CS4093: Special Topics in Computing II
Constraint Programming and Optimisation

Things this module will help you avoid ...

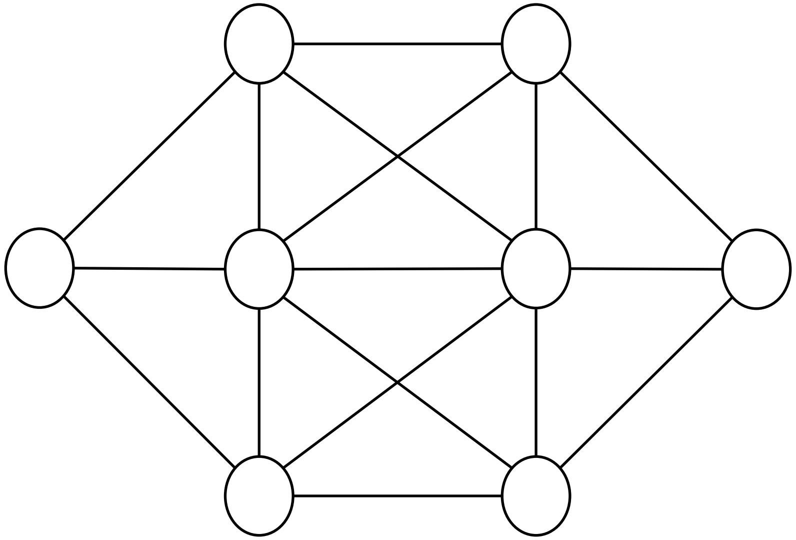
Each plane has a type, and a target time window in which they want to touch down. There may be multiple open runways with different size restrictions.

The *problem*:

- Assign a runway and landing time to each plane so that the time is inside the time window, and the constraints are satisfied

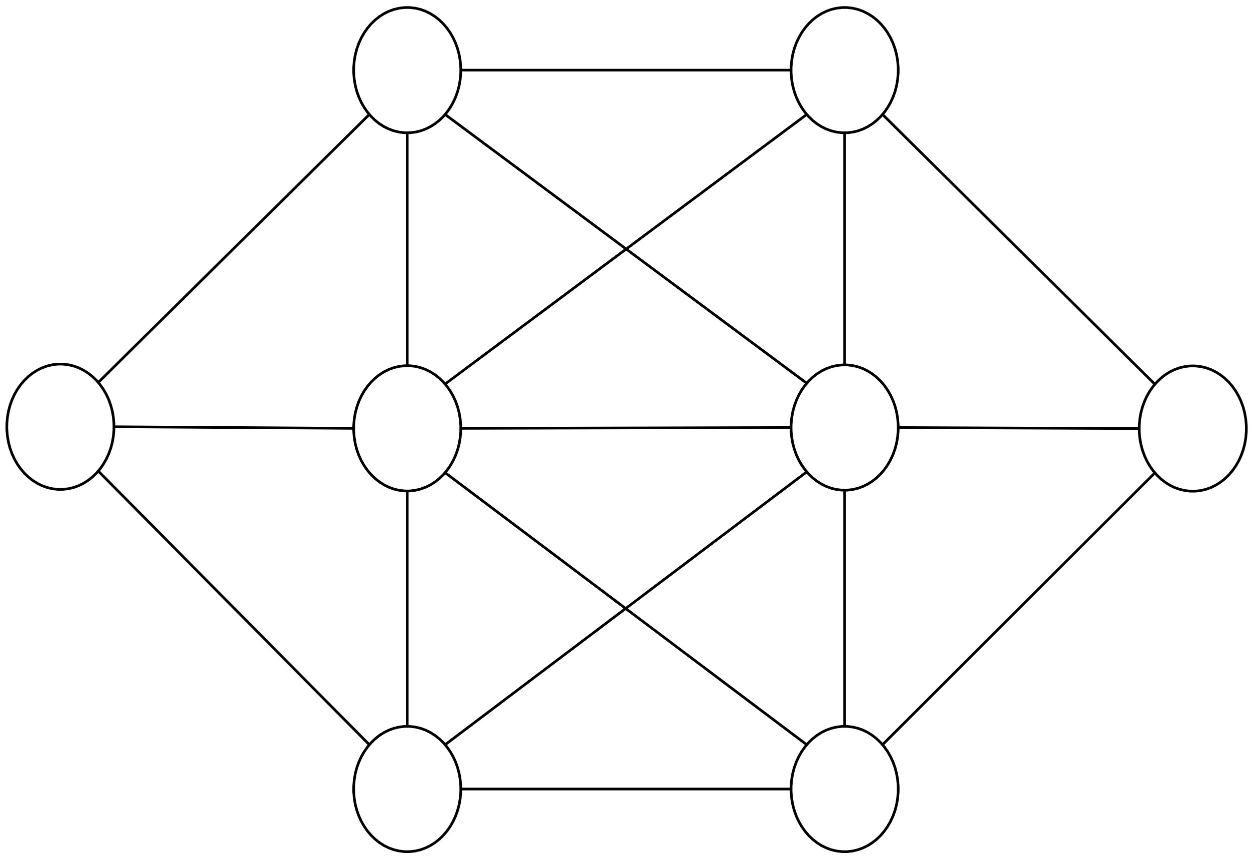
The *constraints*:

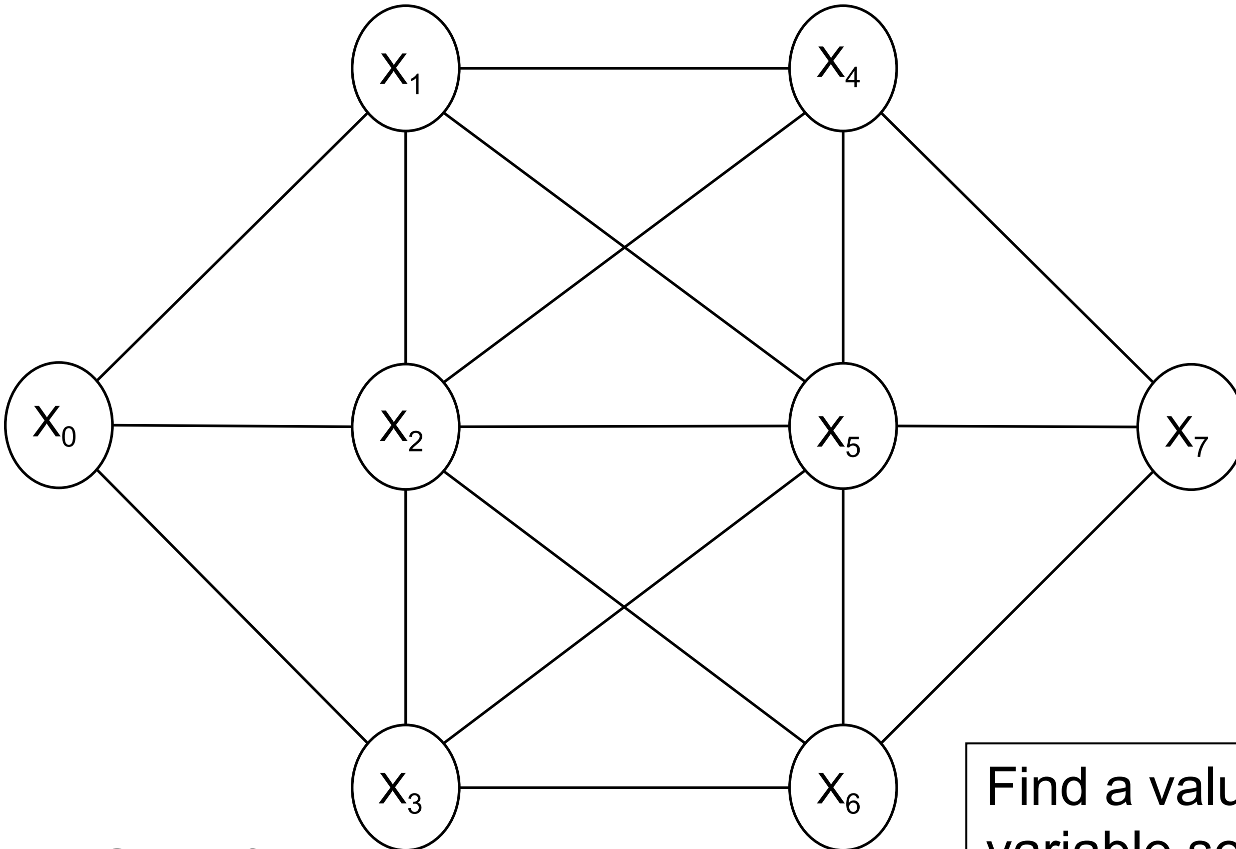
- Make sure no two planes land on the same runway within (e.g.) 240 seconds
- Planes of type 1 cannot use runway 3 (e.g.)



Using only the numbers 1 to 8, put a different number in each circle so that adjacent circles do not have consecutive numbers

You have 8 minutes to complete it.





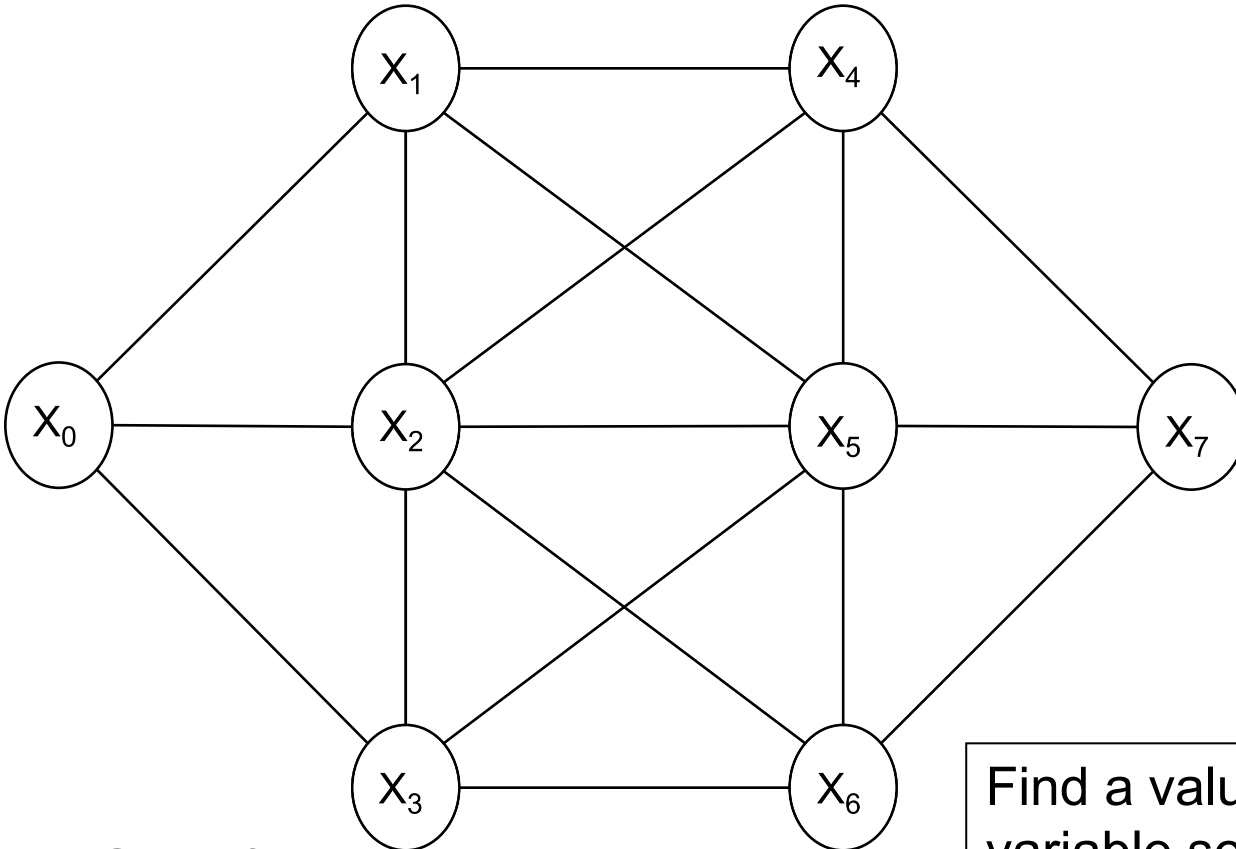
Find a value for each variable so that all constraints are satisfied simultaneously

A *Constraint Satisfaction Problem*:

Variables: $\{X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$

Values: $\{1,2,3,4,5,6,7,8\}$

Constraints: $\{X_0 \neq X_1, X_0 \neq X_2, \ldots, X_0 \neq X_7, \ldots, |X_0 - X_1| > 1, |X_0 - X_2| > 1, \ldots\}$



Find a value for each variable so that all constraints are satisfied simultaneously

A *Constraint Satisfaction Problem*:

Variables: $\{X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$

Values: $\{1,2,3,4,5,6,7,8\}$

Constraints: $\{\text{alldifferent}(\{X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7\}), \ldots,$
$|X_0-X_1|>1, |X_0-X_2|>1, \ldots\}$

The Constraint Satisfaction Problem

Given three sets:

variables $V = \{X_1, X_2, ..., X_n\}$,

domains $D = \{D_1, D_2, ..., D_n\}$ specifying allowable values for each variable, and

constraints $C = \{C_1, C_2, ..., C_m\}$ restricting the values that groups of variables can take simultaneously,

find an assignment for each variable X_i of a value v_i from its domain D_i, (i.e. find an n-tuple $(v_1, v_2, ..., v_n)$, where $v_i \in D_i$), so that all constraints are satisfied.

Constraint

For a given constraint satisfaction problem (V,D,C),

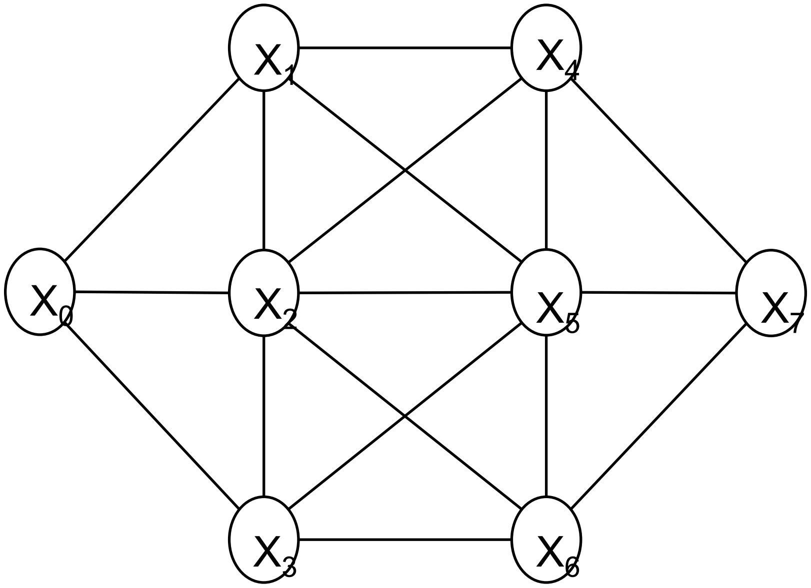
a constraint C_i acts on a set of variables $\{X_{i1}, X_{i2}, ..., X_{it_i}\} \subseteq V$ called its *scope*, and specifies a subset of the cartesian product of its domains

$$C_i \subseteq D_{i1} \times D_{i2} \times \ ... \times D_{it_i}$$

An assignment $(v_{i1}, v_{i2}, ..., v_{it_i})$ to the variables in the scope of C_i *satisfies* C_i if and only if

$$(v_{i1}, v_{i2}, ..., v_{it_i}) \in C_i$$

Example



(X_1,X_4)
(0,2)
(0,3)
:
(1,3)
(1,4)
:
(2,0)
(2,4)
(2,5)
:

The domains D1 and D4 are the set {0,1,2,3,4,5,6,7}
The constraint $|X_1-X_4|>1$ specifies a subset of D1 $\times$ D4
which allows the assignment pairs on the right:

The total assignment (0,1,2,3,4,5,6,7) satisfies the constraint between X1 and X4

(but does not satisfy the constraint between (e.g.) X0 and X1, and so is not a solution.

Constraint programming

Modelling: what are the variables, values and constraints?
Searching: guessing values, backtracking on failure
Heuristics: which variable or value to try next
Inference: ruling out options by reasoning
Symmetry: spotting repeat patterns in the problem space
Complexity: understanding the inherent difficulty
Optimisation: are some solutions better than others?
Applications: what real problems can we solve?
Programming: how to implement all of this in programming languages, so that it is correct and allows us to find solutions efficiently

Each plane has a type, and a target time window in which they want to touch down. There may be multiple open runways with different size restrictions.

Assign a runway and time to each plane so that

(Problem 1): the time is inside the time window

(Problem 2): the deviation from the time windows is minimised

needs an optimisation function specifying the cost of each deviation, and a way of combining all deviation costs

Variables: the planes

Values: ordered pairs (runway, time)

Constraints: no two planes can land on the same runway within (e.g.) 120 seconds

Planes of type 1 cannot use runway 3 (e.g.)

The Constraint *Optimisation* Problem

Given three sets:

variables $V = \{X_1, X_2, ..., X_n\}$,

domains $D = \{D_1, D_2, ..., D_n\}$ of allowable values for each variable, and

constraints $C = \{C_1, C_2, ..., C_m\}$ restricting the values that groups of variables can take simultaneously,

and a function $f : D_1 \times D_2 \times ... \times D_n \rightarrow \mathbb{R}$,

find an assignment for each variable X_i of a value v_i from its domain D_i, (i.e. find an n-tuple $(v_1, v_2, ..., v_n)$, where $v_i \in D_i$), so that all constraints are satisfied and $f(v_1, v_2, ..., v_n)$ is minimised.

IBM ILOG CPLEX Optimization Studio

Features | Components | Downloads

Optimization model development toolkit for mathematical and constraint programming

IBM ILOG CPLEX Optimization Studio Preview Edition Trial

Download the trial software

Download the trial

IBM® ILOG® CPLEX® Optimization Studio is an analytical decision support toolkit for rapid development and deployment of optimization models using mathematical and constraint programming. It combines an integrated development environment (IDE) with the powerful Optimization Programming Language (OPL) and high-performance ILOG CPLEX optimizer solvers.

IBM ILOG CPLEX Optimization Studio enables you to:

- **Optimize your business decisions** with high-performance optimization engines.
- **Develop and deploy optimization models quickly** by using flexible interfaces and prebuilt deployment scenarios.
- **Create real-world applications** that can significantly improve business outcomes.

Best solution found
632
Best possible optimal solution
621.48
Percent from optimal
1.67%

```
maximize TotalAll

subject to {
  forall(fl in Fl
    ctCapLimit:
      sum(it in I
```

Solutions | Conflicts
Element (2)
ctRequiredAssignmentCon
ctRequiredAssignmentCon

Description
Power7 server (http:/
bidding_2DF3809241
55
sched_atfm_58CAAE7
51

Identify the most cost-effective solution for any

Write clear and maintainable optimization

Identify precisely which constraint and which data

Launch and monitor your optimization

Not in United Kingdom?

Select another country.

Considering a purchase?

View prices & buy

Contact IBM

Chat now

E-mail IBM

Request a quote

Or call us at: 01475 898688
Priority code: 101K803W

Product support

Featured offers

Information Kit: Getting started manual for CPLEX Optimization Studio V12.6.1

Online Demo: Getting Started With CPLEX Optimization Studio or-tools - The Google Op: ×

https://code.google.com/p/or-tools/

Apps | Free Hotmail | Suggested Sites | Web Slice Gallery | Favorites

k.brown@cs.ucc.ie ▼ | My favorites ▼ | Profile | Sign out

Research at Google

or-tools

The Google Operations Research Suite

Search projects

Project Home | Wiki | Issues | Source

Summary | People

Project Information

Starred by 368 users
Project feeds

Code license
Apache License 2.0

Labels
Algorithm, CPlusPlus, Python, Optimization, OperationsResearch, ConstraintProgramming, Java, MathProgramming, Graph, Knapsack, CSharp, VehicleRouting, Scheduling

Members
lper...@google.com, vfur...@google.com, laurent...@gmail.com, b...@google.com, orw...@google.com
5 committers
4 contributors

Links

Blogs
Google+ OR-Tools
Google OR-Tools Blog
external non Google blog: My Constraint Programming Blog
external non Google blog: Optimization Zen
external non Google blog: Operations Research Competitions
external non Google blog: CP is fun

External links
Getting Started
OR Tools Online Documentation
Download Binary Archives
Download Documentation

Groups
OR Tools Discussion group

Google's suite of operations research tools, **or-tools**, is available for C++, Python, Java, and .NET (using Mono on non-Windows platforms). It contains:

- A constraint programming solver
- A vehicle routing library
- A wrapper around multiple linear solvers (GLPK, CLP, CBC, SCIP, Sulum, and Gurobi)
- Knapsack algorithms
- Graph algorithms (shortest paths, min-cost flow, max flow, linear sum assignment)

Installation instructions are at Getting Started and detailed documentation at or-tools documentation.

To use or-tools, you can either check out the source code and build the suite from scratch, or use one of our pre-compiled archives for C++, Python, Java, and .NET platforms here (mirror). The Python modules are also available on Pypi under the name ortools. Instructions for installing the binary archives are here.

For more information, subscribe to the or-tools-discuss group and the Google+ page. We'll post all news there.

The code is known to compile on:

- gcc 4.7.x and above on Ubuntu 12.04 and up (12.04, 12.10, 13.04, 13.10)
- XCode >= 5.0 (clang++ with C++11 support)
- Microsoft Visual Studio 2012 and 2013 (2010 is not supported as it lacks support for C++11)

Both 32-bit and 64-bit architectures are supported, although the code is optimized to run in 64-bit mode.

Follow us on Google+

JSRs
Search JSRs GO
» JSRs by Platform
» JSRs by Technology
» JSRs by Stage
» JSRs by Committee
» List of All JSRs

My JCP
User ID: username
Password: GO
Register for Site
Having problems logging in?

JCP Info
» About JCP
» Get Involved
» Community Resources
» Community News
» FAQ
» Contact Us

JSR | Community | Expert Group
Summary | Proposal | Detail (Summary & Proposal)

JSRs: Java Specification Requests
JSR 331: Constraint Programming API

Stage	Access	Start	Finish
Maintenance Release	Download page	16 Oct, 2012	
Maintenance Draft Review	Download page	30 Aug, 2012	29 Sep, 2012
Final Release	Download page	07 Mar, 2012	
Final Approval Ballot	View results	07 Feb, 2012	20 Feb, 2012
Proposed Final Draft	Download page	30 Sep, 2011	
Public Review Ballot	View results	04 Jan, 2011	10 Jan, 2011
Public Review	Download page	22 Nov, 2010	10 Jan, 2011
Early Draft Review	Download page	30 Mar, 2010	28 Jun, 2010
Expert Group Formation		18 Aug, 2009	10 Nov, 2009
JSR Review Ballot	View results	04 Aug, 2009	17 Aug, 2009

Status: Maintenance
JCP version in use: 2.7
Java Specification Participation Agreement version in use: 2.0

Description:
This specification defines a Java runtime API for constraint programming. The CP API prescribes a set of fundamental operations used to define and solve constraint satisfaction and optimization problems.

Please direct comments on this JSR to the Spec Lead(s)

Team

Specification Leads

Jacob Feldman	Feldman, Jacob

Expert Group

Feldman, Jacob	Jussien, Narendra	Keil, Werner
Samsung Electronics Corporation	Simonis, Helmut	Szymanek, Radoslaw
Vergamini, Didier	Viry, Patrick	

Updates to the Original JSR

choco-solver.org | A Free ...

choco-solver.org

Apps | Free Hotmail | Suggested Sites | Web Slice Gallery | Favorites

CHOCO

choco-solver.org

A Free and Open-Source Java Library for Constraint Programming

Home | Choco3 | Download | Extensions | Forums | Team | Related projects | Log in

What is Choco ?

Download | User Guide | Apidocs

Choco is a *Free and Open-Source Software*[1] dedicated to **Constraint Programming**. It aims at describing real combinatorial problems in the form of Constraint Satisfaction Problems and to solve them with Constraint Programming techniques.

Choco can be used for:

- teaching (a user-oriented constraint solver with open-source code)
- research (state-of-the-art algorithms and techniques, user-defined constraints, domains and variables)
- real-life applications (many application now embed CHOCO)

Choco is easy to manipulate, that's why it is widely used for teaching. And Choco is also performant, and we are proud to count industrial users too.

Choco is developed with IntelliJ IDEA and JProfiler.

l'Europe s'engage en Pays de la Loire

We will use Choco on CS4093

[1]: Choco is distributed under BSD license (Copyright(c) 1999-2014, Ecole des Mines de Nantes).

By the way, what is Constraint Programming?

Such a paradigm takes its features from various domains (Operational Research, Artificial Intelligence, etc). Constraint programming is now part of the portfolio of global solutions for processing real combinatorial problems. Actually, this technique provides tools to deal with a wide range of choco-solver.org | A Free × 1.3. Real examples — or-t × RTÉ Comet-chasing duo Roset ×

www.rte.ie/news/special-reports/2014/1112/658716-rosetta-philae-space/

Apps Free Hotmail Suggested Sites Web Slice Gallery Favorites

Comet-chasing duo Rosetta and Philae

1 1

Wednesday 12 November 2014 12.53

An ESA illustration of the Philae lander (centre) descending onto the comet after a successful separation from the Rosetta probe

Ten facts to sum up orbiter Rosetta and robot lab Philae, Europe's comet-chasing duo.

10 years and eight months: How long Rosetta and its payload Philae spent together after their launch on 2 March 2004.

6.5 billion kilometres: The distance they travelled together before Philae ejected and choco-solver.org | A Free × 1.3. Real examples — or-t × How Were Rosetta/Philae ×

https://www.ibm.com/developerworks/community/blogs/jfp/entry/how_were_rosetta_philae_operations_sche

Apps Free Hotmail Suggested Sites Web Slice Gallery Favorites

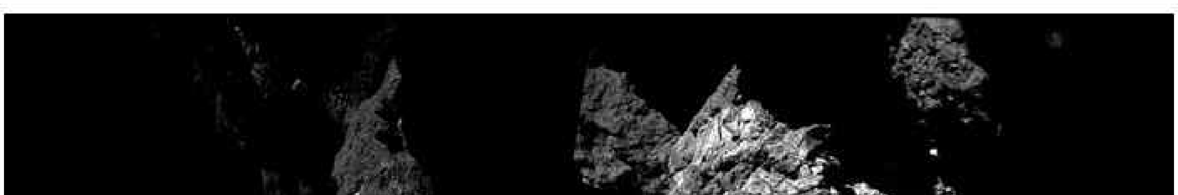
IT Best Kept Secret Is Optimization

Log in to particip...

How Were Rosetta/Philae Operations Scheduled?

JeanFrancoisPuget | *Nov 19* | *Visits (4976)* 6

Unless you live unplugged you certainly saw the astounding pictures of Comet 67P/Churyumov–Gerasimenko taken by the Philae lander. Besides producing nice images, Philae embarked scientific instruments, each developed by a European laboratory, to accomplish scientific experiments when approaching, and after landing on the comet. Given that communication takes about 25 minutes between Earth and Philae once landed, it was very important to carefully plan every operations of the mission in advance. Indeed, there would be little room to adjust and modify plans after landing.



About this blog

Business Analytics and Mathematical Optimization. My Views are my own.

Related posts

IBM libera versão gr...
UpdatedDec 8 0 0

10 tips for CIOs pla...
UpdatedDec 5 0 0

Mobile infrastructur...
UpdatedDec 2 1 0

Join the Twitter #in...
UpdatedDec 3 0 0

Credit : ESA http://blogs.esa.int/rosetta/2014/11/13/comet-with-a-view/

The plans for the approach, landing, and for performing all experiments were elaborated on the ground at the *Science Operations and Navigation Centre* (**SONC**) in Toulouse, France. The problem was modeled as a constraint programming problem. A software (called **MOST**) has been developed on top of IBM Constraint Programming technology (Ilog-Scheduler/Solver) to solve this constraint programming problem.

The above description is based on the Rosetta/Philae blog entry by E. Hebreard. With his colleagues, he has published a scientific paper that describes the use of constraint programming for scheduling Philae operations. We reproduce the paper abstract below.

The Rosetta/Philae mission was launched in 2004 by the European Space Agency (ESA). It is scheduled to reach the comet 67P/Churyumov-Gerasimenko in 2014 after traveling more than six billion kilometers. The Philae module will then be separated from the orbiter (Rosetta) to attempt the first ever landing on the surface of a comet. If it succeeds, it will engage a sequence of scientific exploratory experiments on the comet. In this paper we describe a constraint programming model for scheduling the different experiments of the mission. A feasible plan must satisfy a number of constraints induced by energetic resources, precedence relations on activities, or incompatibility between instruments. Moreover, a very important aspect is related to the transfer (to the orbiter then to Earth) of all the data produced by the instruments. The capacity of inboard memories and the limitation of transfers within visibility windows between lander and orbiter, make the transfer policy implemented on the lander's CPU prone to data loss.We introduce a global constraint to handle data transfers. The goal of this constraint is to ensure that data-producing activities are scheduled in such a way that no data is lost. Thanks to this constraint and to the filtering rules we propose, mission control engineers are now able to compute feasible plans in a few seconds for scenarios where minutes or even hours were previously often required. Moreover, in many cases, data transfers are now much more accurately simulated, thus increasing the reliability of the plans.

The paper is worth a read. i can't resist highlighting this piece: "*Except for the data transfer aspect, all the constraints above can be modeled using the standard methods and algorithms [7] all available in Ilog-Scheduler. Hence, we focus on data transfers and propose a global constraint to reason about this aspect of the problem.*" This shows that we can both have a quite powerful tool (ILOG Scheduler), and a need to extend it with problem specific constraints (data transfer constraint). The ability to extend constraint programming solvers is one of their key value.

Update on Nov 20. Let me conclude with a comment on the IBM product being used here, namely ILOG Scheduler. It is a great product, but we made quite significant evolutions to our constraint programming offering over the past decade. We now recommend to use CP Optimizer for scheduling problems like the above. Converting applications from ILOG Scheduler to CP Optimizer was discussed in our November 19 virtual user group. Slides and replay are available here.

Rosetta/Philae

I am involved in a collaboration with the CNES in Toulouse on scheduling the scientific experiments of the space probe Rosetta/Philae on the comet 67P/Churyumov-Gerasimenko.

We have published a paper at CP 2012. You can see Gilles Simonin's slides of his talk at the conference.

Below is a description of the mission and of our role, taken from a paper currently submitted to Constraints. There is also vulgarisation article, however in French.

Following the fly-by of the comets Halley and Grigg-Skjellerup by the spacecraft Giotto, an even more ambitious mission, including the landing of a robotic module on the comet nucleus, was approved by European Space Agency in 1993. This project involves more than 50 contractors from 14 European countries, Canada and the United States for developing the instruments necessary to a deeper study of the comet. The

choco-solver.org | A Free
1.3. Real examples — or-t
The Insight Centre for Dat

www.insight-centre.org

Apps | Free Hotmail | Suggested Sites | Web Slice Gallery | Favorites

Home | About | Research | Application | Commercialisation | News & Events | People | Work with us | Contact Us

Search | Search

Insight

Centre for Data Analytics

At Insight we undertake high impact research in data analytics that has significant impact on industry and society by enabling better decision making.

Creating a data driven society

Insight Researcher at Mission Control for Rosetta/Philae Landing

Read more

President Higgins meets with Insight researchers and collaborators during State Visit to China

Read more

w.insight-centre.org/content/insight-researcher-mission-control-rosettaphilae-landing-0 compartment, or memory resources for experiments and the lander. The planning around these issues was critical to the battery life and therefore crucial to the mission. Calculation of such plans is analogous to solving a combinatorial optimization or Scheduling problem. Due to low board computing capabilities, design of experiments are made on the ground, and then transmitted to the probe. Engineers from the Scientific Operations and Navigation Centre (SONC) of CNES (Centre National d'Etudes Spatiales) in Toulouse (France), were responsible for the activities planning of the mission. The engineers developed a tool called MOST (Mission Operations Scheduling Tool) for the generation of these plans

Dr Gilles Simonin, currently working as a postdoctoral researcher at Insight, was a member of the ROC team (Recherche Opérationnelle, Optimisation Combinatoire et Contraintes) based at the LAAS laboratory in Toulouse that worked with the SONC team to develop a constraint-programming model into MOST for scheduling the different experiments of the mission. A scheduling plan must satisfy a number of constraints induced by energetic resources, precedence relations on tasks, and incompatibility between instruments. Moreover, a very important aspect is related to the transfer (to the orbiter and then to Earth) of all the data produced by the instruments. The capacity of inboard memories and the limitation of transfers within the visibility windows between lander and Orbiter make the transfer policy implemented on the lander CPU prone to data loss. Gilles' team introduced a global constraint to handle data transfers. The goal of this constraint is to ensure that data producing tasks are scheduled in such a way that no data is lost. As a result of this constraint and the filtering rules the team implemented, mission control was able to compute feasible plans in a few seconds for scenarios where minutes were previously often required.

During the Rosetta landing in November, Gilles was invited back to Toulouse to follow the live Rosetta/Philae landing. Gilles continues to collaborate with his old team on the MOST tool to improve the constraint model and the techniques defined for this kind of missions.

choco-solver.org | A Free   1.3. Real examples — or-t   Home

www.4c.ucc.ie/~ehebrard/Home.html

Apps   Free Hotmail   Suggested Sites   Web Slice Gallery   Favorites

Home   Publications   Research   Software   Co-Authors   CV

Cork Constraint Computation Centre

Emmanuel Hebrard

Embark Post-doctoral Fellow.

Affiliation:
Cork Constraint Computation Centre and
University College Cork

Address:
Cork Constraint Computation Centre,
University College Cork,
Cork, Ireland.

Email: e.hebrard "at" 4c.ucc.ie
Work Phone: +353 (0)21 425 5407
Fax: +353 (0)21 425 5424

UNIVERSITY COLLEGE, CORK
Coláiste na hOllscoile Corcaigh

The exam timetabling problem

Schedule all exams for UCC in two weeks in May.

- We have a set of modules, a set of rooms, and a set of students.
- Each module must be given a room and a start time.
- The length of each exam is known.
- The identity of the students on each module is known.
- The number of desks in each exam room is known.
- All exams must take place between 9 am and 9 pm.
- The distance and travel time between each room is known.
- The time to clear or fill each exam room is known.

Exam timetabling constraints

- No modules with students in common can overlap in time.
- Students must have enough time to move between exams.
- The modules assigned to a room must have total students no more than the capacity.
- Overlapping exams in a room must start and finish at the same time.
- There must be enough time between exams in a room to empty the room and allow new students to enter.

- For each student, exams should be balanced across the two weeks (?)

How hard is the exam timetabling problem?

- How many modules?
- How many students?
- How many overlaps are there?
- How many rooms are there?

How do we represent it as a constraint problem?

How should we solve it?

- Is there necessarily a solution?

How would a person solve it by hand?

Constraint programming applications

Constraint programming is deployed in:

- planning and scheduling in space missions
- logistics and supply chain optimisation
- scheduling production in steel factories
- mobile network frequency assignments
- routing in telecommunication networks
- configuration of complex software products
- scheduling car assembly lines
- aircraft maintenance planning
- workforce rostering
- integrated chip design
- radiology treatment planning
- protein structure determination

and in many more applications.

What are we going to do in CS4093

- explore how to model and solve problems in Choco
- explore the underlying algorithms and methods
- understand the theory of constraint programming
- study existing applications of constraint programming
- explore current research topics

CS4093 Syllabus

- Intro to Choco
- Modelling
- ... and more modelling
- Search
- Propagation
- Global Constraints
- Search Heuristics
- Problem Complexity
- Local Search
- Current Research in Constraint Programming

The formal details for CS4093

Lectures: Tuesday 2pm, WGB G04
Thursday 9am, WGB 304

Continuous assessment: 10% of the total marks available
1st CA: 3 marks
2nd CA: 7 marks

Exam: 90 minutes, standard written exam

What do you need to survive on CS4093?

- you must be able program in Java
- you should be comfortable with everything in CS1105
- you should be capable of clear thinking
- you should enjoy solving problems
- you must be self-motivated and willing to explore libraries and tools for yourself

- it will help if you have taken
  - CS4618 (Artificial Intelligence I)
  - a module on data structures and algorithms

What you can do now

Make sure the following are installed on whichever machine you are using:

- Java 1.8

- A Java IDE
  (I will be using Eclipse Luna (4.4.1))

Next lecture ...

Introduction to Choco

Acknowledgements:
- the Crystal Maze CSP is by Patrick and Zoë Prosser, and many of the ideas in the module about how to explain constraint programming are Patrick's.