_L.$$

# Data Collection

## Aircraft of Different Sizes

We base our computer simulations on three types of airplane of different sizes: Airbus A320 (small—124 seats), Airbus A300 (midsize—266 seats), Airbus A380 (large—555 seats).

## Experimental Data

We could not collect the needed by experimenting or by interviewing airline executives. Fortunately, this work has already been done by van den Briel et al. [2003] as cited by Pan [2006]. They found the following average times:

- Get-on time (time between gate agent and gate—assuming one gate agent): 9.0 s.
- To advance one row: 0.95 s.
- Stowage: 7.1 s.
- Seat interference time: 9.7 s.

# Cellular Automata Simulation Algorithm

In the cellular automata model of boarding analysis, each cell is designated as a passenger, a barrier, a road or a seat. The model restricts individual movements on the plane and computes total boarding time. Time, position, and passenger behavior are each discrete quantities. The passenger compartment is specified as a grid of rectangular cells, while time is incremented using a convenient time step. During one simulation time step (STS), a passenger can move only one cell / row, and all cells representing passengers are processed once and in random order. The simulate iterates time steps and update passengers' state and position until all passengers sit down.

## Call-off Function

Before passengers board the plane, they are usually divided by a gate agent into groups, often by consecutive rows, for boarding efficiency. We develop our call-off function with three steps:

1. Divide different seats into groups according to a specific strategy. For example, in implementing outside-in, we put seats in one column into a group.

2. Generate a random order number in each group.

3. Queue the groups consecutively.

## Enplane Simulation Function

### Simulation of the Next Passenger Boarding

The get-on time has an exponential distribution with mean that we estimate to be 10 STS.

### Individual Behavior Judgments

What do passengers do in each time step?

- Stand still when there is an obstacle.

- Move forward by one cell toward the seat when there is free space in front.

- Stow luggage. This behavior needs a counter to record its STS because it requires more than one step.

- Seat interference when the passenger already seated must stand up and let other passengers move in. It also needs a counter.

## Simulation Results and Analysis

We simulate common boarding strategies, including random, back-to-front, rotating-zone, outside-in, and reverse-pyramid [Finney 2006]. Back-to-front and rotating-zone allow us to choose the number of rows per group; we try 4, 6, and 8 to see how variation affects the strategies. Similarly, reverse-pyramid can also vary in layers, and we choose 2, 3, and 4 layers to analyze.

# Simulation Results

We simulate each boarding strategy 100 times; the results are in **Table 3**.

**Table 3.**
Simulation results for strategies.

| Strategy | Rows (or layers) | Average interference | Seat interference | Aisle |
|---|---|---|---|---|
| Random | | 24 | 72 | 52 |
| Rows | | 32 | 72 | 55 |
| Back-to-front #1 | 4 | 25 | 72 | 51 |
| Back-to-front #2 | 6 | 25 | 72 | 52 |
| Back-to-Front #3 | 8 | 25 | 73 | 53 |
| Rotating #1 | 4 | 25 | 72 | 53 |
| Rotating #2 | 6 | 25 | 73 | 54 |
| Rotating #3 | 8 | 25 | 72 | 54 |
| Outside-in | | 23 | 0 | 42 |
| Reverse-pyramid #1 | 2 | 23 | 0 | 43 |
| Reverse-pyramid #2 | 3 | 23 | 0 | 42 |
| Reverse-pyramid #3 | 4 | 23 | 0 | 42 |

## Analysis of the Simulation Results

- **The more rows in a group, the shorter the boarding time.** This is really unexpected! Usually, we think that if we divide the passengers into more groups before boarding in accordance with a boarding strategy, the passengers will be better organized and board the plane more efficiently. But to our surprise, our simulations run in the opposite direction. Take back-to-front as an example. When a group contains 8 rows, the boarding time is 24.6 min; but when there are 4 rows per group, the boarding time increases to 25.0 min. With the two extremes (i.e., one row per group vs. all the passenger as a group), the contrast is even more obvious: 32 min vs. 24 min.

  How could this happen? Through analysis of the simulation processes, we find that two or more interferences can happen at the same moment (**Figure 1**) without influencing the boarding process adversely. With more rows in a group, multi-interferences increase but boarding time decreases.

- **Dividing passenger groups according to their columns such as outside-in way and reverse-pyramid way avoids seat interference and reduces aisle interference.** This is easy to understand. If we divide the groups by rows, passengers in the same row get on the plane together, and try to stow their luggage at the same time. However, dividing the group by columns staggers the time when passengers stow luggage into the same overhead bin, which lead to a reduced number of aisle interference.

## Optimal Strategy

Based on the above analysis, we draw the conclusion that dividing passenger groups by columns is more efficient than by rows. The two optimal strategies are outside-in (23.0 min) and reverse-pyramid (22.7 min). Although R-P takes a little less time, outside-in is easier to operate both for gate agents and also passengers. Considering this, *we choose outside-in as our boarding strategy*.

## Cross-Validation between Theoretical and Simulation Models

We compare the results from the simulation with the results of our analytical mode, where we had total boarding time as

$$T = \frac{n}{v_{\text{flow}}} + \frac{25}{6} t_s \frac{n}{3} + \frac{n-1}{2} t_L.$$

Using parameter value estimates from van den Briel et al. [2003], we have

$$\bar{t_S} = \frac{25}{6} t_S = 9.7 \text{ s}, \qquad t_L = 7.1 \text{ s}.$$

We also estimate

$$\frac{1}{v_{\text{flow}}} = 4.5 \text{ s}^{-1}.$$

For the A320, we have $n = 126$, for which we calculate the total boarding time to be 23.2 min, a value that agrees closely with our simulation time.

# Mid-size Planes

We extend our simulation model and boarding strategies to midsize aircraft such as the A300; outside-in takes 24.6 min, reverse-pyramid takes 24.4 min.

The A300 has two aisles in economy class, with most (although not all) rows in a 2–4–2 seat configuration. Correspondingly, we adjust our simulation algorithm. Since there are two aisles but only one boarding gate, we divide the passengers into two lines and assume that they don't get into the wrong aisle.

The two strategies are again comparable in average boarding time; again, considering simplicity, we recommend outside-in.

# Large Planes

We extend our simulation model and boarding strategies to large aircraft such as the Airbus A380, with two decks and 555 seats in three classes.

Usually, the A380 opens two gates in front of the plane to let passengers board, one of which leads directly to the upper deck (where all business seats are located and a small portion of the economy seats) and the other goes to the main deck (where most economy seats are located).

Since seats in the upper deck are more spread out, it takes less time to board than the main deck. So we consider only the boarding process on the main deck, which is similar to that of a midsize plane, with two aisles and most rows with a 3–4–3 seat configuration. Both outside-in and reverse-pyramid take 26.8 min. We still recommend outside-in.

# Luggage Distribution Control (LDC)

## A Creative New Boarding Strategy

We offer a brand-new idea to reduce boarding time. During ticket-check time, the passengers are assigned numbers according to how many pieces of luggage they will take onto the plane. Although we do not completely control the order in which passengers board, we can control the distribution of passengers with different amounts of luggage.

A passenger in the last row of the plane blocks nobody when stowing luggage; a passenger in the front row blocks all other passengers behind. Let $P(r)$ denote the probability that a passenger in row $r$ blocks other passengers behind; $P(r)$ is a decreasing function of $r$. The expected aisle interference time that this passenger causes is

$$t_{A:I} = P(r)t_L,$$

where $t_l$ is the time to stow the luggage.. As for seat interference, it has no direct connection with the row number. We simply define the average seat interference time as $t_{S:I}$. So the total expected interference time is

$$T_{\text{total}:I} = \sum_{r=1}^{n}(t_{A:I} + t_{S:I}) = \sum_{r=1}^{n} P(r)t_L + T_{S:I},$$

where $T_{S:I} = \sum_{r=1}^{n} t_{S:I}$ is a constant.

A passenger with more luggage increases the total. To reduce the effect on interference time, we want to put this passenger as far back as possible.

# Simulation Results of LDC

Through simulation, we compare outside-in and reverse-pyramid strategies with our LDC strategy. With our LDC strategy, boarding times for all sizes of aircraft can be reduced by 2–3 min. That is because we send passengers with much luggage to the back of the plane, which reduces the number of interference times.

## How to Implement LDC?

Before passengers board, they exchange their ticket for a boarding card with their seat number. Our strategy is to assign seat numbers according to the amount of carry-on luggage. For the distribution of number of pieces of luggage, we use 60% have one piece, 30% have two pieces, and 10% have three.

We divide the seats from back to front in these proportions. We assign to passengers a seat in the group according to number of pieces of luggage; if seats in that group are exhausted, we still follow our basic principle: the more luggage a passenger takes, the farther back the seat.

# Orderly Deboarding

## Deboarding Strategies

Most airlines conduct deboarding without any organization. As a result, passengers in the front rows can easily get off first, stalling those behind, much like an inverse back-to-front procedure. This process is still faster than boarding. However, if we could adopt a strategy like outside-in, that is, let aisle passengers all get their luggage and get off the plane, then the middle passengers, and finally window passengers, we could fully use the aisle space without interference, leading to higher efficiency.

We put forward the deboarding strategies reversed from boarding strategies: random, front-to-back, inside-out, and V (the strategy derived from the reverse-pyramid boarding strategy).

## Deboarding Simulation Model

We develop a simulation model to compare deboarding strategies. Differing from the boarding process, deboarding has its own characteristics, as follows:

- All passengers start in different positions ("their own seat") and go to the same destination ("outside").
- There is no seat interference, since in most cases passengers in the same row will leave from aisle seat to window seat.
- In the boarding simulation model, passengers enter the plane one by one, forming a queue. During deboarding, the passengers are a crowd and everyone tries to get out of the plane first.

### Rush to One Goal: Object Position

During deboarding, passengers occupy the aisle. We cannot move the passengers according to a certain order, as in the boarding process, but have to consider the conflict that one position is wanted by several passengers. Therefore, we define the concept of *object position*, the position that a passenger wants to get into in the next time step. Our simulation program allows passengers to move forward by one cell in one time step; it can find out passengers' object positions before moving them, determine which passengers want to move to the same object position, determine which passengers cannot move because of obstacles, and then confirm which passengers can move forward and which must stay still. If an object position is wanted by several passengers, we randomly choose one to move and the others have to wait.

### Applicability

Our deboarding strategies are to divide passengers into several groups, and then let the groups deboard in order. We define a *PAD (Passengers Allowed to Deboard) set*, a set of passengers allowed to deboard together.

## Simulation Results and Analysis for Small Planes

We simulate each proposed deboarding strategy 100 times. Inside-out took 9.9 min, V 10.25, random 12.6, and front-to-back 14.0.

Compared with random and front-to-back, inside-out is better because it makes full use of the whole aisle, while the other two strategies only partly use the aisle. The main reason that we think the V-strategy is no better is that it needs to have more groups and it doesn't make full use of the aisle at the beginning and end of deboarding.

Is there any better strategy? Can inside-out be improved? During deboarding, passengers in the plane can still get their luggage as long as the aisle near their seats is empty. But during boarding, passengers who haven't boarded can do nothing but wait. Considering this, we find that there is no need to let the next group of passengers wait to deboard until the previous group is completely off the plane. We modify our model by changing it to when proportion $\alpha$ of the previous group still remains on board, we allow the next group to start to deboard—our *advanced inside-out strategy*. We find that $\alpha = 15\%$ to $20\%$ yields best results, a deboarding time of about 9.4 min instead of 9.9. There is no need to get an exact optimal value of $\alpha$, since it will be almost impossible for the flight crew to implement an optimal strategy exactly.

### Deboarding with Luggage Distribution Control

If the airline is using our LDC boarding strategy, we already know the distribution of luggage. In this case, our simulation program does not need to judge if a passenger has to get luggage and how long it takes. We simulate the inside-out strategy with different values of $\alpha$ under the luggage distribution given by our LDC boarding strategy. Again, $\alpha = 15\%$ to $20\%$ gives best results. The deboarding time too is reduced by 2–3 min; our LDC strategy can reduce not only boarding time but also deboarding time, because we put the passengers who need less time to get their luggage in the front of the plane. (The optimal value of $\alpha$ is not sensitive to the distribution of luggage.)

## Results for Midsize and Large Planes

When we apply the advanced inside-out strategy in midsize and large planes:

- The optimal value of $\alpha$ increases to 20–30%. The reason for this is possibly the increased number of rows in the deck.

# Testing of Simulation Models

Are our simulation results reliable? We apply probability theory.

We ran each simulation model 100 times. The times are independent trials from the same distribution. According to the Central Limit Theorem, the sample mean has approximately a normal distribution. As a result, we can make an interval estimate [Rozanov 1969]:

$$T = \overline{X} \pm \frac{s}{\sqrt{n}}\, t_{\alpha/2,n-1},$$

where $s$ is the sample standard deviation and $n = 100$. We choose 95% confidence. We find for each boarding strategy an interval of ±1 min, meaning that our simulation results are reliably consistent.

# Sensitivity Analyses

In reality, the boarding and deboarding times are influenced by various random events. Will these factors influence our simulation results?

- Occupancy level below 100%, that is, there will be empty seats. To show how occupancy affects our simulation result, we resimulate the strategies under occupancies from 20% to 90%. Result: If occupancy is more than 90%, there are no distinguishable changes in results with variation in time step size. If it is below 90%, the boarding time will be quite short and therefore affect boarding strategies very little.
- Passengers (especially those flying for the first time) may get into the wrong aisle in a midsize or large plane, which has more than one aisle. So we test strategies under a wrong-aisle possibility of 5%. Result: The boarding time increases by an average of 3 min. That is a long time! Proper guidance from the cabin crew is essential on midsize and large planes.

- Our boarding strategies can be implemented on all kinds of aircraft, because the outside-in strategy divides passengers by columns, so small variability in seat numbers won't affect our boarding strategy much.

# Further Discussion

## Passing

Our simulation models assume that passengers do not try to pass other passengers in the aisle. But in reality, research indicates that on average, one person in 10 does this.

## Boarding Stairs

We assume a boarding bridge, but in reality a boarding stairs may be used (e.g., on the Airbus A380). The difference is that the airport must send a bus to take the passengers from the waiting room to the boarding stairs. Airports want to make full use of the bus and take as many passengers as possible. As a result, boarding in groups according to our strategy is hard to implement. But if the number of passengers in the bus equals the number in each group, we can still adopt our boarding strategy. When they are not equal, we adopt the following boarding strategy: Let $R$ be the number of rows in the deck, with $R = pm + q$, where $m$ is the half-capacity of the bus, $p$ and $q$ are integers, and $q < m$. We implement outside-in for $pm$ rows in front; the other passengers are in one group and get on the plane randomly.

## Disobedient Deboarders

Some passengers do not follow directions. We introduce an obedience factor $\beta$, the proportion of obedient passengers, picked at random. Disobedient passengers get off the plane if they get the chance, regardless of whether it is their turn. When obedient passengers are less than 40%, any strategy is useless.

# Strengths and Weaknesses

## Strengths

- We develop a simple theoretical model that gives a rough estimate of airplane boarding time, considering the main factors that may cause boarding delay.

- We design a new boarding strategy that assigns seats according to amount of luggage, which could save about 3 min in boarding.

- With 95% confidence, our simulation results fluctuate by only 1 min.

## Weaknesses

- We don't consider the weight balance of a plane. Usually, the passenger and luggage distribution on the plane should be as uniform as possible.
- There are differences in seat configuration between our model and some actual planes.

# References


CAAC Resource. 2006. First flight of Airbus A380. `http://www.carnoc.com/css/carnoc2`.

Finney, Paul Burnham. 2006. Loading an airliner is rocket science. *New York Times* (14 November 2006). `http://travel2.nytimes.com/2006/11/14/business/14boarding.html`.

Funk, M. 2003. The visualization of the quantification of the commodification of air travel or: Why flying makes you feel like a rat in a lab cage. *Popular Science* 263 (5) (November 2003): 67–74.

Marelli, Scott, Gregory Mattocks, and Remick Merry. 1998. The role of computer simulation in reducing airplane turn time. *Aero Magazine* (4th Quarter 1998). `http://www.boeing.com/commercial/aeromagazine/aero_01/textonly/t01txt.html`.

Pan, Matthew. 2006. [No title.] `http://www.public.asu.edu/~dbvan1/papers/MatthewPanEssay.pdf`.

Rozanov, Y.A. 1969. *Probability Theory*. New York: Dover.

van dan Briel, Menkes H.L., J. René Villalobos, and Gary L. Hogg. 2003. The aircraft boarding problem. *Proceedings of the 12th Industrial Engineering Research Conference (IERC)*, article number 2153. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf`.

________, Tim Lindemann, and Anthony V. Mulé. 2005. America West Airlines develops efficient boarding strategies. *Interfaces* 35 (3) (May-June 2005): 191–200.

Van Landeghem, H. 2000. A simulation study of passenger boarding times in airplanes. `http://citeseer.ist.psu.edu/535105.html`.

# The Unique Best Boarding Plan? It Depends...

Bolun Liu
Xuan Hou
Hao Wang
National University of Singapore

Advisor: Yannis Yatracos

## Summary

We devise and compare strategies for boarding and deboarding planes of varying capacity. We clarify what properties a good strategy should have. We apply the same assumptions regarding basic boarding procedure, inner structure of planes, and behavior of passengers to all the cases.

For boarding, we study prevailing strategies and a seemingly excellent strategy, seat–by–seat, proposed in past literature, and categorize them into two types, assigned–seating and open–seating. We develop a model and a simulation for each type . Our criteria identify two good candidates, reverse-pyramid and open-seating. We develop our own comprehensive strategy, simulate it, and compare it with those two. However, the optimal boarding strategy is not the same for different planes. Some values of parameters, such as the passengers' luggage size and weight, greatly influence the final result. Based on these discoveries, we suggest how to modify a boarding procedure in practice to make it optimal.

For deboarding, a simple strategy beats a complicated one; but we still give a theoretically optimal model, then modify it to achieve a concise strategy applicable in practice.

## Introduction

Planes produce revenue while flying; thus, it is important to minimize turnaround time, the time between flights that an plane spends on the ground.

*The UMAP Journal* 28 (3) (2007) 385–404. 

For many airlines, boarding is the bottleneck. Reduction of boarding time results in profit increases while potentially increasing passenger satisfaction [van den Briel et al. 2003].

A number of different strategies have been implemented: back–to–front, rotating-zone, outside–in, and so on. We consult existing research and find the key issues in describing boarding mathematically and designing a good boarding algorithm.

Finally, we analyze deboarding and see how its time can be minimized.

# Judging Standards

**Efficiency:** Minimizing total boarding time is our primary target.

**Passenger satisfaction:** We use two measures to describe passenger satisfaction: the proportion of passengers satisfied, and average individual boarding time.

**Feasibility:** Whether and how the strategy is applicable in reality. Generally speaking, the more complicated the boarding strategy, the less feasible it is.

Shorter total boarding time is preferred by both airlines and passengers. There are two measures of boarding time: total boarding time (which affects airline profits directly) and average individual boarding time (which influences passengers' satisfaction with the airline and thus indirectly airline profits).

We cannot find industry standards for passenger satisfaction and feasibility. However, we can define them descriptively, rather than merely numerically.

# Assumptions

- We consider only coach and business–class passengers. Passengers with special needs and first–class passengers are only a small portion of all passengers and are seated before the majority, so their boarding time is assumed to be fixed and has little influence on our models.

- Each passenger is allowed one piece of carry–on luggage and one personal item (purse, computer, briefcase, or small tote, etc.) [Airline Carry-On Luggage Regulations 2007]. Each passenger puts any luggage into the overhead compartment and takes any personal item to the seat. Occasional withdrawal of excessive carry–on luggage is not in our scope.

- When boarding starts, all passengers have arrived at the boarding gate. Missing and late passengers are not in our scope.

- The aisles of planes are narrow and do not allow a passenger to pass another passenger in the aisle.

- The boarding time of an individual passenger is from entry into the plane (and thus the aisle) to sitting down.
- The total boarding time is from the entry of the first passenger into the plane (and thus the aisle) to when the last passenger sits down.
- In all our simulations:
  - For an instance of small planes, we take the Boeing 737
  - For an instance of midsize planes, we take the Airbus A340, which has its 264 seats typically arranged 2–4–2 with two aisles.
  - For an instance of large planes, we take the Airbus A380. It is double–deck, typically with 350 seats arranged 3–4–3 in the main deck and 176 seats arranged 2–4–2 in the upper deck. When boarding, passengers of either deck board at the same time, through two doors, either of which is the front door of its deck.

# The Assigned–Seating Model

We establish a model of assigned–seating boarding strategies, first for small planes, with only one aisle, then for larger planes.

## The Boarding Process

- The gate agent announces boarding, then calls groups one after another. The passengers of a called group start queueing at the gate, with occasional conflicts and controversy about position in the queue.
- The gent checks boarding passes before passengers enter the plane.
- Passengers board through the front door. For a long-haul plane, passengers may board through an additional rear door.
- Passengers enter the plane. Because the aisles in the plane are narrow, when a passenger is putting luggage into the overhead compartment, or stops for another reason, the passengers behind have to line up. As soon as the passenger enters the seat row, the passengers behind can move again.
- Each passenger switches from moving in the aisle to moving in the seat row, after putting luggage into the overhead compartment, and finally sits. If there are other passengers between the passenger and the assigned seat, the passenger has to wait to pass through.
- When the last passenger gets seated, boarding ends.

# Detailed Algorithm

## Parameters

- $x_1, \ldots, x_n$ The sequence of passengers in the boarding queue.
- $l$ The distance between rows (leg room), including the thickness of the seats. It is a fixed value for one plane but it may differ for different models of planes or different airlines.
- $r$ The number of rows of seats in the plane.
- $S$ The aisle length of the plane, where $S = lr$.
- $m$ The number of boarding groups.
- $G_1, \ldots, G_m$ The sequence of boarding groups, according to the position of their assigned seats. The smaller the subscript, the earlier the group boards.
- $r_q$ The row of the $q$th passenger's assigned seat.
- $t_q$ The time when the $q$th passenger enters the plane (thus enters the aisle). We assume that $t_1 = 0$, that is, we set the time to be 0 when the first passenger enters the plane.
- $T_q$ A random variable denoting the time difference between the $q$th and $(q+1)$st passengers' entry into the plane; $T_q = t_{q+1} - t_q$, with mean $\overline{T}$ and variance $\sigma_T^2$.
- $S_q$ The position of the $q$th passenger's assigned seat. We define it as $S_q = lr_q$.
- $w_q$ The aisle space that the $q$th passenger occupies, luggage and safe distance between two passengers included. We assume that $w_q$ is a random variable following a normal distribution with mean and $\bar{w}$ and variance $\sigma_w^2$.
- $v_q$ The moving speed of the $q$th passenger in the aisle; it follows a normal distribution with mean $\bar{v}$ and variance $\sigma_w^2$.
- $a_q T_{q,L}$ The time that the $q$th passenger spends putting luggage into the overhead compartment. $T_{q,l}$ follows a triangular distribution [Van Landeghem and Beuselinck 2000] with mean $\overline{T}_L$ and variance $\sigma_L^2$. The $a_q$ is a coefficient random variable relevant to the size, weight and shape of $x_q$'s luggage; it follows a normal distribution with mean $\bar{a}$ and variance $\sigma_a^2$. The variables $a_q$ and $T_{q,L}$ are independent.
- $T_{q,0}$ The time that the $q$th passenger spends to pass through an empty seat when moving in a row. It follows a triangular distribution with mean $\overline{T}_0$ and variance $\sigma_0^2$.

- $T_{q,1}$ The time that the $q$th passenger spends to pass through a seat in which another passenger is sitting when moving in a row; it follows a triangular distribution with mean $\overline{T}_1$ and variance $\sigma_1^2$. This includes the time that the seated passenger stands up and that the $q$th passenger passes through.

- $\overline{T}_{q,S}$ The time that the $q$th passenger takes to sit; it follows a triangular distribution with mean $\overline{T}_S$ and variance $\sigma_S^2$.

- $x_q(t)$ The position in the aisle of the $q$th passenger at time $t$. We define $x_1(t_0)$ as the position of the aisle's entrance and set $x_1(t_0) = 0$.

## Mathematical Assumptions

- We divide passengers into groups $G_1, \ldots, G_m$. Passengers in the same group queue randomly. Then all the group queues connect with each other, in order of subscript from small to large, to form a total queue. The two extremes are: just one group (so the position of each passenger in the total queue is randomly determined), and the number of groups equals the number of passengers (each group consists of only one passenger, and the position of each passenger in the total queue is fixed by the assigned seat).

- There is no waiting time between two consecutive groups.

- When the $q$th passenger has entered the plane (thus enters the aisle), the passenger is in one of two states: moving with a constant speed $v_q$ or standing still. Standing still can further be divided into three categories:
  - Someone ahead is putting luggage in, so the aisle is congested and the queue cannot move forward.
  - The passenger is stowing luggage.
  - The passenger is finished stowing luggage, but the aisle seat of the row of the passenger's seat is now occupied by someone else.

- The $q$th passenger has three states when in the seat row
  - Passing through an empty seat (with time $T_{q,0}$);
  - Passing through a seat occupied by another passenger (with time $T_{q,1}$);
  - Waiting (standing still).

- The passenger is sitting down (with time $T_{q,S}$).

## The Algorithm

We now propose a detailed algorithm to calculate both total boarding time and individual boarding time for any assigned–seating boarding strategy.

### Motion in the Aisle

The $q$th passenger's motion is determined if we know the situation in the aisle ahead, which is determined only by the passengers entering earlier, namely, the 1st to the $(q-1)$st passengers. We record the position and some other information about each passenger and use iteration to do the calculation.

Let $W_Q(t)$ be the interval of space in the aisle that passenger $q$ occupies at time $t$:

$$W_q(t) = \begin{cases} \left[\max\{0, x_q(t) - \frac{1}{2}w_q\},\ \min\{S, x_q(t) + \frac{1}{2}w_q\}\right], & x_q(t) \neq 0; \\ \emptyset, & x_q(t) = 0. \end{cases}$$

We assume that $x_q(t) = 0$ from the moment that the $q$th passenger enters the seat row (the passenger "disappears from the aisle").

Let $C_q(t)$ be the space in the aisle occupied by all the passengers in front of $x_q$ at time $t$:

$$C_q(t) = \begin{cases} \cup_{p=1}^{q-1} W_p(t), & q > 1; \\ \emptyset, & q = 1. \end{cases}$$

From the time $t_q$ when $x_q$ enters the aisle, we define

$$t_{q,k} = \begin{cases} 0 & k = 0; \\ t_q + 0.1k, & k = 1, \ldots. \end{cases}$$

We let the computer) calculate and record $W_q(t)$ and $C_q(t_{q,t})$ every 0.1 s.

Let

$$x_q(t_{q,k+1}) = \begin{cases} \min\{x_q(t_{q,k}) + 0.1v_q,\ S_q\}, & W_q(t_{q,k}) \cap C_q(t_{q,k}) = \emptyset; \\ x_q(t_{q,k}), & \text{otherwise}. \end{cases}$$

In the first case, passenger $q$ moves at some speed during the next 0.1 s until reaching the assigned seat row within the next 0.1 s. In the second case, passenger $q$ stays unmoved during the next 0.1 s.

We denote the time when $x_q(t_{q,k}) = S_q$ (the passenger has reached the assigned row) by $t_{q,k_0}$. When $x_q$ reaches the assigned row, $x_q$ starts to stow luggage, taking time $a_q T_{q,L}$.

**Motion in a Row**

We say a seat is "occupied" only if a passenger is passing through it or the passenger to whom this seat belongs is getting seated; otherwise, the seat is not occupied, *even if the passenger to whom this seat belongs has already been seated* (since another passenger can pass through it). Then, $x_q$ at time $(t_{q,k_0} + a_q T_{q,L})$ finishes stowing luggage. We now check whether the aisle seat of the row is occupied:

- If it is occupied, $x_q$ waits in the aisle until it is clear.
- If not occupied, $x_q$ enters the row (thus is no longer in the aisle) and occupies the space of the aisle seat at time $(t_{q,k_0} + a_q T_{q,L}) + 0.1$.

- If this is the assigned seat, $x_q$ then spends time to get seated. During this time period, nobody else can pass though or occupy this seat. After $x_q$ sits down, another passenger could pass through this seat
- If this is not the assigned seat, $x_q$ spends time $T_{q,0}$ (if nobody is sitting in it) or $T_{q,1}$ (if someone is sitting on it) to pass through it. During this time period, nobody else can pass though or occupy this seat. After $x_q$ passes through this seat, we check whether the next seat is occupied, in the same manner, until $x_q$ gets seated.

# Simulation

## The Seven Candidates

Most of the airlines use one of the following six boarding strategies [Airplane Boarding . . . 2007]:

**Back-to-front** (US Airways, Air Canada, British Airways). Divide the seats into 4–6 blocks and board passengers from the back block to the front block.

**Rotating-zone** (AirTran) Divide the seats into 4–6 blocks, and board passengers in the sequence (back block, front block, next back block, the next front block, etc.).

**Random** (Northwest) Impose no sequence and let passengers enter randomly.

**Block** (Delta) Divide the seats into two or three blocks, then divide each block into a window-seats block and non-window-seats block. Label the back window-seats as block 1, the back non-window-seats as block 2, the second back window-seats as block 3, etc.

**Reverse-pyramid** (US Airways) Rear window and middle first, front window and middle next, followed by rear aisle, then front aisle.

**Outside-in** (United) Window seats first, followed by middle, then aisle seats.

**Figure 1** shows the patterns of the above six boarding strategies. The numbers are group numbers.

## The 7th Approach

Van Landeghem and Beuselinck [2000] include one of the extreme boarding strategies that we raised earlier—seat-by-seat. In their simulation, this strategy performed rather well. Thus, we also include this strategy in our simulation.

To be concrete, we use the following boarding sequence.

> In the first round, one seat from each row is called once; in second round, a second seat of each row is called; etc. In each round, the gate agent always calls from back row to front row. This eliminates aisle delay (delay due

to staying in the aisle), except for when the last passenger in the previous round (having a front seat) has not finished stowing luggage when the first passenger in the next round (having a back seat) enters the plane.

Van Landeghem and Beuselinck claim that this is the best strategy. But they assume that the time for a passenger to pass through an empty seat is the same as to pass through a seat with a person in it. In our model, we distinguish these two times.

In addition, we let window seats be filled first, followed by middle seats, and then aisle seats. Following this strategy, we have no row delay whatsoever.

## The Simulation

- We simulate the seven boarding strategies stated with only small planes, using a Boeing 737.
- We assume variable and parameter values as in **Table 1**.

**Table 1.**
Assumed variable and parameter values.

| Symbol | Value | Symbol | Value | Symbol | Value | Symbol | Value |
|---|---|---|---|---|---|---|---|
| $l$ | 0.7 m | $r$ | 23 | $\bar{w}$ | 1 m | $\sigma_w$ | 0.2 m |
| $\bar{v}$ | 1 m/s | $\sigma_v$ | 0.2 m/s | $\bar{a}_q T_{q,L}$ | 30 s | $\sigma_{a_q T_{q,L}}$ | 10 s |
| $\overline{T}$ | 9 s | $\sigma_T$ | 3 s | $\overline{T}_0$ | 1 s | $\sigma_T$ | 0.2 s |
| $\overline{T}_1$ | 5 s | $\sigma_{T_1}$ | 5 s | $\overline{T}_S$ | 0.5 s | $\sigma_{T_s}$ | 0.1 s |

- We iterate the simulation 100 times, obtaining the sample mean and sample variance of total boarding time and average individual boarding time. **Table 2** and **Figures 2–3** show the results.

**Table 2.**
Simulation results for boarding times for the strategies, in order of increasing average total time.

| Strategy | Total | | Average | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Seat-by-seat | 15.6 | 0.11 | 1.6 | 0.05 |
| Reverse-pyramid | 22.4 | 0.56 | 2.5 | 0.24 |
| Outside-in | 22.8 | 1.06 | 2.2 | 0.26 |
| Block | 24.7 | 0.35 | 2.6 | 0.15 |
| Random | 25.2 | 0.51 | 2.5 | 0.14 |
| Back-to-front | 26.2 | 0.39 | 2.9 | 0.18 |
| Rotating-zone | 26.9 | 0.83 | 3.1 | 0.22 |

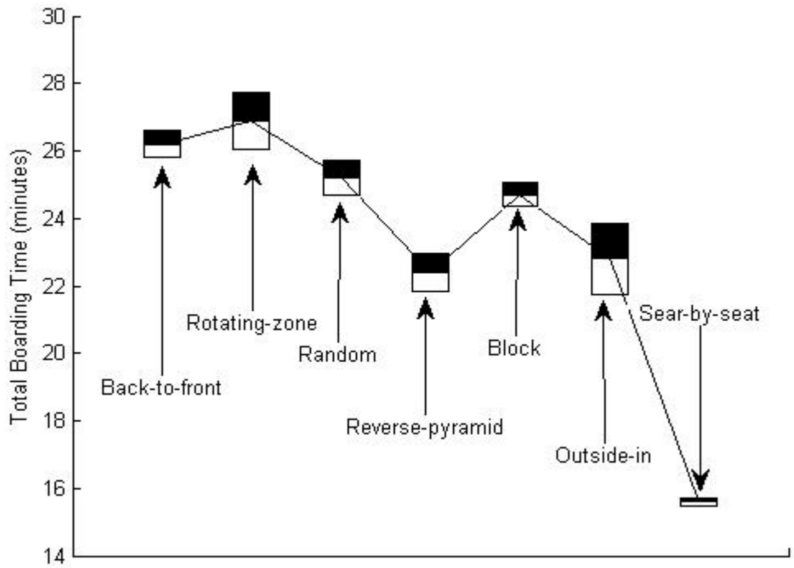



**Figure 2.** Average total boarding times for the strategies.

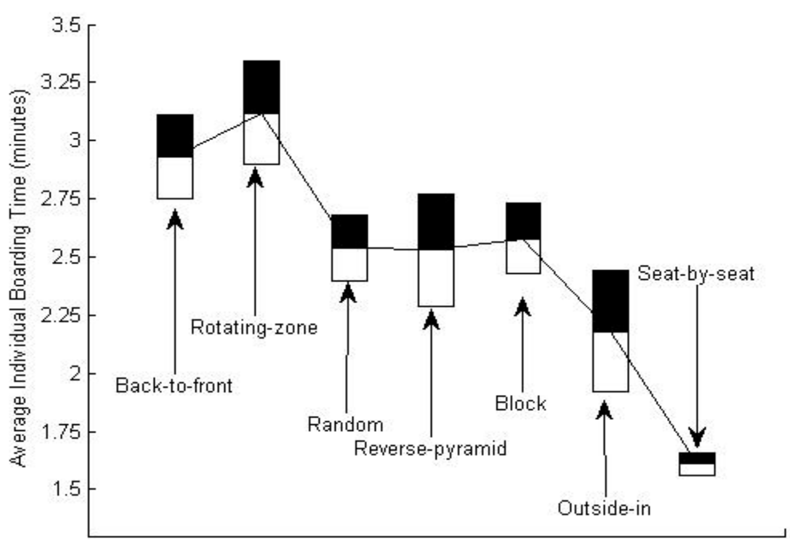



**Figure 3.** Average individual boarding times under the strategies.

### Results Analysis

The results of the above simulation indicate that

- Seat-by-seat is the most efficient boarding strategy. However, its boarding process is too complicated Thus, it is hard to execute in practice, and we eliminate it.
- Reverse-pyramid and outside-in are the next most efficient. Reverse-pyramid is better in having smaller variation in boarding time.

In addition, since reverse-pyramid and outside-in are widely used, they are feasible and satisfactory, hence have the best comprehensive quality.

## More on Reverse-Pyramid

Seat-by-seat and outside-in are two extremes of reverse-pyramid: Seat-by-seat is reverse-pyramid with the most boarding groups, while outside-in is reverse-pyramid with the fewest boarding groups.

This discovery inspires us to study reverse-pyramid further and identify the following properties of a reverse-pyramid-type strategy:

- Each boarding group has approximately the same number of passengers.
- After each group finishes boarding, there must be a passenger sitting behind each passenger seated (except for back-row passengers).
- After each group finishes boarding, the numbers of passengers are decreasing from the window columns to the aisle columns. Moreover, unless all the seats in a column have been occupied, they are strictly decreasing.
- Every passenger is only permitted to board at least one round after the one who will sit abreast and closer to the window.
- The seats for each group are symmetric with respect to the axis of the plane.

We claim that the above five properties are necessary and sufficient conditions for a reverse-pyramid type strategy.

Additionally, we suggest the following three criteria when grouping the seats for a reverse-pyramid type strategy:

- For a single-aisle plane, just follow the above five points.
- For a double-aisle plane, first divide the seats into two halves by the axis of the plane. If the seats in the middle part have an odd number of columns: For the seats lying on the axis, assign every second seat (from the first one) to either half, and the rest to the other half. After that, for either half, group the seats following the above five points.

- The number of seats in each group is at least about 30. Divide all the seats into 4–8 groups.

With these grouping criteria, we can group seats for any size plane, thereby simulating reverse-pyramid for midsize and large planes.

# The Open-Seating Model

## Description

Passengers board in the order of their check-in: the earlier you check-in, the earlier you can board. In addition, since seats are not assigned to passengers in advance, a passenger can choose from the open seats at boarding time. Thus, passengers boarding earlier have a wider range of choices and are more likely to select a satisfactory seat.

Southwest Airlines uses this open-seating policy. Their boarding procedure is described as follows [2007]:

> Customers get assigned to Groups A, B or C on their boarding passes, in the order in which the passenger checks in. Groups are called in alphabetical order, with passengers rushing to occupy the seat of their choice.

## Gains and Losses

### Gains

Compared to assigned-seating strategies, open-seating is more efficient [Finney 2006], for two reasons:

- Passengers in the same group compete with one another for seats, so passengers hurry.

- Most passengers have wide preferences for particular kinds of seats. Therefore, when there is congestion, a passenger will probably choose a seat before the congestion, rather than wait for the aisle to clear.

### Losses

Although minimizing boarding time is our primary goal, the level of satisfaction is also important. Customer reviews of Southwest Airlines reveal that many people do not like open-seating. The main types of dissatisfaction can be categorized as follows.

- People may want to sit next to friends and relatives, but with open-seating this may not happen.

- Some people are used to boarding with seats assigned in advance. They do not want to rush or "compete" with others.
- Some people not distinguished as having "special needs" still have comparatively low capability of "competing" for seats.

# Algorithm

To compare open-seating with assigned-seating strategies, we need to calculate the boarding time for open-seating strategy. With open-seating, however, there may be unpredictable events. For example, two passengers may have conflict when they want to choose the same seat. To propose a systematic algorithm, we must make more assumptions to simplify the behaviors of passengers.

## Assumptions

- We use Southwest's open-seating strategy of three groups (A, B and C). Group A passengers enter the plane before Group B, followed by Group C.
- All passengers prefer the same types of seats (we specify these later).
- All passengers behave rationally and politely. That is, when a seat a passenger prefers is taken by someone else, the passenger finds another seat.
- Once a passenger is seated, the passenger does not move again.
- The moving speed of passengers is faster than in assigned-seating boarding.

## Passenger Preferences

We categorize seats into three classes of preference for each passenger: most preferred, preferred, and not preferred. A passenger first checks for most preferred seats; if there are none, the passenger considers seats in the next class.

For each passenger, every preference class consists of only two types of seats: a certain column (for example, window seats), and/or a certain range of adjacent rows (for example, seats in Row 10–15). A passenger who has both types of preference in a class has a preference between them.

## The Boarding Process for a Passenger

- For a Group A passenger:
  - When the passenger just enters the plane (and thus the aisle), if there is someone stowing luggage at row $i$, the passenger checks whether there are most-preferred seats in front of row $i$, chooses the nearest one if there

is one, and otherwise moves forward to wait just behind the passenger stowing luggage at row $i$.

- When the passenger at row $i$ finishes stowing luggage and clears row $i$, our passenger moves on and repeats the above if there is someone stowing luggage at row $j > i$ and iterates until there is no one in the way.
- The passenger now checks whether there are any most-preferred seats among all the seats from the row where he/she is to the last row, chooses the nearest available one or if none are available checks whether there are preferred seats in this area. If there are, the passenger chooses the nearest one; if not, the passenger choose the nearest empty seat.

- A Group B passenger's behavior is similar, except that the passenger also checks whether there are preferred seats every time after checking for most-preferred seats and not finding one.
- A Group C passenger's behavior is similar, except that the passenger also checks whether there are empty seats every time after checking for preferred" seats and not finding one.

## Simulation

**Table 3** shows the results of our computer simulations with the same parameter values as before, together with results for a combined strategy to be developed in the next section. We do not simulate open-seating for large planes. Since they are for long voyages, a larger proportion of passengers prefer assigned-seating and boarding time has less impact on airline profits.

**Table 3.**

Simulation results for boarding times for the open-seating and reverse-pyramid strategies.

| | Small plane | | | | midsize plane | | | |
|---|---|---|---|---|---|---|---|---|
| | Total | | Average | | Total | | Average | |
| Strategy | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| Open-seating | 19.6 | 0.3 | 1.7 | 0.2 | 37.4 | 1.5 | 1.5 | 0.3 |
| Reverse-pyramid | 22.4 | 0.6 | 2.5 | 0.2 | 40.8 | 1.2 | 1.5 | 0.3 |
| Combined | 21.1 | 0.6 | 2.3 | 0.3 | 38.9 | 1.3 | 1.6 | 0.2 |

# A Comprehensive Model

## Motivation

A good boarding strategy should perform well in three aspects: efficiency (short total boarding time), passenger satisfaction (short average individual boarding time), and feasibility.

Although the three representative boarding strategies—reverse-pyramid (outside-in), seat-by-seat, and open-seating—each have advantages, they also have drawbacks:

- Reverse-pyramid / outside-in has a longer boarding time than open-seating.
- Open-seating dissatisfies a notable number of passengers.
- Seat-by-seat is not feasible in practice.

We are then motivated to develop a comprehensive model, with the aim of including all their advantages and eliminating their drawbacks.

# Boarding Strategy

We group the seats as in reverse-pyramid. We divide the groups and their corresponding seats into two categories, with $G_1, \ldots, G_i$ in Category 1 (passengers who board in the manner of open-seating) and $G_{i+1}, \ldots, G_m$ in Category 2 (passengers who board based on assigned-seating).

By devising the boarding strategy in this way, we ensure that

- Passengers who prefer assigned-seating are assured a fixed seat at check-in.
- Passengers who prefer open-seating boarding can select seats at will.
- This will beat the open-seating policy in regard to passenger satisfaction

When the boarding starts:

- Passengers who chose open-seating board first, group by group. We let the number of these groups now be $\min\{3, i\}$. But we first mark out certain seats, which are specially for open seating passengers.
- Then the assigned-seating passengers also board group by group.

# Fixing the Number of Groups

In our new model, the new parameter is $i$, the number of groups for open-seating. The value of $i$ will vary with the number of groups (that reverse-pyramid has initially), model of plane, and ratio of passengers preferring the two boarding manners.

We assume that a proportion $A$ of passengers prefer open-seating and a proportion $(1 - A)$ prefer assigned-seating. Since a fixed optimal grouping scheme of reverse-pyramid is determined by the seat plan of the plane, there is a unique $i_0$ that satisfies

$$\frac{\sum_{j=1}^{i_o} n_j}{n} \le 1 - A, \qquad \frac{\sum_{j=1}^{i_o+1} n_j}{n} \ge 1 - A,$$

where $n_j$ is the number of passengers in group $j$ and $n$ is the total number of passengers.

The value of $i$ can be set to either $i_0$ or $(i_0 + 1)$; the choice depends on the airline's weighting between efficiency and passengers' satisfaction. If the airline considers efficiency more important, it should let more passengers board according to open-seating and set $i = i_0 + 1$; if the airline considers passenger satisfaction more important, it should let more passengers board with assigned seats and set $i = i_0$.

When $i = 0$ or $i = 1$, we do not provide for open-seating, for which the value of $A$ is supposed to be large (for example, 85%). The size of a plane is usually related to the length of the trip (small planes for short trips, bigger planes for longer trips). Usually, the longer the trip, the more passengers prefer assigned seating and the less impact of boarding time on profits. Hence, for larger planes, the portion of seats for open-seating is smaller.

## Advantages

- Based on the simulation results, reverse-pyramid-type strategies have the highest efficiency in the assigned-seating model. In addition, the grouping in reverse-pyramid boarding strategy is good because
  - It has comparatively more groups. This lets us have more flexibility to arrange the seats for the two boarding types.
  - If the value of $i$ does not go to extremes, each boarding type will have seats of all features (against the window, beside the aisle, in the front, in the back, next to each other, and so on). This enhances the range of choices for passengers from both boarding types.
- This boarding strategy considers preferences of passengers, so it is quite likely to have higher customer satisfaction.
- Its boarding process is open-seating passengers first, followed by passengers boarding in reverse-pyramid. This process actually takes into account possible confusions during open-seating passengers' boarding and prevents most passengers from witnessing the confusions.
- Both boarding manners are feasible and are used, so we infer that the new combined strategy too is feasible.

# Testing the New Model

## Comparison

**Table 3** earlier also displays results for the combined strategy, which has shorter total boarding time and shorter average individual boarding time than reverse-pyramid but is longer on both counts than open-seating.

Therefore, we suggest:

- For a small plane (85–210 passengers), use open-seating or the combined strategy.
- For a midsize plane (210–330 passengers), use the combined strategy.
- For a large plane (450–800 passengers), results not displayed show little difference between reverse-pyramid and the combined strategy (both take over an hour).

# Sensitivity Analysis

We repeated the simulation many times, using different values for the parameters. The resulting data shows that, no matter which strategies we use, the following two input parameters have the biggest impact on boarding time:

- $\overline{T}$, the expected time difference between two adjacent passengers' entry to the plane.
- $\bar{a}$, the expected value of the coefficient random variable that is relevant to the size, weight, and shape of the passengers' luggage.

We give the following suggestions for decreasing total boarding time:

- For small planes, to the extent that it is bearable for passengers, lower the limits for carry-on luggage size and weight.
- For midsize and large planes, have more training for the gate agents or more gate agents. Since large planes are usually for long flights, passengers often need more luggage, so it is not proper to set the luggage limit too low.

# Deplaning

## An Ideal Model

Total deplaning time is the time from the first passenger standing up to the last passenger leaving the plane.

We assume:

- The speed of all passengers moving in the aisle when deplaning is constant.
- The time that all passengers spend in picking luggage is constant.

The *deplaning queue* is an imaginary queue that is formed if we join all passengers in a line in the order that they deplane, and if the passengers who have left the plane would continue moving forward in the queue.

We now propose our ideal deplaning strategy:

- Passengers in aisle seats on one side of the aisle (say all the C seats) as a whole stand up at the same time, which is the beginning of the deplaning process. They take their luggage and leave the plane as a whole.
- As soon as the last passenger (23C) leaves the 23rd row, the passenger in seat 23D moves into the aisle, occupies it in no time, and takes luggage—but does *not* (yet) move forward.
- As soon as the 23C passenger passes row 22, the 22D passenger moves into the aisle, occupies it in no time, and takes luggage—but does *not* (yet) move forward.
- The passengers in the D seats behave in the same manner, until the 1D passenger is in the aisle with luggage. Then all the D-seat passengers move to leave the plane as a whole.
- Thereafter, all the B, E, A, and F passengers repeat what the D passenger did.
- When the last passenger (F23) leaves the plane, the deplaning ends.

Using the above strategy, total deplaning time is minimized. The argument is as follows.

> There are only five segments of unoccupied space in the deplaning queue, and all appear in front of the row 1 passengers. Every passenger has to spend time in the aisle taking luggage. When a row 1 passenger is taking luggage, the passenger who deplanes before that passenger is moving in the deplaning queue. Thus, these five segments of unoccupied space cannot be eliminated, no matter what deplaning strategy we use.

We then say that all the passengers are divided into six "rounds" by those five segments of unoccupied space.

## Reality

The airlines have little control of the behavior of passengers deplaning, so any detailed strategy would be very difficult to implement. Therefore, instead, we now propose a concise criterion for passengers to deplane.

> We use again the concept of "rounds," but with a small modification: Each round has 23 passengers, coming from all the 23 rows; for each row, the passenger stepping out is either of the ones on both sides who are nearest to the aisle.

In practice, the crew can announce the criterion before passengers deplane. Even with occasional violations, we do better.

# Conclusion

Open-seating beats the existing assigned-seating strategies in total boarding time but it loses in terms of passenger satisfaction. Seat-by-seat wins in both aspects; but it is not feasible in reality, thus useless.

We combine opening-seating with the most efficient feasible assigned-seating strategy, namely, reverse-pyramid (outside-in). We expect the combined strategy to to be feasible and good on of our criteria.

As for deplaning, airlines do not have as much control as they do in boarding, so a complex plan cannot be executed well. Thus, it's better have a simple procedure than a detailed but unfeasible one.

# Strengths and Weaknesses

## Reverse-Pyramid Boarding Strategy

### Strengths

- The most efficient of the prevailing assigned-seating strategies.
- Methodical, so there will be little confusion during boarding.
- Passengers can probably sit next to friends and relatives as they wish.

### Weaknesses

- Not as efficient as open-seating.
- Requires more staff to execute and control boarding.

## Open-Seating Boarding Strategy

### Strengths

- Higher efficiency than all the assigned-seating boarding strategies.
- For a few people, especially the young, it is attractive.
- Looks simple and requires less staffing to execute.

### Weaknesses

- Dissatisfaction of passengers is the vital drawback.

## Combined Boarding Strategy

### Strengths

- More efficient than traditional assigned-seating boarding strategies.
- Meets the needs of different types of passengers, thus probably making the airline more satisfactory and popular.

### Weaknesses

- It might be tedious for airlines to set the portions, and surveys may be needed.

# References


Finney, Paul Burnham . 2006. Loading an airliner is rocket science. `http://travel2.nytimes.com/2006/11/14/business/14boarding.html` . Accessed 10 February 2007.

Luggage Online, Inc. 2007. Airline carry-on luggage regulations. `http://www.luggageonline.com/about_airlines.cfm` . Accessed 9 February 2007.

Marelli, Scott, Gregory Mattocks, and Remick Merry. 1998. The role of computer simulation in reducing airplane turn time. *Aero Magazine* (4th Quarter 1998). `http://www.boeing.com/commercial/aeromagazine/aero_01/textonly/t01txt.html` .

Southwest Airlines. 2007. Boarding process. `http://www.southwest.com/travel_center/boarding_process.html` . Accessed 10 February 2007.

van den Briel, Menkes H.L. n.d. Airplane boarding. `http://www.public.asu.edu/~dbvan1/projects/boarding/boarding.htm` . Accessed 9 February 2007.

________, J. René Villalobos, and Gary L. Hogg. 2003. The aircraft boarding problem. In *Proceedings of the 12th Industrial Engineering Research Conference (IERC-2003)*, No. 2153, CD-ROM. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf` .

Van Landeghem, H., and A. Beuselinck. 2000. Reducing passenger boarding time in airplanes: A simulation based approach. *European Journal of Operational Research* 142: 294–308. Copy on request from `menkes@asu.edu` .

Bolun Liu, Hao Wang, advisor Yannis Yatracos, and Xuan Hou.

# Airliner Boarding and Deplaning Strategy

Linbo Zhao
Fan Zhou
Guozhen Wang
Peking University
Beijing, China

Advisor: Xufeng Liu

## Summary

To reduce airliner boarding and deplaning time, we partition passengers into groups that board in an arranged sequence. We assume that first-class and business-class passengers board first; our model treats only economy class. Since deplaning is the converse process of boarding, a strategy for boarding gives a strategy for deplaning.

We develop a model of interferences among passengers, which determine boarding time. We try to find a strategy with the least interferences. By running Lingo, we tackle the resulting nonlinear integer programming problem and obtain near-optimal strategies for fixed numbers of groups. This model supports the outside-in and reverse-pyramid strategies.

We develop another model to give a global lower bound for interferences. We also prove that individual boarding sequence, which boards passengers one by one in a particular order, attains that lower bound.

We develop code in C++ to simulate boarding strategies and test various strategies for three airliners: Canadair CRJ-200 (small), Airbus A320 (midsize) and Airbus A380 (large). Individual boarding sequence, reverse-pyramid, and outside-in are the best three strategies in terms of both average boarding time and its standard deviation.

We test strategies under various luggage loads and levels of occupancy, with and without late passengers and those with special needs. Outside-in and reverse-pyramid are stable under variation of parameters, whereas individual boarding sequence is extremely sensitive, though not to luggage.

*The UMAP Journal* 28 (3) (2007) 405–419. 

Our conclusions discredit traditional back-to-front strategies and support individual boarding sequence, reverse-pyramid, and outside-in. The more groups, the worse the situation with back-to-front. Taking cost into consideration, random sequencing should also be recommended.

Finally, we analyze deplaning and see how its time can be minimized.

# Introduction

Airliner turnaround time is an important factor in determining airplane productivity [van den Briel et al. 2005], and boarding time constitutes an important part of turnaround time. *Boarding time* is the period between when the first passenger enters the plane and when the last passenger is seated.

Deplaning is also an essential part of turnaround time. We regard deplaning as the converse process of boarding, so an efficient boarding strategy brings out an equally efficient deplaning strategy.

A common approach in boarding is to partition passengers into several groups and board the groups in a sequence:

- **Back-to-front:** This is a traditional strategy.
- **Outside-in:** Window seats first, middle seats second, and aisle seats last.
- **Reverse-pyramid:** Discovered by van den Briel et al. [2005]. It is in use.
- **Individual boarding sequence:** Passengers are called to board one by one according to their seat number. This strategy is criticized as impractical; no known airline uses it.
- **Random:** There is no sequencing at all.
- **Rotating:** The deck is divided into blocks from back to front. The back block is called first, then the front block, the next back block third, continuing until the blocks meet in the middle.
- **Free seat choice:** Some airlines do not preassign seats to passengers; passengers choose their seat after boarding [Ferrari and Nagel 2005].

## Previous Work

The back-to-front procedure is used by many major airlines [Finney 2006]. Recent research [Ferrari and Nagel 2005; Van Landeghem n.d.; Van Landeghem and Beuselinck 2002; van den Briel et al. 2005] discredits the effectiveness of back-to-front and recommends a version of outside-in, such as reverse-pyramid. These versions perform with similar efficiency, as substantiated by simulations.

Previous work tends to compare known boarding strategies. Research based purely on simulation is not fully satisfactory because of its lack of rigor. Among previous work, van den Briel et al. [2005] seems to offer the only well-rounded research. They combine analytical model, simulation, and practical implementation, but use a complex nonlinear integer programming problem. We simplify their model and do similar nonlinear integer programming.

Ferrari and Nagel [2005]:

- point out deficiencies in the simulation of Van Landeghem and Beuselinck [2002];
- offer more details on passenger behavior, such as a bin occupancy model and seating model, which they use to describe the effect of seat interferences;
- consider the passengers' seat preferences, while van den Briel et al. do not; and
- pay special attention to robustness.

Van Landeghem and Beuselinck [2002] is an excellent work, with data from the national carriers' database and from interviewing gate agents and method engineers. They give distributions for walking speed, time to store luggage, and time to sit. They also consider the distribution of the number of luggage items. They conclude that the most effective strategy is the outside-in by-seat strategy but do not offer a rigorous argument to defend that conclusion. They find that random sequencing, used frequently today, preforms well compared to most other strategies, which is somewhat surprising; only 9 of the 46 strategies that Van Landeghem and Beuselinck study do better than random. They conclude that in taking a structured approach to boarding, one should beware of making things far worse by choosing a wrong way of sequencing.

The data from Van Landeghem and Beuselinck [2002] and Van Landeghem [n.d.] have a strong impact on our simulation.

# Problem Analysis and Basic Approach

It is difficult to arrange passengers precisely in a designated sequence. One basic solution is to partition the passengers into several groups, such as by row number or column letter.

Our basic boarding strategy is based on group partition. Groups board in a particular order, but passengers in the same group enter in a random sequence.

# Assumptions

- Within a group, all sequences of passengers are equally likely; and permutations within different groups are independent of one another.

- We simplify the behavior of passengers. In reality, a passenger walks at a speed between zero and an upper bound; in our models, a passenger either keeps still or moves at a constant speed, but different passengers walk at their personal constant walking speeds. A static passenger does not require any time to accelerate to usual walking speed.
- The airliner deck is a rectangle. In reality, the shape resembles a rectangle with its corners truncated.
- We consider deplaning to be the converse of boarding except in the cases of random sequencing and free-seat-choice strategy. Thus, we consider only boarding.
- We follow the traditional strategy to assign a constant time for first-class and business-class passengers to board before economy class.
- Some large airliners, such as the Boeing 747 and Airbus A330-300, have two parallel aisles. We divide a two-aisle airline into two equal halves and treat each half as a single plane with only one aisle. (Sometimes the two halves are not exactly symmetric, as with the Boeing 747.)

# Model I: Interferences

We describe passenger behavior with two parameters, the walking speed and the expected time to stow luggage. It seems impossible to obtain an explicit closed formula for expected boarding time, so we use expected boarding interference instead.

We use the definitions of van den Briel et al. [2005]. *Boarding interference* is a passenger blocking another passenger's access to his or her seat. There are two types: *seat interferences* and *aisle interferences*. Seat interferences occur when passengers seated close to the aisle block other passengers to be seated in the same row. Aisle interferences occur when passengers stowing luggage block other passengers' access to seats.

## Model Description

Suppose that the plane has a single aisle. Let the groups be numbered and let groups enter the airliner in sequence 1, 2, . . . .

**Assumption** To avoid seat interference as often as possible, if passenger A of Group $i$ and passenger B of Group $j$ sit on the same half of a row (the same left row or the same right row), with $i < j$, we assume that A sits closer to the window than B.

With this assumption, we can describe a group partition and a sequence of groups by two matrices:

$$(x^j_{i,\mathrm{r}}),\ (x^j_{i,\mathrm{l}}), \qquad j = 1, \dots, n;\ i = 1, \dots, m,$$

where $n$ is the number of groups and $m$ is the number of rows. Group $j$ consists of $x^j_{i,\mathrm{r}}$ passengers from the right half of row $i$th and $x^j_{i,\mathrm{l}}$ passengers from the left half of the row $i$. The sum of all the entries in the two matrices is the number of passengers on the plane

Due to the assumption, seat interferences occur only within the same group, so we have:

$$\text{E[seat interferences]} = \sum_{j=1}^{n} \sum_{i=1}^{m} \left[ \tfrac{1}{2} \binom{x^j_{i,\mathrm{r}}}{2} + \tfrac{1}{2} \binom{x^j_{i,\mathrm{l}}}{2} \right], \tag{1}$$

$$\text{E[interferences within Group } i] = \frac{1}{s_j} \left[ \binom{s_j}{2} + \sum_{i=1}^{m} \binom{x^j_{i,\mathrm{r}} + x^j_{i,\mathrm{l}}}{2} \right], \tag{2}$$

where

$$s + j = \sum_{i=1}^{m} (x^j_{i,\mathrm{r}} + x^j_{i,\mathrm{r}}),$$

and the expected number of aisle interferences between consecutive groups Group $j$ and Group $j+1$ is

$$\frac{1}{s_j s_{j+1}} \sum_{i=1}^{m} \left[ \left( x^j_{i,\mathrm{r}} + x^j_{i,\mathrm{l}} \right) \sum_{t=1}^{m} \left( x^{j+1}_{t,\mathrm{r}} + x^{j+1}_{t,\mathrm{r}} \right) \right]. \tag{3}$$

Then the expected number of aisle interferences is the sum of expected aisle interferences within groups plus the sum of expected aisle interferences between groups.

Equation **(1)** is interpreted as each pair of passengers in the same group and the same row on the same side (both right or both left) have probability 1/2 of seat interference, since exactly one of their two possible boarding orders causes a seat interference.

For aisle interference, for each ordered pair of passengers there are $s_j - 1$ positions for the two passengers to board one after another, leaving $(s_j - 2)!$ ways for the remaining passengers in this group to board. Thus, the probability of such a situation is $(s_j - 1)(s_j - 2)!/s_j! = 1/s_j$, and there are

$$\binom{s_j}{2} + \sum_{i=1}^{m} \binom{x^j_{i,\mathrm{r}} + x^j_{i,\mathrm{l}}}{2}.$$

pairs that can cause interference; this gives **(2)**. In a similar way, we can calculate the expected aisle interferences of two consecutive groups. The only case of two passengers in different groups to cause interference is that they are the last of the previous group and the first of the next one. This happens with probability $1/s_j s_{j+1}$. Calculating all the interfering pairs gives **(3)**.

We define the evaluation of a strategy to be

$$\text{(expected aisle interferences)} + \lambda\text{(expected seat interferences)}. \qquad \textbf{(4)}$$

where $\lambda$ is a positive number determined by the time needed to stow luggage and the constant walking speed of passengers.

# Optimal Strategy in a Weak Sense

Because the number of aisle interferences between each pair of consecutive groups is no more than 1, the total of aisle interferences between different groups is less than the number of groups. When the number of groups is not very large, which is often the case, we neglect aisle interferences between different groups and concentrate on seat interferences and aisle interferences within the same group.

The first term in **(2)** is in fact constant when summed over $j$. The other terms in **(1)** and **(2)** are convex and monotonically increasing functions with respect to $x^j_{i,\mathrm{r}}$ and $x^j_{i,\mathrm{l}}$.

With aisle interferences between different groups neglected, and the number of groups and number of passengers in each group all fixed, the total number of expected interferences is a sum of convex functions. Therefore, the strategy with the fewest interferences must have the property that in the same group the difference of passengers in different half-rows is no more than 1. Moreover, the difference of passengers in different *rows* is also no more than one in the best strategy.

For instance, outside-in and reverse-pyramid strategies have the above properties. This indicates that outside-in and reverse-pyramid strategies might be optimal strategies when the number of groups is not large.

# Results

As in van den Briel et al. [2005], let $\lambda = 1$. For each strategy, we could compute the total expected interferences using **(4)**. We use Lingo 8.0 to search for the optimal strategy with the least expected interferences with the number of groups fixed. The task is to determine the two matrices representing a strategy with the least interferences. We are faced with a nonlinear integer programming problem. The objective function is

$$\text{(expected aisle interferences)} + \text{(expected seat interferences)}.$$

Such a problem is NP-hard [van den Briel et al. 2005].

Due to the limitation of our computers, we could not determine the global optimal solution for a 60-passenger or larger plane, even in several hours.

Nevertheless, we ran Lingo, stopped the software after some tens of minutes of search, and observed the best solution found to that point. In rare cases, the lower bound of the objective function equaled the least interferences, which means that our computer found a global minimum. In many cases, the interferences of the best strategies found by computer is slightly greater than the lower bound of objective function.

**Table 1** gives the results from Lingo for different airliner structures. Triples in the table denote structures of airliners; for instance, "2,3,11" means an airliner with 2 columns of seats on one side of aisle, 3 columns of seats on the other, and 11 rows. The incompleteness of the results of computation set aside, the best strategies found, as anticipated, are consistent with the theoretical results of the previous subsection.

**Table 1.**

Results from LINGO.

| Airliner type | Structure | Number of groups | Best known strategy | Bound by Lingo |
|---|---|---|---|---|
| Airbus A380 | 2,3,15 | 2 | 47.0 | 46.4 |
| | | 3 | 39.6 | 37.5 |
| | | 4 | 40.4 | 33.1 |
| | | 5 | 40.7 | 30.8 |
| | | 6 | 40.3 | 29.2 |
| | 2,3,11 | 2 | 35.0 | 35.0 |
| | | 3 | 29.7 | 28.5 |
| | | 4 | 29.8 | 25.3 |
| | | 5 | 30.1 | 23.5 |
| | 2,3,9 | 2 | 29.1 | 29.1 |
| | | 3 | 24.6 | 24.1 |
| | | 4 | 24.8 | 21.3 |
| | | 5 | 24.9 | 19.4 |
| Airbus A320 | 3,3,26 | 2 | 106.5 | 101.0 |
| | | 3 | 80.5 | 80.5 |
| | | 4 | 81.0 | 71.4 |
| | | 5 | 81.3 | 65.6 |
| | | 6 | 81.8 | 61.7 |
| | | 7 | 81.4 | 58.9 |
| | | 8 | 83.3 | 56.8 |
| Canadair CRJ-200 | 2,2,14 | 2 | 29.5 | 29.5 |
| | | 3 | 29.8 | 25.1 |
| | | 4 | 29.1 | 22.7 |
| | | 5 | 30.1 | 21.5 |
| | | 6 | 30.6 | 20.3 |
| Part of Boeing 747 | 2,2,19 | 2 | 39.5 | 39.5 |
| | | 3 | 39.8 | 33.3 |
| | | 4 | 40.3 | 30.1 |
| | | 5 | 40.8 | 28.2 |
| | | 6 | 45.1 | 27.0 |

Surprisingly, we got three exactly outside-in strategies, with the rest resembling either reverse-pyramid or outside-in.

# Model II: Individual Boarding Sequence

Practical problems set aside, the most efficient strategy comes out of the finest group partition, in which each passenger corresponds to a group and each group consists of exactly one passenger. Passengers are arranged to enter the airliner in an expected order. Using this partition, the best solution is the individual boarding sequence strategy.

In minimizing interferences, there is an obvious lower bound: the back-seat passenger in each column must be blocked when the one just before is stowing luggage. Also, the front-seat passenger while stowing luggage must block the next passenger in sequence. Hence, the interference at least occurs $(n-1)$ times, where $n$ is the number of columns, since every back-seat passenger of a column causes an interference except the first to board. Such a minimum can be attained actually when each back-seat passenger follows a front-seat passenger and there are no other interferences—which is an individual boarding sequence.

Certainly, this strategy is often considered seriously impractical. Van Landeghem and Beuselinck [2002] argue that comparable systems exist today. We devise a system that can be used to make the finest partition: At the airport, there is plenty of time between check-in and when passengers are allowed to enter. Airlines can assign numbers and letters to waiting seats at the airport. Passengers can be seated there according to their seats on the airliner. The airline can call passengers to board in sequence by the numbered seats.

# Stochastic Simulations

To establish a simulation that can test various strategies and treat various kinds of airliners, we wrote a program in C++.

To test boarding strategies, our simulation should reproduce the boarding process as closely as impossible, so that the assumptions about human behavior are tenable and data describing passenger boarding behavior accords with reality. Our simulation is based on discrete time and continuous space; each time step is 0.5 s.

## Assumptions and Details in the Simulation

- Business/economy assumption: First-class and business-class passengers board before economy-class passengers, in an assigned constant time.

- The aisle of economy class is narrow and cannot contain two passengers abreast. Thus, if a passenger is stowing luggage, the passenger behind in the aisle must wait and cannot pass.

- Between consecutive boarding groups, there is no time interval. That is, Group $i$ boards in the wake of Group $(i-1)$.

- In free-seat-choice boarding strategies, passengers prefer seats more toward the window. Thus, the window seat is passengers' favorite and the aisle seat is the most unpopular. However, there is no preference for a particular row.
- There is only one way to reach each seat. Later, we discuss validity of this assumption in multi-aisle airliners.
- In the aisle, passengers can move only toward the back. If a passenger's seat is more toward the front than where the passenger is, the passenger has to go to the back and wait for the aisle to clear at the end of boarding.
- A passenger walks at an individual constant walking speed, but different passengers have different walking speeds. The walking speed distribution is a triangular distribution with lower limit 0.28 m/s, mode 0.365 m/s, and upper limit 0.45 m/s, based on observations by Van Landeghem and Beuselinck [2002]. This distribution is the sum of two continuous uniform distributions.
- The distance between consecutive passengers must not be smaller than 0.6 m. A passenger who walks fast enough to violate this limitation stops where the minimum is attained.
- We exclude several low-probability events: a passenger falls down, passenger mistakes another's seat for the passenger's own, or a seated passenger leave the seat voluntarily (e.g., for the toilet).
- The distance between rows is 33.25 in = 0.84 m [Wikipedia n.d.].

## Bin Occupancy Model

As in Ferrari and Nagel [2005], we suppose that there is an overhead bin for each row on each side of the aisle and every passenger is assigned a random number of pieces of luggage, according to the probabilities in **Table 2**.

**Table 2.**
Luggage distribution at normal and high load.

| | Number of pieces | | | |
|---|---|---|---|---|
| | 0 | 1 | 2 | 3 |
| Normal load | 5% | 55% | 30% | 10% |
| High load | 5% | 20% | 55% | 20% |

The time that a passenger needs to stow luggage depends on how much luggage and the occupancy of the overhead bin, as follows:

$$t_{\text{sl}} = 2.4\left(2 + \frac{n_{\text{bin}} + n_{\text{l}}}{2} \times n_{\text{l}}\right)$$

when $n_{\text{l}}$ is positive, with

- $t_{\text{sl}}$ is the time to stow all pieces of luggage (seconds),
- $n_{\text{bin}}$ is the number of pieces of luggage already in the bin, and
- $n_{\text{l}}$ is the number of pieces of luggage carried by the passenger.

We let $t_{\text{sl}} = 0$ when $n_{\text{l}} = 0$.

Fractional results for $t_{\text{sl}}$ are rounded to the nearest half-integer. The values of $n_{\text{bin}}$ refer to the corresponding half-rows overhead; passengers always put their luggage into the bin corresponding to their half-row. In reality, if the overhead bin gets full, passengers have to move to other rows to find a bin. This fact is not reproduced directly by the simulation; however, note that $t_{\text{sl}}$ becomes rather large for full bins.

## The Seating Model

Our seating model too is inherited from Ferrari and Nagel [2005]. The time that passengers need to sit down depends on the number of interfering passengers (seat interferences) who are already seated. Those interfering passengers passengers have to get out of their row and then sit down again after the new passenger sits. The mathematical form of this is

$$t_s = t_p + 2t_p n_s = t_p(1 + 2n_s)$$

when $n_s$ is positive, where

- $t_s$ is the total time for seating (seconds);
- $t_p$ is time used to get from the seat into the aisle or back (seconds), $tp = 3.6$; and
- $n_s$ is the number of occupied seats in front of the passenger's seat.

We let $t_s = 0$ when $n_s = 0$.

## Additional Assumptions of Free Seat Choice

Modeling free-seat-choice strategies is not easy. We need more assumptions.

- Passengers are supposed to be sagacious. That is to say, they know the best kind of seats that they can obtain under the worst situation. They know instantly how many of such kind of seats are guaranteed to be available to them. They sit at a seat of such kind with possibility of $1/n$, where $n$ is the estimated number of available seats of that kind after then.
- Queued passengers may lose patience and accept a "bad" seat.
- If a passenger arrives at the last row with free seats, the passenger sits there.
- Passengers do not change their walking direction to find seats.
- We do not consider free seat choice in a two-aisle airliner, where two passengers could reach a seat from two aisles at the same time.

## Simulation Results

Using the simulation software that we built in C++, we simulated each boarding strategy 50 times. Average boarding time indicates the performance of a strategy, while the standard deviation reflects its robustness.

The airliner in these simulations has full occupancy. Passengers are considered to carry luggage of normal load as indicated in **Table 2**.

We tested 14 strategies for Canadair CRJ-200, 16 for Airbus A320, and 9 for Airbus A380. These strategies come from back-to-front, outside-in, reverse-pyramid, random sequencing, rotating-zone, free-seat-choice, individual sequencing, and two strategies produced by computer from Model I. Except for individual boarding sequence, all strategies are currently practicable, with several in wide use. The notation "back-to-front 3" means a back-to-front strategy with 3 groups.

### Canadair CRJ-200

We consider the CRJ-200 as a typical small airliner, with a rectangular 14-row deck and two columns on either side of the aisle. We tested 14 boarding strategies, with the results in **Table 3**.

**Table 3.**

Statistics of simulation of the boarding time of strategies toward CRJ-200 (min).

| Strategy | Average | SD |
|---|---|---|
| Individual boarding sequence | 3.7 | 0.3 |
| Reverse-pyramid 3 | 6.5 | 0.7 |
| Strategy 1 from Model I | 6.8 | 0.6 |
| Strategy 2 from Model I | 6.8 | 0.5 |
| Free seat choice | 6.8 | 0.3 |
| Outside-in 2 | 6.8 | 0.6 |
| Outside-in 4 | 7.7 | 0.6 |
| Random | 7.8 | 0.8 |
| Back-to-front 2 | 8.1 | 0.6 |
| Back-to-front 3 | 8.2 | 0.7 |
| Back-to-front 4 | 8.4 | 0.7 |
| Rotating-zone 4 | 8.5 | 0.9 |
| Rotating-zone 3 | 8.6 | 0.8 |
| Rotating-zone 5 | 9.1 | 0.6 |

- The simulation supports the claim from Model II that individual boarding sequence has the shortest boarding time, only 3.7 min; all other strategies need at least 6 min. Moreover, individual boarding sequence's standard deviation is the smallest.

- The simulation results of Strategy 1 and Strategy 2 from Model I are both satisfactory, which substantiate Model I greatly.

- Among all strategies currently in use, reverse-pyramid, outside-in, and free-seat-choice are the soundest, with boarding time approximately 6.5 min. The standard deviation for free-seat-choice is quite small.
- The traditional back-to-front strategy is most disappointing, with average boarding time over 8 min.
- Considering the ease to perform free-seat-choice strategy and random sequencing (do not require any extra effort), they are acceptable choices.

### Airbus A320 (Midsize)

The Airbus A320 is a typical midsize airliner. We consider it to have a rectangular 26-row deck with three columns on either side of aisle. We tested 16 boarding strategies; the two new ones are back-to-front 5 and back-to-front 6.

The results are completely analogous to those for the CRJ-200. Ironically, the more groups, the worse the situation is for the rotating-zone strategy and for back-to-front. Both are even worse than random sequencing.

### Airbus A380 (Large-Size)

We take the Airbus A380 as a typical large airliner, with two decks. We divide its upper deck into two halves; we divide the lower deck into three parts horizontally and then divide each of three into two halves. Thus, the two-deck economy class is divided into eight parts and each part is treated as a single airliner. We assume that the A380 has two entrances in the front. Passengers are not allowed to cross between halves, so there is only one way to reach each seat.

Our strategy is a combination of strategies for each of the eight parts of the A380. Thus, there are many possible combinations, of which we selected nine to test. Individual boarding sequence performs best again, with the least standard deviation. We also find that large airliners are less sensitive to boarding strategies than smaller ones.

# Sensitivity Analysis

In our simulations to this point, we assume that the airliner is full and passengers carry a normal load of luggage. Also, we exclude the possibility that passengers are late to board and neglect passengers with special needs, who are usually board first. Here we analyze the effect of these possibilities.

## More Luggage

We compare normal load with high load of luggage for a full A320 with no late passengers. Individual boarding sequence still performs excellently with great robustness; its increase is below 10%. The remaining strategies all increase by 25% to 30%. These results indicate that sensitivity to luggage load should not be considered an important criterion in choosing boarding strategies.

## Occupancy

We ran simulations with an A320 at 62.5% occupancy, as in Van Landeghem and Beuselinck [2002], with no late passengers and normal load of luggage. We randomly selected 37.5% of seats to be unoccupied. All strategies perform well. In comparison with full occupancy, however, individual boarding sequencing show the least sensitivity. Among the remaining strategies, reverse-pyramid, outside-in, and strategies from Model I have greater sensitivity; and back-to-front and rotating strategies are quite sensitive.

## Late Passengers

A late passenger is one who does not board when the passenger's group is allowed to board but who reaches the gate before boarding ends. We assume that late passengers are not allowed to board before non-late passengers. We randomly choose passengers to be late, with probability 25%, for a full A320 with normal load of luggage. As expected, individual boarding, which requires the most precise sequencing, is the most sensitive to late passengers, requiring 70% more time than without late passengers—but still requiring less time than other strategies, all of which vary by at most 12% (increase or decrease) from their times without late passengers. One can think of increasing late passengers as making a strategy more like random sequencing; in fact, random sequencing has almost the same average boarding time with or without late passengers.

## Passengers with Special Needs

We follow the tradition that passengers with special needs, such as the handicapped and mothers with children, board first. We compare an A320 with 5% special-need passengers with one with no such passengers, with full occupancy and normal load of luggage. At this level of passengers with special needs, average boarding time is not sensitive.

# Strengths and Weaknesses

## Strengths

- Our simulation can deal with different types of airlines, ranging from a 20-passenger jet to a two-deck Airbus-380.

- We give theoretical proof to support the excellence of individual boarding sequence; this proof is not found elsewhere.

## Weaknesses

- Interferences overestimate boarding time.
- In Model I, our computer did not allow us to make a complete computation to obtain the global minimum.
- The data that we use to simulate passenger boarding behavior is specifically intended to model a certain type of airliner. However, passenger behavior, such as walking speed and time to stow luggage, varies in different types of airliner [Marelli et al. 1998].
- In the simulation of free-seat-choice, a passenger is unrealistically expected to foresee blocking ahead.
- We oversimplify the preferences for seats and rows, neglecting the possibility that some passengers love aisle seats and some passengers prefer front rows.
- Considering the need to maintain balance in flight, passengers are not allowed to be seated randomly in a non-full occupancy airliner. However, we neglect this point.

# Conclusions

- With Model I, we translate the original problem into a nonlinear integer programming. Running Lingo, we almost get the optimal strategies with the number of groups fixed. The results from Lingo are outside-in strategies and many of them are only slightly different from reverse pyramid.
- With Model II, we give a proof that individual boarding sequence is the best strategy except for its impracticability.
- We test several kinds of boarding strategies. The results are in accordance with results from Model I and II, in terms of average and standard deviation of boarding time and in terms of robustness. Moreover, random sequencing is acceptable.
- Unfortunately, the traditional back-to-front is worst in many major aspects.
- Sensitivity toward luggage load should not be considered important.
- The more late passengers, the more a best strategy moves toward random sequencing. Individual sequencing is extremely vulnerable to late passengers.

# References

Ferrari, P., and K. Nagel. 2005. Robustness of efficient passenger boarding in airliners. Transportation Research Board Annual Meeting, paper number 05-0405. Washington, DC.

Finney, Paul Burnham. Loading an airliner is rocket science. *New York Times* (14 November 2006) `http://travel2.nytimes.com/2006/11/14/business/14boarding.html` .

Marelli, Scott, Gregory Mattocks, and Remick Merry. 1998. The role of computer simulation in reducing airplane turn time. *Aero Magazine* (4th Quarter 1998). `http://www.boeing.com/commercial/aeromagazine/aero_01/textonly/t01txt.html` .

van den Briel, Menkes H.L., J. René Villalobos, Gary L. Hogg, Tim Lindemann, and Anthony V. Mulé. 2005. America West Airlines develops efficient boarding strategies. *Interfaces* 35 (3) (May-June 2005): 191–201.

Van Landeghem, H. n.d. A simulation study of passenger boarding times in airliners. `http://citeseer.ist.psu.edu/535105.html` .

________, and A. Beuselinck. 2002. Reducing passenger boarding time in airliners: A simulation approach. *European Journal of Operations Research* 142: 294–308.

Wikipedia. n.d. Airline seat. `http://en.wikipedia.org/wiki/Airline_seat` .

Fan Zhou, Guozhen Wang, and Linbo Zhao.

# Best Boarding Uses Buffers

Kevin D. Sobczak
Eric J. Hardin
Bradley J. Kirkwood
Slippery Rock University
Slippery Rock, PA

Advisor: Athula R. Herat

## Summary

By constructing a mathematical model of human behavior, we find:

- **Back-to-front block loading is the least efficient boarding method.** As passengers enter the aircraft in groups, aisle congestion becomes greatest at the front of the plane, consequently increasing the time required for the next group to enter and take their seats. Aisle congestion in this case is primarily attributed to the time for a passenger to navigate the aisle and reach the assigned seat if obstructed by another passenger sitting in the same row.

- **Small planes and large planes exhibit minimal turnaround times.** Small planes have a single aisle but few passengers, hence little congestion. In large planes, multiple aisles and decks offset the congestion found in single-aisle midsize planes; a large plane can be modeled as several small planes.

- **Boarding strategies are optimized when 10% of the passengers are late.** Fewer passengers enter initially, so there is less congestion. When passengers enter late, congestion that would otherwise have occurred is averted.

Our first observation concurs with researchers who suggest abandoning back-to-front boarding in favor of more-elaborate schemes [Finney 2006; van den Briel et al. 2004; Ferrari and Nagel 2005]; however, these new models make erroneous assumptions about human behavior. A comprehensive scheme must include the time to navigate a congested aisle, stow luggage, and maneuver through a filled row if necessary. We recommend the following:

- **Abandon back-to-front block boarding and consider alternatives.** We suggest a hybrid group-boarding method utilizing a rotating seating arrange-

*The UMAP Journal* 28 (3) (2007) 421–434. 

ment that incorporates back-to-front and window-to-aisle seating. This approach decreases congestion that otherwise might accumulate near the front of the airplane, by loading a first group of passengers into rear seating while another group is consecutively loading into the front seats. This trend is carried out until the last group is seated in the center of the aircraft.

- **Incorporate a second aisle into midsize aircraft.** Midsized aircraft tend to display the worst boarding times due to the absence of a second aisle. A second aisle would ease congestion and cut turnaround time nearly in half.
- **Reduce carry-on luggage.** Reducing the amount of carry on luggage greatly decreases aisle congestion greatly decreases.
- **Queue passengers into lines prior to gangway entry.** Increased order greatly reduces aisle congestion.

# Introduction

It is common to board an aircraft in zones or groups. The most-used method is boarding blocks of seats, from the rear moving toward the front. This method is more efficient than boarding the entire plane at once but is one of the least efficient schemes that we tested. Several factors have to be taken into consideration for determining boarding times; passengers entering the plane are assumed to be unsorted within their boarding groups, and not all passengers arrive on time. Our model that incorporates all these factors and provides surprising and consistent results.

Our computer simulation can evaluate various boarding schemes on aircraft of varying sizes. It has factors for late passengers and will search for worst- and best-case boarding times based on randomly-arranged passengers. The most efficient method that we found moves passengers onto the plane from window to aisle, back to front, implementing rotation buffers. This method even allows for some passengers to be late without severely affecting efficiency. We report results for boarding time, tolerance to passenger arrival, and boarding time predictability for several boarding schemes and aircraft.

# Boarding Strategies and Terminology

We define boarding strategies by rows, columns, and groups.

**Block boarding:** Each block is a group and a block consists of a number of rows. Typically, groups of passengers are seated in blocks sequentially from the back to the front.

**Buffering:** Buffering places empty seats between sequentially seating groups, so that congestion and delay of the first group does not interfere with seating the second group. For example, a plan may seat block 1 consisting of rows 30–25 and then sequentially block 2 of rows 22–17. The aisle by rows 23 and 24 will be filled temporarily by busy passengers from block 1 and will not interfere with block-2 passengers. Rows 23 and 24 will be seated later.

**Column Seating:** Column seats by columns instead of by rows. This strategy is typically implemented from the window to the aisle to minimize row congestion.

**Reverse-Pyramid Scheme:** The reverse-pyramid scheme seats passengers in V-shaped groups starting back in the aisle and propagating forward to a window. This method minimizes row congestion while maximizing group size.

# Assumptions

## Logistical Assumptions

We assume that all planes are entered exclusively from the front and that passengers sit one to a seat.

Our model does not account for the time that it takes a person to seat themselves in an empty row or a seat unobstructed by another passenger. This is because the moment that the passenger leaves the aisle, they can no longer add to aisle congestion and the total seating time for the group.

According to the U.S. Department of Transportation [2007], airplanes fly at 79% capacity on average. With this information, we make three assumptions:

- Since passengers are randomly seated, the empty seats are randomly dispersed, and thus there is no need to reseat passengers for balance purposes.
- Boarding times are based on 100% capacity, but group size and buffers can be adjusted to minimize boarding times are based on expected capacities.
- All two-level airplanes are boarded with two-level jetways and board both levels simultaneously.

Large two-aisle planes with possibly two decks have similar configurations to several small airplanes; hence the strategies for small planes will have comparable efficiencies for large planes.

We assume that the time required to seat those with special needs is nearly constant. Although "9.7% . . . of men and women, aged 16–64 report a sensory, physical, mental, or self-care disability in the United States" [Employment and Disability Institute 2007], the percentage among travelers will be lower, due to monetary limits and ability to travel. A midsize plane with capacity 300 running at 79% capacity would carry 237 passengers, of whom less than 9.7%—perhaps

15—need special assistance. We assume that seating strategies for special-needs groups are unnecessary due to their small size.

We also recognize that column boarding is efficient but may separate parties traveling together, costing the airline in terms of customer inconvenience. The goal of column seating is to minimize row congestion. We contend that allowing parties to board the airplane together, even though this may deviate from the seating plan, would not degrade the advantage. Parties flying together will enter a row in order, maintaining minimal row congestion.

We assume that as the overhead compartments fill, stowing luggage becomes increasingly difficult. We also assume that each passenger has the same amount of carry-on luggage requiring the same volume. To this effect, the time for a passenger to stow luggage depends only on the number of people seated in the row and the number of luggage-volume units already taken up.

## Behavioral Assumptions

A fundamental assumption is that passengers are willing to bypass localized aisle congestion, resulting in a time cost. But when aisle congestion becomes large-scale, passengers become averse to bypassing larger numbers of people. This assumption of human behavior is accounted for by disallowing groups of passengers to bypass other groups who are blocking the way to their seats, but allowing passengers in the same group to pass each other in the aisle. Our model of localized passing predicts seating times better than popular models that assume that passengers do not pass others in the aisle.

All constants have been estimated to the best of our ability but would require experimental determination. Due to the nature of our passenger time model, discrepancies between our constants and actual values will have little effect on the relative efficiencies of the strategies that we investigate.

We assume that every passenger walks with a constant speed, since the speed of a line is dictated by the slowest passenger.

We assume that when business- and first-class passengers seat themselves, they have a greater average passenger speed and smaller constants corresponding to bypassing others in the aisle, stowing luggage, and traversing occupied seats. The reasons are that business- and first-class aisles are wider, first-class passengers carry less luggage because their trips tend to be shorter, and individual seating areas in business- and first-class are about 4.5 $m^3$ as opposed to 1.2 $m^3$ in coach [Ferrari and Nagel 2005].

# Methods

We devised a highly dynamic object-oriented model in Java, which can import group assignments from a text file. We collected data from the simulation from several different configurations and aircraft sizes.

## Group Boarding Time Model

We assume that the marginal time increase of a group to be seated with respect to each additional person is:

$$t(p) = C_1\tilde{p} + C_2\alpha + C_3\beta + C_4\gamma,$$

where

- $p$ is the number of people who need to cross in the aisle,
- $\alpha$ is the number of people who need to move to reach the seat from the aisle,
- $\beta$ is the amount of luggage stored in the overhead compartment, and
- $\gamma$ is the number of rows that a passenger must traverse.

We let $w$ and $l$ be the width and length of the aircraft. Conceiving of our model as a continuous model, We have

$$t(\bar{p}) = \frac{dt}{dp} = C_1\bar{p} + C_2\bar{\alpha} + C_3\bar{\beta} + C_4\bar{\gamma},$$

where $\bar{p}$ is linear and $\bar{\alpha}, \bar{\beta}, \bar{\gamma}$ are constant. Integrating both sides, we find that time per group $T(p)$ is

$$T(p) = \frac{C_1}{2}\,\bar{p}^2 + C_2\bar{\alpha}\bar{p} + C_3\bar{\beta}\bar{p} + C_4\bar{\gamma}\bar{p}.$$

Because our model is discrete and $\tilde{p}, \alpha, \beta, \gamma$ can fluctuate randomly depending on the arrangement of passengers within the group, $dt/dp$ cannot be represented continuously in an accurate manner.

However, by taking a Riemann sum, $T(p)$ can be found as

$$T(p) = \sum_{n=0}^{p} (C_1\tilde{p} + C_2\alpha + C_3\beta + C_4\gamma)\Delta p,$$

where $\Delta p = 1$ person.

## Simulation Algorithm

Our simulation models boarding as a queue while mathematically modeling human behavior. The dynamic simulation can compute aircraft boarding times for different grouping configurations with no modification to the code. The model also can loop through several different configurations and can iterate each configuration to acquire an average seating time with error bounds. Because the problem assumes that passengers are not arranged within the group, passengers are randomly shuffled in the groups and each run yields slightly different results.

### Running the Model

Parameters are the number of rows, number of seats per row, location of aisle, number of groups, group configuration file, and number of iterations to average the solution (in most cases, 1,000).

### Late Passengers

We assume that a specified percentage of randomly-selected passengers arrive late.

### Group Boarding Time

The total time to board aircraft is related the time to board each group, which in turn is based on the time to board each individual within that group and on interactions between consecutive boarding groups. There are two cases for which we make provision:

- (the more common case) A *waiting condition* occurs when a group boarding the plane is directly adjacent or farther back in the aircraft than the previous boarded group. In this case, the second group waits until the previous group is seated. The total time for aircraft boarding will have the previous group seating time added minus the time required for the latter group to approach the previous group.
- The other condition occurs when there is a gap between two consecutive boarding groups. This is known as a *buffer condition*, and the time added to the total time is the entering time of the previous group.

The two exceptions are the first group and last group. The first group has no interactions with any another group initially and will not contribute to the total time. The time for the first group will factor in after the second group is determined. The last group interaction is determined using one of the two methods and then the time for the final group is added to the total time.

## Results

After considering the implementation of buffering zones, block boarding methods, and variations regarding column-boarding techniques, we were able to use our model to represent the most efficient boarding schemes. When cabin configurations are kept constant between different sized aircraft, different boarding schemes are better suited for different sizes of aircraft. We show the results for midsize aircraft in **Figure 1**.

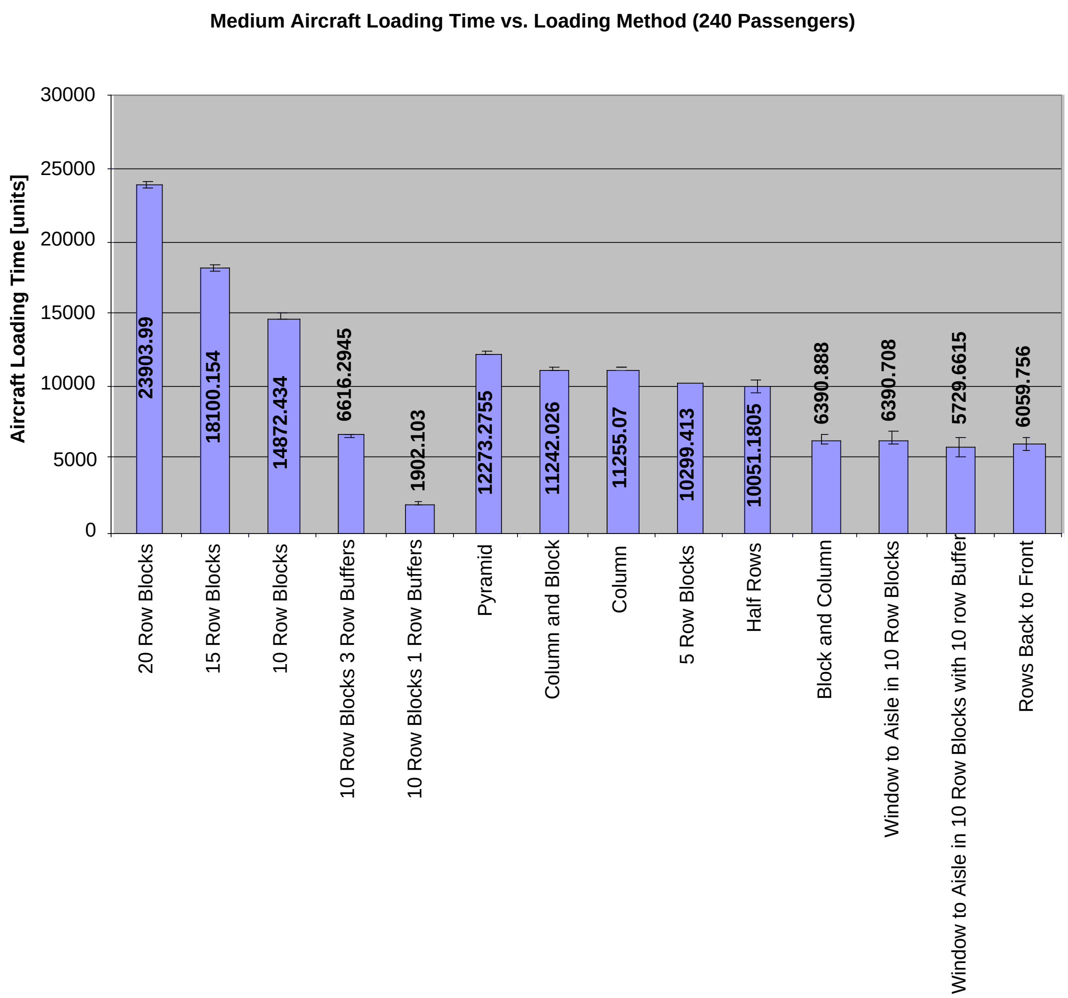



**Figure 1.** Boarding times for various boarding methods, for a 240-passenger plane.

## Back-to-Front Block Boarding

A group of passengers entering the aircraft in four groups takes twice as long as a boarding party of the same size with a five-row buffer zone. As passengers board the airplane, congestion from passengers heading toward the back of the airplane and stowing luggage slows the advance of entering passengers. Due to our assumption that individuals passing in the aisle require 1.5 time steps to maneuver past a passenger, a consequently higher group seating time develops. This prolonged seating period prevents the next group from entering, since members from different groups cannot interfere with one another. With a buffer zone, congestion is reduced: As the group advances towards the rear of the airplane and proceeds to take seats, the second group enters, leaving a five-row buffer zone, allowing the second group to get seated as well without any additional interference from the first group, consequently reducing the overall boarding time.

## Half-Block Boarding

A variation back-to-front is the half-block boarding method, which uses the aisle to divide the passengers into more groups. Because it is more organized, it is nearly twice as efficient as the traditional block boarding system.

## Column Boarding

Column boarding is very inefficient if the number of columns is the maximum number of groups present in the plane. Column-filling with an unbuffered group-boarding scheme yields only marginally better results. Only when columns are broken into smaller groups themselves are the advantages of column-filling evident. The implementation of column-filling completely eliminates aisle congestion caused by seat crossovers. The incorporation of segmented columns into our model decreases boarding time because increasing the number of groups enhances the order of the scheme.

## Reverse-Pyramid

A hybrid system based on a back-to-front rotating column arrangement, seating passengers from the window to aisle seats, is dubbed the reverse-pyramid. A simulation based on this scheme shows only marginal increases in boarding time compared to block boarding from back to front. The fundamental principle behind reverse-pyramid is reducing aisle congestion by permitting groups to enter in staggered column configurations to minimize seat crossings. A significant drawback, however, is that for it to be most effective, a column that is sent in must be further sectioned. Most airlines use a six-group configuration that most resembles a "V" as the airplane begins to fill up. So if a more robust model is desired, further complication is needed in properly dividing the entering segments of the reverse pyramid.

## Buffer Zone Implementation

Schemes that incorporate a buffer are sensitive to most variables. Buffer systems are successful in small and midsize aircraft, because more travelers can be boarded without interfering with other groups; however, the introduction of late arrivals into a buffer system causes the scheme to fail.

### Back-to-Front Boarding

This most-common strategy is ironically the most undesirable. As passengers enter the plane, congestion immediately builds from back to front as passengers in the back must pass others in the aisles, wait to stow luggage, and maneuver past other passengers to access a seat.

### Hybrid Boarding Methods

The most effective boarding scheme is one that encourages the simultaneous boarding of passengers from window to aisle, and from back to front of the aircraft, while incorporating a buffer. Due to the ordered nature of this scheme, the buffer zone is affected very little by late passengers. Our hybrid scheme has virtually no aisle congestion due to seat crossovers and luggage stowage, because passengers file in from the windows toward the aisle and from back to front. Late passengers are considered as an independent group unto their own, hence do not interfere with groups currently boarding.

### Deboarding

Since passengers are already in an ordered system, aircraft deboarding lends itself well to random passenger exiting. This assumption can be based on the fact that there will be no congestion caused by persons crossing other passengers within the same row, because everyone has the same incentive to deboard the airplane. Our results show that a more ordered system boards the fastest; therefore the most ordered system will deboard in the most efficient manner.

### Tardiness

Buffer systems are sensitive to tardiness, since each successive late passenger tends to increase the boarding time. Block boarding methods actually experience an increase in efficiency when late passengers board! This results from an absence of congestion that would otherwise be present; however, block boarding methods are the least efficient methods in general. Any improvement on the block boarding scheme is merely making an inefficient system slightly more efficient, but is still not preferable to other methods available.

## Sensitivity

### Random Passenger Order within Groups

There is a weak inverse power relation between the number of groups and the difference between the maximum and minimum boarding times compared to the average boarding time. This relationship can be realized intuitively: As the number of groups increases, the randomness decreases until the number of groups equals the number of passengers. In that case, there is no randomness; every passenger is sent to their seat, and every boarding simulation will provide the same result. To this end, strategies with more groups have greater predictability.

Taking the number of groups for each strategy into consideration, random entry within groups affected our strategies in the following ways:

- **Block methods** have a minimum number of groups to board; thus, we show that the current method presently used by airlines is unpredictable.
- **Column-seating methods** have minimal group numbers and, when taking group number into account, high predictability. The goal of column-seating is to minimize time cost by minimizing row congestion resulting from passengers randomly entering the rows out of order. In other words, column-seating is designed to be resistant to random entry order.
- **Buffering systems** are by far the most sensitive to random passenger order. This is no surprise, because the benefit of the buffer system is maximized when the overflow from a leading group fills up the exact number of rows to where the following group seating begins. As random amounts of passenger bypassing increases or decreases, congestion due to the leading group increases or decreases as well, resulting in more or less overflow, and thus decreasing the efficiency.
- **Pyramid schemes** show predictability greater than block seating but less than column-seating. This may be due to the column-oriented advantages shared with column-seating, but decreased due to the increase in group size.

## Airplane Configurations

Typically, efficient models are efficient despite plane configuration. Differences in relative efficiencies are noticeable but generally negligible, with some exceptions.

The primary exception is buffer strategies. Buffer-strategy effectiveness depends on congestion overflow into the aisle, which in turn depends on plane configuration. Column-seating methods also are a surprising exception. In one simulation of boarding a small plane of 208 passengers, column-boarding performed the worst of any strategy. It also performed poorly in a midsize plane of 300 passengers, whereas it performed well in an airplane of only 240 passengers. These results are as a direct result of plane configuration; while uncertainties in column-boarding strategies are noticeable, they are not enough to influence this trend based on average times in 1,000 boarding trials.

## Late Passengers

Inevitably, some passengers arrive late. We assume that all late passengers are called to board as a final group to be seated without a strategy. With this in mind, we can analyze the sensitivity of different strategies to the number of late passengers. Late passengers have the following effects (see **Figure 2**):

- **Block methods** become more efficient as passengers arrive late. As block size increases, block methods become maximally efficient at a higher rate of late passengers. The following example will illustrate the inefficiencies of block methods and the sheer magnitude that late passengers have on time efficiency. In midsize aircraft (240 passengers), blocks of 10 to 20 rows have minimal boarding times when late passenger percentages are 21% to 35%. The boarding time of every block method tested decreases as the number of late passengers increased from zero. Not only does the boarding time decrease, but some block configurations became as much as 15% faster at late percentages resulting in maximum efficiencies.
- **Column-seating** becomes slightly more efficient as the rate of late passengers increased but column-seating is resistant to late arrivals.
- **Buffering methods** are the most susceptible to late passengers; they decrease utilization of the buffer rows. Adding rows that are not being seated or used to store overflow decreases efficiency. The efficiency of buffer systems quickly decreases when parameters change.
- **Pyramid schemes** act much like column-seating with respect to late passengers. This is no surprise, because pyramid schemes are a form or column-seating.

## Buffer Size

The buffer system is the most sensitive of all the seating strategies. Counterintuitively, it can be one of the most efficient methods if the buffer size is chosen properly. If the buffer is too small, the following group waits. If the buffer is too big, the rows between the last overflow passenger and the first passenger of the following group are not utilized. Rows not utilized for seating or storing overflow decrease efficiency. Because buffer size depends on the overflow of a leading group due to congestion, and the size of the overflow depends on plane configuration and flight capacity, efficiencies of buffer systems are extraordinarily delicate.

## Vacant Seats

The percentage of vacant seats has little effect on relative strategy efficiencies.

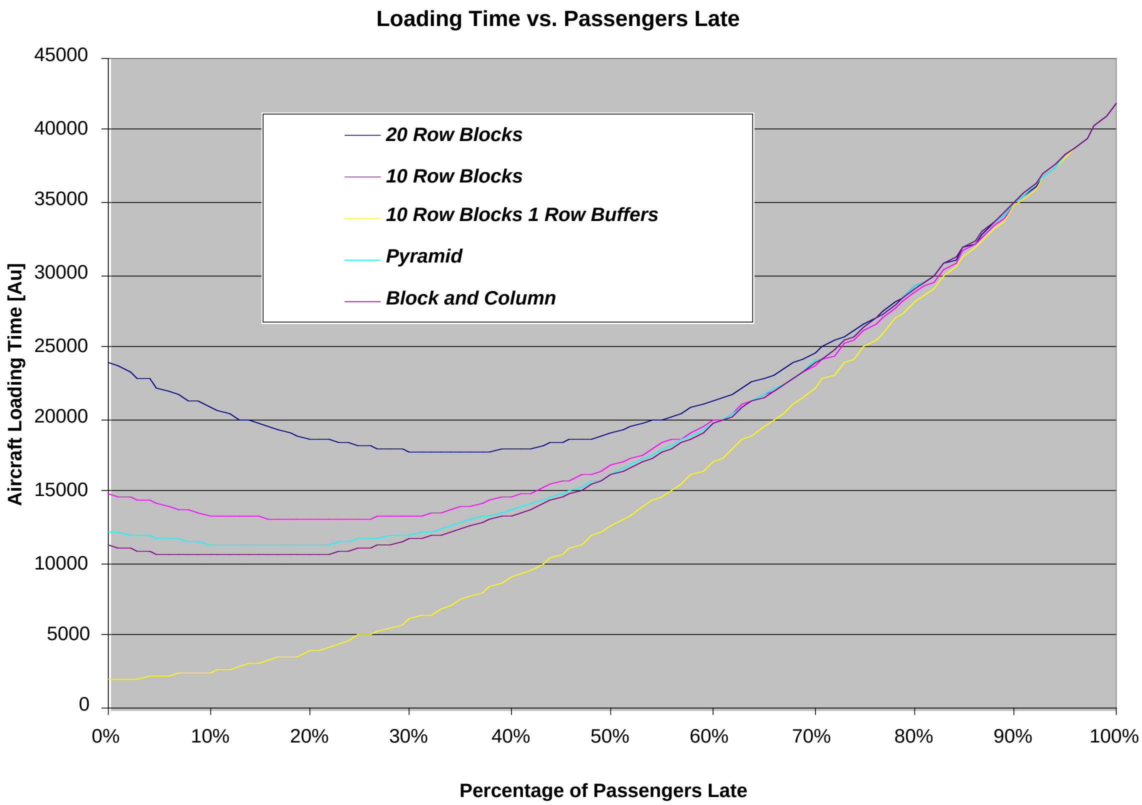



**Figure 2.** Effect of late passengers on boarding time, for various boarding methods.

# Conclusion

## Future Work

We suspect that the aisle congestion if groups of passengers were allowed to pass each other can be modeled using a constant times the number of people in the group divided by the number of row the group spans.

Realistically, passengers may not appreciate column-seating because it separates parties. Further complexity can be added to our model to allow parties to sit together. To this end, we could test if airlines truly must trade customer convenience for the benefits of column-seating. We strongly suspect that allowing parties to board together does not undermine the benefits of column-seating.

Due to the sensitivity but great potential of buffer systems, quickly applicable algorithms for determining optimal buffer zones would be valuable. These algorithms would depend on group size, expected tardiness, expected carrying capacity, and plane configuration.

Additionally, it would be useful to investigate the effects of adding preliminary order to passengers waiting in the terminal. Using a calling system (such as colored signs) to add preliminary order in the terminal could allow for greater order entering the plane without customer confusion and inconvenience.

## Closing Remarks

Not only are presently-employed block strategies inefficient, but they are usually the *most* inefficient strategies. They lack order to minimize row congestion, and they facilitate accumulation of aisle congestion.

The best method for boarding airplanes is to board primarily from window to aisle, secondarily from back to front, and furthermore use a buffer system. We assume that preliminary order can be applied to passenger groups still in the terminal to alleviate the large number of groups required to employ our strategy. Our method is resilient to random entry within groups, thus is predictable compared to other methods. We assume that predictability is heavily valued by airlines and recommend our strategy for that reason as well as maximum time efficiency.

# References

Employment and Disability Institute. 2007. Disability statistics: Online resource for US disabilities statistics. `http://www.ilr.cornell.edu/edi/disabilitystatistics/` .

Ferrari, Pieric, and Kai Nagel. 2005. Robustness of efficient passenger boarding in airplanes. *Transportation Research Record: Journal of the Transportation Research Board* (Issue Number 1915) (July 2004): 44–54. `http://fgvsp01.vsp.tu-berlin.de/biblio/53/01/15nov04.pdf` .

Finney, Paul Burnham. 2006. Loading a plane is rocket science. *New York Times* (14 November 2006). `http://travel2.nytimes.com/2006/11/14/business/14boarding.html` .

U.S. Department of Transportation. 2007. Bureau of Labor Statistics. `http://www.bts.gov` .

van den Briel, Menkes H.L., J. René Villalobos, and Gary L. Hogg. 2004. The aircraft boarding problem. In *Proceedings of the 12th Industrial Engineering Research Conference (IERC-2003)*, No. 2153, CD-ROM. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf` .

Eric Hardin, Kevin Sobczak, and Bradley Kirkwood.

# Modeling Airplane Boarding Procedures

Bach Ha
Daniel Matheny
Spencer Tipping
Truman State University
Kirksville, MO

Advisor: Steven J. Smith

## Summary

We describe two models that simulate the process of passengers boarding an an aircraft and taking their seats. Using these models, we simulate common boarding procedures on popular aircraft to analyze efficiency. The second model is more ambitious and tries to model the situation more accurately, but even the first one addresses the major problems involved in boarding an airplane.

From running the simulations and analyzing the data, we find that the fastest and most consistent procedures are outside-in and reverse-pyramid. Both allow those closest to the windows to be seated first and proceed inward (though reverse-pyramid is slightly more complex). Reverse-pyramid is slightly faster.

## Introduction

It would seem that the quickest way load passengers onto a plane would be simply to line all the passengers up in order of seat assignment, starting with the back-row window seats and working up to the front-row aisle seats, and march them onto the plane in that order. However, this is far from "simple"; the logistics would be extremely difficult to manage, not to mention that customers would dislike being forcibly lined up.

The response of airlines has been to try to control the randomness in passenger boarding order by seating passengers in groups, thereby localizing the

*The UMAP Journal* 28 (3) (2007) 435–450. 

disorder to a particular part of the plane. The traditional approach is back-to-front boarding, where a certain number of rows are seated, starting from the back and working forward. Other procedures include:

- **Outside-in:** Also called WilMA (for Window, Middle, Aisle), passengers with window seats are seated first, followed by those with middle seats, and finally those with aisle seats.
- **Rotating-zone:** Similar to back-to-front, except that after a set of rows in the back of the plane are seated, a set of rows in the front are seated. Back rows and front rows are alternated until the plane is full.
- **Reverse-pyramid:** Reverse pyramid resembles a mix of outside-in and back-to-front, giving preference to seats as far back and to the outside as possible. First seated would be the back half of window seats, then the back middle seats and the front window seats, then the back aisle seats and the front middle seats, then finally the front aisle seats.
- **Random:** Some airlines purposefully do not try to control the order of passengers on the plane. Random seating can be done with or without some seats assigned. Often the plane is still boarded in stages, with the passengers lumped into groups by check-in time or by another method.

Much research has been done to determine what procedure is fastest. While some studies have been analytic in nature [Bachmat et al. 2006a, 2006b; Van den Briel et al. 2003], most have adopted the approach of simulating the phenomenon [Ferrari and Nagel 2004; Bazargan et al. 2006; Van den Briel n.d.]. One problem with available simulations is that they focus on at most one plane size and type. In particular, in many models, a small plane with one aisle and three seats on each side of the aisle is taken to be representative of all aircraft. In reality, most planes that carry more than a small number of passengers have two aisles, and some have two floors as well.

We describe a model that aims to address these problems. We simulate aircraft boarding for any size plane (number of passengers), any layout (number of aisles, number of seats in each row), and most importantly, any order for seating passengers. We use the model to estimate the relative efficiency of various boarding procedures.

# Motivating the Model

The process is slowed when boarding passengers have to interact, events we call *interferences*. *Aisle interference* occurs when a passenger cannot continue down the aisle to the seat because the aisle is blocked. *Seat interference* occurs when a passenger can only reach the seat by going past already occupied seats. Seat interferences increase the time it takes to sit down and may also lead to prolonged aisle interference if the people currently sitting down must get into the aisle to let a person in.

# Assumptions

- **The plane is full.** While this is not always true, if a plane is far below capacity, any boarding method will probably work well.

- **All passengers are in the economy class and there are no special-needs passengers.** First-class passengers pay a premium price and expect to be seated first. We model only the seating of economy passengers. Likewise, we do not take into account pre-boarding by passengers with special needs such as the disabled, the elderly, and those traveling with small children.

- **There are no late passengers.** All passengers in a particular boarding group get in line to board as soon as they are called.

- **All pieces of carry-on luggage are the same and there is always enough room for them.** Passengers enter the plane with a randomly assigned number of bags (within a reasonable range such as 0–3). If there are already passengers seated in a particular row, it may take longer for an arriving passenger to stow bags, but the passenger can eventually do so.

- **No one can pass a passenger in the aisle.** This is the principal cause of aisle interference. In reality, a passenger might be able to squeeze past another, but we do not allow this. This is a reasonable assumption because in general passing a person in the narrow aisle of an airplane is a difficult and slow task anyway.

- **There is only one entrance.** Some airlines allow boarding from two entrances, but the majority of airlines have only a single entrance. Moreover, planes that allow multiple entrances tend to be small, where the boarding is already not as difficult as in large planes.

- **All seats are assigned.** This assumption primarily affects the random boarding procedure, for which we assume that the order of passengers entering the plane is completely random but each passenger has a unique seat that they are headed for when they enter. In unassigned random seating, there is the added problem that the passengers likely do not have a specific seat in mind when they get on the plane. When they enter, they head to either a seat that they consider "desirable" or to a seat that is easy to get to. We do not model this choice process.

- **Every passenger sits in their assigned seat.** Any seat-switching happens after take-off and so does not affect the boarding time.

- **Deboarding is always the same.** This is probably the most significant assumption that we make. We assume that the passenger unloading process will be the same no matter what the boarding process was, or the reverse of what the boarding process was, an assumption that seems to match the way airlines currently operate.

## Estimating Parameters

There are few data for the speed at which passengers board planes, the time that it takes to stow luggage, and so on (though Bazargan et al. [2006] have some estimates). Thus, we had to estimate these quantities ourselves; consequently, the boarding times from in our model probably do not correspond directly to actual times. Our model strives to compare boarding procedures, which can be done by taking a standard set of "reasonable" values. We later discuss what happens to the model when parameter values are varied.

# Modeling Airplane Boarding Procedures

We built two separate simulation models, which we will refer to as the Array Model (or AM) and the Graph Model (or GM), both implemented in Python. In general, the AM is more simplistic, while the GM tries to simulate the situation more accurately.

In the Array Model, the plane is represented internally by a two-dimensional array. Some columns of the array are aisles, and the rest are seats. The seats are either occupied are unoccupied, as are aisle cells. In this model, a person in the aisle completely blocks all the people behind. The AM can simulate different boarding procedures as well as different plane sizes and types, but with only one type of plane geometry at a time.

The Graph Model represents the plane internally by a graph, where the nodes and edges are weighted to represent the delay associated with crossing that node or traversing that edge. While more complicated, the GM is more flexible, allowing passengers to pass one another in the aisle (subject to a corresponding time delay, of course), and allowing for different plane geometries in different sections of the plane. This gives the possibility of modeling first-class seating as well, where the seating structure is different from economy seating. Also, in larger planes there is a second level of seating, which might have a different configuration from the primary level.

In the following two sections, we give the details of the two models and the relevant parameters in use. Then we give results for the tests that we did using the AM and the GM.

# The Array Model

The array model is built based on the motivation of the Game of Life, devised by mathematician John Horton Conway. We treat the layout of seats and aisles as a matrix of four different values: occupied and unoccupied seats, occupied and unoccupied locations on aisles. The only objects that interact with this matrix are passengers. By moving up and down, right and left, inside the matrix, passenger change cell values in the matrix. Of course, passengers cannot move freely; they must follow certain rules that depend on the current layout of the matrix.

## Parameters

The AM is based on several parameters. To make the model more accurate to the real world situation, we assign most parameters a normal probability distribution so that they vary slightly from run to run.

- **Interval boarding time**: This is the time that the airline staff checks a passenger's boarding pass. Default mean 4 s, standard deviation 1 s.
- **Time sensitivity**: This is the interval of time for which our model will update itself. The default value is 0.25 s.
- **Luggage stowing time**: This value depends strongly on the number of passengers there already (which also means that their luggage is already there). The more passengers, the longer it takes for a new passenger to stow luggage. By default, mean time and its standard deviation are 4 s and 1 s for an empty row, 8 s and 2 s for a row with one passenger, and 14 s 3 s for a row with more than one passenger. (In all the airplanes that we examined, no seat has more than two seats between it and an aisle, so we need to consider at most two passengers already seated.)
- **Seating time**: Similar to the luggage stowing time parameter, this value depends greatly on the number of passengers already there. By default, we set the mean time and its standard deviation to 3 s and 1 s for an empty row, 7 s and 2 s for a row with one passenger, and 17 s and 3 s for a row with more than one passengers.

The behavior of the model as it handles seat and aisle interference can be seen best by examining a screen shot of the AM simulating a plane using the back-to-front procedure (**Figure 1**).

# The Graph Model

The graph model builds the airplane seating from a graph of nodes, each connected bidirectionally to applicable adjacent nodes in one of the four cardinal directions. Each node contains an occupant. Aisles have connections in all directions and seats have connections horizontally.

Passengers are tracked as they pass through the plane. Each is randomly assigned with uniform probability zero, one, or two carry-on bags that must be stowed before the passenger is seated. It is assumed that the passenger takes the aisle closest to the assigned seat in every case, crossing no more seats than is absolutely necessary.


**Figure 1.** Screenshot for AM model.

Storage bins are considered shared among several rows, usually two or three. The time required to load one additional bag into an overhead bin is proportional to the square root of the number of bags currently there. Thus, time to load bags is on the order of the 3/2 power of the number of bags.

Because of the structural flexibility present in the model, it can emulate planes with inconsistent geometries, such as the Boeing 767-400 with 2-2-2 in the front and 2-3-2 in the back. We can also, although with more difficulty, implement planes with two floors, such as the Airbus 380.

The graph model can use a smaller sample size because of the recomputation of random data. Every time a node's delay is computed, it is re-randomized; thus, a single run incorporates a broadly normalized set of random data. For this reason, we consider 200 runs per configuration to be sufficient to represent accurately the performance of a configuration.

# Parameters

- **Aisle-aisle movement delay:** How long it takes for a person to move one node through an aisle. Default: 2 s.

- **Aisle-seat movement delay:** How long it takes to move from an aisle to a seat. Default: 3 s.

- **Seat-aisle movement delay:** How long it takes to move from a seat to an aisle. Default: 3.5 s.

- **Seat-seat movement delay:** How long it takes to move from one seat to another. Default: 7 s.

## Strengths

One strength includes accurate simulation of shared luggage bins: A passenger loading a bag into a bin two rows ahead may influence the loading time of a piece of luggage elsewhere. In addition, luggage bins are shared for both sides of an aisle, which accurately models people's tendency to put luggage on either side of the aisle.

Another is that aisle spaces are allocated for people moving across already taken seats, simulating the requirement of everyone clearing the occupied seats for the newcomer to move in. This accentuates the effectiveness of modifications to the strategies, such as the even-odd variation or the staggered variation.

A third strength is that if there is an aisle, an empty seat, a filled seat, and the target seat, in that order, and a passenger moves into the empty seat on the way to the target seat, the passenger can get to the target seat only when the aisle is clear, according to the rationale that all swapping must be done through an aisle.

## Weaknesses

One weakness of the model is that it does not simulate people traveling very far to get to an empty luggage bin. Rather, luggage bins are assumed to have unlimited capacity, and people do not prefer those with smaller delays.

Further, in our model, when a passenger enter the aisle to make room for someone who needs access to a inner seat, the passenger does not move toward the front of the plane; if the cell toward the back cannot be taken, then there is extra delay. This is somewhat inaccurate.

# Using the Model

## Plane Configurations

To analyze how the model reacts to different types of planes, we developed several plane configurations based on actual popular plane configurations.

**Small Planes:**

- **S1:** 3–3 (three seats, an aisle, and three more seats), 23 rows, 138 seats. (Based on the Airbus 320)
- **S2:** 2–3–2, 25 rows, 175 seats. (Based on the Boeing 767-200)

**Midsize Planes:**

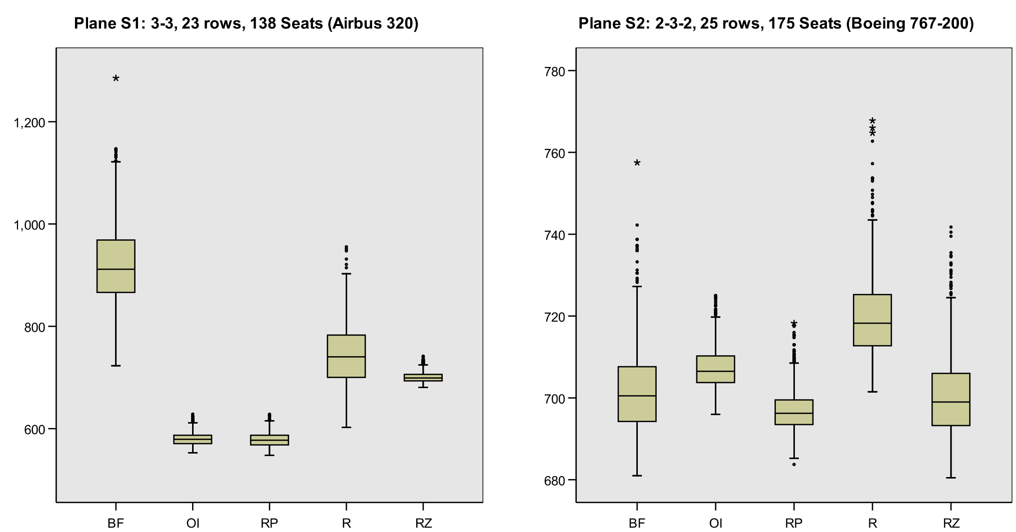
- **M1:** 2–3–2, 35 rows, 245 seats. (Based on the Boeing 767-400)
- **M2:** 2–4–2, 40 rows, 320 seats. (Based on the Airbus A300-600)

**Large Planes:**

- **L1:** 3–4–3, 40 rows, 400 seats. (Based on the Boeing 747)
- **L2:** 3–4–3, 40 rows on bottom level. 2–4–2, 30 rows on top level. 640 seats total.
  (Based on the Airbus 380)

## Array Model Tests

For each plane configuration, we ran the simulation for the following boarding styles: back-to-front (BF), outside-in (OI), reverse-pyramid (RP), random (R), and rotating-zone (RZ). Data were gathered for 1,000 runs of each procedure. Shown in **Figure 2** are a boxplot and the mean and standard deviation for two of each plane size and each boarding procedure.




**Plane S1:**

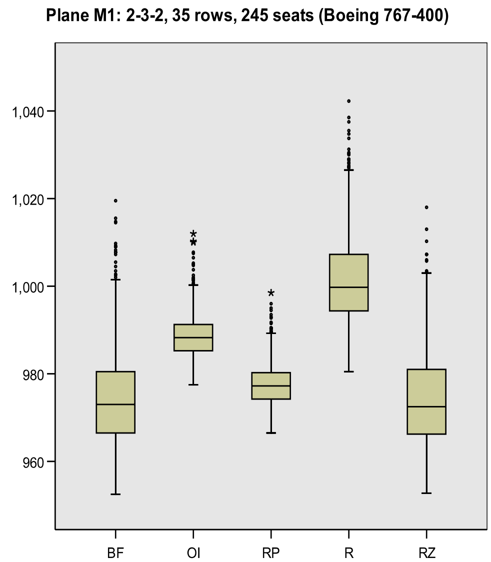
| Procedure | Mean | SD |
|---|---|---|
| BF | 919 | 77 |
| OI | 580 | 12 |
| RP | 578 | 14 |
| R | 743 | 58 |
| RZ | 701 | 10 |

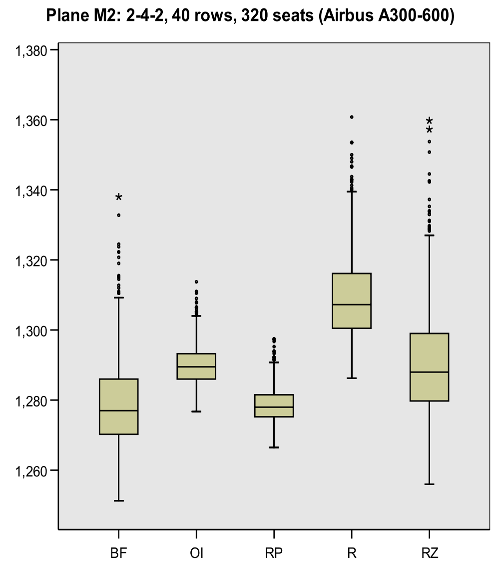
**Plane S2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 702 | 10 |
| OI | 707 | 5 |
| RP | 697 | 5 |
| R | 720 | 10 |
| RZ | 701 | 10 |

**Figure 2.** Results based on the array model: Small planes.

Except for Plane S1, where it performed horribly, back-to-front boarding actually performed quite well. This suggests that the conventional wisdom of the airline carriers is well-founded. Random boarding did not perform particularly well on any configuration, contrary to the opinions of airlines that are beginning to adopt it. The only procedures that consistently performed the



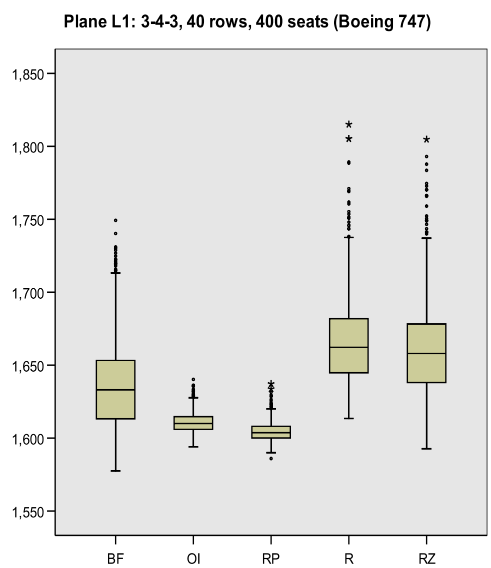



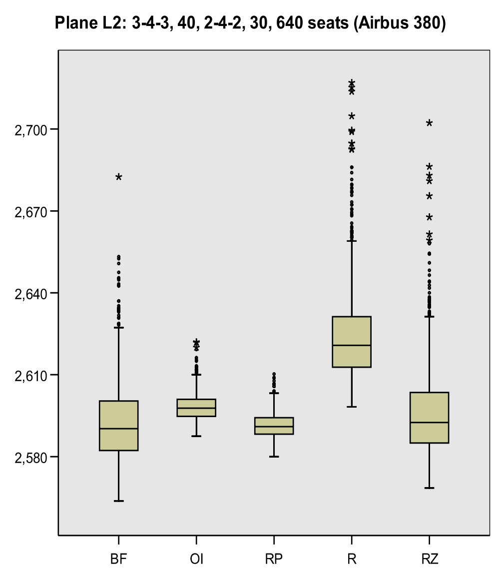

**Plane M1:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 974 | 10 |
| OI | 989 | 5 |
| RP | 978 | 5 |
| R | 1001 | 10 |
| RZ | 974 | 10 |

**Plane M2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 1279 | 12 |
| OI | 1290 | 5 |
| RP | 1279 | 5 |
| R | 1309 | 12 |
| RZ | 1290 | 16 |




**Plane L1:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 1636 | 30 |
| OI | 1611 | 7 |
| RP | 1605 | 6 |
| R | 1666 | 29 |
| RZ | 1661 | 33 |

**Plane L2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 2593 | 15 |
| OI | 2598 | 5 |
| RP | 2592 | 5 |
| R | 2624 | 17 |
| RZ | 2596 | 17 |

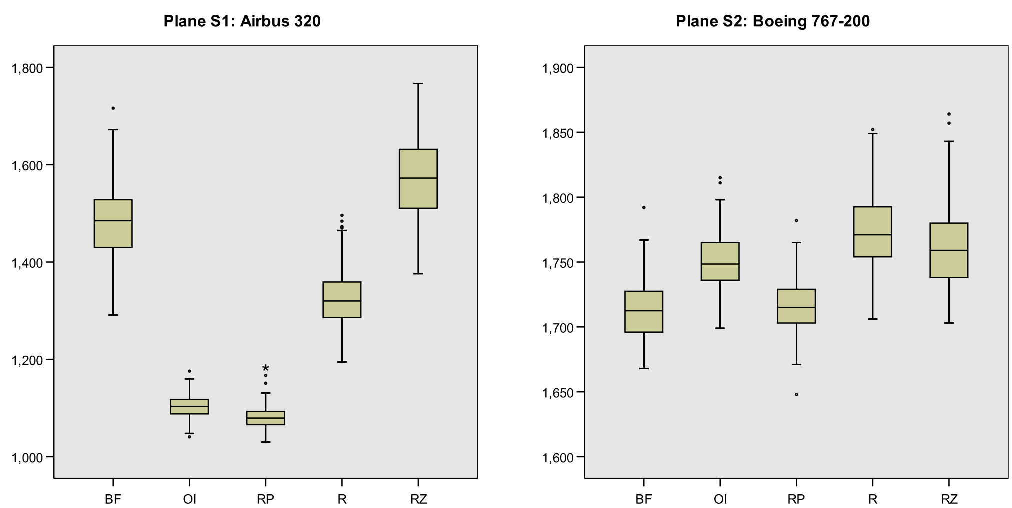
**Figure 2 (continued).** Results based on the array model: Midsize and large planes.

fastest were outside-in and reverse-pyramid, with reverse-pyramid having a slight advantage.

However, the most interesting aspect of the data is not the mean times, but rather the standard deviations. Outside-in and reverse-pyramid consistently had standard deviations almost half that of the other procedures. This is very important, because besides wanting to make boarding as fast as possible, airlines need to keep on schedule, so they do not just need the fastest but the most consistently fast. With this in mind, outside-in and reverse-pyramid seem clear winners according to the AM.

# Graph Model Tests

As with the AM, we ran the five boarding strategies on each of the six plane configurations (**Figure 3**). Since the GM includes more randomization internally, we ran each simulation only 200 times, which still gives a 95% confidence interval about the mean with a radius of less than five time units, which is enough precision for our purposes.




**Plane S1:**

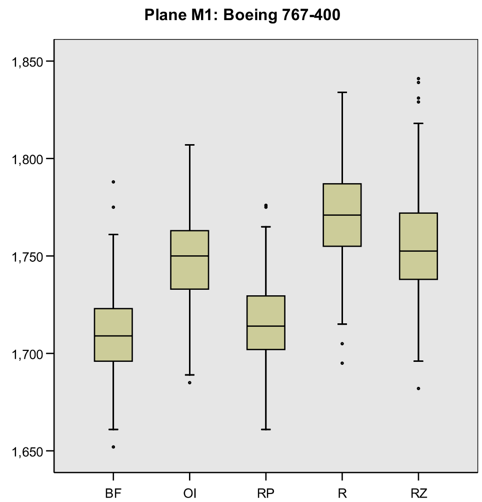
| Procedure | Mean | SD |
|---|---|---|
| BF | 1479 | 74 |
| OI | 1103 | 22 |
| RP | 1081 | 22 |
| R | 1326 | 58 |
| RZ | 1575 | 80 |

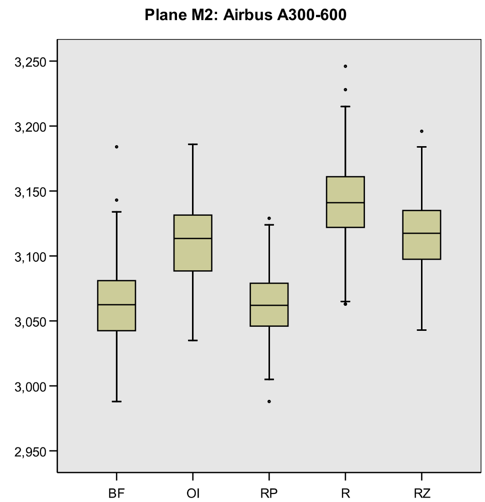
**Plane S2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 1712 | 21 |
| OI | 1749 | 21 |
| RP | 1715 | 20 |
| R | 1773 | 29 |
| RZ | 1760 | 28 |

**Figure 3.** Results based on the graph model: Small planes.

The data generated by the Graph Model are not easily interpretable. As in the Array Model, back-to-front does quite well; outside-in does not do quite as well as before. However, reverse-pyramid is still the best boarding strategy.



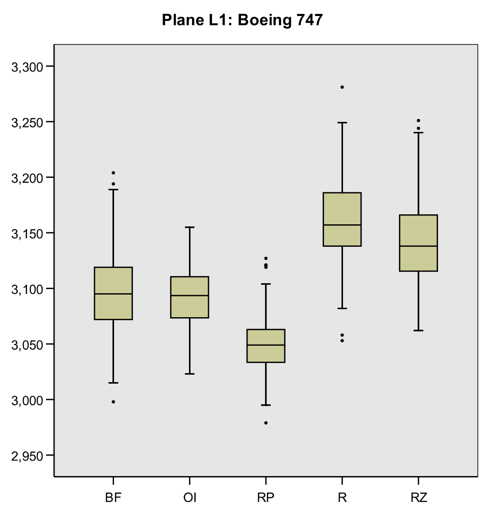



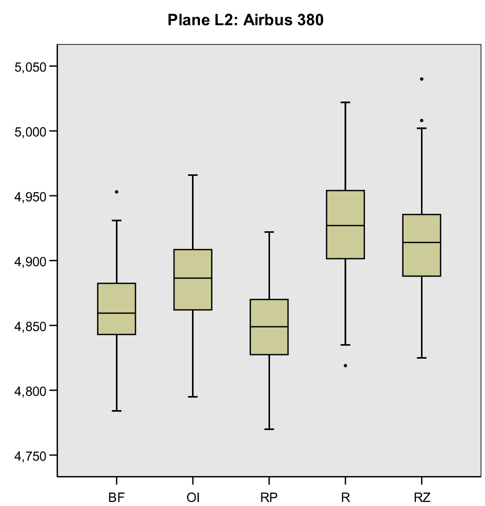

**Plane M1:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 1710 | 21 |
| OI | 1749 | 22 |
| RP | 1716 | 19 |
| R | 1771 | 25 |
| RZ | 1756 | 27 |

**Plane M2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 3063 | 31 |
| OI | 3112 | 29 |
| RP | 3062 | 24 |
| R | 3142 | 33 |
| RZ | 3117 | 27 |




**Plane L1:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 3098 | 38 |
| OI | 3093 | 27 |
| RP | 3050 | 25 |
| R | 3161 | 37 |
| RZ | 3141 | 3 |

**Plane L2:**

| Procedure | Mean | SD |
|---|---|---|
| BF | 4862 | 30 |
| OI | 4885 | 31 |
| RP | 4849 | 31 |
| R | 4927 | 39 |
| RZ | 4913 | 37 |

**Figure 3 (continued).** Results based on the graph model: Midsize and large planes.

according to this model. The only plane for which it performed worse than another model was the M1 plane, where it was beaten out by back-to-front by less than 7 time units. The standard deviations for reverse-pyramid are in general less than those of the other strategies, though by not nearly as much as in the Array Model.

One very perplexing aspect of the Graph Model data is the actual numerical values returned. Even more so than the Array Model, the Graph Model has not been tuned to actual time, so the time units in the results cannot really be taken as seconds. Still, planes S1, S2, and M1 all have simulation values in the range of 1,000 to 1,800 despite different plane sizes, yet the values for M2 and L1 jump dramatically up to the range of 3,000 to 3,300. We had expected a more gradual growth as plane size increased.

## New Boarding Strategies

One benefit of the flexibility of the Graph Model is that we can devise our own boarding strategies and see how they compare with common ones. We tried taking an existing strategy and modifying it so that for a given group, the passengers in even-numbered rows board, followed by those in odd rows. We also considered the "staggered" modification, which is similar to the even-odd modification, except that even-numbered rows are boarded on one side of an aisle and odd-numbered on the other side. These modifications are to an existing strategy, so the original groupings still remain. For instance, in OI.EO (outside-in, modified by even-odd), the even-numbered window seats are boarded, then the odd numbered window seats, then the even middle seats, and so on.

The performance of these alternative strategies generally shaves a few time units off of the simulation time, but the relative results are the same. Shown in **Figure 4** is a graph comparing the 15 boarding procedures on the M2 plane.

# Evaluating the Model

## Array Model Sensitivity

To test the sensitivity of the Array Model, we ran the simulation 100 times each on the M2 configuration for a value higher and a value lower than the default on each of interval boarding time, luggage stowage time, and seating time. The changes alter the boarding time, but we were primarily concerned with whether the qualitative assessment of the different procedures changes as well.

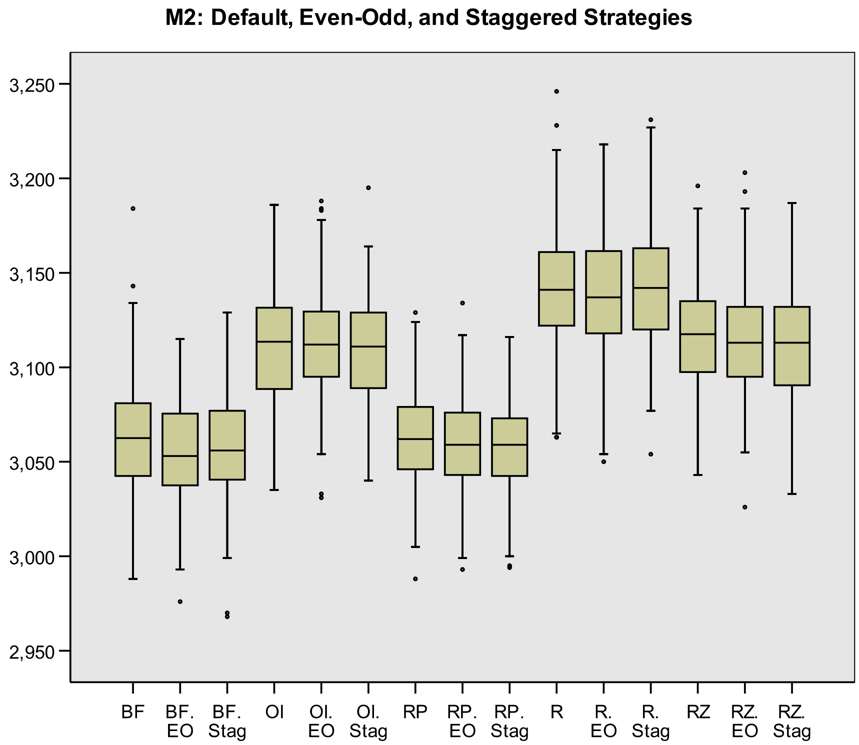



**Figure 4.** Comparison of modified and original strategies.

## Boarding Time

We ran the simulation with a parameter values of (2,0.5) (that is, a mean of 2 and a standard deviation of 0.5), and a value of (7,3), and compared them to the default value of (4,1). The relationship between the outside-in, reverse-pyramid, and random procedures remains about the same; back-to-front and rotating-zone become comparatively faster as the boarding time increases (meaning that passengers are entering the plane at a slower rate). However, we must keep in mind that in the example where the mean boarding time is 7 (one passenger boards every 7 s), the total time to load the plane is over 2,000 s (more than 30 min). The difference in means is less than 1 min, which is not significant. What this means is that as the boarding time increases, the gains of one procedure over another become relatively less.

## Stowage Time

For luggage stowage time, we ran the model with parameter values (2,0.5), (5, 1), (10,2)—meaning (2,0.5) for an empty row, (5,1) for a row with one person already seated, and (10,2) for a row in which two people are already seated—and the values (8,2), (14, 3), (20, 4), in addition to the default values (4,1), (8,2), (14,3).

The model is not very sensitive to changes in luggage stowage time. The times returned by the simulation rose slightly as stowage time increased, but not by much, especially compared to how the simulation times changed with boarding time.

### Seating Time

The default seating time is (3,1), (7,2), (17,3) and the others parameter values run were (2,0.5), (4,1), (8,2) and (6,2), (12,3), (25,5). As with stowage time, the Array Model is not very sensitive to changes in seating time.

## Graph Model Sensitivity

Sensitivity testing for the Graph Model involved choosing six parameters and running the simulation on each plane for each parameter low, high, and normal. More specifically, all but one parameter were normal, and that parameter would be set either low or high. Low was defined to be 50% of normal and high was 175%. The parameters that we chose were:

- Seat-to-seat movement delay per node
- Aisle-to-seat movement delay per node
- Seat-to-aisle movement delay per node
- Aisle-to-aisle movement delay per node
- Luggage bin loading delay per bag
- Delay between successive passengers boarding the aircraft

We ran 25 tests for each configuration, a configuration being a setting of the six variables, a plane, and a loading system. Since the analysis for the Array Model showed that there is little variation, we tested only the original version of each boarding system. Although we found some outliers in the sensitivity analysis, we believe that the results are adequately representative to draw our conclusions, due to the degree of randomness internal to the model.

The primary source of variation in the Graph Model is the delay between passengers boarding the aircraft. Total boarding time is almost directly proportional to the time between individual passengers, suggesting that the main bottlenecks occur outside, rather than inside, the plane. When we decreased boarding time to have a mean of 3.5 s instead of 7 s, the average boarding time was reduced by nearly half. This is the same situation as in the Array Model, where the largest variation was found by modifying the passenger boarding rate.

Other sensitivities were far less noticeable. The aisle-to-aisle transfer time has the next-largest impact on the result, but the variation of the boarding times is well within 15% for a 50% change in the variable. After aisle-to-aisle transfer time comes luggage loading time, whose variation is closer to 10%. Others quickly drop off to below 8%.

# Strengths and Weaknesses

## Array Model

The strength of the Array Model is mostly in its conceptual simplicity. It represents a fairly simple view of an aircraft and its passengers. It is also easy to modify to accommodate different plane configurations and boarding strategies.

The main weakness is likewise its conceptual simplicity. There is a lot more that can be done to model the aircraft boarding process more accurately. For instance, instead of having the parameters be decided according to a normal distribution at the beginning of each run, a more accurate version might have each individual have their own randomly chosen parameter values. Also, a more accurate model might be able to get rid of some of the assumptions that we made, such as allowing passengers to pass in the aisle in certain situations, allowing for late passengers, and modeling first and business class passengers.

## Graph Model

The purpose of the Graph Model was to address the major weaknesses of the Array Model. The Graph Model is even more flexible in allowing different plan geometries, handles passenger interactions more intelligently, and incorporates randomness at the individual level. However, it is not without its problems. It still treats some aspect of passenger behavior and luggage storage in a naive fashion. More importantly, it is even less tuned to actual times than the Array Model, so it would take a large amount of effort to use the model to generate precise time estimates for the boarding procedures.

# Conclusions

Despite these problems, we still feel that both our models capture the essence of the plane boarding process. From the Array Model data, we can make a fairly confident conclusion that the best boarding strategies are reverse-pyramid and outside-in, due to their fast times and low amount of variation. However, from results on the Graph Model, we had to slightly revise our conclusions regarding the outside-in strategy, which did not perform particularly well. The reverse-pyramid strategy still performed best in the Graph Model, so it remains our primary recommendation.

Reverse-pyramid is a bit complicated to implement, so outside-in might still be a good strategy. We must also remember that the traditional back-to-front boarding performed well too, so it might not be worthwhile for an airline to switch away from it. Further, a small speed increase can be gained by implementing an even-odd or staggered variation. For an airline trying to squeeze every last bit of efficiency out of their boarding procedures, a variation of the reverse-pyramid is the best bet.

# References

Bachmat, E., D. Berend, L. Sapir, S. Skiena and N. Stolyarov. 2006a. Analysis of airplane boarding times. Working paper. `http://www.public.asu.edu/~dbvan1/papers/orsubmit.pdf` .

________ . 2006b. Optimal boarding policies for thin passengers. Working paper. `http://www.public.asu.edu/~dbvan1/papers/thin.pdf` .

Bazargan, Massoud, Juan Ruiz, and Victor Cole. 2006. Aircraft boarding strategies: Simulation study. AGIFORS Airline Operations 2006 Conference. `www.agifors.org/document.go?documentId=1586\&action=download` .

Demerjian, Dave. 2006. Airlines try smarter boarding. *Wired News* (9 May 2006) `http://www.wired.com/news/technology/0,70689-0.html?tw=wn_index_1` .

Ferrari, P., and K. Nagel. 2004. Robustness of efficient passenger boarding in airplanes. `http://www.vsp.tu-berlin.de/publications/airplane_boarding/15nov04.pdf` .

Finney, Paul B. 2006. Loading an airliner is rocket science. *New York Times* (14 Nov 2006). `http://travel2.nytimes.com/2006/11/14/business/14boarding.html` .

van dan Briel, Menkes H.L. n.d. Airplane boarding. http://www.public.asu.edu/ dbvan1/projects/boarding/boarding.htm . Accessed 9 February 2007.

________ , J. René Villalobos, and Gary L. Hogg. 2003. The aircraft boarding problem. In *Proceedings of the 12th Industrial Engineering Research Conference (IERC-2003)*, No. 2153, CD-ROM. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf` . Accessed 9 February 2007.

# American Airlines' Next Top Model

Sara J. Beck
Spencer D. K'Burg
Alex B. Twist
University of Puget Sound
Tacoma, WA

Advisor: Michael Z. Spivey

## Summary

We design a simulation that replicates the behavior of passengers boarding airplanes of different sizes according to procedures currently implemented, as well as a plan not currently in use. Variables in our model are deterministic or stochastic and include walking time, stowage time, and seating time. Boarding delays are measured as the sum of these variables. We physically model and observe common interactions to accurately reflect boarding time.

We run 500 simulations for various combinations of airplane sizes and boarding plans. We analyze the sensitivity of each boarding algorithm, as well as the passenger movement algorithm, for a wide range of plane sizes and configurations. We use the simulation results to compare the effectiveness of the boarding plans. We find that for all plane sizes, the novel boarding plan Roller Coaster is the most efficient. The Roller Coaster algorithm essentially modifies the outside-in boarding method. The passengers line up before they board the plane and then board the plane by letter group. This allows most interferences to be avoided. It loads a small plane 67% faster than the next best option, a midsize plane 37% faster than the next best option, and a large plane 35% faster than the next best option.

## Introduction

The objectives in our study are:

- To board (and deboard) various sizes of plane as quickly as possible.

*The UMAP Journal* 28 (3) (2007) 451–461. 

- To find a boarding plan that is both efficient (fast) and simple for the passengers.

With this in mind:

- We investigate the time for a passenger to stow their luggage and clear the aisle.
- We investigate the time for a passenger to clear the aisle when another passenger is seated between them and their seat.
- We review the current boarding techniques used by airlines.
- We study the floor layout of planes of three different sizes to compare any difference between the efficiency of a given boarding plan as plane size increases and layouts vary.
- We construct a simulator that mimics typical passenger behavior during the boarding processes under different techniques.
- We realize that there is not very much time savings possible in deboarding while maintaining customer satisfaction.
- We calculate the time elapsed for a given plane to load under a given boarding plan by tracking and penalizing the different types of interferences that occur during the simulations.
- As an alternative to the boarding techniques currently employed, we suggest an alternative plan and assess it using our simulator.
- We make recommendations regarding the algorithms that proved most efficient for small, midsize, and large planes.

# Interferences and Delays for Boarding

There are two basic causes for interference—someone blocking a passenger in an aisle and someone blocking a passenger in a row. *Aisle interference* is caused when the passenger ahead of you has stopped moving and is preventing you from continuing down the aisle towards the row with your seat. *Row interference* is caused when you have reached the correct row but already-seated passengers between the aisle and your seat are preventing you from immediately taking your seat. A major cause of aisle interference is a passenger experiencing row interference.

We conducted experiments, using lined-up rows of chairs to simulate rows in an airplane and a team member with outstretched arms to act as an overhead compartment, to estimate parameters for the delays cause by these actions. The times that we found through our experimentation are given in **Table 1**.

**Table 1.**
Delays caused by common boarding activities.

| Boarding activity | Time (s) |
|---|---|
| Walking 1 row of seats | 1 |
| Carry-on stowage | 6 |
| Clearing aisle when you must get by someone seated in the aisle seat | 4 |
| Clearing aisle when you must get by people seated in the aisle seat and adjacent seat | 4 |
| When person seated on the aisle must get up | 6 |
| When person seated in middle seat must get up | 6 |
| When two people must get up | 7 |
| When no one is in the aisle and you can squeeze by the middle person | 1 |

We use these times in our simulation to model the speed at which a plane can be boarded. We model separately the delays caused by aisle interference and row interference. Both are simulated using a mixed distribution defined as follows:

$$Y = \min\{2, X\},$$

where $X$ is a normally distributed random variable whose mean and standard deviation are fixed in our experiments. We opt for the distribution being partially normal with a minimum of 2 after reasoning that other alternative and common distributions (such as the exponential) are too prone to throw a small value, which is unrealistic. We find that the average row interference time is approximately 4 s with a standard deviation of 2 s, while the average aisle interference time is approximately 7 s with a standard deviation of 4 s. These values are slightly adjusted based on our team's cumulative experience on airplanes.

# Typical Plane Configurations

Essential to our model are industry standards regarding common layouts of passenger aircraft of varied sizes. We use an Airbus 320 plane to model a small plane (85–210 passengers) and the Boeing 747 for a midsize plane (210–330 passengers). Because of the lack of large planes available on the market, we modify the Boeing 747 by eliminating the first-class section and extending the coach section to fill the entire plane. This puts the Boeing 747 close to its maximum capacity. This modified Boeing 747 has 55 rows, all with the same dimensions as the coach section in the standard Boeing 747. Airbus is in the process of designing planes that can hold up to 800 passengers. The Airbus A380 is a double-decker with occupancy of 555 people in three different classes; but we exclude double-decker models from our simulation because it is the larger, bottom deck that is the limiting factor, not the smaller upper deck.

# Current Boarding Techniques

We examine the following industry boarding procedures:

- random-order
- outside-in
- back-to-front (for several group sizes)

Additionally, we explore this innovative technique not currently used by airlines:

- "Roller Coaster" boarding: Passengers are put in order before they board the plane in a style much like those used by theme parks in filling roller coasters. Passengers are ordered from back of the plane to front, and they board in seat-letter groups. This is a modified outside-in technique, the difference being that passengers in the same group are ordered before boarding. **Figure 1** shows how this ordering could take place. By doing this, most interferences are avoided.

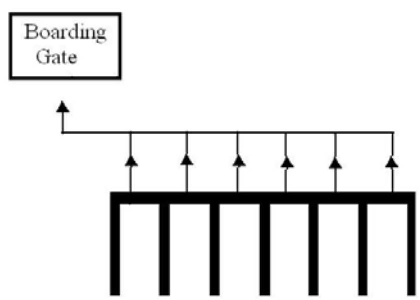



**Figure 1.** Roller Coaster boarding before passengers reach the boarding gate.

# Current Deboarding Techniques

Planes are currently deboarded in an aisle-to-window and front-to-back order. This deboarding method comes out of the passengers' desire to be off the plane as quickly as possible. Any modification of this technique could lead to customer dissatisfaction, since passengers may be forced to wait while others seated behind them on the plane are deboarding.

# Boarding Simulation

We search for the optimal boarding technique by designing a simulation that models the boarding process and running the simulation under different plane configurations and sizes along with different boarding algorithms. We then compare which algorithms yielded the most efficient boarding process.

## Assumptions

The environment within a plane during the boarding process is far too unpredictable to be modeled accurately. To make our model more tractable, we make the following simplifying assumptions:

- **There is no first-class or special-needs seating.** Because the standard industry practice is to board these passengers first, and because they generally make up a small portion of the overall plane capacity, any changes in the overall boarding technique will not apply to these passengers.
- **All passengers board when their boarding group is called.** No passengers arrive late or try to board the plane early.
- **Passengers do not pass each other in the aisles;** the aisles are too narrow.
- **There are no gaps between boarding groups.** Airline staff call a new boarding group before the previous boarding group has finished boarding the plane.
- **Passengers do not travel in groups.** Often, airlines allow passengers boarding with groups, especially with younger children, to board in a manner convenient for them rather than in accordance with the boarding plan. These events are too unpredictable to model precisely.
- **The plane is full.** A full plane would typically cause the most passenger interferences, allowing us to view the worst-case scenario in our model.
- **Every row contains the same number of seats.** In reality, the number of seats in a row varies due to engineering reasons or to accommodate luxury-class passengers.

## Implementation

We formulate the boarding process as follows:

- The layout of a plane is represented by a matrix, with the rows representing rows of seats, and each column describing whether a row is next to the window, aisle, etc. The specific dimensions vary with each plane type. Integer parameters track which columns are aisles.
- The line of passengers waiting to board is represented by an ordered array of integers that shrinks appropriately as they board the plane.

- The boarding technique is modeled in a matrix identical in size to the matrix representing the layout of the plane. This matrix is full of positive integers, one for each passenger, assigned to a specific submatrix, representing each passenger's boarding group location. Within each of these submatrices, seating is assigned randomly to represent the random order in which passengers line up when their boarding groups are called.

- Interferences are counted in every location where they occur within the matrix representing the plane layout. These interferences are then cast into our probability distribution defined above, which gives a measurement of time delay.

- Passengers wait for interferences around them before moving closer to their assigned seats; if an interference is found, the passenger will wait until the time delay has finished counting down to 0.

- The simulation ends when all delays caused by interferences have counted down to 0 and all passengers have taken their assigned seats.

# Strengths and Weaknesses of the Model

## Strengths

- It is robust for all plane configurations and sizes. The boarding algorithms that we design can be implemented on a wide variety of planes with minimal effort. Furthermore, the model yields reasonable results as we adjust the parameters of the plane; for example, larger planes require more time to board, while planes with more aisles can load more quickly than similarly-sized planes with fewer aisles.

- It allows for reasonable amounts of variance in passenger behavior. While with more thorough experimentation a superior stochastic distribution describing the delays associated with interferences could be found, our simulation can be readily altered to incorporate such advances.

- It is simple. We made an effort to minimize the complexity of our simulation, allowing us to run more simulations during a greater time period and minimizing the risk of exceptions and errors occurring.

- It is fairly realistic. Watching the model execute, we can observe passengers boarding the plane, bumping into each other, taking time to load their baggage, and waiting around as passengers in front of them move out of the way. Its ability to incorporate such complex behavior and reduce it are key to completing our objective.

## Weaknesses

- It does not account for passengers other than economy-class passengers.
- It cannot simulate structural differences in the boarding gates which could possibly speed up the boarding process. For instance, some airlines in Europe board planes from two different entrances at once.
- It cannot account for people being late to the boarding gate.
- It does not account for passenger preferences or satisfaction.

# Results and Data Analysis

For each plane layout and boarding algorithm, we ran 500 boarding simulations, calculating mean time and standard deviation. The latter is important because the reliability of plane loading is important for scheduling flights.

We simulated the back-to-front method for several possible group sizes. Because of the difference in the number of rows in the planes, not all group size possibilities could be implemented on all planes.

## Small Plane

For the small plane, **Figure 2** shows that all boarding techniques except for the Roller Coaster slowed the boarding process compared to the random boarding process. As more and more structure is added to the boarding process, while passenger seat assignments continue to be random within each of the boarding groups, passenger interference backs up more and more. When passengers board randomly, gaps are created between passengers as some move to the back while others seat themselves immediately upon entering the plane, preventing any more from stepping off of the gate and onto the plane. These gaps prevent passengers who board early and must travel to the back of the plane from causing interference with many passengers behind them. However, when we implement the Roller Coaster algorithm, seat interference is eliminated, with the only passenger causing aisle interference being the very last one to board from each group.

Interestingly, the small plane's boarding times for all algorithms are greater than their respective boarding time for the midsize plane! This is because the number of seats per row per aisle is greater in the small plane than in the midsize plane.

## Midsize Plane

The results experienced from the simulations of the mid-sized plane are shown in **Figure 3** and are comparable to those experienced by the small plane.

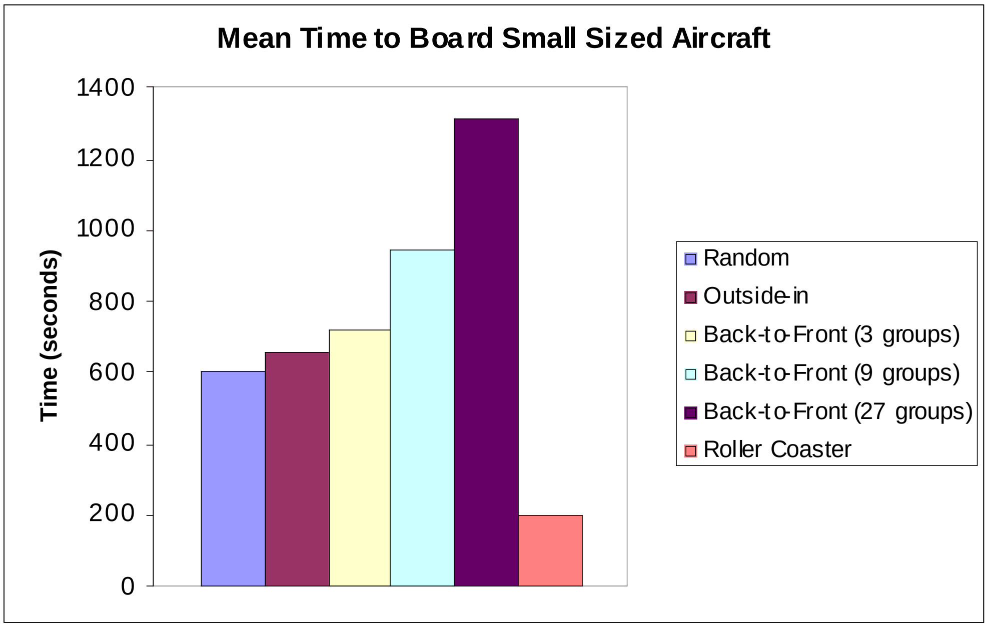



**Figure 2.** Results of boarding strategies on small aircraft.

Again, the Roller Coaster method proved the most effective.

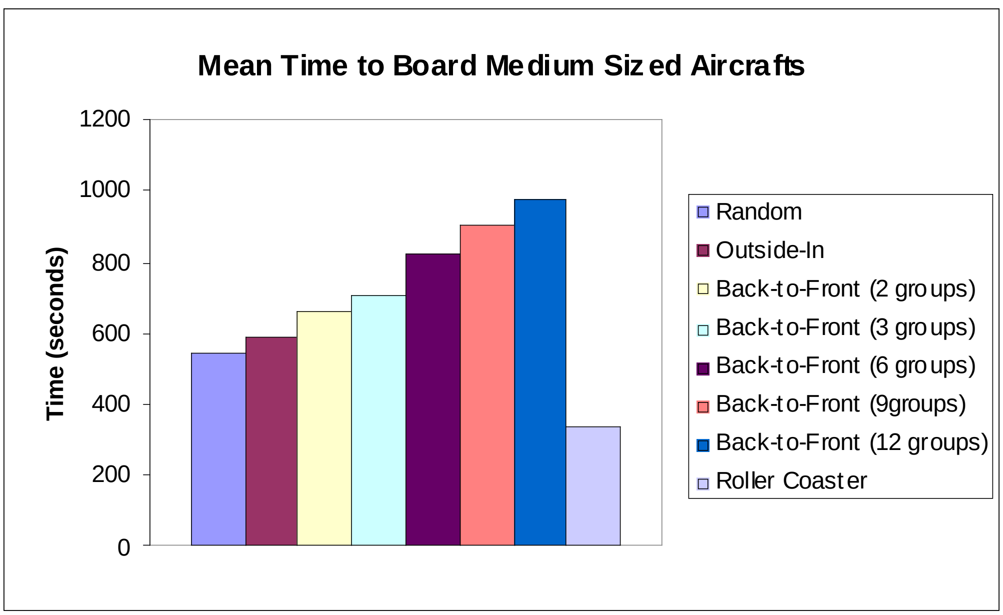



**Figure 3.** Results of boarding strategies on midsize aircraft.

## Large Plane

**Figure 4** shows that the boarding time for a large aircraft, unlike the other plane configurations, drops off when moving from the random boarding algorithm to the outside-in boarding algorithm. Observing the movements by the passengers in the simulation, it is clear that because of the greater number of passengers in this plane, gaps are more likely to form between passengers in the aisles, allowing passengers to move unimpeded by those already on board. However, both instances of back-to-front boarding created too much structure to allow these gaps to form again. Again, because of the elimination of row interference it provides for, Roller Coaster proved to be the most effective boarding method.

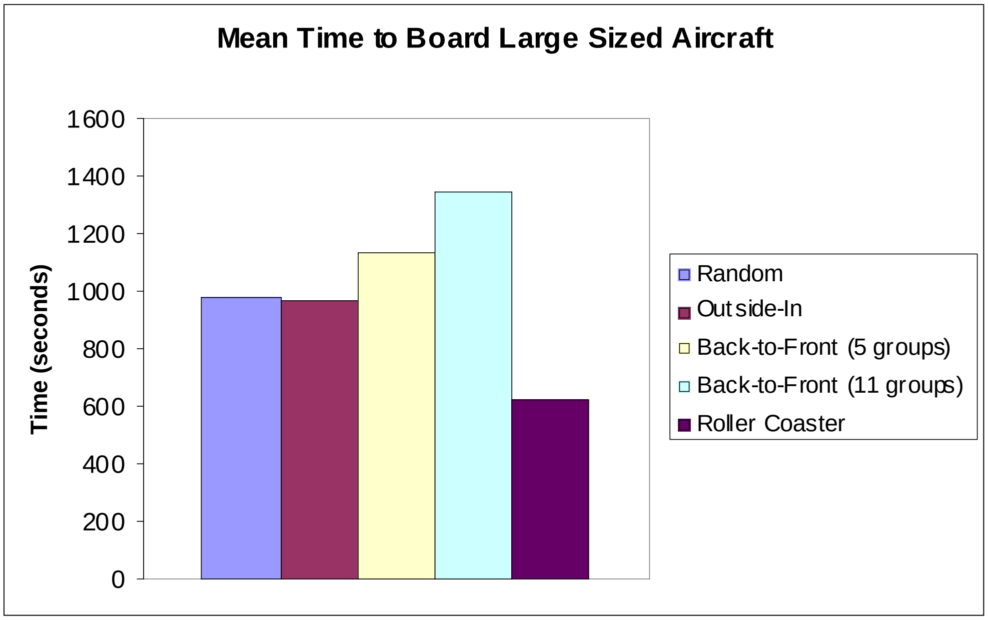



**Figure 4.** Results of boarding strategies on large aircraft.

# Overall

*The Roller Coaster boarding algorithm is the fastest algorithm for any plane size.* Compared to the next fastest boarding procedure, it is 35% faster for a large plane, 37% faster for a midsize plane, and 67% faster for a small plane. The Roller Coaster boarding procedure also has the added benefit of very low standard deviation, thus allowing airlines a more reliable boarding time. The boarding time for the back-to-front algorithms increases with the number of boarding groups and is always slower than a random boarding procedure.

The idea behind a back-to-front boarding algorithm is that interference at the front of the plane is avoided until passengers in the back sections are already on the plane. A flaw in this procedure is that having everyone line up in the plane can cause a bottleneck that actually increases the loading time. The outside-in ("Wilma," or window, middle, aisle) algorithm performs better than the random boarding procedure only for the large plane. The benefit of the random procedure is that it evenly distributes interferences throughout the plane, so that they are less likely to impact very many passengers.

# Validation and Sensitivity Analysis

We developed a test plane configuration with the sole purpose of implementing our boarding algorithms on planes of all sizes, varying from 24 to 600 passengers with both one or two aisles.

We also examined capacities as low as 70%; the trends that we see at full capacity are reflected at these lower capacities. The back-to-front and outside-in algorithms do start to perform better; but this increase in performance is relatively small, and the Roller Coaster algorithm still substantially outperforms them. Under all circumstances, the algorithms we test are robust. That is, they assign passenger to seats in accordance with the intention of the boarding plans used by airlines and move passengers in a realistic manner.

# Recommendations

We recommend that the Roller Coaster boarding plan be implemented for planes of all sizes and configurations for boarding non-luxury-class and non-special-needs passengers. As planes increase in size, its margin of success in comparison to the next best method decreases; but we are confident that the Roller Coaster method will prove robust. We recommend boarding groups that are traveling together before boarding the rest of the plane, as such groups would cause interferences that slow the boarding. Ideally, such groups would be ordered before boarding.

# Future Work

It is inevitable that some passengers will arrive late and not board the plane at their scheduled time. Additionally, we believe that the amount of carry-on baggage permitted would have a larger effect on the boarding time than the specific boarding plan implemented—modeling this would prove insightful. We also recommend modifying the simulation to reflect groups of people traveling (and boarding) together; this is especially important to the Roller Coaster boarding procedure, and why we recommend boarding groups before boarding the rest of the plane.

# References

Airbus S.A.S. 2007. Aircraft families / A320 family. `http://www.airbus.com/en/aircraftfamilies/a320/`.

Boeing. 1995. 747 family. `http://www.boeing.com/commercial/747family/index.html`.

van dan Briel, Menkes H.L., J. René Villalobos, and Gary L. Hogg. 2003. The aircraft boarding problem. *Proceedings of the 12th Industrial Engineering Research Conference (IERC)*, article number 2153. `http://www.public.asu.edu/~dbvan1/papers/IERC2003MvandenBriel.pdf`.

Wikipedia. 2007. Boeing 747. `http://en.wikipedia.org/wiki/Boeing_747`.

Alex Twist, Spencer K'Burg, Sara Beck, and advisor Mike Spivey.

Pp. 463–478 can be found on the *Tools for Teaching* 2007 CD-ROM.

# Boarding—Step by Step: A Cellular Automaton Approach to Optimising Aircraft Boarding Time

Chris Rohwer
Andreas Hafver
Louise Viljoen
University of Stellenbosch
Stellenbosch, South Africa

Advisor: Jan H. van Vuuren

## Summary

We model the boarding time for the aircraft using a cellular automaton. We investigate possible solutions and present recommendations about effective implementation.

The cellular automaton model is implemented in three stages:

- Initialisation of the seating layout for a chosen aircraft type and assignment of seats to passengers
- The sorting of passengers according to various proposed boarding methods
- "Propagating" the passengers through the aisle(s) of the aircraft and seating them at their assigned places.

The rules governing the automaton take into account various factors. Among these are the load factor (percentage filled) of the craft, different walking speeds of passengers walking through the aisle, and time delays from stowing luggage and obstructions by other passengers during the seating process. The algorithm accommodates predefined aircraft layouts of common aircraft and also user-defined aircraft layouts.

We modeled and tested various boarding strategies for efficiency with regard to total boarding time and average boarding time per passenger. Thus, our approach focuses not only on optimisation of the process in favour of

*The UMAP Journal* 28 (3) (2007) 463–478. 

the airlines, but also yields information regarding convenience to passengers. Random boarding (where passengers with assigned seat numbers enter the plane in a random sequence) was used as a point of reference. Among other strategies tested were boarding the plane in groups from either end, boarding from seats farthest from the aisles toward the aisles, and combinations of these approaches.

We conclude that boarding strategies starting farthest away from the entrance or farthest away from the aisles yield shorter boarding times than random boarding. The most successful methods are combinations of these strategies, their detailed implementation depending on the exact layout/size of the aircraft. The method yielding the shortest total boarding time is not necessarily the one with shortest average boarding time per passenger. By considering standard deviations of total and individual boarding times over many iterations of the simulation, we can derive conclusions regarding the stability/consistency of the specific boarding strategies and how evenly the waiting time is distributed amongst the passengers.

By selecting appropriate strategies, time savings of 2–3 min for small and medium aircraft could be achieved. For a custom 800-seat aircraft with two aisles, more than 6 min could be saved compared to random boarding. Having compared our results to actual turnaround times quoted by airlines, we believe them to be realistic.

# Automata Theory and Its Relevance

A cellular automaton is an algorithm that determines the time development of a given system. If the algorithm is fed an initial configuration of the system, a finite set of fixed rules determines how the system develops. A time-step structure is used, such that the algorithm advances incrementally with all its rules being implemented at every time-step.

We used this approach to model various boarding strategies. We create a set of rules to govern how passengers move in the aisle(s) of a plane and what happens when they take their seats. Then we tested various strategies for boarding by changing the order in which passengers entered the plane. Ultimately, we made a comparison of relative boarding times for different strategies (averaged over many iterations), to select the most time-effective strategy. The algorithm was implemented in Matlab.

# The Algorithm

The simulation consists of three main parts:

- an input vector of the passengers,
- a set of rules describing the behaviour of passengers in the plane, and

- a seating plan of the plane (flexible for various sizes/layouts of planes) represented as a matrix.

The arrangement of the input vector determines in which sequence passengers enter the plane. For instance, if the strategy is to board window-seat passengers first, the input vector would be sorted / arranged so that these passengers are at the "front" end of this vector. The vector (which is essentially a lookup-table) also contains the following information for each passenger:

- passenger number (to track elements moving through the matrices);
- seat number;
- walking speed of the passenger (dependent on whether the passenger is a healthy adult, a child, or a passenger with a disability);
- class of the passenger (first class, economy class etc.); and
- the passenger's individual boarding time (determined when passenger is seated).

The rules governing the behaviour (or rather propagation) of the passengers in the plane takes into consideration the passengers' walking speed. We assumed that in the space of one seat-row, two consecutive passengers can stand in the aisle. Thus, the aisle of the plane (also modelled as a vector) was created such that it has two elements for every seat row it bypasses. According to the rules, a passenger could move ahead in the aisle only if the element in front is unoccupied.

# Assumptions

- The aircraft has a single entrance. Most airports have facilities for only one boarding entrance to each plane.
- Not all passengers walk at the same speed. We created three categories of passengers who move through the plane at different speeds. The notion of speed is difficult to implement in cellular automata, due to the finite time-step nature of the algorithm. Thus, we used probabilities. Rules were constructed in such a way that a healthy adult definitely advances one matrix- (or rather aisle-) element per time step. Since children would move slightly slower, they only advance with a probability of 0.7. Lastly, disabled, frail or handicapped people would move the slowest, and were thus forced to advance with a probability of 0.3. In this way, an idea of speed is introduced, where slow passengers hold up the faster ones in the aisle. It is also assumed that passengers do not pass one another in the aisle.

- The distribution of the three categories of passengers is: 2% disabled, frail, or handicapped; 10% children; and the remaining 88% healthy adults. These assumptions are based on semi-educated guesses, since very few data on this matter are available.

- When a passenger gets to the row of the allocated seat, the passenger must stow hand luggage, blocking the aisle for 5 time steps.

- If a passenger reaches the row of the allocated seat, a similar time-penalty is introduced, depending on how many seated people the passenger has to pass over to reach the seat. This time allows for passengers to move out of their seats and into the aisle to permit the given passenger to pass. During this time, obstruction occurs in the aisle, leading to a time delay. This time delay was implemented using a quadratic method: A fixed time delay was multiplied by the square of the number of seated passengers in the way. We considered this to be a realistic model, since several people moving into the aisle would cause a larger time delay for other passengers trying to pass them.

- The time units quoted in the results section are arbitrary and represent individual steps of the cellular automaton. Nonetheless, the time delays are scaled so that their magnitude, in terms of movement of passengers in the aisle, is reasonable. The scale was calculated as follows:

  - A healthy adult passenger advances one element in the aisle vector during each time step if not obstructed. This would be approximately 0.5 m.
  - The average walking speed in an aircraft of a healthy adult is taken to be 0.75 m/s.
  - Thus, one algorithmic time-step would is roughly 0.67 s.

  Based on these assumptions, the delays were calculated as described above.

- Most planes have more space per person in first class and business class than in economy class. Thus, we implement large time delays for luggage stowing in economy class, smaller ones in business class, and the smallest in first class.

- We assume that passengers move in only one direction in the aisle during boarding, since they all have allocated seat numbers and can (we hope) read.

The model is later expanded to accommodate larger planes with two aisles, where similar assumptions are made.

# Step-by-Step Explanation of the Algorithm

First, a seating-plan is loaded, in the form of a matrix, in which the elements represent the seats in the plane numbered sequentially. Our code was constructed such that a fixed, predetermined seating plan could be loaded (for specified aeroplane layouts) or that a seating plan with a chosen number of rows and seats per row could be used.

The passenger vector is then initialized. A load factor is chosen, which determines what fraction of available seats is occupied. This, of course, affects the length of the passenger vector. (Length of the passenger vector is equal to (load factor) $\times$ (total number of available seats)). Each element in this vector has a passenger number and corresponding values for this passenger's seat, speed, and class. This vector is rearranged in different ways for the various boarding strategies, by changing the sequence of the passengers before they enter the plane, so that, for instance, passengers with window seats board first.

Next, a vector representing the aisle is created. This vector has two elements per seat-row. Each vector element can contain a single passenger. As passengers move into the aisle, their passenger numbers are stored in this vector.

The propagation / motion of passengers through the aisle is implemented in finite time-steps. The aisle is checked element-by-element from the rear of the plane. When a passenger is encountered in an element, a check is carried out whether another passenger is present in the aisle element directly ahead. If that element is unoccupied, the passenger moves into this aisle-element with the probability (speed) associated with that individual. This check continues through the entire aisle until the entrance of the plane is reached. If the element of the aisle at the entrance to the plane is unoccupied, another passenger is extracted from the passenger vector and fed into the aisle vector. Then another time-step iteration is initiated. The process can be summarised as a sequential checking of the entire aisle (and propagation of passengers through it) during each time-step, and the feeding of new passengers from the passenger vector into the aisle vector.

After any passenger advances one element in the aisle, a check is carried out for whether the passenger reached the row of the assigned seat. If so, the row containing the seat is checked for seated passengers obstructing the path, and the described time delays are implemented. A delay for the loading of each passenger's hand luggage is also initiated when the passenger reaches the row of the allocated seat. Qualitatively, these delays are instituted in such a way that they result in this passenger spending a number of time steps stationary in the aisle.

When time delays expire, the passenger is "seated" and is removed from the aisle.

The entire algorithm is iterated until all passengers are in their allocated seats and the aisle is empty. The time taken for this entire process is recorded and stored.

For a given initial setup of the passenger vector, the entire simulation was run over several iterations to obtain statistically relevant values for:

- Average time taken for the entire boarding process. This is an indication of how effective the boarding process is, since it is in the interest of the airlines to minimize the total boarding time.
- Standard deviation for this average total boarding time over all iterations. The absolute standard deviation is a quantitative measure of the consistency of the boarding procedure, i.e., how sensitive the strategy is to randomness. The relative standard deviation (absolute standard deviation divided by average total boarding time) is a qualitative measure of the consistency of the boarding times and allows comparison between the various strategies and aircraft types.
- Average time (and standard deviation time) that it takes each passenger from when entry until being seated. The standard deviations show how uniform / consistent the boarding time per passenger is.

For larger planes and for custom layouts of the seating plan, we implemented an option for a second aisle vector. The algorithm is carried out as above, with the sequential checking procedure simply being carried out in both aisles during each time-step. Yet some modifications are required: Still only one line of passengers enters the plane; this line has to split into the two aisles. This is done by checking whether the passenger at the entrance to the plane sits on the left half or the right half of the plane, and feeding the passenger into the relevant aisle vector. If the seating layout of the plane has an odd number of seats per row, passengers sitting on the middle seats enter the aisle that first has an unoccupied first element. If the first elements of both aisles are open, the passenger enters either aisle with a 50% probability. The rest of the algorithm progresses as for the single-aisle case.

# Description, Implementation and Results

We describe the various boarding strategies, and the qualitative results for each, in detail. **Table 1** gives numerical results.

The load factors used in the simulations are based on statistics obtained from Transport Canada [2004]. In-depth analysis of the results follows the descriptions.

1. **Random boarding**
   *Description and algorithmic implementation*
   Here the seat numbers in the passenger vector are arranged at random, so that the sequence of passengers entering the aisle at the front of the aircraft is random. Random boarding is common and will thus be used as a reference for comparison with other methods.

**Table 1.** Results for various aircraft and boarding methods (originally in an Appendix).

**Table 1 - Results for various aircraft and boarding methods**

| | Method: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fokker 50 (54 seats) | Ave total boarding time | 207.4 | 166.5 | 290 | 196.7 | 151.2 | 269.7 | 155.6 | 197.2 | 149.1 | 157.9 |
| | Abs. std dev total boarding time | 28.7 | 33.5 | 32.4 | 25.5 | 26.1 | 31.8 | 23.4 | 28.3 | 24.2 | 24.4 |
| | Rel. std dev total boarding time | 0.138 | 0.201 | 0.112 | 0.13 | 0.173 | 0.118 | 0.15 | 0.144 | 0.163 | 0.154 |
| | Ave individual boarding time | 110.1 | 102.7 | 152.5 | 106.7 | 94.5 | 143.8 | 89.8 | 106.7 | 92.3 | 92.1 |
| | Abs. std individual boarding time | 51.6 | 32.6 | 83.0 | 48.2 | 29.4 | 78.2 | 37.9 | 48.5 | 29 | 38.3 |
| | Rel. std individual boarding time | 0.469 | 0.317 | 0.546 | 0.452 | 0.311 | 0.544 | 0.421 | 0.454 | 0.314 | 0.416 |
| Boeing 737-400 (153 seats) | Ave total boarding time | 485.5 | 431.8 | 671.8 | 454 | 404 | 616.6 | 377 | 442.5 | 392.9 | 372.1 |
| | Abs. std dev total boarding time | 51.9 | 62.1 | 59.2 | 46.6 | 54.2 | 50.1 | 53.1 | 39.9 | 50.5 | 41.6 |
| | Rel. std dev total boarding time | 0.107 | 0.148 | 0.088 | 0.103 | 0.134 | 0.081 | 0.141 | 0.09 | 0.129 | 0.112 |
| | Ave individual boarding time | 244.2 | 251.4 | 336 | 233.6 | 236.6 | 313.4 | 219.3 | 232.8 | 233.3 | 207.8 |
| | Abs. std individual boarding time | 119.3 | 90.6 | 193.5 | 111.7 | 82.9 | 177.5 | 97.2 | 109.2 | 81 | 94.9 |
| | Rel. std individual boarding time | 0.489 | 0.36 | 0.576 | 0.478 | 0.351 | 0.566 | 0.443 | 0.469 | 0.347 | 0.456 |
| Boeing 777-200 (340 seats) | Ave total boarding time | 828.4 | 741 | 1058.2 | 801.7 | 697.4 | 1000.6 | 684 | 767.4 | 691.1 | 699.1 |
| | Abs. std dev total boarding time | 60.5 | 70.5 | 64.1 | 62.5 | 63.8 | 61 | 59.5 | 66.1 | 59.3 | 74.1 |
| | Rel. std dev total boarding time | 0.073 | 0.095 | 0.061 | 0.078 | 0.091 | 0.061 | 0.087 | 0.086 | 0.086 | 0.106 |
| | Ave individual boarding time | 412.2 | 430.6 | 509.7 | 397.7 | 413.6 | 479.3 | 404.1 | 401.6 | 421.2 | 392.8 |
| | Abs. std individual boarding time | 191.8 | 144.3 | 292.4 | 184.1 | 135.6 | 275.5 | 150 | 174.4 | 135.4 | 158.6 |
| | Rel. std individual boarding time | 0.465 | 0.335 | 0.574 | 0.463 | 0.328 | 0.575 | 0.371 | 0.434 | 0.321 | 0.404 |
| Custom Large Plane (792 seats) | Ave total boarding time | 1956 | 1837.6 | | 1902.3 | 1803.8 | 2189.8 | 1732 | 1752.3 | 1730.9 | 1661.3 |
| | Abs. std dev total boarding time | 113.7 | 114 | | 105 | 113 | 107.5 | 115.4 | 115 | 105.5 | 121.5 |
| | Rel. std dev total boarding time | 0.058 | 0.062 | | 0.055 | 0.063 | 0.045 | 0.067 | 0.066 | 0.061 | 0.073 |
| | Ave individual boarding time | 948.5 | 964.4 | | 902.2 | 945.1 | 1124 | 896.5 | 937.2 | 974.5 | 929.6 |
| | Abs. std individual boarding time | 483 | 403.1 | | 477.4 | 394.6 | 549.8 | 417.3 | 425.7 | 399.5 | 406.3 |
| | Rel. std individual boarding time | 0.509 | 0.418 | | 0.529 | 0.418 | 0.578 | 0.465 | 0.454 | 0.410 | 0.437 |

worst performance

best performance

关注数学模型

获取更多资讯

*Results*
Random boarding yielded results which were, in general, only faster than those methods that board the plane from front to back. This method never obtained the worst results in any of the measured categories.

2. **Dividing passengers into three groups and beginning boarding with the rear group**
   *Description and algorithmic implementation*
   First the passenger vector is arranged randomly. By finding the highest available seat number, the seating plan is divided into three equal groups. The seat numbers in the passenger vector are arranged in these three groups. The group at the back of the plane boards first, the middle group second, and the front group boards last. Thus, the three groups are internally still arranged at random.
   *Results*
   For all aircraft sizes, this method yielded faster total average boarding times than random boarding. For larger aircraft, the average boarding time per passenger was larger than for random boarding. This may be explained due to each individual's initial seating time. For random boarding, very soon after boarding commences rapid front-positioned seating occurs as people seat themselves randomly; but for this method, people start seating themselves only after the rear passengers move to the back of the plane. Thus, the aisle has to be traversed before any seating occurs. This effect is greatly pronounced if the plane is large and the aisle is longer. The relative deviation of passenger boarding times was lower than that of the random method, which implies that these individual times are more uniformly distributed.

3. **Dividing passengers into three groups and beginning boarding with the front group**
   *Description and algorithmic implementation*
   As before, the available seats are divided into three equal groups. The passenger vector is arranged in such a way that boarding commences with the front group. The middle group boards second and the rear group last. As before, the three groups are internally still arranged at random.
   *Results*
   This method performs worst in almost all aspects. However, the relative standard deviations of the total boarding time are among the best. The poor performance of this particular method can be explained by the congestion of passengers near the entrance to the plane, since the front-seated passengers board first and obstruct flow through the aisle(s). This method was not tested on the largest aircraft, since it was evident that it was the most ineffective boarding strategy.

4. **Beginning boarding by filling window seats first**
   *Description and algorithmic implementation*
   First, the passenger vector is arranged randomly. By checking which seat

numbers are in the first and last column of the seating matrix, passengers with window seats (arranged randomly) are extracted from the passenger vector, which is then rearranged in such a way that these passengers board first. The rest of the passengers are queued behind them at random.
*Results*
This method is faster than random seating but is out-performed by seating in groups from the back of the craft to the front. The standard deviation in total boarding time is small for all aircraft and is the best in this category for the largest aircraft.

5. **Beginning boarding by filling window seats first, and dividing passengers into three groups and beginning boarding with the rear group**
   *Description and algorithmic implementation*
   As above, the window seats are extracted and placed at the front of the passenger vector. Then the passenger vector is then divided into three groups (front, middle and back), and boarding commences with the group at the back of the craft.
   *Results*
   This method is a good improvement on merely commencing boarding with window seats. Thus far, it yields the best results for average total boarding time; but the average time per passenger is not the best.

6. **Beginning boarding by filling window seats first, and dividing passengers into three groups and beginning boarding with the front group**
   *Description and algorithmic implementation*
   As above, passengers with window seats are placed at the front of the passenger vector. Then the passenger vector is then divided into three groups (front, middle and back), and boarding commences with the group at the front of the craft.
   *Results*
   Especially with large aircraft, this method performed poorly. As with method 3, this can be attributed to the congestion at the entrance of the plane.

7. **Dividing passengers into three groups, beginning with the back group, and extracting window seats**
   *Description and algorithmic implementation*
   Again the passengers are grouped into front, middle and back, with the back group at the beginning of the passenger vector. The window seats are then extracted and placed at the front of the vector. Boarding begins with window seats (arranged back to front) and then with normal seats (grouped back to front).
   *Results*
   This method is the best of the methods mentioned thus far, with overall good performance in all aspects.

8. **Filling seats inwards towards the aisle(s)**
   *Description and algorithmic implementation*

For planes with a single aisle: Each passenger's seat is located in the seating plan matrix, and its distance (in terms of seats) from a window seat is calculated. The passenger vector is then rearranged in such a way that the passengers are arranged in terms of this distance from the window seats, beginning with the smallest distance (i.e., with the window seats themselves). The plane fills up from the window seats towards the middle of the plane (which is the aisle).

For planes with two aisles: Essentially the plane is divided into two halves, each aisle being the centre of one half. For simplicity, planes with even numbers of seats between the two aisles were considered, to simplify the location of the middle of the plane. Each half of the plane is then treated as in the previous boarding strategy (that is, as if it were an individual plane with one aisle), and the passenger vector is arranged such that seating begins with passengers furthest from the aisles, and ends with passengers closest to the aisles.

*Results*

This method is an improvement on the strategy of boarding window seats first (method 4). For all aircraft except the smallest (Fokker 50), the average total boarding time is shorter than for method 4. However it is not among the best methods in any particular aspect, though both the average boarding time and the total boarding time are fairly stable (that is, they have relatively small standard deviations).

9. **The passengers are first sorted in groups from back to front, and these groups are further sorted towards the aisle(s)**

   *Description and algorithmic implementation*

   As in strategy 8, the seats are arranged to fill towards the aisle(s). The seats are then further sorted into three groups, and boarding commences with the back group. The table in the left of **Figure 1** shows the way in which the passenger vector is sorted before boarding for a simple aircraft layout with one aisle. The numbers in the figure show in which order seats from the various sections are sorted in the passenger vector.

   *Results*

   This method performs best in most points (as is clear from inspection of **Table 1**). For small aircraft, it is the fastest method. Throughout, the standard deviation of passenger boarding times is good, as is the absolute standard deviation of total boarding time.

10. **The passengers are first sorted towards the aisle(s) and then further divided into groups from back to front**

    *Description and algorithmic implementation*

    Again the three groups are created, from the back of the craft to the front. Then the passengers are sorted within the groups such that the seats farthest from the aisles board first and those closest to the aisle board last. The right-hand table in **Figure 1** shows how the passenger vector is sorted for a simple aircraft layout with one aisle; low numbers seat first.

*Results*

For total boarding time of the largest aircraft, this method yields the best result. For other aircraft, it also performs well in this regard. However this strategy is not very consistent, since the standard deviations of the total boarding times are among the highest, especially for large aircraft. For all aircraft sizes, this method yields shorter average boarding time per passenger than method 9.

| Method 9 | | | | |
|---|---|---|---|---|
| 5 | 6 | Aisle | 6 | 5 |
| 3 | 4 | Aisle | 4 | 3 |
| 1 | 2 | Aisle | 2 | 1 |

| Method 10 | | | | |
|---|---|---|---|---|
| 3 | 6 | Aisle | 6 | 3 |
| 2 | 5 | Aisle | 5 | 2 |
| 1 | 4 | Aisle | 4 | 1 |

**Figure 1.** Illustration of seating strategies 9 and 10 (low numbers seat first).

## Short Summary of Results

For small aircraft (roughly 50 seats), methods 9, 5, and 7 yield the best average total boarding times.

For slightly larger aircraft (roughly 150 seats), methods 10, 7, and 9 yield the best average total boarding times.

For medium aircraft (roughly 300 seats), methods 7, 9, and 5 yield the best average total boarding times.

For large aircraft (roughly 800 seats), methods 10, 9, and 7 yield the best average total boarding times.

It is thus clear that

*methods 5, 7, 9 and 10 are the most efficient strategies.*

They have in common that they begin boarding with passengers seated in the rear of the plane. Furthermore, they implement a further sorting criterion (for instance, boarding window seats first or filling the columns of the plane towards the aisles).

Random boarding was among the three most *in*efficient methods for all plane classes.

# Sensitivity Analysis

To see how sensitive the algorithm is to variations in its parameters, we carried out additional simulations.

- **Changing the percentage of disabled / frail / handicapped passengers**

  We carried out several simulations with varying percentages of disabled or handicapped passengers. Some methods were affected more strongly by these changes than others.

  As an example, we provide results from a simulation with a Boeing 777-200 (midsize aircraft), with the same load factor as used previously (0.78). We change from 2% of the total passengers assumed to be handicapped to 6%.

  The percentage increases in average total boarding time for methods 1, 9, and 10 were 16%, 13%, and 19%. The percentage change does not vary too greatly for the various methods.

- **Investigating the effect of various load factors on the total boarding time** From the results obtained, we chose one method (method 9) and ran the simulation over a range of load factors. We found a strong linear relation between load factor and average total boarding time. We conclude that it is fairly irrelevant what load factor is used to compare boarding procedures, provided the same load factor is used for all.

# Advantages of Our Model

The model can be customised to accommodate any seat plan specification (an aircraft with either one or two aisles). Two decks on a plane could simply be modelled as two separate aircraft with their individual seating configurations.

Our measurements are sensible in that they provide information from 200 iterations of each boarding procedure. Thus, the accuracy of the specific boarding measurements is reliable.

We strove to keep all parameters fixed during the simulation of each boarding strategy. This ensures that if a parameter has been allocated an unrealistic value, then this fault has a reduced effect on the outcome of the experiment and, more specifically, on the comparison of boarding strategies.

Many different boarding strategies were implemented. Some were combined with others to yield a more substantial result. In essence, any of the boarding strategies can be combined in the model to produce many more procedures. We did, however, select strategies that we assume to be realistic and that yield a spread of data and results that are informative.

# Improvements, Comments

- We do not address deboarding strategies. We are convinced, though, that it is significantly more challenging to set up effective boarding strategies, since this involves arranging passengers before they board the plane. In a sense, deboarding is inherently a more structured process. We also believe that a simple reversal of the more effective boarding procedures should save time during deboarding.

- The speeds of passengers are strongly discrete in the model. Distinctions are made only among frail/handicapped passengers, children, and healthy adults. It would have been more realistic to implement a probability distribution of walking speeds.

- Throughout the model, the assumption is made that passengers move in only one direction in the aisle. This does not accommodate the fact that sometimes people move opposite to the stream of passengers in the aisle.

- In our model, passengers cannot pass in the aisle. This is not realistic, though in most cases passengers would probably wait for a person ahead of them to stow luggage before passing. This could slow boarding and would affect the outcomes of some boarding strategies (for instance, boarding from the front would have yielded a significantly shorter boarding time). Nonetheless, we believe that our model is inherently systematic and that the obtained results are meaningful and believable.

- It would be sensible to allow for two doors into the plane. However, one door at the front of the plane and one at the rear could be likened to boarding two separate planes, each from one entrance.

- We could have investigated how division into more groups would have affected the simulations. Nonetheless, we believe that it is not reasonable for airport staff to have to divide passengers into so many groups, as this process itself would be time-consuming.

- Special seat allocations and boarding strategies for disabled people could have been considered, but this would not have affected the final outcome of the simulations greatly, due to the small percentage of disabled passengers. Perhaps one strategy should have involved seating all disabled passengers in the front of the plane so that they do not obstruct motion through the aisles and that they do not have to walk as far in the plane.

- Many of our boarding strategies do not allow for passengers to board in groups (for instance, mothers with their children), and this could cause inconveniences in its implementation in the real world.

- The various delays in the boarding algorithm were based on guesses that we made by comparing the average walking speed of a healthy person to the average time that we assumed would be needed to stow hand luggage. These delays were then implemented in the finite time-step nature of the algorithm, and could have been researched more accurately.

# Conclusion

The task assigned to us was to devise and test various strategies for boarding procedures for various classes of aircraft. Using the approach of an algorithm based on a finite time-step cellular automaton, we obtained some clear results.

From our simulations, we find that certain boarding procedures result in significant savings of time. The most efficient strategies all apply two filters to the passengers before the plane is boarded. One filter involves sorting passengers so that that those farthest from the entrance board first, and the other sorts passengers according to seat columns. Nonetheless, methods that disrupt boarding of groups of adjacently seated passengers may be logistically difficult to implement or even irritate passengers (e.g., method 10).

The most outstanding of these are implemented as follows (see **Figure 2**):

**Method A:** The aircraft is divided in groups seated from the rear to front, and each of these groups is further sorted from window (and centre) seating towards the aisle(s).

**Method B:** The passengers are first sorted from window seats (and centre seats) towards the aisle(s). They are then further divided into groups from rear to front. (The figure illustrates how this would be implemented in a two-aisle aircraft)

Time differences in total average boarding time between the most efficient and least efficient strategies are up to 3 min for small planes and 5–6 min for larger planes.

Essentially, an optimised seating strategy merely divides the plane into sections and dictates which sections are boarded first. Thus, once a strategy is selected, it would be easy to include these sections on the tickets of passengers, and to have the passengers group themselves as desired (for instance, by allotting areas at the boarding gates to the various passenger groups, or calling them separately for boarding). In this way, the organisation of the passengers into the desired sequence need not imply a significant time delay at all. The trade-off between shortened boarding time and required organisation time is especially pronounced with small aircraft.

Methods with the best total boarding times do not necessarily guarantee the shortest average boarding time per individual passenger. This is important to note, since the implementation of such a strategy may infringe on convenience to the passengers. The average boarding time per individual passenger for some methods was greater than that for random boarding, especially

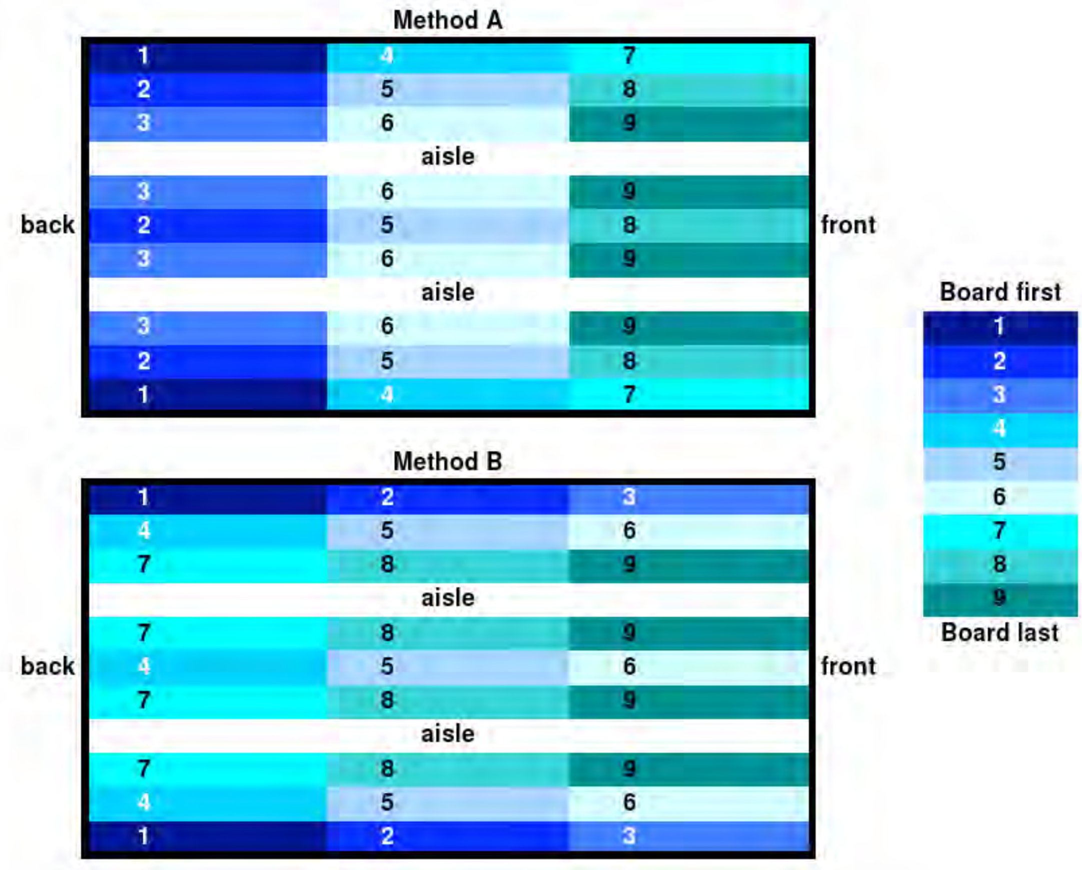



**Figure 2.** Implementation of Methods A and B in a two-aisle aircraft.

for large aircraft. This is not necessarily relevant, since random boarding has a larger spread in individual boarding times. Furthermore, the differences in average boarding time per passenger in relation to the total boarding times of these methods were relatively small. Yet a rigorous sorting of passengers prior to boarding may very well be perceived as irritating by many passengers. This effect should be minimised by performing the sorting efficiently and in a simple manner (as suggested above).

It is easier to group people by seat row of the aircraft than by column, since passengers often travel in groups and some sorting methods would disrupt these groups. Thus Method A would most probably be more practical to enforce than Method B. The event of passengers not abiding to the desired procedure could cause disruption of the boarding process.

In conclusion, *we recommend Method A*.

Airports and airlines could further shorten boarding times by making infrastructure changes, such as allowing passengers to board from both ends of the plane.

We do not believe that implementation of these structured boarding strategies in the real world would result in an administrative waste of time that outweighs the potential boarding time savings.

# References

Finney, Paul Burnham. 2006. Loading an airliner is rocket science. *New York Times* (14 November 2006). http://travel2.nytimes.com/2006/11/14/business/14boarding.html .

KLM Airways. n.d. http://www.klm.com/travel/fi_en/travel_information/on_board/seating_plans/index.htm .

Transport Canada Aviation Forecasting. 2004. Passenger load factors. http://www.tc.gc.ca/pol/en/airforcasting/assumptionreport2004/assum20047.htm .

Top, from left: Advisor J.H. Van Vuuren, Chris Rohwer; bottom: Andreas Hafver, Louise Viljoen.

# Judges' Commentary: The Fusaro Award Airplane Seating Paper

Marie Vanisko
Dept. of Mathematics
Carroll College
Helena, MT
mvanisko@carroll.edu

Peter Anspach
National Security Agency
Ft. Meade, MD
anspach@aol.com

The Ben Fusaro Award for the 2007 Airplane Seating Problem went to a team from Rowan University in Glassboro, New Jersey. Their paper was designated Meritorious; it fell just short of the Outstanding designation due to an error in one of their equations and some questionable results. However, this paper exemplified some outstanding characteristics:

- it presented a high-quality application of the complete modeling process;
- it demonstrated noteworthy originality and creativity in their modeling effort; and
- it was well written, in a clear expository style, making it a pleasure to read.

The students were asked to devise and compare procedures for boarding and deboarding planes with varying numbers of passengers. They were also asked to prepare an executive summary for an audience of airline executives, gate agents, and flight crews, in which they explained their findings.

Addressing real-world problems involves formulating a mathematical description of the problem, solving the mathematical model, interpreting the mathematical solution, and critically evaluating the model.

Before a team could formulate a mathematical description of the problem, it was necessary to do research to estimate reasonable values for parameters to be

*The UMAP Journal* 28 (3) (2007) 479–481. 

used. The Rowan University team began by looking at current boarding procedures and came up with a detailed list of sources of boarding delays, including the storing of carry-on luggage. Based on their assumptions, it was clear that the team members had considered many issues associated with the boarding process, and that they justified each assumption. Certain assumptions, for example, in terms of the implication on boarding time "random seating and assignment seating can be thought to be equivalent," might be questionable. However, as long as they used this assumption consistently in their simulation models, it was considered allowable. It should be mentioned that such assumptions help to distinguish Outstanding papers from Meritorious papers. In setting up their simulation model, the Rowan University team considered

- the time that it takes passengers to walk to their seats;
- the service time, which includes time for stowing luggage, based on the size and quantity of luggage; and
- seating time.

The rather important detail of distinguishing separate times for these activities was overlooked by many other teams. The Rowan team determined walking speed using an accelerometer and included the results in their model, using a uniformly-distributed random variable, together with a factor that allowed for a decrease in walking speed as the number of passengers increased. They used a uniform distribution over the interval 11.5 to 14.5 seconds to estimate stowing time for luggage. The number of passengers with carry-on luggage was estimated with a log function of a uniformly distributed random variable. Seating time was a function of which column (window, middle, aisle) passengers were in. Although the level of mathematics used in this model may not have been as high as some, the team utilized it very well. Overall, the Rowan model was quite simple, but the description of parameters used was very clearly spelled out. This is what judges look for when simulations are done.

Their simulation models consisted of four different seating methods:

- open seating, where passengers are lined up randomly;
- back-to-front seating;
- outside-in Seating (WilMA); and
- modified reverse-pyramid seating, in which the outer columns are seated first, followed by open seating of the rest of the plane.

To test the efficiency of their model, the team used Matlab simulations with several types of small, medium, and large aircraft. Reports were given for the mean, median, and variance of the simulated results for each type of seating on each type of plane. Frequency histograms were also given for each category. This type of reporting clearly demonstrated the results of their simulations.

However, the judges did not feel that all the results were reasonable, and this was a reason for the Meritorious designation rather than Outstanding. If the team had acknowledged the unreasonableness of some of their results, that would have been more acceptable. Nevertheless, this paper is a very good example of mathematical modeling. The team is to be congratulated for using mathematics to create their own model to solve the problem at hand, in a clear and solid example of the modeling process.

# About the Authors

Marie Vanisko has retired from Cal State Stanislaus and moved back to Montana, where she taught for 31 years at Carroll College and was a visiting professor at the U.S. Military Academy at West Point. She chairs a College Board committee for the SAT Subject Tests in Mathematics and serves on a national joint committee of the National Council of Teachers of Mathematics and the Mathematical Association of America (MAA). For each of the past two years, Marie has co-directed an MAA Tensor Foundation grant project for high school girls, entitled Preparing Women for Mathematical Modeling, with the hope of encouraging more young women to select careers that involve mathematics. She serves as a judge for the COMAP MCM and HiMCM has also been active in the MAA PMET (Preparing Mathematicians to Educate Teachers) project.

**UNITED STATES POSTAL SERVICE**

# Statement of Ownership, Management, and Circulation (All Periodicals Publications Except Requester Publications)

| 1. Publication Title | 2. Publication Number | 3. Filing Date |
|---|---|---|
| The UMAP Journal | 0 1 9 7 _ 3 6 2 2 | 9/7/07 |
| **4. Issue Frequency** | **5. Number of Issues Published Annually** | **6. Annual Subscription Price** |
| Quarterly | Four (4) | $104.00 |

7. Complete Mailing Address of Known Office of Publication *(Not printer) (Street, city, county, state, and ZIP+4®)*

COMAP, Inc., 175 Middlesex Tpk., Suite 3B, Bedford, MA 01730

Contact Person
John Tomicek

Telephone *(Include area code)*
781/862-7878 x130

8. Complete Mailing Address of Headquarters or General Business Office of Publisher *(Not printer)*

SAME

9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor *(Do not leave blank)*

Publisher *(Name and complete mailing address)*

Solomon Garfunkel, 175 Middlesex Tpk., Suite 3B, Bedford, MA 01730

Editor *(Name and complete mailing address)*

Paul Campbell, 700 College St., Beloit, WI 53511

Managing Editor *(Name and complete mailing address)*

Tim McLean, 175 Middlesex Tpk., Suite 3B, Bedford, MA 01730

10. Owner *(Do not leave blank. If the publication is owned by a corporation, give the name and address of the corporation immediately followed by the names and addresses of all stockholders owning or holding 1 percent or more of the total amount of stock. If not owned by a corporation, give the names and addresses of the individual owners. If owned by a partnership or other unincorporated firm, give its name and address as well as those of each individual owner. If the publication is published by a nonprofit organization, give its name and address.)*

| Full Name | Complete Mailing Address |
|---|---|
| Consortium for Mathematics and | 175 Middlesex Tpk., Suite 3B, Bedford, MA 01730 |
| Its Applications, Inc. (COMAP, Inc.) | |
| | |
| | |
| | |

11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities. If none, check box → ☐ None

| Full Name | Complete Mailing Address |
|---|---|
| | |
| | |
| | |
| | |
| | |

12. Tax Status *(For completion by nonprofit organizations authorized to mail at nonprofit rates) (Check one)*
The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes:

☐ Has Not Changed During Preceding 12 Months

☐ Has Changed During Preceding 12 Months *(Publisher must submit explanation of change with this statement)*

| 13. Publication Title | 14. Issue Date for Circulation Data | |
|---|---|---|
| The UMAP Journal | 11/16/07 | |

| 15. **Extent and Nature of Circulation** | | | **Average No. Copies Each Issue During Preceding 12 Months** | **No. Copies of Single Issue Published Nearest to Filing Date** |
|---|---|---|---|---|
| a. Total Number of Copies *(Net press run)* | | | 720 | 730 |
| b. Paid Circulation *(By Mail and Outside the Mail)* | (1) | Mailed Outside-County Paid Subscriptions Stated on PS Form 3541 *(Include paid distribution above nominal rate, advertiser's proof copies, and exchange copies)* | 650 | 670 |
| | (2) | Mailed In-County Paid Subscriptions Stated on PS Form 3541 *(Include paid distribution above nominal rate, advertiser's proof copies, and exchange copies)* | 0 | 0 |
| | (3) | Paid Distribution Outside the Mails Including Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Paid Distribution Outside USPS® | 30 | 35 |
| | (4) | Paid Distribution by Other Classes of Mail Through the USPS (e.g. First-Class Mail®) | 0 | 0 |
| c. Total Paid Distribution *(Sum of 15b (1), (2),(3), and (4))* | | | 680 | 705 |
| d. Free or Nominal Rate Distribution *(By Mail and Outside the Mail)* | (1) | Free or Nominal Rate Outside-County Copies lincluded on PS Form 3541 | 0 | 0 |
| | (2) | Free or Nominal Rate In-County Copies Included on PS Form 3541 | 15 | 21 |
| | (3) | Free or Nominal Rate Copies Mailed at Other Classes Through the USPS (e.g. First-Class Mail) | 0 | 0 |
| | (4) | Free or Nominal Rate Distribution Outside the Mail *(Carriers or other means)* | 0 | 0 |
| e. Total Free or Nominal Rate Distribution *(Sum of 15d (1), (2), (3) and (4)* | | | 15 | 21 |
| f. Total Distribution *(Sum of 15c and 15e)* ▶ | | | 695 | 726 |
| g. Copies not Distributed *(See Instructions to Publishers #4 (page #3))* ▶ | | | 5 | 16 |
| h. Total *(Sum of 15f and g)* ▶ | | | 700 | 742 |
| i. Percent Paid *(15c divided by 15f times 100)* ▶ | | | 98 | 98 |

16. Publication of Statement of Ownership

☒ If the publication is a general publication, publication of this statement is required. Will be printed in the Third issue of this publication.

☐ Publication not required.

| 17. Signature and Title of Editor, Publisher, Business Manager, or Owner | Date |
|---|---|
| Solomon Garfunkel | 11/16/07 |

I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanctions (including fines and imprisonment) and/or civil sanctions (including civil penalties).