Advanced Mixed Integer Programming Formulation Techniques

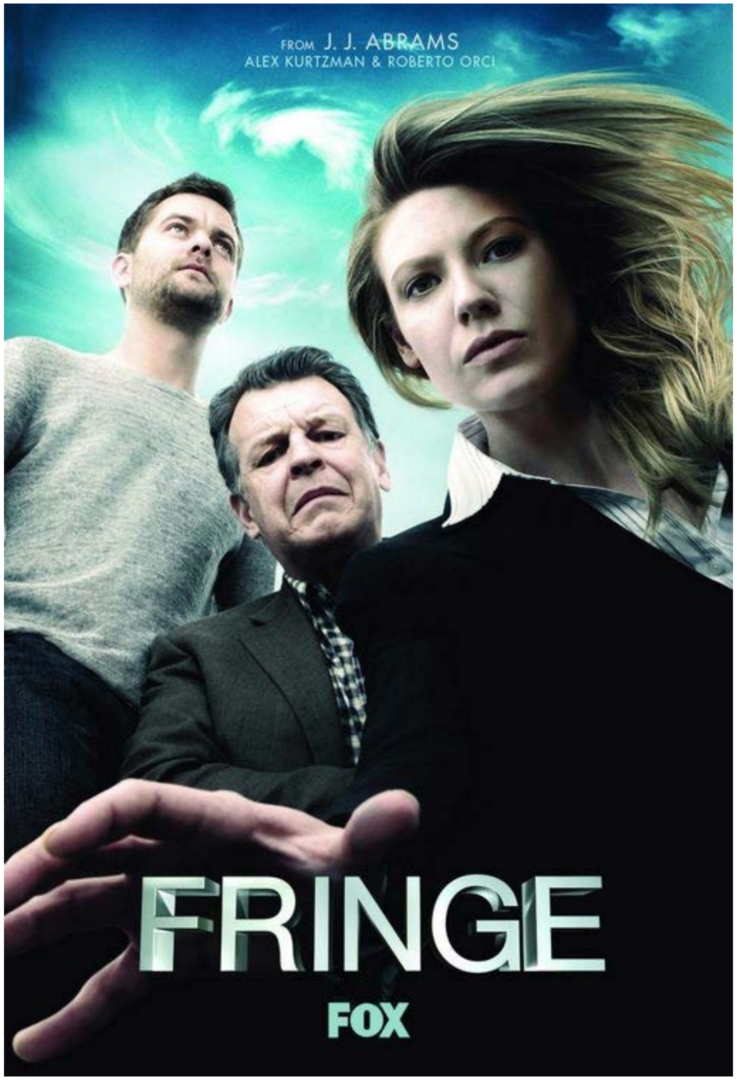
Joey Huchette and Juan Pablo Vielma

Massachusetts Institute of Technology

Spring School of the International Symposium on Combinatorial Optimization (ISCO 2018)

Marrakesh, Morocco, April, 2018

Why MIP?


FROM J. J. ABRAMS
ALEX KURTZMAN & ROBERTO ORCI
FRINGE
FOX


YOU'LL ALSO NEED
SOMEONE WHO CAN
WORK WITH MIXED
INTEGER PROGRAMS

(Nonlinear) Mixed Integer Programming (MIP)

$$\min \quad f(x)$$

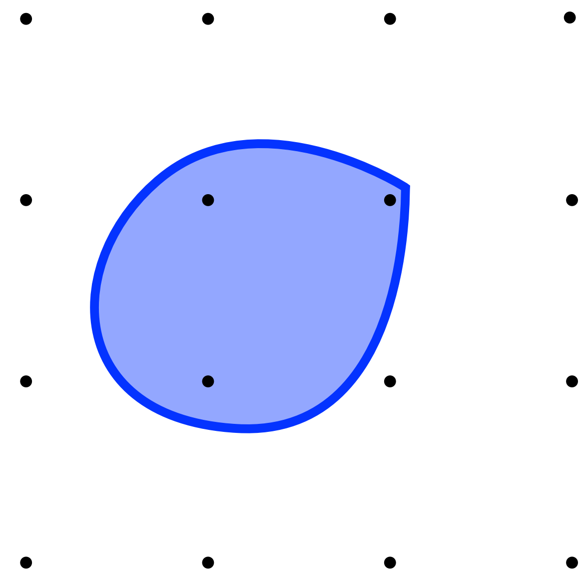
$$s.t.$$

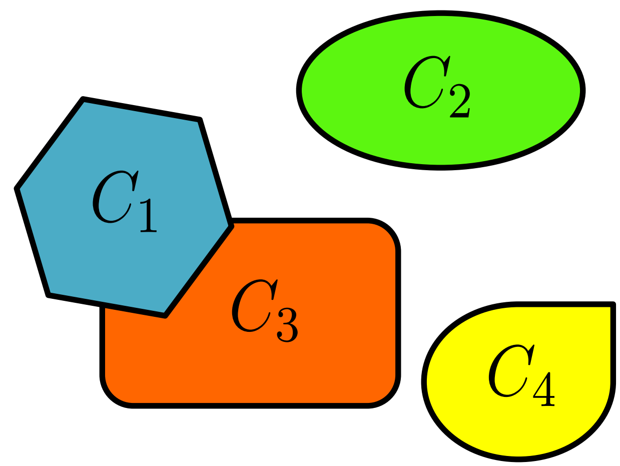
$$x \in C$$

$$x_i \in \mathbb{Z} \quad i \in I$$

convex f and C.

NP-hard = Challenge Accepted!





Pure 0-1 : Traveling Salesman Problem

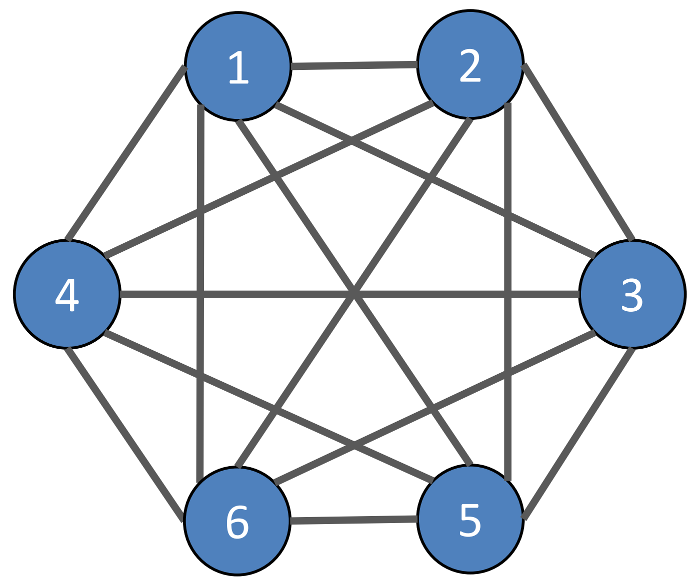
$$\min \sum_{e \in E} t_e x_e$$

$$s.t.$$

$$\sum_{e \in \delta(v)} x_e = 2 \qquad \forall v \in V$$

$$x_e \in \{0,1\} \quad \forall e \in E$$

$$\sum_{e \in E(S)} x_e \le |S| - 1 \quad \forall S \subsetneq V$$



General Integer : e.g. Knapsack

$$\max \quad \sum_{i=1}^{n} b_i x_i$$

$$s.t.$$

$$\sum_{i=1}^{n} a_i x_i \leq c$$

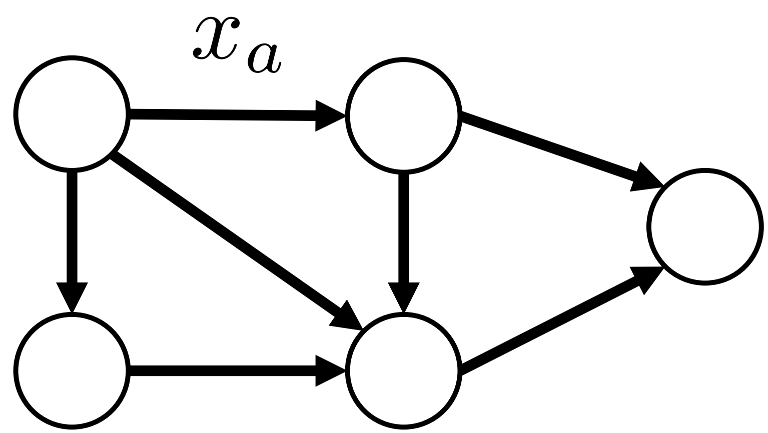
$$x_i \geq 0 \quad \forall i \in \{1, \ldots, n\}$$

$$x_i \in \mathbb{Z} \quad \forall i \in \{1, \ldots, n\}$$

Mixed-Integer or Mixed-Binary

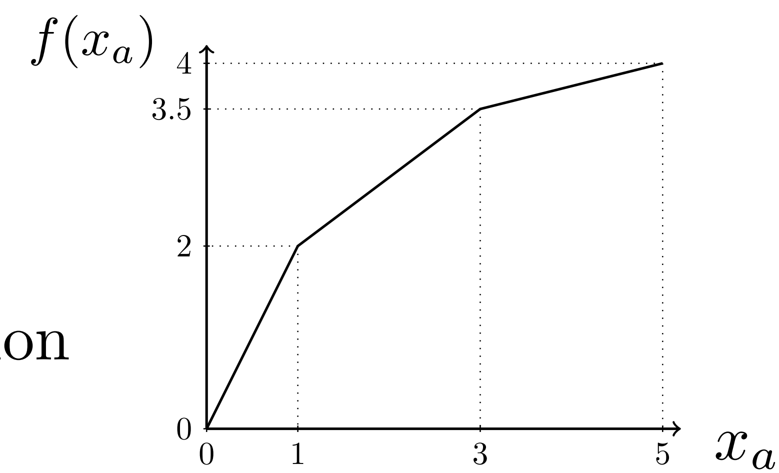
- Fixed-charge transportation problems:

$$
\begin{aligned}
&\min \quad \sum\nolimits_{a\in A} v_a x_a + f_a y_a \\
&s.t. \\
&\quad \{x_a\}_{a\in A} \text{ satisfy flow conservation} \\
&\quad x_a \le u_a y_a \quad \forall i \in \{1,\ldots,n\} \\
&\quad y_a \in \{0,1\} \quad \forall i \in \{1,\ldots,n\}
\end{aligned}
$$



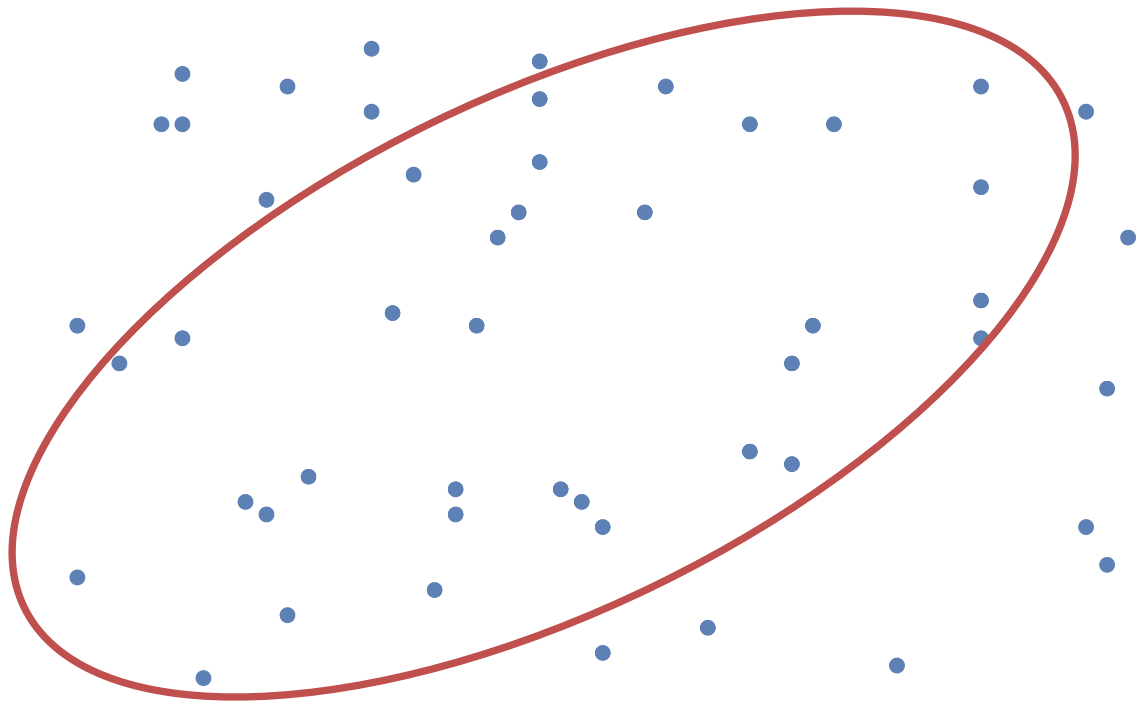
- Also transportation problems with concave costs.

$$
\begin{aligned}
&\min \quad \sum\nolimits_{a\in A} f(x_a) \\
&s.t. \\
&\quad \{x_a\}_{a\in A} \text{ satisfy flow conservation}
\end{aligned}
$$

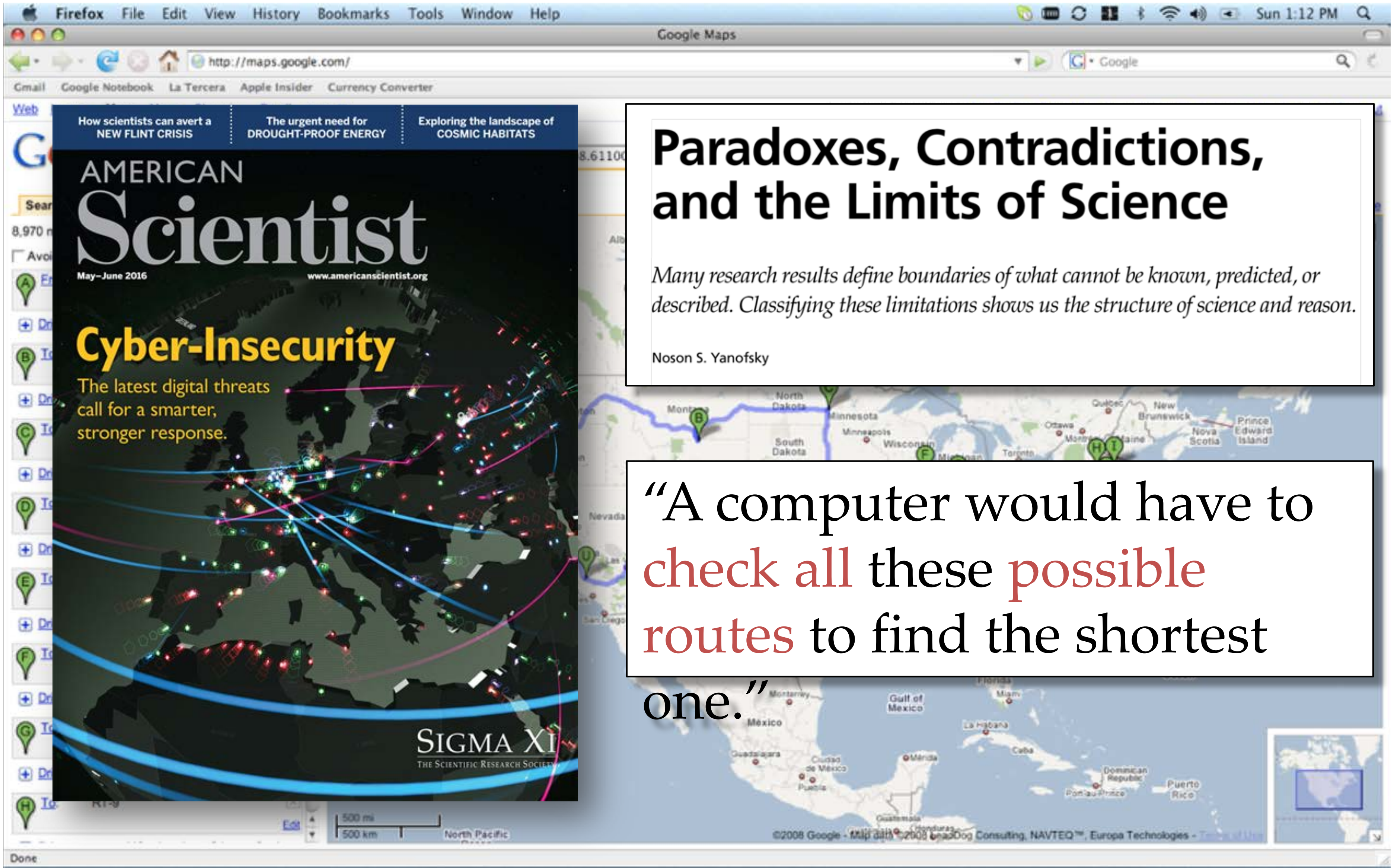


Also Nonlinear Mixed Integer Programming (MIP)

- Example: Find minimum volume ellipsoid that contains 90% of data points



How hard is MIP: Traveling Salesman Problem ?



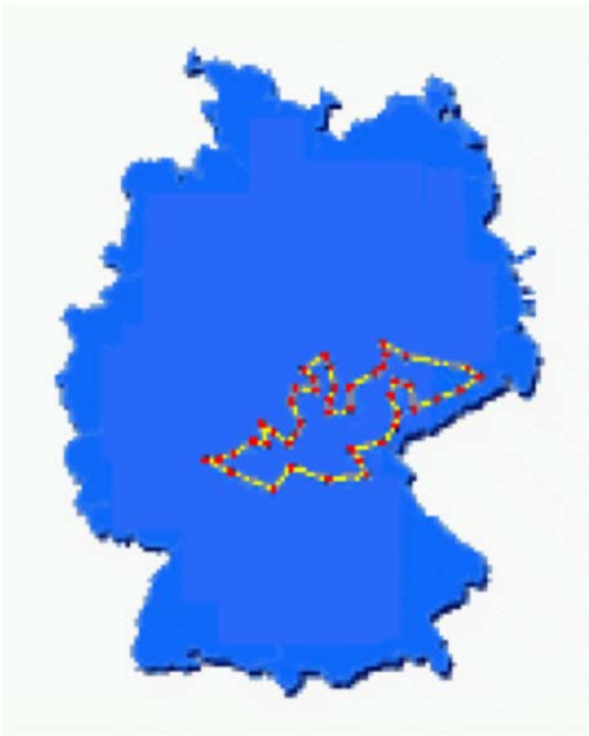
Deep Learning and TSP?

- “we implemented the Held-Karp algorithm [18] which finds the optimal solution in $O(2^n n^2)$ (we used it up to **$n = 20$**). For **larger n, producing exact solutions is extremely costly**, therefore we also considered algorithms that produce approximated solutions:”
- “In practice, **TSP solvers rely on handcrafted heuristics** that guide their search procedures to find competitive tours efficiently. Even though these heuristics work well on TSP, once the **problem statement changes slightly, they need to be revised**.”
- “While most successful machine learning techniques fall into the family of supervised learning, where a mapping from training inputs to outputs is learned, supervised learning is not applicable to **most combinatorial optimization problems because one does not have access to optimal labels**.” (considers problems **up to 100 nodes**)

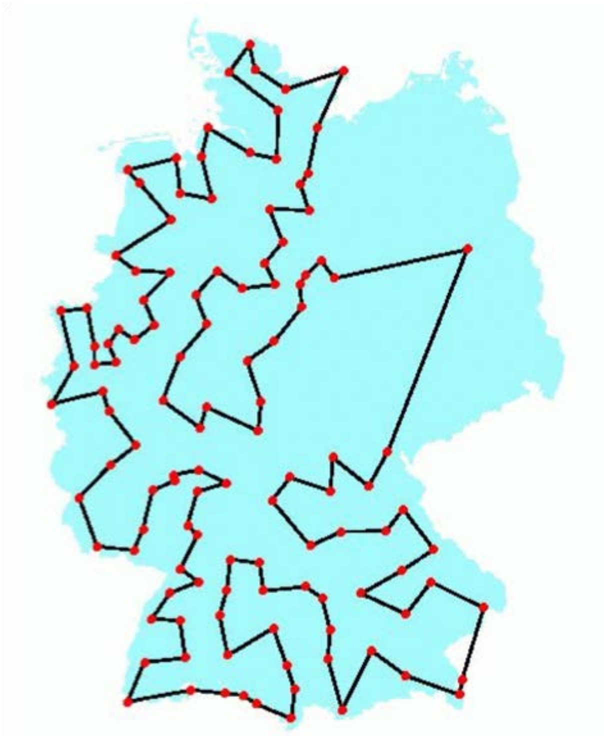
MIP = Avoid Enumeration

- Number of tours for 49 cities $= 48!/2 \approx 10^{60}$
- Fastest supercomputer $\approx 10^{17}\text{flops}$
- Assuming one floating point operation per tour:
  $> 10^{35}$ years $\approx 10^{25}$ times the age of the universe!
- How long does it take on an iphone?
  - Less than a second!
  - **4 iterations** of **cutting plane** method!
  - Dantzig, Fulkerson and Johnson 1954 did it by hand!
  - For more info see tutorial in ConcordeTSP app
  - **Cutting planes** are the key for effectively solving (even NP-hard) MIP problems in practice.

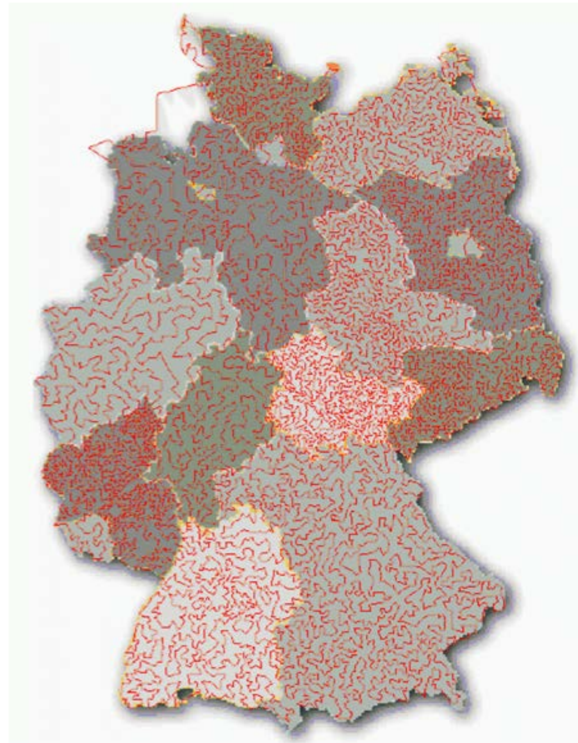
Using MIP to visit Germany



45 cities(1832)



120 cities(1977)



15,112 cities(2004)

But, what About General MIP !?

- “In practice, TSP solvers rely on handcrafted heuristics that guide their search procedures to find competitive tours efficiently. **Even though these heuristics work well on TSP, once the problem statement changes slightly, they need to be revised**.”

50+ Years of MIP = Significant Solver Speedups

- Algorithmic Improvements (**Machine Independent**):
  - CPLEX → ILOG → IBM
    - v1.2 (1991) – v11 (2007): **29,000 x** speedup
  - GUROBI OPTIMIZATION
    - v1 (2009) – v6.5 (2015): **48.7 x** speedup

≈ **1.9 x / year**

- Also **convex nonlinear**:
  - GUROBI OPTIMIZATION
    - v6.0 (2014) – v6.5 (2015) quadratic: **4.43 x**
      (**V.**, Dunning, Huchette, Lubin, 2015)

Widespread Use of Linear/Quadratic MIP Solvers

http://www.gurobi.com/company/example-customers

State of MIP Solvers

- Mature: Linear and Quadratic (Conic Quadratic/SOCP)
  - Commercial:

GUROBI OPTIMIZATION IBM CPLEX optimized by Xpress MP FICO

  - “Open Source”

SCIP COIN|OR CBC GNU GLPK

- Emerging: Convex Nonlinear (e.g. SDP)
  - Open-Source + Commercial linear MIP Solver > Commercial

KNITRO < COIN|OR Bonmin < 

Accessing MIP Solvers = Modelling Languages

- User-friendly algebraic modelling languages (AML):

GAMS
AMPL

PYOMO ↔ python™
CVX RESEARCH ↔ MATLAB®

Standalone and Fast

Based on General Language and Versatile

- Fast and Versatile, but complicated
  - Proprietary low-level C/C++ solver interphases.
  - C/C++ Coin-OR interphases and frameworks
- 21st Century AMLs: JuMP ↔ julia

What Do You Need to Access this Technology

- The right modeling language (easy)

- A “good” formulation

JuMP

julia

Why julia and JuMP ?

- julia http://julialang.org
  - 21st century programming language
  - MIT licensed (and developed): free and open source
  - (Almost) as fast as C (LLVM JIT) and as easy as Matlab
    - “Floats like python/matlab, stings like C/Fortran”
  - Easy to use and wide library ecosystem (specialized and frontend)
  - Only language besides C/Fortran to scale to 1 Petaflop!
    - 10^{15} floating point operations per second on NERSC Cori Phase II (9,300 nodes and 650,000 cores)

Why julia and JuMP ?

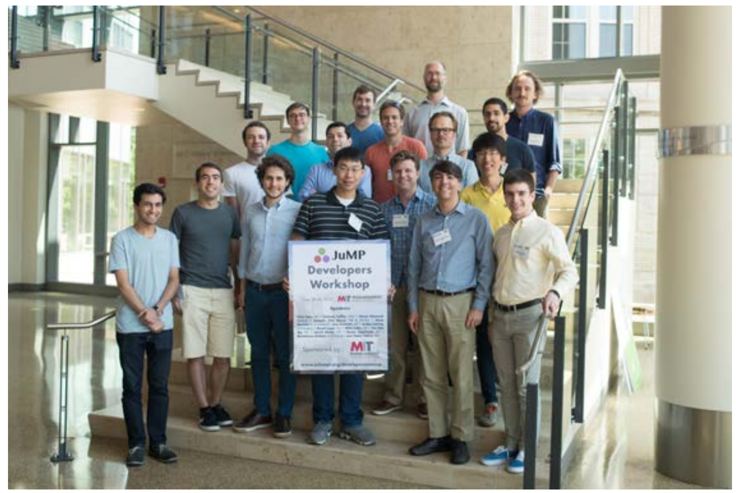
- JuMP https://github.com/JuliaOpt/JuMP.jl
  - Also open-source and free.
  - Julia-based algebraic modelling language for optimization
  - Easy and natural syntax for linear, quadratic and conic (e.g. SDP) mixed-integer optimization.
  - Modular, extensible, easy to embed (e.g. simulation, visualization, etc.) and FAST.
  - Solver-independent access to advanced MIP features (e.g cutting plane callbacks)
  - Advanced automatic differentiation of user defined functions

JuMP

Started by students

Iain Dunning, Miles Lubin and Joey Huchette

Current Community



OPERATIONS RESEARCH CENTER

MIT MANAGEMENT SLOAN SCHOOL

Google Research

PUC RIO

UCL Université catholique de Louvain

THE UNIVERSITY OF AUCKLAND NEW ZEALAND

PSR

Los Alamos NATIONAL LABORATORY

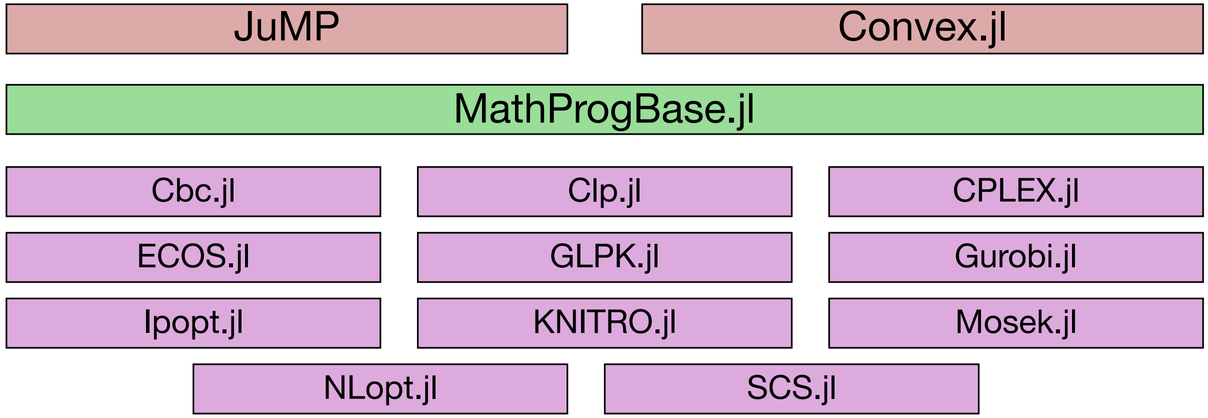
Lawrence Livermore National Laboratory

WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

mosek

Extensive Stack of Modelling and Solver Packages

JuliaOpt



Solvers

http://www.juliaopt.org

- JuMP extensions for: block stochastic optimization, robust optimization, chance constraints, piecewise linear optimization, polynomial optimization, multi-objective optimization, discrete time stochastic optimal control, sum of squares optimization, etc.
- Useful Julia Packages: Multivariate Polynomials, etc.

JuMP Awards

2016 ICS Prize: "JuMP's design leverages advanced features of the Julia language to offer distinctive functionality while achieving performance in instance creation often similar to commercial modeling tools."

- "Computing in operations research using Julia", M. Lubin, I. Dunning. INFORMS Journal on Computing 27 (2), 238-248.
- "JuMP: A modeling language for mathematical optimization", I. Dunning, J. Huchette, M. Lubin. SIAM Review 59 (2), 295-320
- Also COIN-OR Cup and MIT's Operations Research Center's best student paper award

JuMP-dev Workshop

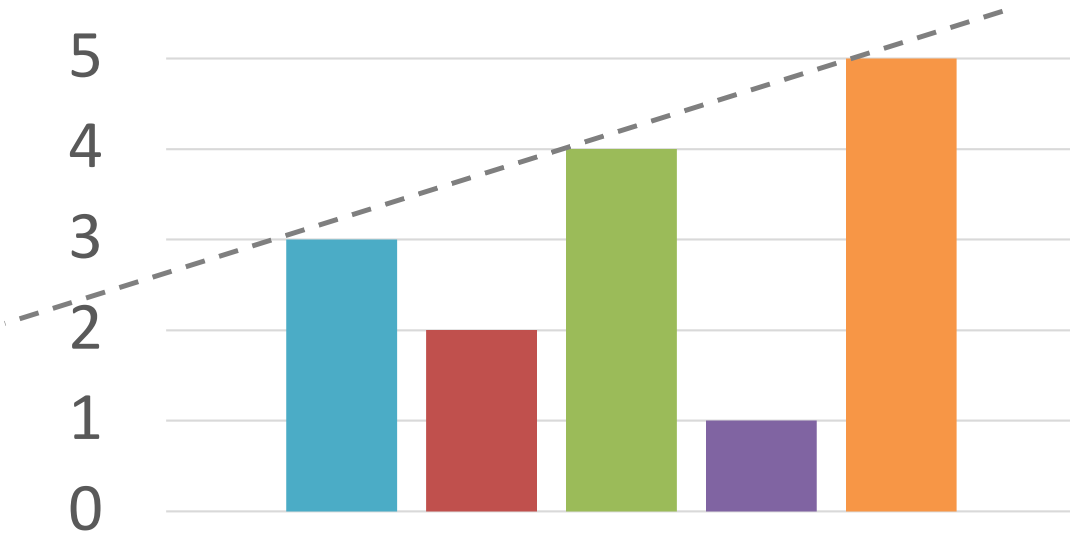
JuMP Developers Workshop. MIT Sloan, Cambridge, MA. June, 2017.

The Second Annual JuMP-dev Workshop
June 27-29, 2018, University of Bordeaux.
The week before ISMP.
http://www.juliaopt.org/meetings/bordeaux2018/

MIP Formulations

Exercise (Challenge?): Formulation for Tower Puzzle

- $x_{\{i,j\}} \in [5] \quad \forall i, j \in [5]$
- $\left(x_{\{i,j\}}\right)_{i=1}^{5}$ are all different $\forall j \in [5]$
- $\left(x_{\{i,j\}}\right)_{j=1}^{5}$ are all different $\forall j \in [5]$
- Visibility constraints:



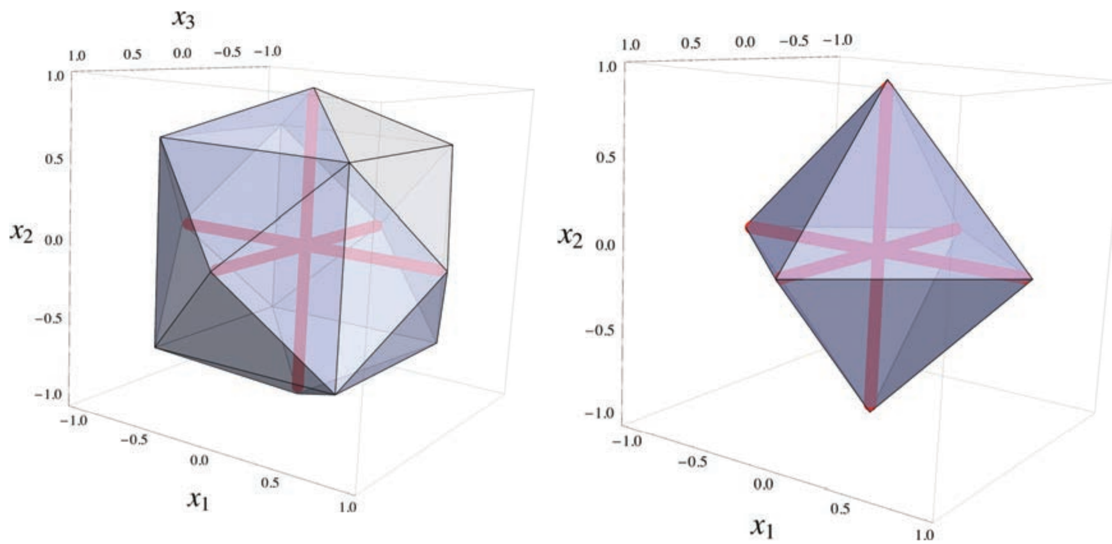
	2	1	3	2	2	
2	1	5	2	4	3	3
1	5	4	1	3	2	4
3	3	2	4	1	5	1
2	4	3	5	2	1	3
3	2	1	3	5	4	2
	3	4	2	1	2	

$[5] := \{1, \ldots, 5\}$

Overview of Spring School

- Theoretical background (polyhedral theory), definition of "good" formulations, etc.
- Introduction to JuMP
- Constructing Formulations Using Convex Hulls
- Combinatorial Techniques for MIP Formulations
- Other Computational Properties of MIP Formulations (e.g. branching)
- Nonlinear MIP formulation
- MIP Representability

General References



- Mixed integer linear programming formulation techniques. J. P. Vielma. SIAM Review 57, 2015. pp. 3-57.
- M. Conforti, G. Cornuejols, and G. Zambelli. Integer Programming. Graduate Texts in Mathematics. Springer, 2014.

Other References

- Advanced linear MIP formulations
  - Embedding Formulations and Complexity for Unions of Polyhedra. J. P. Vielma. To appear in Management Science, 2017.
  - A combinatorial approach for small and strong formulations of disjunctive constraints. J. Huchette and J. P. Vielma. Submitted for publication, 2016.
  - Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools. J. Huchette and J. P. Vielma. Submitted for publication, 2017.
  - A combinatorial approach for small and strong formulations of disjunctive constraints. J. Huchette and J. P. Vielma. Submitted for publication, 2016.

Other References

- Advanced Nonlinear MIP Formulations:
  - Small and strong formulations for unions of convex sets from the Cayley Embedding. J. P. Vielma. To appear in Mathematical Programming, 2018. DOI:10.1007/s10107-018-1258-4
- Nonlinear MIP representability
  - Mixed-integer convex representability. M. Lubin, I. Zadik and J. P. Vielma. In F. Eisenbrand and J. Könemann, editors, Proceedings of the 19th Conference on Integer Programming and Combinatorial Optimization (IPCO 2017), Lecture Notes in Computer Science 10328, 2017. pp. 392-404.
  - Regularity in mixed-integer convex representability. M. Lubin, I. Zadik and J. P. Vielma. Submitted for publication, 2017.

Example Application of MIP

Ellipsoidal Methods for Adaptive Choice-Based Conjoint Analysis

Denis Sauré
University of Chile, dsaure@dii.uchile.cl

Juan Pablo Vielma
MIT Sloan School of Management, jvielma@mit.edu

Adaptive Choice-Based Conjoint Analysis

Feature	SX530	RX100
Zoom	50x	3.6x
Prize	$249.99	$399.99
Weight	15.68 ounces	7.5 ounces
Prefer	☑	☐

Feature	TG-4	G9
Waterproof	Yes	No
Prize	$249.99	$399.99
Weight	7.36 lb	7.5 lb
Prefer	☐	☑

Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	$249.99	$399.99
Viewfinder	Electronic	Optical
Prefer	☑	☐

Estimate of preference **parameter**

- Today: Minimize **variance** of **parameter** estimates

Parametric Model = Logistic Regression

Feature	Chewbacca	BB-8
Wookiee	Yes	No
Droid	No	Yes
Blaster	Yes	No
Prefer?	☑	☐

$$x^1 \qquad x^2$$

Product profile

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = x^2$$

MNL Random **Linear** Utility

$$U_j = \underbrace{\beta \cdot x^j}_{\sum_{j=1}^{d} \beta_i x_i^j} + \epsilon_j$$

$$\longleftrightarrow \quad z = x^1 - x^2$$

Question:

$$x^1 \succ x^2 \Leftrightarrow \quad U_1 \text{ “}>\text{” } U_2$$

$$\Leftrightarrow \beta \cdot z \text{ “}>\text{” } 0$$

$$\boxed{\mathbb{P}\left(x^1 \succ x^2 \mid \beta\right) = \frac{1}{1 + e^{-\beta \cdot z}}}$$

Experimental Design for **Linear Regression**

Model: $y^i = \beta \cdot z^i + \epsilon_i, \quad \epsilon_i \sim N(0,1)$

Questions: Answers:

$Z = \left[z^1 | \ldots | z^q\right]^T \in \mathbb{R}^{q \times n}$ $\quad Y = \left[y_1 | \ldots | y_q\right]^T$

Minimize "variance" of estimator of $\beta \in \mathbb{R}^n$:

$$\max \left(\det \left(Z^T Z\right)\right)^{1/q}$$

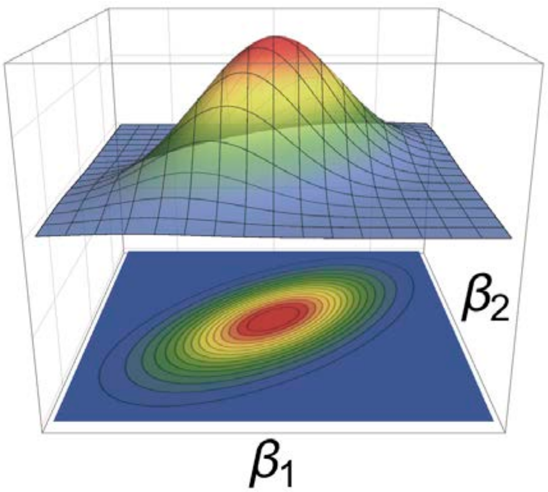
$$x^{1,i}, x^{2,i} \in \{0,1\}^n$$

$$z^i = x^{1,i} - x^{2,i} \in \{-1,0,1\}^n$$

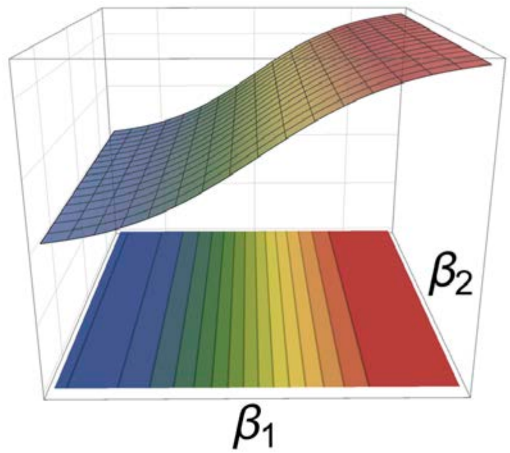
MINLP problem similar to minimum volume ellipsoid

Bayesian **Logistic Regression** with Normal Prior

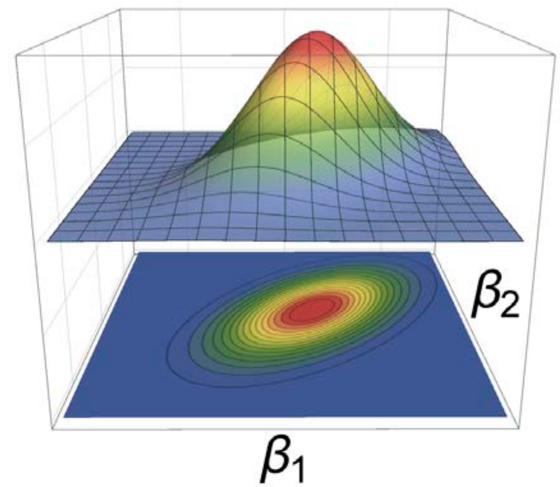
Prior distribution



Answer likelihood



Posterior distribution



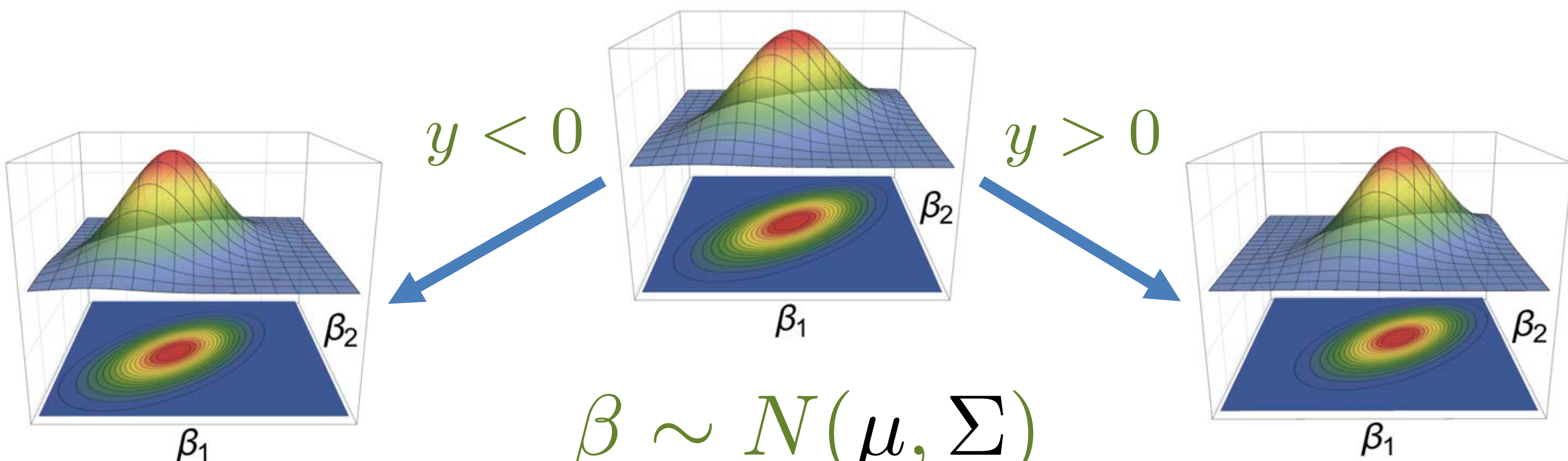
$$\beta \sim N(\mu, \Sigma) \qquad L\,(y \mid \beta, z) \qquad g\,(\beta \mid y, z\,)$$

$$y = \operatorname{sign}(\beta \cdot z) \qquad L\,(y \mid \beta, z) = \left(1 + e^{-y\beta\cdot z}\right)^{-1}$$

$$g\,(\beta \mid y, z\,) \propto \phi\,(\beta\,;\; \mu, \Sigma)\,L\,(y \mid \beta, z)$$

Expected Posterior Variance of Estimator

$$f(z,\mu,\Sigma) := \mathbb{E}_{y,\beta}\left\{\left(\det \operatorname{cov}(\beta \,|\, y, z)\right)^{1/m}\right\}$$



$$\beta \sim N(\mu, \Sigma)$$

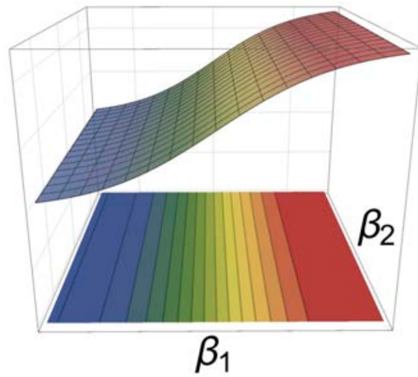
$\operatorname{cov}(\beta) = \Sigma_1$ $\qquad$ $\operatorname{cov}(\beta) = \Sigma_2$

$$\min_{z \in \{-1,0,1\}^n \setminus \{\mathbf{0}\}} f(z,\mu,\Sigma)$$

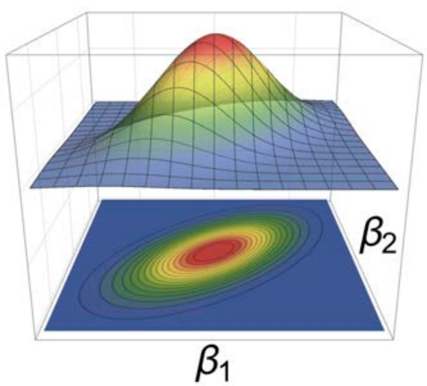
- $f(z,\mu,\Sigma)$ is hard to evaluate, non-convex and n large

1st Step: Moment-Matching Approximate Bayes

Answer likelihood



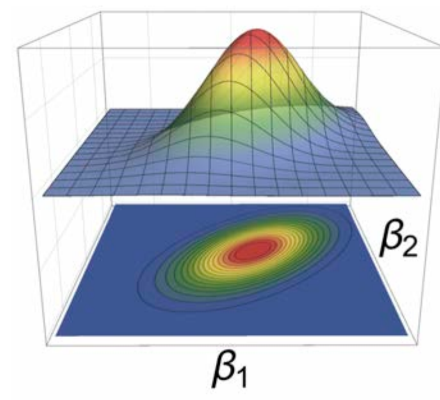
Prior distribution



$$\beta \sim N\left(\mu^{i}, \Sigma^{i}\right)$$

Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	$249.99	$399.99
Viewfinder	Electronic	Optical
Prefer	✓	□

Posterior distribution



$$\beta \overset{approx.}{\sim} N\left(\mu^{i+1}, \Sigma^{i+1}\right)$$

- $\mu^{i+1} = \mathbb{E}\left(\beta \,|\, y, x^1, x^2\right)$
- $\Sigma^{i+1} = \mathrm{cov}\left(\beta \,|\, y, x^1, x^2\right)$

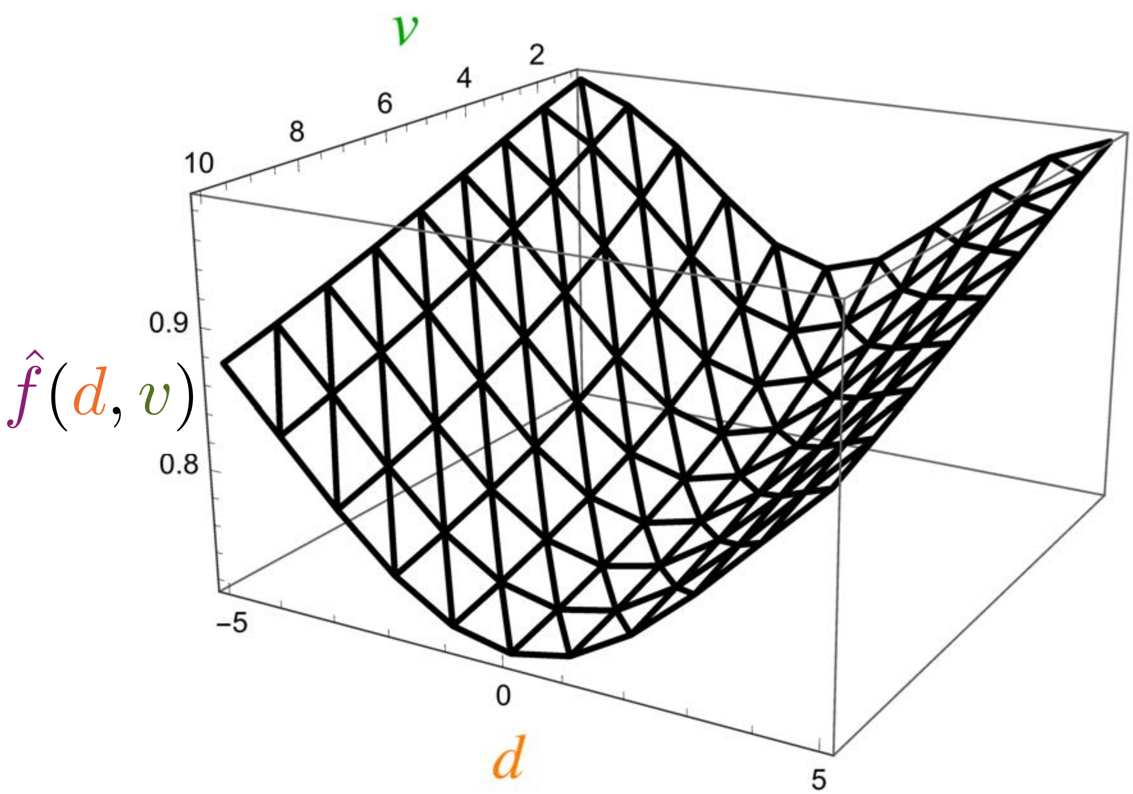
- Linear Algebra + 1-d numerical integration (e.g. BDA3)

Can Simplify with Some Linear Algebra

- D-efficiency $f(z)$ = Non-convex function $f(d, v)$of

$$\text{mean:} \quad d := \mu \cdot z$$

$$\text{variance:} \quad v := z' \cdot \Sigma \cdot z$$



Can evaluate $f(d, v)$ with 1-dim integral

Piecewise Linear (PWL) Interpolation $\hat{f}(d, v)$

“Almost” Direct Linear MIP Formulation

$$z = x^1 - x^2$$

MIP formulation for PWL function

$$\min \quad \boxed{\hat{f}(d, v)} \quad \checkmark$$

$$s.t.$$

$$\mu \cdot \left(x^1 - x^2\right) = d$$

$$\left(x^1 - x^2\right)' \cdot \Sigma \cdot \left(x^1 - x^2\right) = v \quad \checkmark$$

$$\text{linearize } x_i^k \cdot x_j^l \quad \left\|x^1 - x^2\right\|_2^2 \geq 1 \quad \left(x^1 \neq x^2\right)$$

$$x^1, x^2 \in \{0, 1\}^n$$

Step 1: Binary Quadratic $\quad x^1, x^2 \in \{0,1\}^n$

$$(x^1 - x^2)' \cdot \Sigma \cdot (x^1 - x^2) = v$$

$$X^l_{i,j} = x^l_i \cdot x^l_j \quad (l \in \{1,2\}, \quad i,j \in \{1,\ldots,n\}):$$

$$X^l_{i,j} \le x^l_i, \quad X^l_{i,j} \le x^l_j, \quad X^l_{i,j} \ge x^l_i + x^l_j - 1, \quad X^l_{i,j} \ge 0$$

$$W_{i,j} = x^1_i \cdot x^2_j :$$

$$W_{i,j} \le x^1_i, \quad W_{i,j} \le x^2_j, \quad W_{i,j} \ge x^1_i + x^2_j - 1, \quad W_{i,j} \ge 0$$

$$\sum_{i,j=1}^{n} \left(X^1_{i,j} + X^2_{i,j} - W_{i,j} - W_{j,i}\right) \Sigma_{i,j} = v$$

Step 1: Binary Quadratic $x^1, x^2 \in \{0,1\}^n$

$$x^1 \neq x^2 \quad \Leftrightarrow \quad \|x^1 - x^2\|_2^2 \geq 1$$

$$X^l_{i,j} = x^l_i \cdot x^l_j \quad (l \in \{1,2\}, \quad i,j \in \{1,\ldots,n\}):$$

$$X^l_{i,j} \leq x^l_i, \quad X^l_{i,j} \leq x^l_j, \quad X^l_{i,j} \geq x^l_i + x^l_j - 1, \quad X^l_{i,j} \geq 0$$

$$W_{i,j} = x^1_i \cdot x^2_j :$$

$$W_{i,j} \leq x^1_i, \quad W_{i,j} \leq x^2_j, \quad W_{i,j} \geq x^1_i + x^2_j - 1, \quad W_{i,j} \geq 0$$

$$\sum_{i,j=1}^{n} \left(X^1_{i,j} + X^2_{i,j} - W_{i,j} - W_{j,i}\right) \geq 1$$

Step 2: Advanced Formulations for PWL Functions

$$(x_{i,j}, z_{i,j}) = \sum_{k=1}^{N+1} v_{i,j}^k \lambda_k^{i,j} \qquad \forall i \in S, j \in D$$

$$\lambda_3^{i,j} + \lambda_4^{i,j} + 2\lambda_5^{i,j} + 2\lambda_6^{i,j} + 3\lambda_7^{i,j} + 3\lambda_8^{i,j} + 4\lambda_9^{i,j} \le y_1^{i,j} \qquad \forall i \in S, j \in D$$

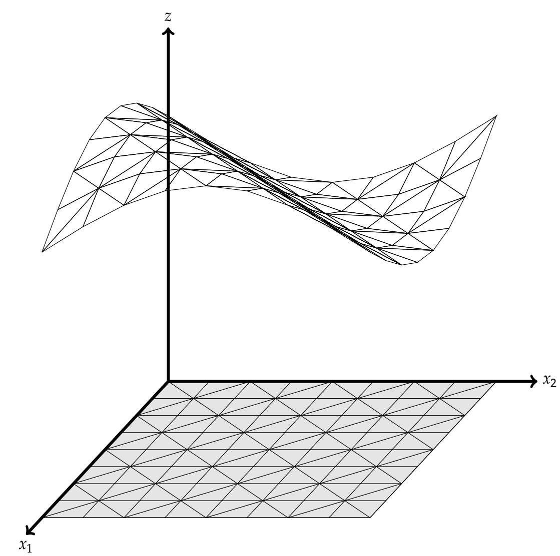
$$\lambda_2^{i,j} + \lambda_3^{i,j} + 2\lambda_4^{i,j} + 2\lambda_5^{i,j} + 3\lambda_6^{i,j} + 3\lambda_7^{i,j} + 4\lambda_8^{i,j} + 4\lambda_9^{i,j} \ge y_1^{i,j} \qquad \forall i \in S, j \in D$$

$$\lambda_4^{i,j} + \lambda_5^{i,j} + \lambda_6^{i,j} + \lambda_7^{i,j} + 2\lambda_8^{i,j} + 2\lambda_9^{i,j} \le y_2^{i,j} \qquad \forall i \in S, j \in D$$

$$\lambda_3^{i,j} + \lambda_4^{i,j} + \lambda_5^{i,j} + \lambda_6^{i,j} + 2\lambda_7^{i,j} + 2\lambda_8^{i,j} + 2\lambda_9^{i,j} \ge y_2^{i,j} \qquad \forall i \in S, j \in D$$

$$\lambda_6^{i,j} + \lambda_7^{i,j} + \lambda_8^{i,j} + \lambda_9^{i,j} \le y_3^{i,j} \le \lambda_5^{i,j} + \lambda_6^{i,j} + \lambda_7^{i,j} + \lambda_8^{i,j} + \lambda_9^{i,j} \qquad \forall i \in S, j \in D$$

$$(\lambda^{i,j}, y^{i,j}) \in \Delta^9 \times \{0,1,2,3,4\} \times \{0,1,2\} \times \{0,1\} \qquad \forall i \in S, j \in D$$



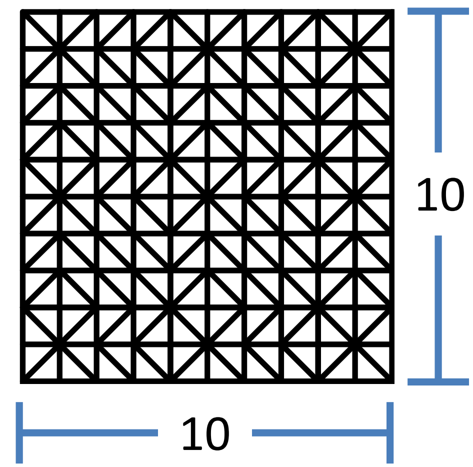
Easy to Build through julia & JuMP

- PiecewiseLinearOpt.jl (**Huchette** and V. 2017)

$$
\begin{aligned}
\min \quad & \exp(x+y) \\
s.t. \quad & \\
& x, y \in [0,1]
\end{aligned}
$$

Automatically select Δ

Automatically construct formulation (easily chosen)



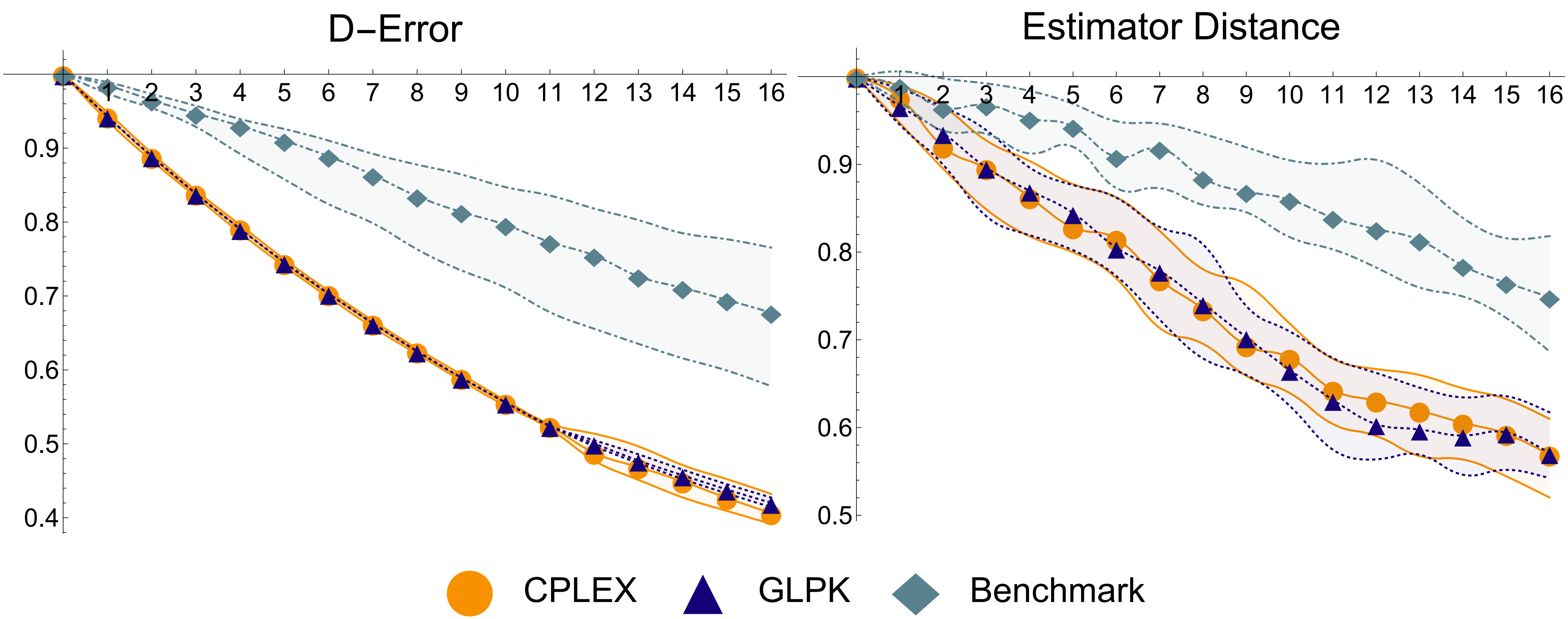

```julia
using JuMP, PiecewiseLinearOpt
m = Model()
@variable(m, x)
@variable(m, y)

z = piecewiselinear(m, x, y, 0:0.1:1, 0:0.1:1, (u,v) -> exp(u+v))
@objective(m, Min, z)
```

Easy to Build through julia & JuMP

```julia
function getquestion(μ,Σ,variancefuction)
  n = size(Σ,1)
  m = Model()
  # define variables for linearization
  @variable(m, 0 <= x[1:n] <= 1, Int)
  @variable(m, 0 <= y[1:n] <= 1, Int)
  # x ≠ y
  @constraint(m, linquad(m,(x-y)⋅(x-y)) >= 1)
  # v = x-y, β ~ 𝒩(μ,Σ), v⋅β ~ 𝒩(μᵥ,σ²), μᵥ = μ⋅v, σ² = v'*Σ*v
  @variable(m, μᵥ)
  @constraint(m, μᵥ == μ⋅(x-y) )
  @variable(m, σ² >=0)
  @constraint(m, σ² == linquad(m,(x-y)⋅(Σ*(x-y))))
  # (x-y)'*Σ*(x-y) <= eigmax(Σ) ||x-y||₂ <= eigmax(Σ)*n
  σ̄² = eigmax(Σ)*n
  # (x-y)'*Σ*(x-y) >= eigmin(Σ) ||x-y||₂ >= eigmin(Σ)   ( x ≠ y )
  σ̲² = eigmin(Σ)
  μ̄ᵥ = norm(μ,1)
  μᵥnpoints = 2^k - 1
  μᵥpoints = 0:μ̄ᵥ/μᵥnpoints:μ̄ᵥ+(μ̄ᵥ/μᵥnpoints)/2
  σ²range = σ̄² - σ̲²
  σ²npoints = 2^k-1
  σ²points = σ̲²:σ²range/σ²npoints:σ̄²+(σ²range/σ²npoints)/2
  pwl = BivariatePWLFunction(μᵥpoints, σ²points, (μᵥ,σ²) -> variancefuction(μᵥ,sqrt(σ²)))
  obj = piecewiselinear(m, μᵥ, σ², pwl)
  @objective(m, Min,  obj )
  status = solve(m)
  return [  round(Int64,getvalue(x)),  round(Int64,getvalue(y))]
end
```

MIP v/s Best Benchmark (Toubia et al. ‘03,’04)

- 16 questions, 2 options, 12 features, 100 individual β^* sampled from known prior N(μ,Σ)
- Best **benchmark** v/s MIP + Moment Matching
- CPLEX: ≤ 1 s (0.2 s Avg.), GLPK: ≤ 5 s (1.7 s Avg.)



Easy To Add Questionnaire Rules

Feature	Chewbacca	BB-8
Wookiee	Yes	No
Droid	No	Yes
Blaster	Yes	No
Prefer?	☑	☐

Product profile

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = x^2$$

- Realism is important: Wookiees are not Droids!

$$x^1_{\text{Wookie}} + x^1_{\text{Droid}} \leq 1$$

- Partial Profiles:
  - Limit # of feature differences and assume those not shown are the same (e.g. both are members of the resistance)

$$\left\| x^1 - x^2 \right\|_1 \leq 3$$

Full v/s Partial Profiles (# Feature Differences)

- 16 questions, 2 options, 12 features, 100 individual β^* sampled from known prior N(μ,Σ)
- Best **benchmark** v/s MIP + Moment Matching (CPLEX)
- Full: ≤ 1 s (0.2 s Avg.), Partial (5 diff.): ≤ 66 s (8 s Avg.)