Global and Robust Optimization for Engineering Design

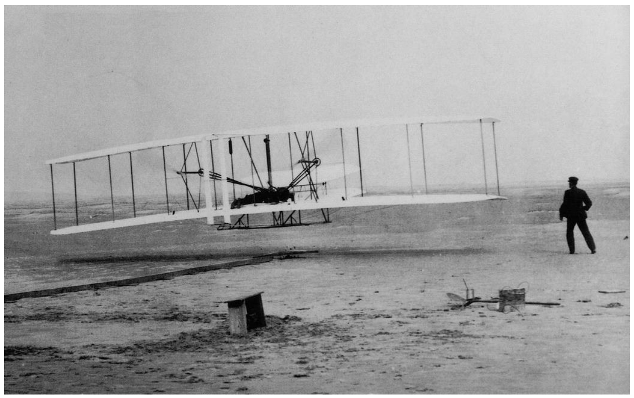
01/06/2022

Berk Öztürk

Committee Members:
Prof. Dimitris Bertsimas (Research Advisor)
Prof. Mark Drela (Committee Chair)
P.R.E. Robert Haimes

Thesis Readers:
Prof. Olivier De Weck
Dr. Jack Dunn

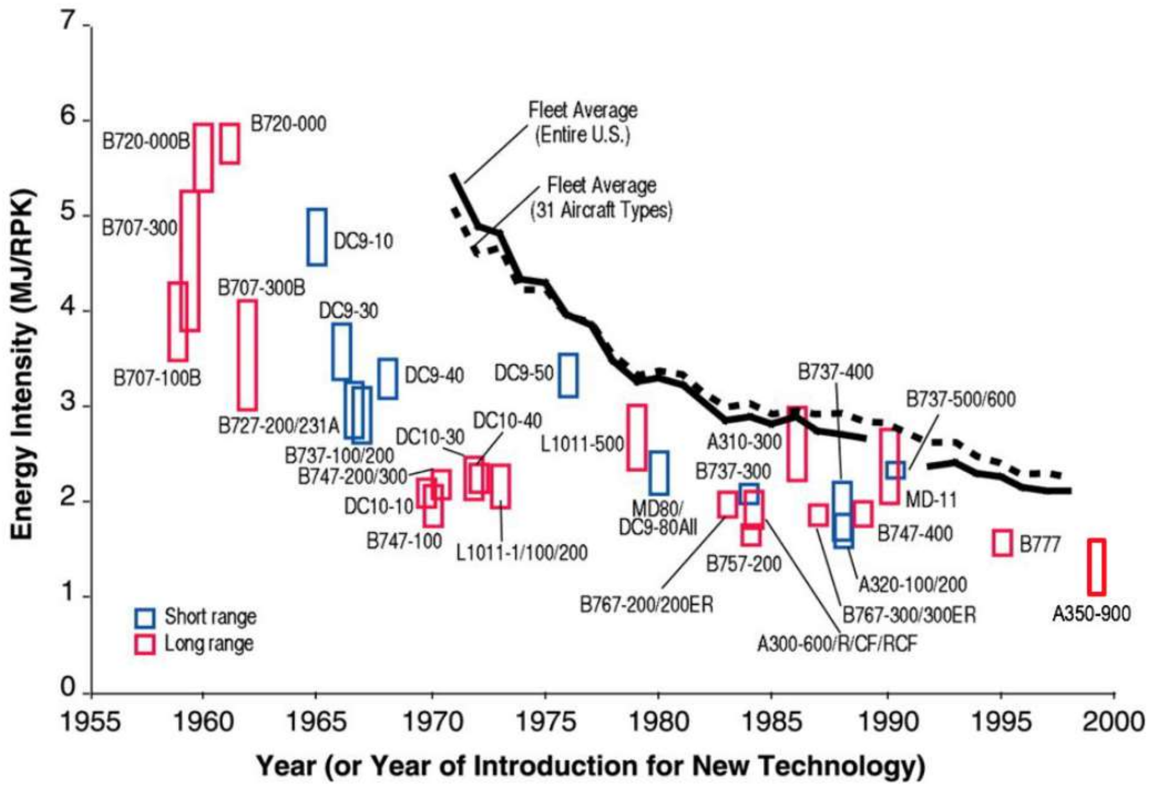
Aerospace engineering is only 120 years old, but been instrumental for expanding human frontiers.



Wright Flyer, 1903 - First heavier-than-air flight **on Earth** featuring **efficient, high-speed propellers** and **lightweight propulsion.**

Ingenuity, 2021- First heavier-than-air flight **on Mars** featuring **efficient, high-speed propellers** and **lightweight propulsion.**

Aerospace concepts are limited by the 2nd Law!



Commercial aerospace products have had a rate of improvement of 3.3% since 1950s.

For the configuration and technologies, it becomes harder to squeeze performance from each unit of energy.

US DOT, Bur. Transp. Stat., Off. Airl. Inf. 1999. Form 41 Schedule P-5.2 and Schedule T-2 for 1968–98. Washington, DC: Dep. Transp.

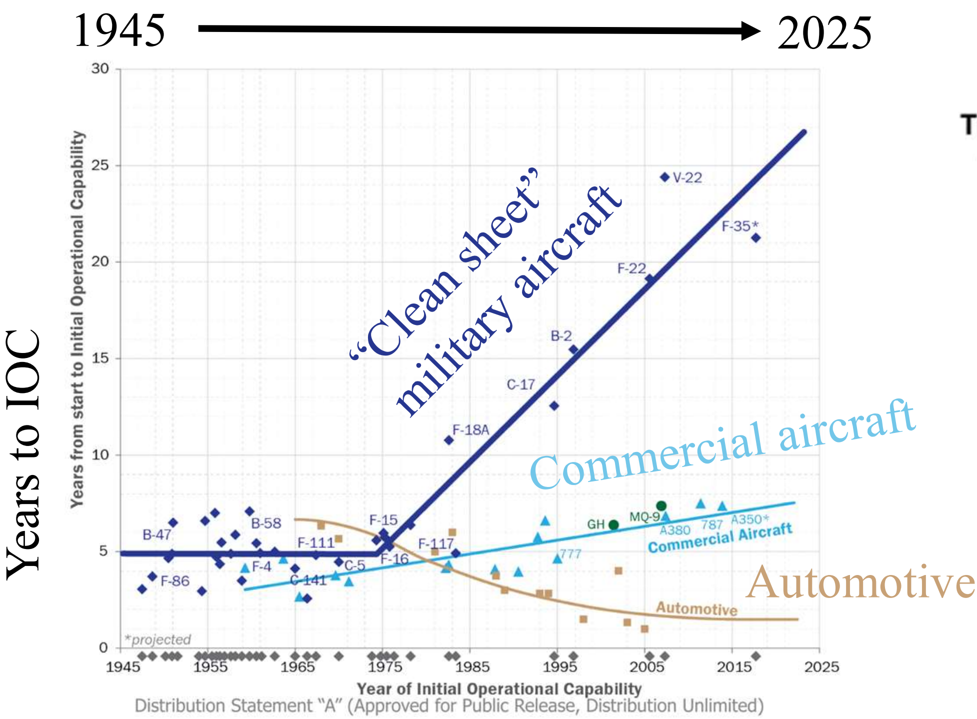
Design tools are critical for novel, increasingly complex concepts where experience is lacking.

Aurora D8

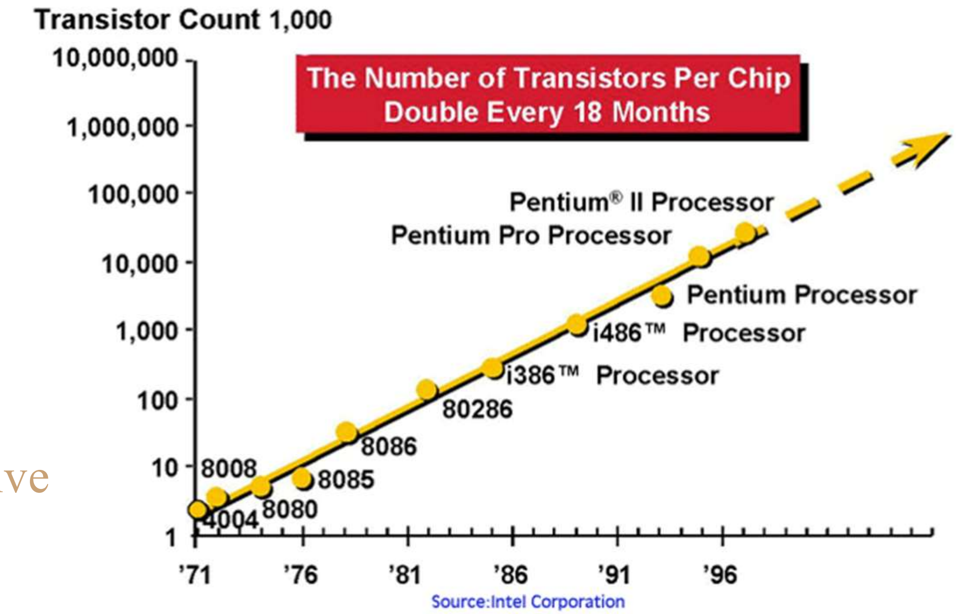
Virgin Hyperloop

Electra.aero

Improved computation has yet to meet challenges in design.



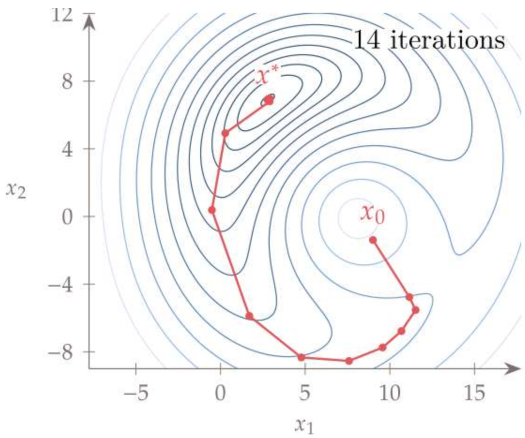
Year of initial operational capability (IOC)



Aerospace literature has primarily focused on:

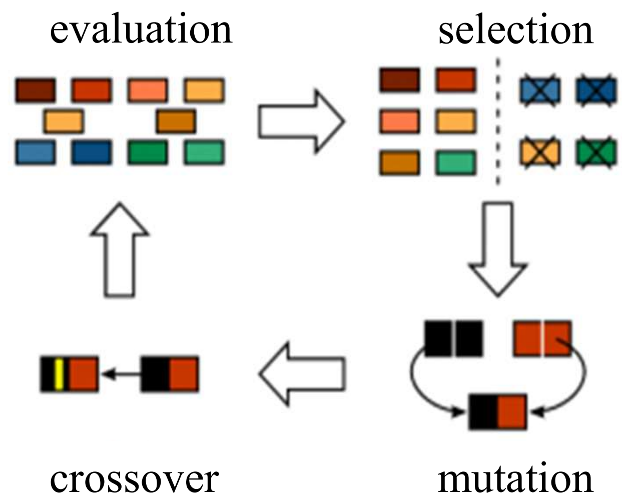
Gradient-based methods
(eg. Quasi-Newton descent)

$$\tilde{f}\left(x_k + p\right) = f_k + \nabla f_k{}^{\intercal} p + \frac{1}{2} p^{\intercal} \tilde{H}_k p \,,$$



- Limited to local optimization, often of a few disciplines or subsystems.
- Requires a good initial guess.

Heuristic methods
(eg. genetic algorithm)

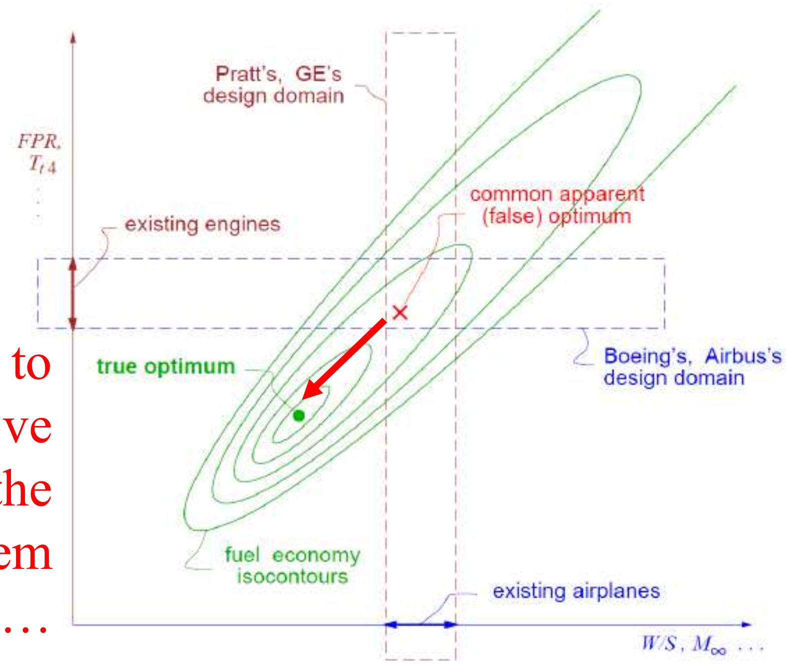


- Limited to low-dimensional optimization.
- No guarantees of optimality.

Martins, J. R. R. A., & Ning, A. (2021). *Engineering Design Optimization* (Issue September).

We need all-at-once optimization to improve the conceptual design process!

We want to move towards the system optimum…



[21] Mark Drela. Low-Order Modeling for Conceptual Aircraft Design and Development of the D8 Transport Concept. *Stanford AA295 Seminar*, pages 1–77, 2011.

Instead of having a push-pull design process…

Thesis theme: Address conceptual design using all-at-once mixed-integer linear and convex optimization.

Take advantage of

- Mathematical guarantees,
- Ability to make both discrete and continuous decisions,
- Speed and scalability,

but also leverage new literature in

- Machine learning (ML): making predictions and prescriptions from data,
- Robust optimization (RO): decision making under uncertainty sets,

to improve the ability of tools to address complexities of new concepts.

My thesis addresses two specific opportunities in aerospace design using linear and convex optimization.

Ch. 2:
Optimization over constraints and objectives with arbitrary mathematical primitives, using machine learning (ML) and mixed-integer optimization (MIO).

Global Optimization via Optimal Decision Trees
(30 mins)

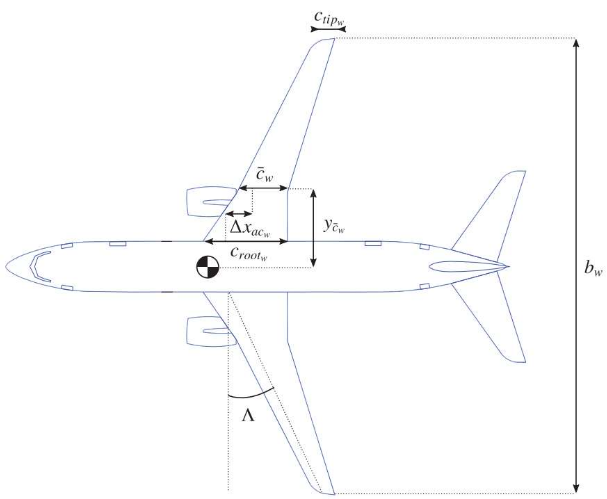
Ch. 3:
Design optimization while protecting against uncertainty in a tractable, deterministic manner, using robust optimization (RO).

Engineering Design Decisions Under Uncertainty via RSPs
(10 mins)

Global Optimization via Optimal Decision Trees

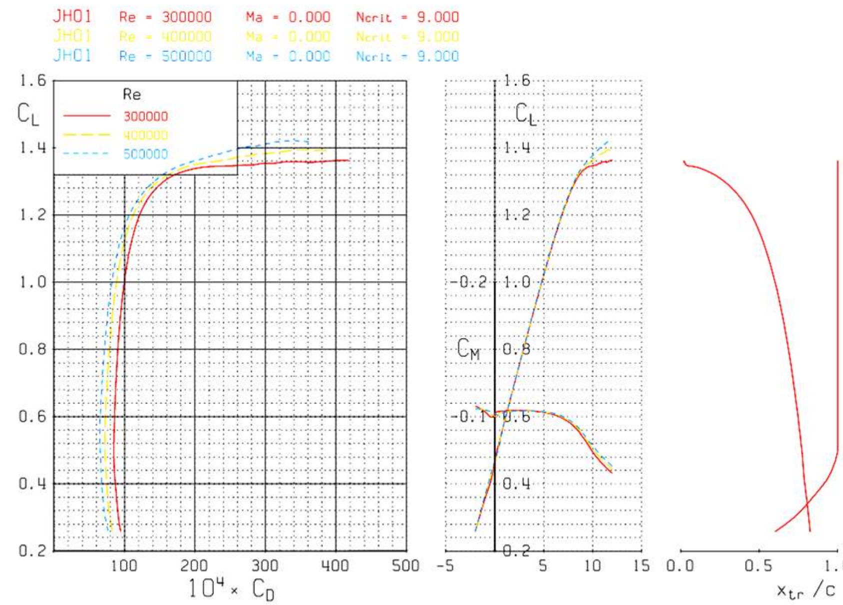
Constraints in aerospace engineering come in many forms.

Explicit



$$M_r c_{\text{root}} \geq \Big(L_{\text{wing,max}} - N_{\text{lift}} (W_{\text{wing}} + f_{\text{fuel,wing}} W_{\text{fuel,total}}) \Big) \times \left(\frac{b_w^2}{12 S_w} (c_{\text{root}} + c_{\text{tip}}) \right) - N_{\text{lift}} W_{\text{engine}} y_{\text{engine}}$$

Inexplicit



Existing global optimizers are not sufficiently general to address design problems.

CONOPT: A generalized reduced gradient method, assumes
- Continuous variables,
- Smooth, well-scaled constraints and first derivatives.

IPOPT: An interior point method, assumes
- Continuous variables,
- Twice-continuously-differentiable constraints and objective.

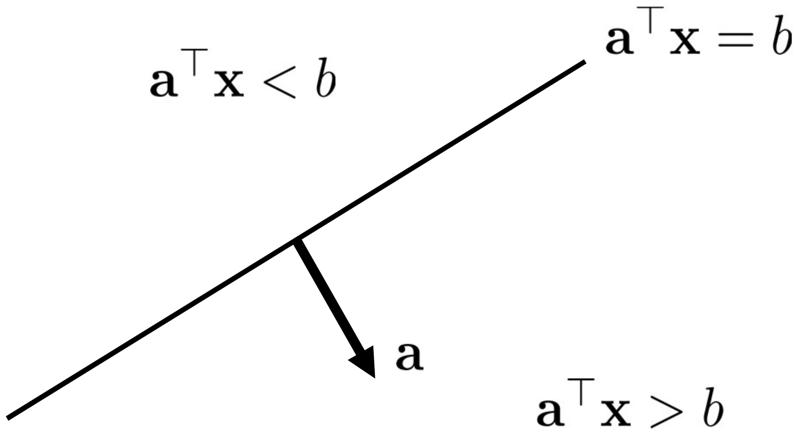
BARON: A branch-and-reduce MIO method that assumes
- Allowable primitives "exp(x), ln(x), x^{α} for real α, and β^{x} for real β".

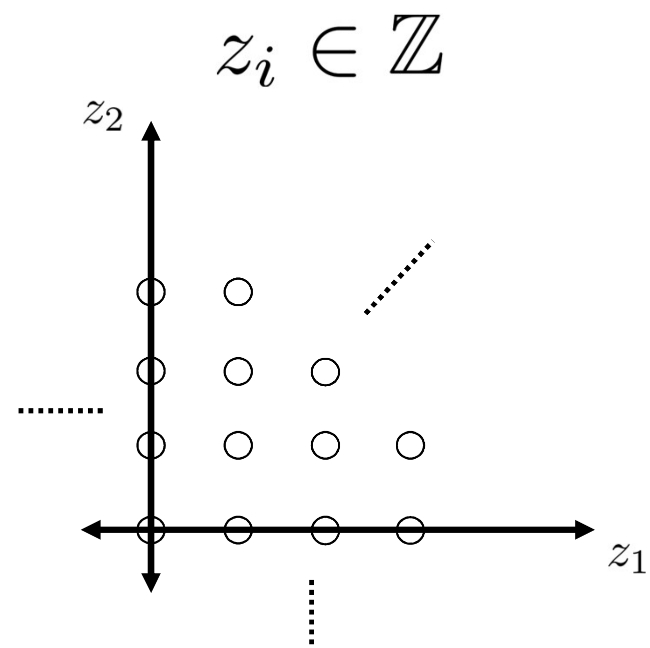
$$
\begin{aligned}
\min_{x} \quad & f(\mathbf{x}) \\
\text{s.t.} \quad & g_i(\mathbf{x}) \geq 0, \;\; i \in I, \\
& h_j(\mathbf{x}) = 0, \;\; j \in J, \\
& \mathbf{A}\mathbf{x} \geq \mathbf{b}, \;\; \mathbf{C}\mathbf{x} = \mathbf{d}, \\
& x_k \in [\underline{x}_k, \overline{x}_k], \;\; k \in [n].
\end{aligned}
$$

Proposed approach: Optimize over objectives and constraints with **arbitrary mathematical primitives** by learning MIO-compatible approximations of f, g_i and h_j.

Only requirements are **bounded domain** for decision variables in difficult constraints.

MIO-compatible approximation can be described by a combination of <u>separating hyperplanes</u> and <u>integer variables</u>.





Parameter learning [PL] (i.e. fitting) has received attention, but has limits.

- Similarity: PL methods assume some underlying structure of the data (e.g. convex polynomial-ness, piecewise linearity, or convexity).
- Difference: The methods consider fitting without consideration for data generation.

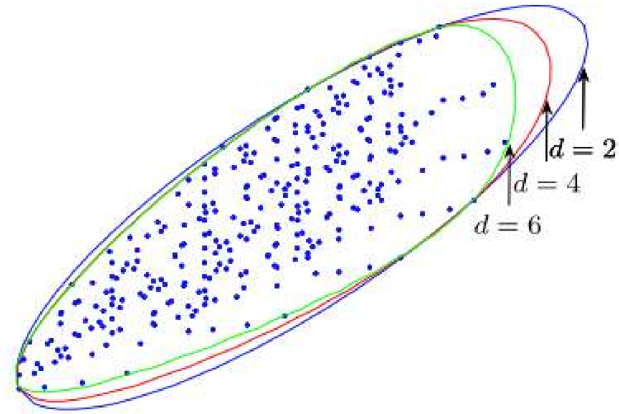
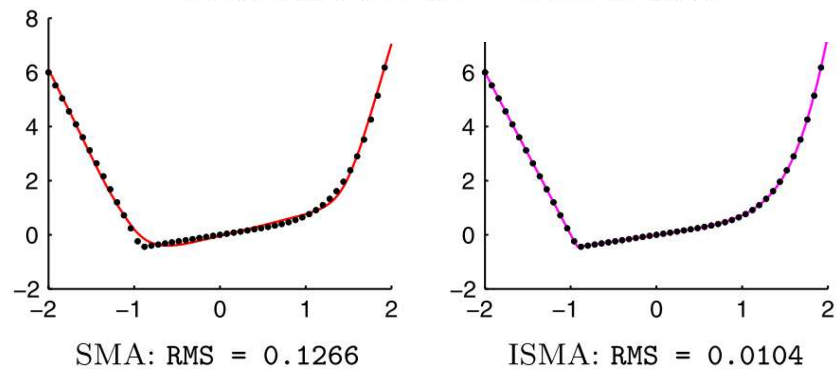


Fig. 2: Pseudo minimum volume example.



1. Magnani, A., Lall, S., & Boyd, S. (2005). Tractable fitting with convex polynomials via sum-of-squares. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, *2005*, 1672–1677.
2. Hoburg, W., Kirschen, P., & Abbeel, P. (2016). Data fitting with geometric-programming-compatible softmax functions. *Optimization and Engineering*, *17*(4), 897–918.

Constraint learning expands the scope of PL to arbitrary constraints, models or data.

In ML jargon: Can we sample data for and train MIO-compatible binary classifiers to approximate difficult constraints?

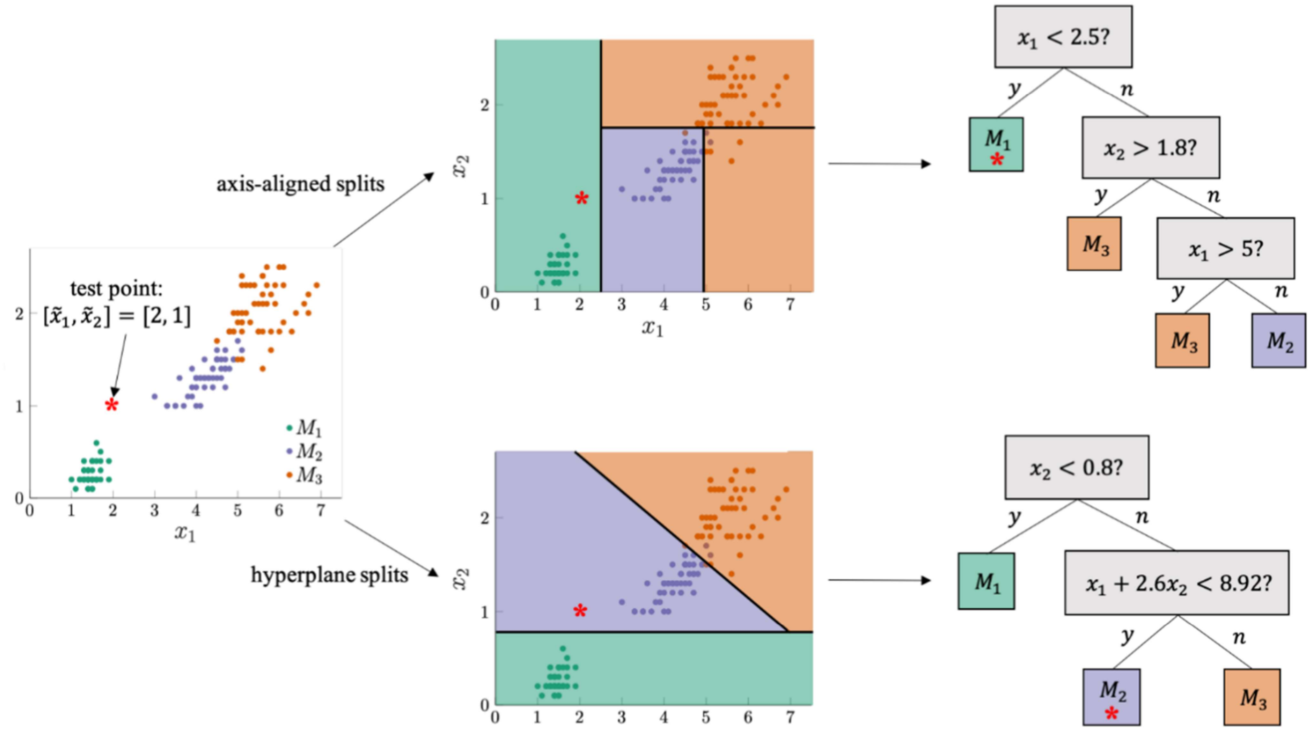
If so, confirming whether $\mathbf{x}$ satisfies $g_i(\mathbf{x}) \geq 0$ would be equivalent to querying the trained classifier M.

$$\mathbb{I}(g_i(\mathbf{x}) \geq 0) \leftrightarrow M(\mathbf{x})$$

More importantly, if classifier is *MIO-compatible*, then we could use the classifier to optimize approximately over functions and/or data.

$$g_i(\mathbf{x}) \geq 0 \leftrightarrow \mathbf{x} \in \{\mathbf{X} : M(\mathbf{X}) = 1\}$$

Decision trees are MIO-compatible nonlinear classifiers.



Optimal Classification Tree (OCT)

Optimal Classification Tree with Hyperplanes (OCT-H)

OCTs/ORTs are superior to other ML methods.

- **Tunable:** Depth and sparsity can be adjusted. Training time/optimality can be traded off as well for dynamic applications.
- **Accurate:** Achieve low misclassification/MSE error without overfitting.
- **Interpretable:** Each split of a tree defines an easy-to-apply decision rule.
- **MIO compatible:** Predictions can be represented using linear constraints and binary variables!

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning*, *106*(7), 1039–1082.

Contributions

1. Ensemble of methods for sampling constraints for learning.
2. Learning of nonlinear constraints using decision trees (DTs).
3. A representation of DTs that is compatible with mixed-integer optimization.
4. A projected gradient descent method to check and repair near-optimal, near-feasible solutions from the approximations.
5. Application of DT-driven global optimization to a set of benchmark and real world problems.

Method and Demonstrative Example

OCT-HaGOn [OCT-H for Global Optimization] takes the following steps.

1. Generates standard form problem.
2. Samples and evaluates nonlinear constraints.
3. Trains DTs over constraint data.
4. Generates MIO representations of DTs.
5. Solves MIO approximation.
6. Checks and repairs solution.

Consider the following (modified) mixed-integer nonlinear problem from Duran and Grossmann, 1986.

$$
\begin{aligned}
\min\ & f(\mathbf{x}) = 10x_1 - 17x_3 - 5x_4 + 6x_5 + 8x_6 \\
\text{s.t. } & g_1(\mathbf{x}) = 0.8\log(\mathrm{x}_2 + 1) + 0.96\log(\mathrm{x}_1 - \mathrm{x}_2 + 1) - 0.8\mathrm{x}_3 \geq 0, \\
& g_2(\mathbf{x}) = \log(\mathrm{x}_2 + 1) + 1.2\log(\mathrm{x}_1 - \mathrm{x}_2 + 1) - \mathrm{x}_3 - 2\mathrm{x}_6 + 2 \geq 0, \\
& x_1 - x_2 \geq 0, \quad 2x_4 - x_2 \geq 0, \\
& 2x_5 - x_1 + x_2 \geq 0, \quad 1 - x_4 - x_5 \geq 0, \\
& 0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 2,\ 0 \leq x_3 \leq 1, \\
& x_4, x_5, x_6 \in \{0, 1\}^3.
\end{aligned}
$$

Duran, M. A., & Grossmann, I. E. (1986). An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming*, *36*, 307–339.

1. It generates a standard form problem, separating the "easy" and "difficult" constraints.

Objective function	$\min_{x} \; f(\mathbf{x})$
Difficult constraints	$\text{s.t.} \; g_i(\mathbf{x}) \geq 0, \; i \in I,$ $h_j(\mathbf{x}) = 0, \; j \in J,$
Easy constraints	$\mathbf{Ax} \geq \mathbf{b}, \; \mathbf{Cx} = \mathbf{d},$
Variables	$x_k \in [\underline{x}_k, \overline{x}_k], \; k \in [n].$

1. Most global optimization problems are in this standard form by construction.

Objective function

$$\min \ f(\mathbf{x}) = 10x_1 - 17x_3 - 5x_4 + 6x_5 + 8x_6$$

Difficult constraints

$$\text{s.t. } g_1(\mathbf{x}) = 0.8\log(\mathrm{x}_2 + 1) + 0.96\log(\mathrm{x}_1 - \mathrm{x}_2 + 1) - 0.8\mathrm{x}_3 \geq 0,$$

$$g_2(\mathbf{x}) = \log(\mathrm{x}_2 + 1) + 1.2\log(\mathrm{x}_1 - \mathrm{x}_2 + 1) - \mathrm{x}_3 - 2\mathrm{x}_6 + 2 \geq 0,$$

Easy constraints

$$x_1 - x_2 \geq 0, \quad 2x_4 - x_2 \geq 0,$$

$$2x_5 - x_1 + x_2 \geq 0, \quad 1 - x_4 - x_5 \geq 0,$$

Variables

$$0 \leq x_1 \leq 2, \ 0 \leq x_2 \leq 2, \ 0 \leq x_3 \leq 1,$$

$$x_4, x_5, x_6 \in \{0, 1\}^3.$$

2. It generates samples efficiently for each difficult constraint over dom(**x**), in two steps.

1. **Optimal Latin Hypercubes (OLH)**

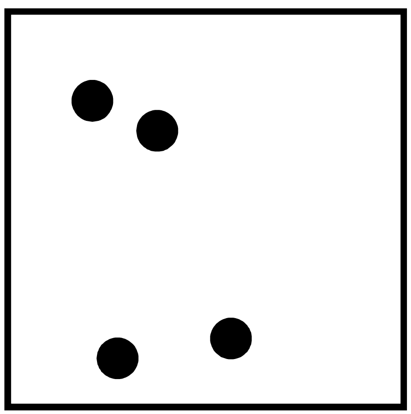
Generating space-filling samples for accuracy over the whole domain of **x**, using an off-the-shelf package.

2. **k-Nearest Neighbors (kNN) Sampling**

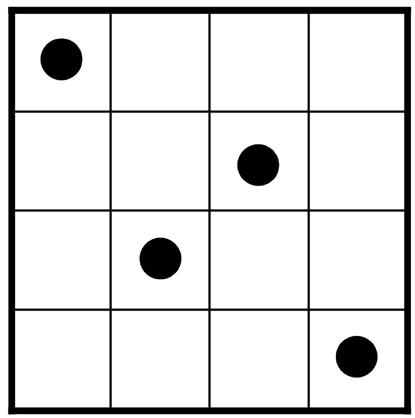
Sampling near the constraint boundary for local accuracy, using a new algorithm.

Bates, S. J., Sienz, J., & Toropov, V. V. (2004). Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm. *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, *7*(April 2004), 5217–5223.

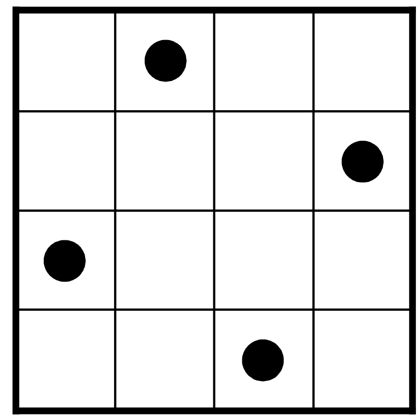
2. Optimal Latin Hypercubes is space-filling, to ensure good global approximation accuracy.



Randomization

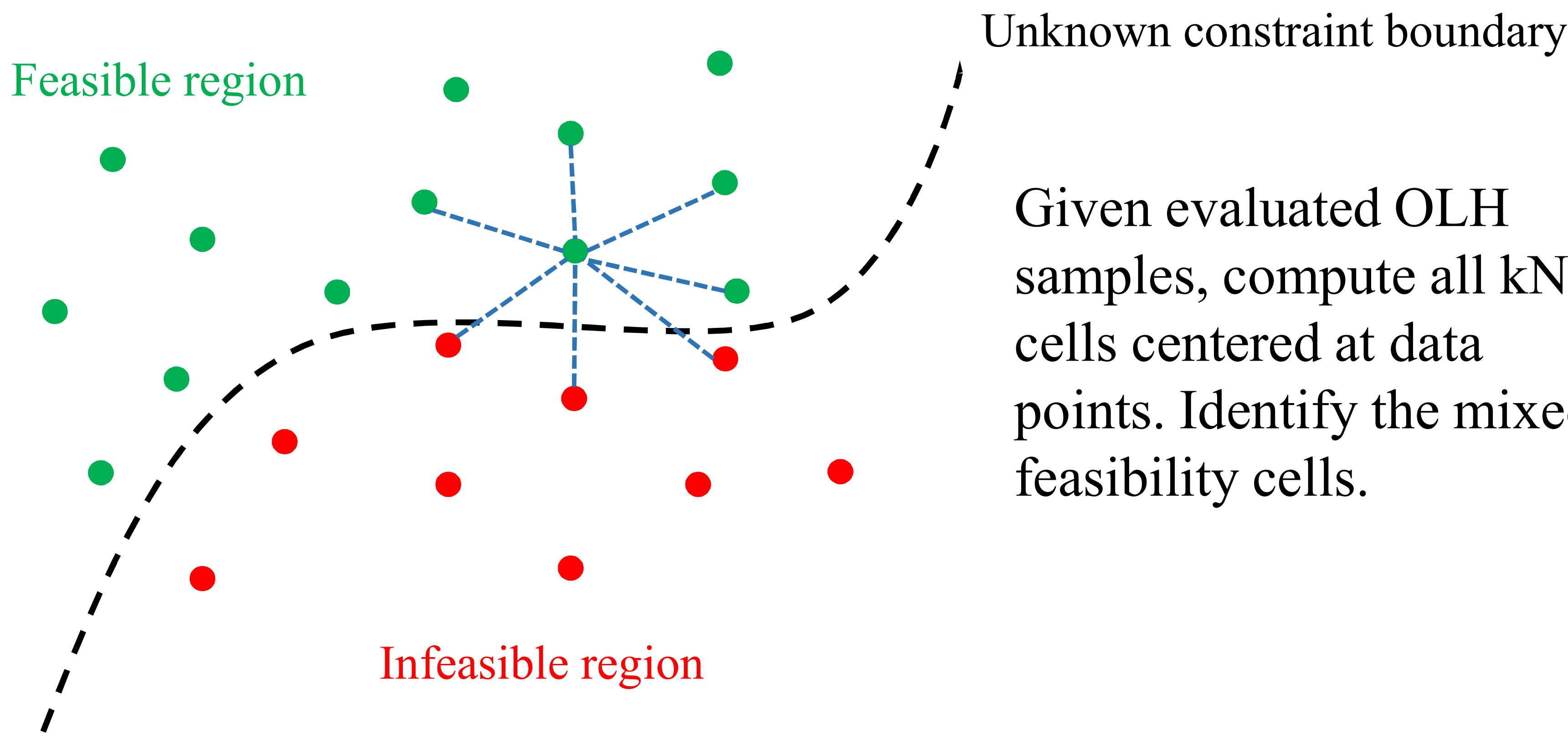


Latin Hypercubes



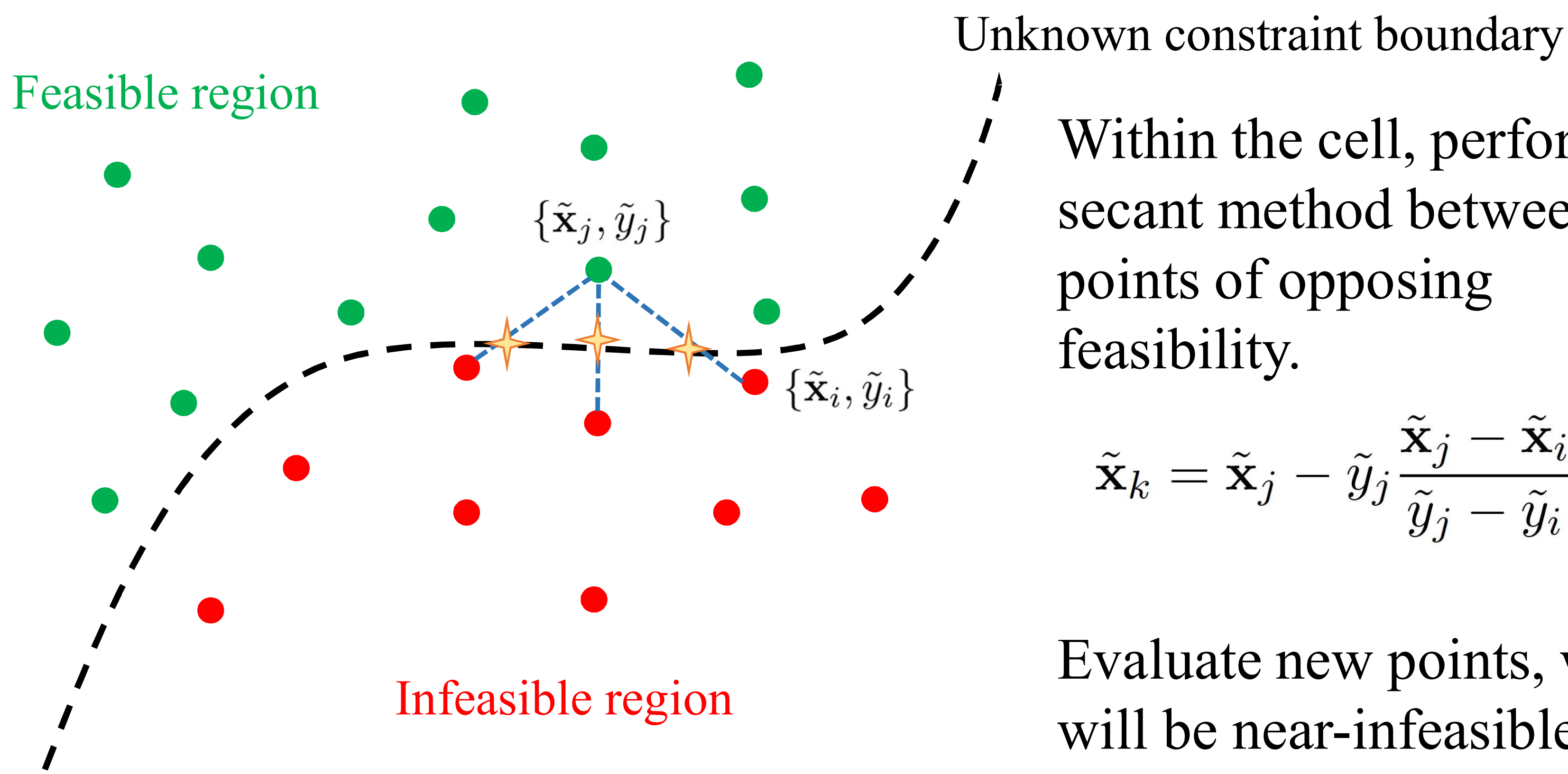
Optimal Latin Hypercubes

2. kNN is an ML algorithm that can be used for sampling x-domains of interest.



Given evaluated OLH samples, compute all kNN cells centered at data points. Identify the mixed-feasibility cells.

2. Root finding ($g_i(\tilde{\boldsymbol{x}}) = 0$) is approximated via the secant method.

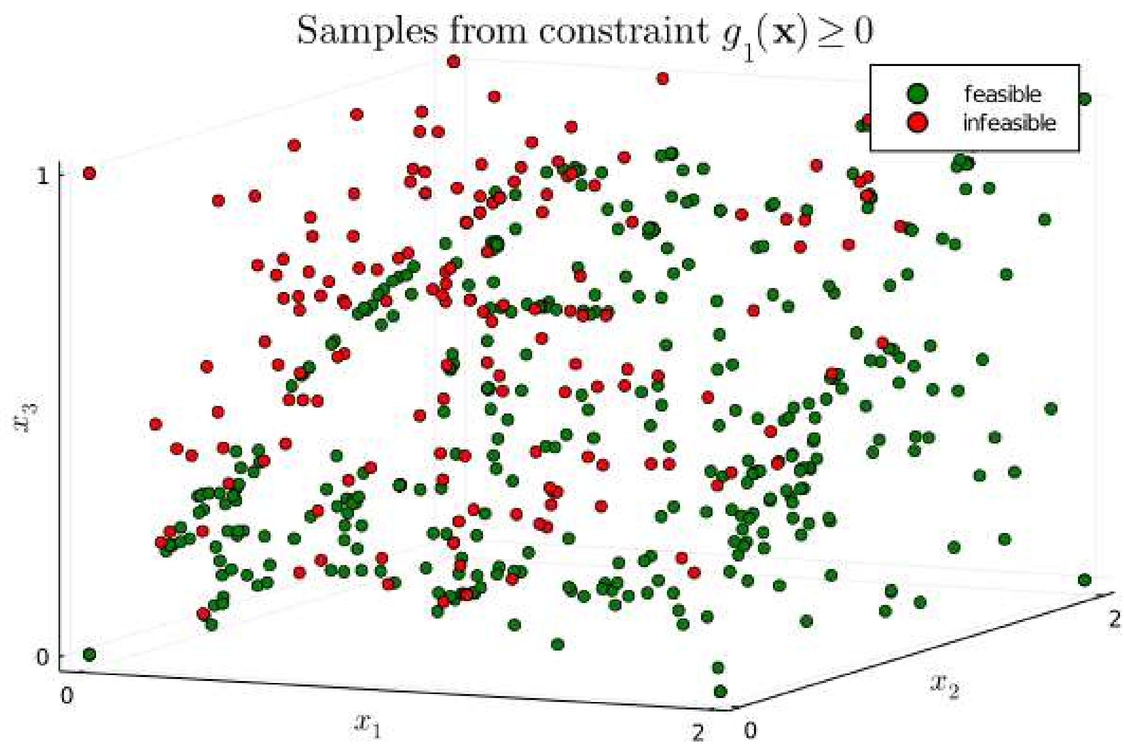


Within the cell, perform secant method between points of opposing feasibility.

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_j - \tilde{y}_j \frac{\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i}{\tilde{y}_j - \tilde{y}_i}$$

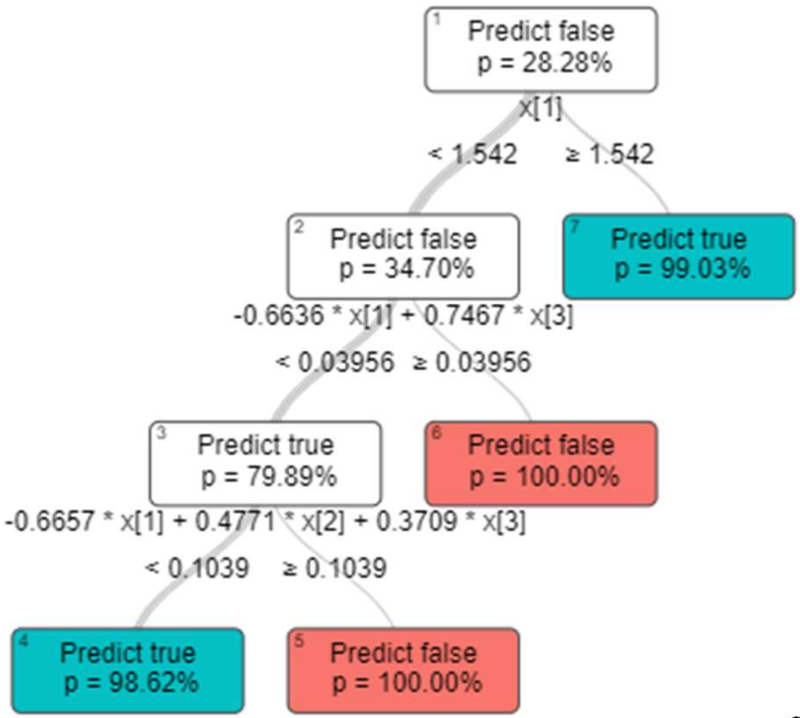
Evaluate new points, which will be near-infeasible.

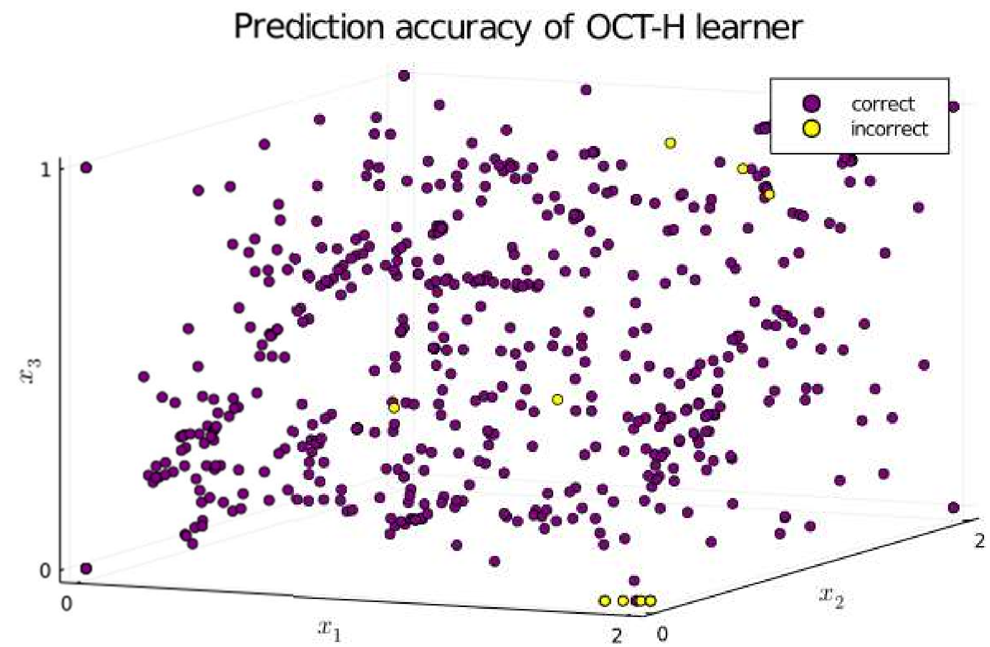
2. The points for $g_1(\boldsymbol{x}) \geq 0$ are given below.



Distribution of points are not obviously beneficial, but improve both the global and local accuracy of approximation.

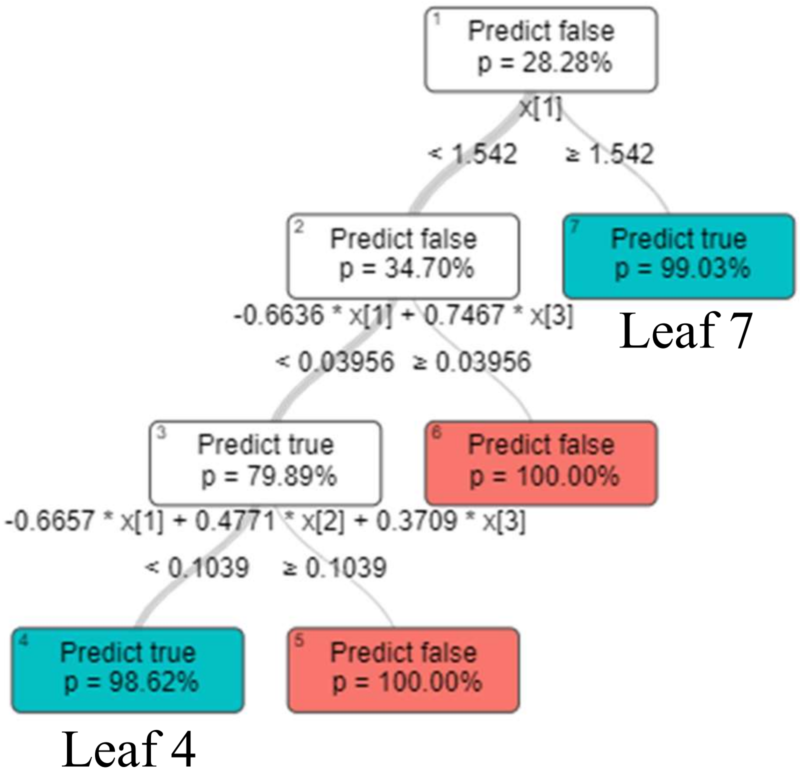
3. OCT-HaGOn trains decision trees over feasibility data, using Interpretable AI.





~99% accuracy in 600 samples over dom($\mathbf{x}$).

4. g_1 has the following big-M free, locally ideal[1] MIO representation.

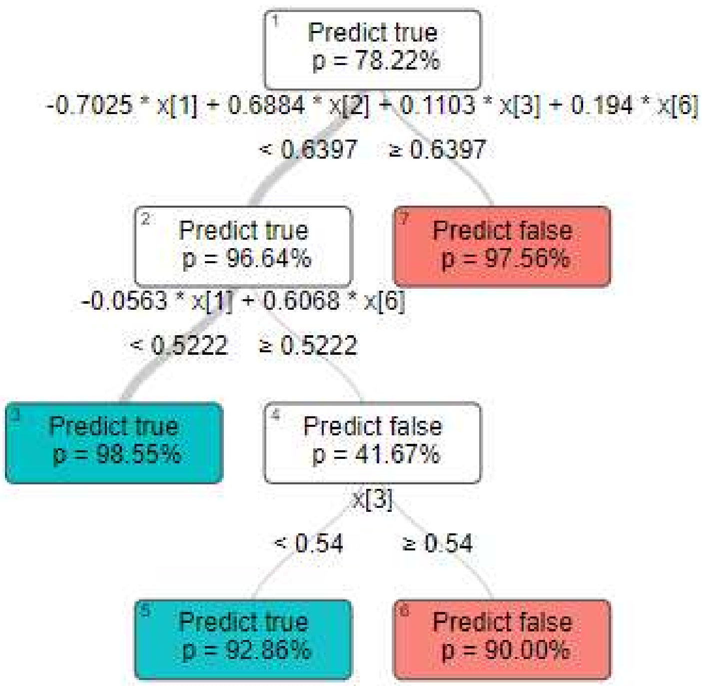


$$
\begin{aligned}
[1, 0, 0] \cdot \mathbf{y}_{1,7} &\geq 1.542 z_{1,7}, \\
[1, 0, 0] \cdot \mathbf{y}_{1,4} &\leq 1.542 z_{1,4}, \\
[-0.6636, 0, 0.7467] \cdot \mathbf{y}_{1,4} &\leq 0.03956 z_{1,4}, \\
[-0.6657, 0.4771, 0.3709] \cdot \mathbf{y}_{1,4} &\leq 0.1039 z_{1,4}, \\
\mathbf{y}_{1,4} + \mathbf{y}_{1,7} = [x_1, x_2, x_3],\ & z_{1,4} + z_{1,4} = 1, \\
[0, 0, 0] z_{1,4} \leq \mathbf{y}_{1,4} &\leq [2, 2, 1] z_{1,4}, \\
[0, 0, 0] z_{1,7} \leq \mathbf{y}_{1,7} &\leq [2, 2, 1] z_{1,7}, \\
z_{1,4}, z_{1,7} &\in \{0, 1\}.
\end{aligned}
$$

Book-keeping note: Auxiliary variables have constraint and leaf indices, in this case constraint 1 and leaves 4 and 7.

1. Vielma, J. P. (2015). Mixed Integer Linear Programming Formulation Techniques. *SIAM Review*, *57*(1), 3–57.

4. g_2 is similarly approximated.



$$
\begin{aligned}
[-0.7025, 0.6884, 0.1103, 0.194] \cdot \mathbf{y}_{2,3} &\leq 0.6397 z_{2,3}, \\
[-0.0563, 0, 0, 0.6068] \cdot \mathbf{y}_{2,3} &\leq 0.5222 z_{2,3}, \\
[-0.7025, 0.6884, 0.1103, 0.194] \cdot \mathbf{y}_{2,5} &\leq 0.6397 z_{2,5}, \\
[-0.0563, 0, 0, 0.6068] \cdot \mathbf{y}_{2,5} &\geq 0.5222 z_{2,5}, \\
[0, 0, 1, 0] \cdot \mathbf{y}_{2,5} &\leq 0.54 z_{2,5}, \\
\mathbf{y}_{2,3} + \mathbf{y}_{2,5} = [x_1, x_2, x_3, x_6], &\ \ z_{2,3} + z_{2,5} = 1, \\
[0,0,0,0] z_{2,3} \leq \mathbf{y}_{2,3} &\leq [2,2,1,1] z_{2,3}, \\
[0,0,0,0] z_{2,5} \leq \mathbf{y}_{2,5} &\leq [2,2,1,1] z_{2,5}, \\
z_{2,3}, z_{2,5} &\in \{0,1\}.
\end{aligned}
$$

$$\min \; f(\mathbf{x}) = 10x_1 - 17x_3 - 5x_4 + 6x_5 + 8x_6$$

$$\text{s.t. } g_1(\mathbf{x}) = 0.8\log(x_2 + 1) + 0.96\log(x_1 - x_2 + 1) - 0.8x_3 \geq 0,$$

$$\begin{aligned}
[1, 0, 0] \cdot \mathbf{y}_{1,7} &\geq 1.542 z_{1,7}, \\
[1, 0, 0] \cdot \mathbf{y}_{1,4} &\leq 1.542 z_{1,4}, \\
[-0.6636, 0, 0.7467] \cdot \mathbf{y}_{1,4} &\leq 0.03956 z_{1,4}, \\
[-0.6657, 0.4771, 0.3709] \cdot \mathbf{y}_{1,4} &\leq 0.1039 z_{1,4}, \\
\mathbf{y}_{1,4} + \mathbf{y}_{1,7} = [x_1, x_2, x_3], &\;\; z_{1,4} + z_{1,4} = 1, \\
[0, 0, 0] z_{1,4} \leq \mathbf{y}_{1,4} &\leq [2, 2, 1] z_{1,4}, \\
[0, 0, 0] z_{1,7} \leq \mathbf{y}_{1,7} &\leq [2, 2, 1] z_{1,7}, \\
z_{1,4}, z_{1,7} &\in \{0, 1\}.
\end{aligned}$$

$$g_2(\mathbf{x}) = \log(x_2 + 1) + 1.2\log(x_1 - x_2 + 1) - x_3 - 2x_6 + 2 \geq 0,$$

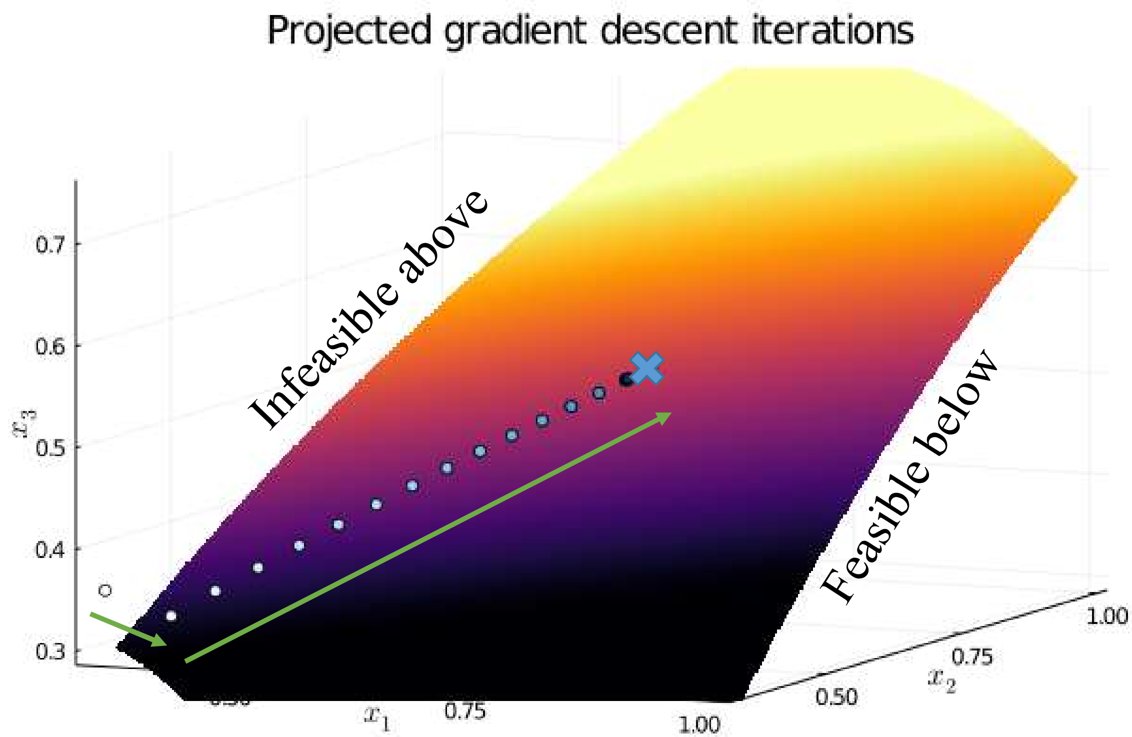
$$\begin{aligned}
[-0.7025, 0.6884, 0.1103, 0.194] \cdot \mathbf{y}_{2,3} &\leq 0.6397 z_{2,3}, \\
[-0.0563, 0, 0, 0.6068] \cdot \mathbf{y}_{2,3} &\leq 0.5222 z_{2,3}, \\
[-0.7025, 0.6884, 0.1103, 0.194] \cdot \mathbf{y}_{2,5} &\leq 0.6397 z_{2,5}, \\
[-0.0563, 0, 0, 0.6068] \cdot \mathbf{y}_{2,5} &\geq 0.5222 z_{2,5}, \\
[0, 0, 1, 0] \cdot \mathbf{y}_{2,5} &\leq 0.54 z_{2,5}, \\
\mathbf{y}_{2,3} + \mathbf{y}_{2,5} = [x_1, x_2, x_3, x_6], &\;\; z_{2,3} + z_{2,5} = 1, \\
[0, 0, 0, 0] z_{2,3} \leq \mathbf{y}_{2,3} &\leq [2, 2, 1, 1] z_{2,3}, \\
[0, 0, 0, 0] z_{2,5} \leq \mathbf{y}_{2,5} &\leq [2, 2, 1, 1] z_{2,5}, \\
z_{2,3}, z_{2,5} &\in \{0, 1\}.
\end{aligned}$$

$$\begin{aligned}
& x_1 - x_2 \geq 0, \quad 2x_4 - x_2 \geq 0, \\
& 2x_5 - x_1 + x_2 \geq 0, \quad 1 - x_4 - x_5 \geq 0, \\
& 0 \leq x_1 \leq 2, \; 0 \leq x_2 \leq 2, \; 0 \leq x_3 \leq 1, \\
& x_4, x_5, x_6 \in \{0, 1\}^3.
\end{aligned}$$

5. OCT-HaGOn solves the mixed-integer linear approximation to find a near-feasible, near-optimal solution, **x***.

6. It checks and repairs the solution using a projected gradient descent method.

- The MI approx. is replaced by an auto-differentiated local gradient.
- It takes steps (i.e. solves quadratic optimization problems) that restore feasibility as well as descend the objective.
- Example projection and descent steps on the right (shown on of g_1).



Benchmark problems

Success in small benchmarks gives confidence in the method.

Problem Name	Continuous Variables	Integer Variables	Linear Constraints	Nonlinear Inequalities	Nonlinear Equalities	Nonlinear Objective
`minlp`	3	1	4	2	0	Y
`pool1`	7	0	2	4	0	N
`nlp1`	2	0	0	1	0	N
`nlp2`	3	0	0	0	3	N
`nlp3`	10	0	3	1	3	Y

Problem name	Objective		Time (s)		Solution	
	BARON	OCT-HaGOn	BARON	OCT-HaGOn	BARON	OCT-HaGOn
`minlp`	6.0098	6.0098	0.120	29.9	[0,1,0,1.3,0,1]	[0,1,0,1.3,0,1]
`pool1`	23.0	23.0	0.082	3.90	[4.0, 3.0, 1.0, 4.0, 0.0 2.12, 0.0]	[4.0, 3.0, 1.0, 4.0, 0.0 6.63, 0.0]
`nlp1`	-6.667	-6.667	0.106	0.461	[6, 0.667]	[6, 0.667]
`nlp2`	201.16	201.16	0.092	2.75	[6.29, 3.82, 201.16]	[6.29, 3.82, 201.16]
`nlp3`	-1161.34	-1161.34	1.265	17.7	[...]	[...]

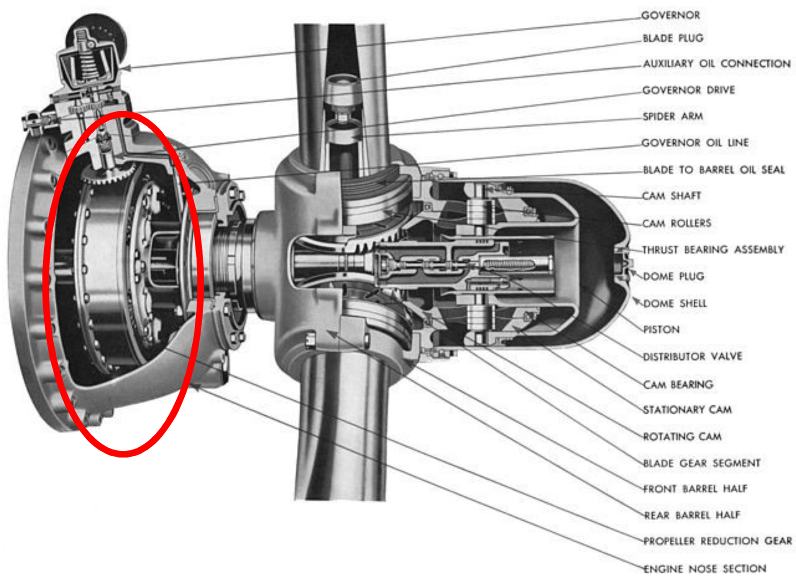
OCT-HaGOn is competitive solving much larger problems as well.

Problem name	Continuous Variables	Integer Variables	Linear Constraints	Nonlinear Inequalities	Nonlinear Equalities	Nonlinear Objective
himmel16	19	0	1	15	6	N
kall_circles_c6b	18	0	54	21	1	N
pointpack08	17	0	41	28	0	N
flay05	23	40	61	5	0	N
fo9	111	72	326	18	0	N
o9_ar4_1	109	72	418	18	0	N

Problem name	Objective			Time (s)		GO
	GO	OCT-HaGOn	BK	GO	OCT-HaGOn	
himmel16	-0.6798	−0.8660*	-0.8660	0.055	109.575	CONOPT
kall_circles_c6b	2.8104	2.1583*	1.9736	0.355	38.503	IPOPT
pointpack08	-0.2574	-0.2500	-0.2679	13.483	91.805	IPOPT
flay05h	64.498	64.499	64.498	0.212	9.515	CONOPT
fo9	23.464	23.464	23.464	959.090	29.534	BARON
o9_ar4_1	236.138	236.138	236.138	2283.281	1255.598	BARON

Real world problems

Golinski's [1970] speed reducer is an NLO problem from aerospace literature.



$$\min \quad f(\boldsymbol{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

(Minimize weight.)

$$x_1 \in [2.6,\ 3.6],\ x_2 \in [0.7,\ 0.8],\ x_3 \in [17,\ 28]$$

$$x_4, x_5 \in [7.3,\ 8.3],\ x_6 \in [2.9,\ 3.9],\ x_7 \in [5,\ 5.5]$$

$$g_1(\boldsymbol{x}) = (27 - x_1x_2^2x_3)/27 \le 0$$
$$g_2(\boldsymbol{x}) = (397.5 - x_1x_2^2x_3^2)/397.5 \le 0$$
$$g_3(\boldsymbol{x}) = (1.93 - (x_2x_6^4x_3)/x_4^3)/1.93 \le 0$$
$$g_4(\boldsymbol{x}) = (1.93 - (x_2x_7^4x_3)/x_5^3)/1.93 \le 0$$
$$g_5(\boldsymbol{x}) = [(745x_4/(x_2x_3))^2 + 16.91 \times 10^6]^{0.5}/0.1x_6^3 - 1100 \le 0$$
$$g_6(\boldsymbol{x}) = [(745x_5/(x_2x_3))^2 + 157.5 \times 10^6]^{0.5}/0.1x_7^3 - 850 \le 0$$
$$g_7(\boldsymbol{x}) = x_2x_3 - 40 \le 0$$
$$g_8(\boldsymbol{x}) = (5 - x_1/x_2)/5 \le 0$$
$$g_9(\boldsymbol{x}) = (x_1/x_2 - 12)/12 \le 0$$
$$g_{10}(\boldsymbol{x}) = (1.9 + 1.5x_6 - x_4)/1.9 \le 0$$
$$g_{11}(\boldsymbol{x}) = (1.9 + 1.5x_7 - x_5)/1.9 \le 0$$

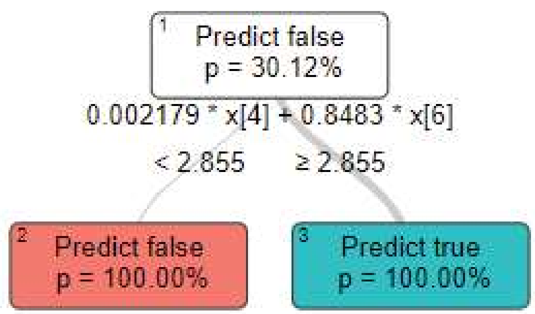
(subject to specifications; and stress, deflection, geometry and manufacturability constraints.)

Each NL constraint is approximated. (1)

Some are straightforward, eg:

$$g_5(\mathbf{x}) = \left[\left(745\frac{x_4}{x_2x_3}\right)^2 + 16.91 \times 10^6\right]^{0.5} 0.1x_6^3 - 1100 \leq 0,$$

where a single hyperplane is able to approximate the constraint in the relevant $\text{dom}(\mathbf{x})$ with 100 % accuracy over 602 uniform samples.

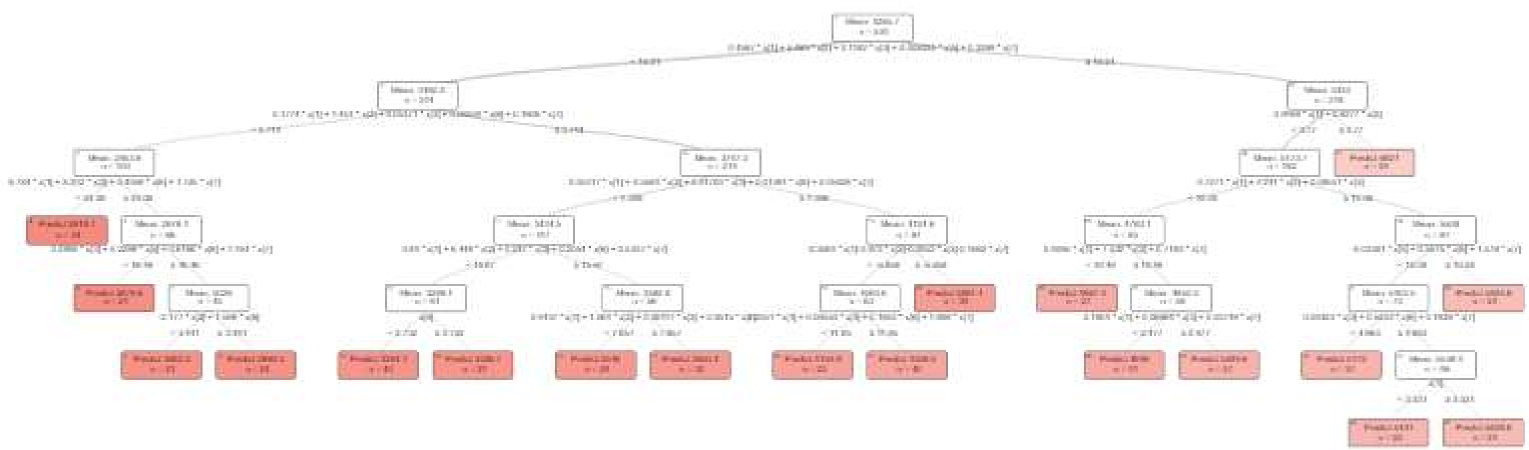


Each NL constraint is approximated. (2)

Other constraints have more complicated approximations:

$$f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934)$$
$$- 1.5079x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2),$$

where the objective is represented by a regression inside 19 unique polyhedra with R^2 of 0.995 over 531 samples.



OCT-HaGOn converges to a solution better than the best known (BK) optimum.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Objective	Time (s)	Error
BK	3.5	0.7	17	7.3	7.7153	3.3503	5.2867	2994.472	476	10^{-6}
OCT-HaGOn	3.5	0.7	17	7.3	7.7153	3.3502	5.2867	2994.355	32.6	0
IPOPT	3.5	0.7	17.0*	7.3	7.7153	3.3502	5.2867	2994.355	4.2	10^{-7}

- Notes:
  - Error is on constraints and objective.
  - IPOPT requires relaxing integrality constraint.

Also applied methods to a satellite on-orbit servicing (OOS) optimization problem.

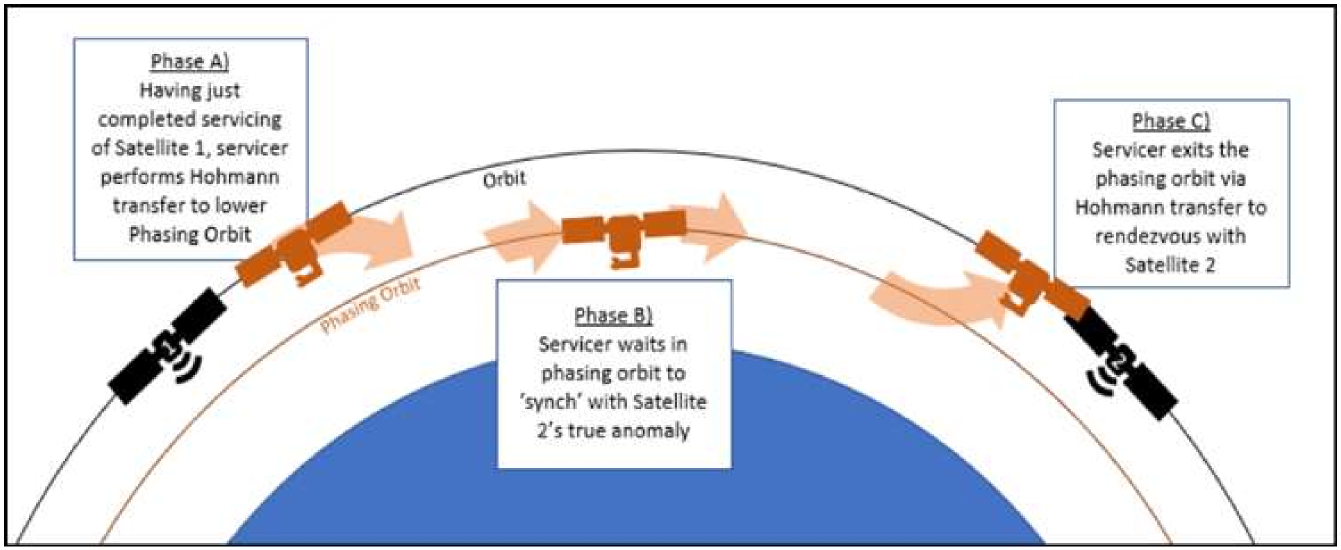
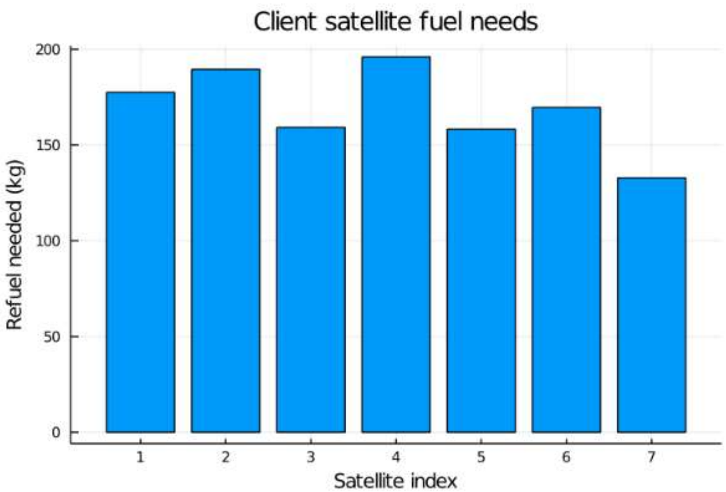


Fig. 5 Concept of operation for orbital phasing to align the true anomaly of the OOS with the client sat.

One *servicer satellite* needs to refuel 7 *client satellites* in orbit through a series of orbital transfers.

OOS problem description and parameters



Parameter	Value	Units
Satellite dry mass	500	kg
Propulsor specific impulse	230	(Ns)/kg
Number of client satellites	7	-
Client satellite altitude	780	km
Servicer satellite altitudes	[760,800]	km
Maximum service time	0.35	years

Objective: minimize total fuel,
Subject to:

- Transfer orbit entry burn,
- Transfer orbit exit burn,
- Mass conservation,
- Phasing orbit period,
- Transfer time,
- Transfer orbit revolutions,
- Total maneuver time constraints.

Problem size:

- 123 continuous variables,
- 49 binary variables,
- **60 nonlinear equalities (!).**

OOS problem is solved via two ways.

Metric	Values						
OCT-HaGOn solution							
Wet mass (kg)	1725.9						
Total maneuver time (years)	0.350						
Satellite order	4	3	2	1	7	6	5
Refuel mass (kg)	196.0	159.2	189.5	177.4	132.9	169.6	158.2
Transfer orbit altitude (km)		765.8	765.8	765.8	765.8	765.8	767.6
Maneuver fuel (kg)		9.60	8.74	7.73	6.79	6.08	4.17
Maneuver time (days)		20.7	20.7	20.7	20.7	20.7	24.1
Orbital revolutions		297.0	297.0	297.0	297.0	297.0	345.3
Discretized MI-bilinear solution							
Wet mass (kg)	1724.4						
Total maneuver time (years)	0.350						
Satellite order	4	3	2	1	7	6	5
Refuel mass (kg)	196.0	159.2	189.5	177.4	132.9	169.6	158.2
Transfer orbit altitude (km)		768.0	768.0	766.0	765.0	765.0	762.0
Maneuver fuel (kg)		8.46	7.53	7.51	6.77	5.80	5.51
Maneuver time (days)		24.9	24.9	21.4	19.9	19.9	16.6
Orbital revolutions		357.1	357.1	306.1	285.7	285.7	238.1

Table 2.10: The discretized and OCT-H formulations come up with the same optimal satellite schedule, although the MI-bilinear approximation is 0.1% better.

OCT-HaGOn solution has same optimal schedule as a discretized MI-bilinear solution, with slight suboptimality.

Both cases took around 15s to solve
(not to mention the reformulation time for MI solution).

Conclusions

- The OCT-HaGOn constraint sampling and learning approach is powerful to solve a wide range of design optimization problems.
- OCT-HaGOn may currently be the only MIO approach to handle explicit and inexplicit constraints in one framework.
- Constraint learning could leverage other MIO-compatible ML methods, such as neural networks and tree ensembles[1].
- In thesis, I discuss future work for DT-based optimization in length, including:
  - Different sampling and training methods,
  - Complexity theory,
  - Using MI-convex formulations,
  - Improving the speed and reliability of the method.

1. Maragno, D., Wiberg, H., Bertsimas, D., Birbil, S. I., den Hertog, D., & Fajemisin, A. (2021). Mixed-Integer Optimization with Constraint Learning. *ArXiv.*

When can you use OCT-HaGOn?

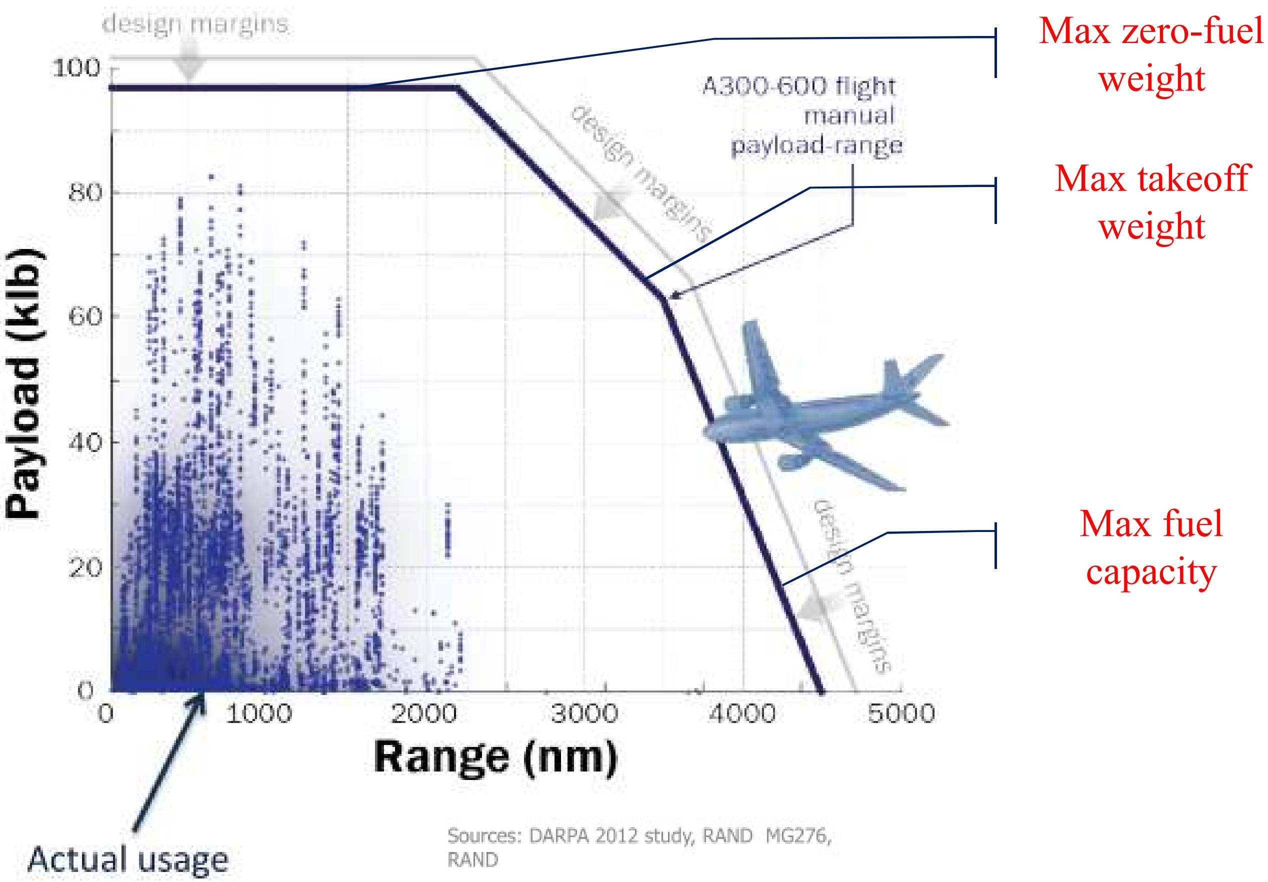
- Work in progress, with George and Dimitris.
- After paper submission, it will be public (January 2022).
- Now at 2700 active lines of Julia code and over 100 pull requests.
- Requires an academic license for IAI. Compatible with any JuMP.jl-compatible MIO solver; uses CPLEX by default.

Interpretable AI

IBM CPLEX

Engineering Design under Uncertainty via Robust Optimization

Motivation: Legacy design methods do not adequately consider the risk-performance tradeoff.



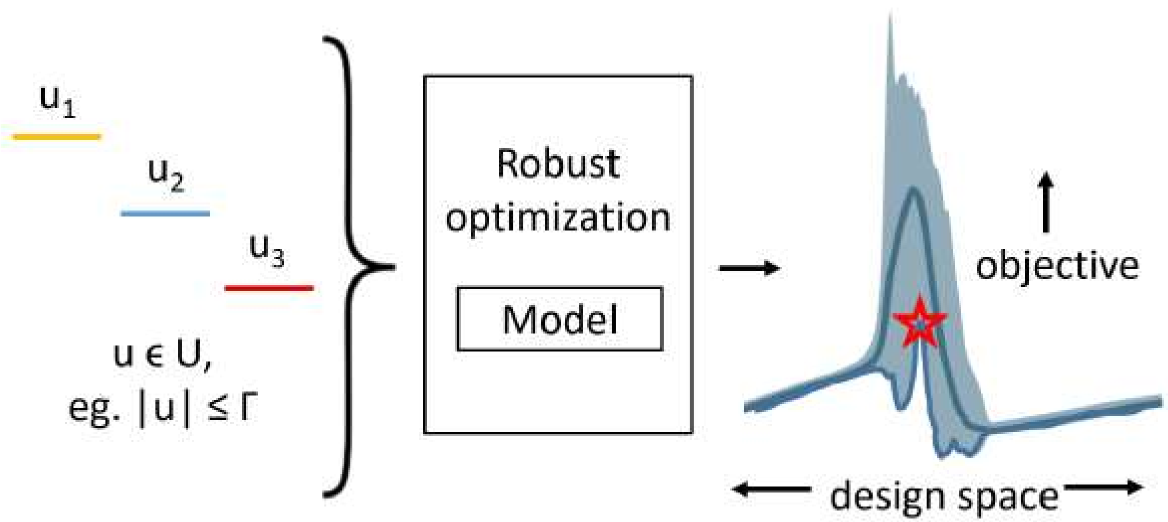
Sources: DARPA 2012 study, RAND MG276, RAND

There is no such thing as a free lunch. Conservative margins leave performance on the table.

How about:

- Technological capabilities?
- Manufacturing quality?
- Regulatory environment?

Robust optimization is a tractable, deterministic method for optimization under uncertainty (OUU).



RO makes sure all constraints are feasible for all parameter outcomes from an uncertainty set, while minimizing the worst-case objective.

Primary Contributions

1. A tractable robust signomial programming (RSP) formulation for design under uncertainty, that is sufficiently general to address aerospace design problems.
2. Application of RO to an aircraft design problem, showing its practicality, tractability and ability to consider uncertainty with mathematical rigor.

Problem of interest: aircraft design that captures important multidisciplinary tradeoffs.

- Unmanned, gas-powered aircraft
- Without uncertainty: 176 variables and 154 constraints
- Monolithic: optimizes aircraft and flight trajectory concurrently through disciplined SP form

Wing

- Structure
- Fuel volume
- Profile drag
- Stall constraint

Fuselage

- Fuel and payload
- Profile drag

Engine

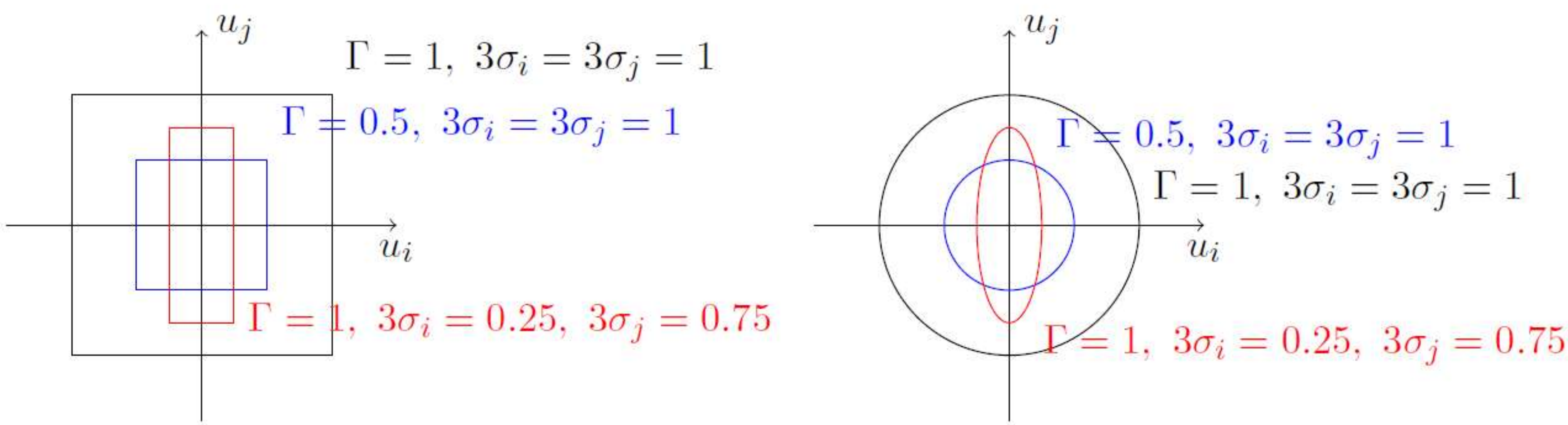
- Data-based sizing
- Lapse rate
- BSFC fits
- T/O and TOC constraints

We determine uncertain parameters, and expected variances.

Table 3.1: Parameters and uncertainties (increasing order)

Parameters	Description	Value	% Uncert. (3σ)
e	span efficiency	0.92	3
μ	air viscosity (SL)	1.78×10^{-5} kg/(ms)	4
ρ	air density (SL)	1.23 kg/m^3	5
$C_{L,\text{max}}$	stall lift coefficient	1.6	5
k	fuselage form factor	1.17	10
$C_{f,\text{ref}}$	reference fuselage skin friction factor	0.455	10
ρ_p	payload density	1.5 kg/m^3	10
N_ult	ultimate load factor	3.3	15
V_min	takeoff speed	35m/s	20
W_p	payload weight	3000 N	20
$W_\text{coeff,strc}$	wing structural weight coefficient	2×10^{-5} 1/m	20
$W_\text{coeff,surf}$	wing surface weight coefficient	60 N/m^2	20

The uncertainty is defined by box and ellipsoidal sets.

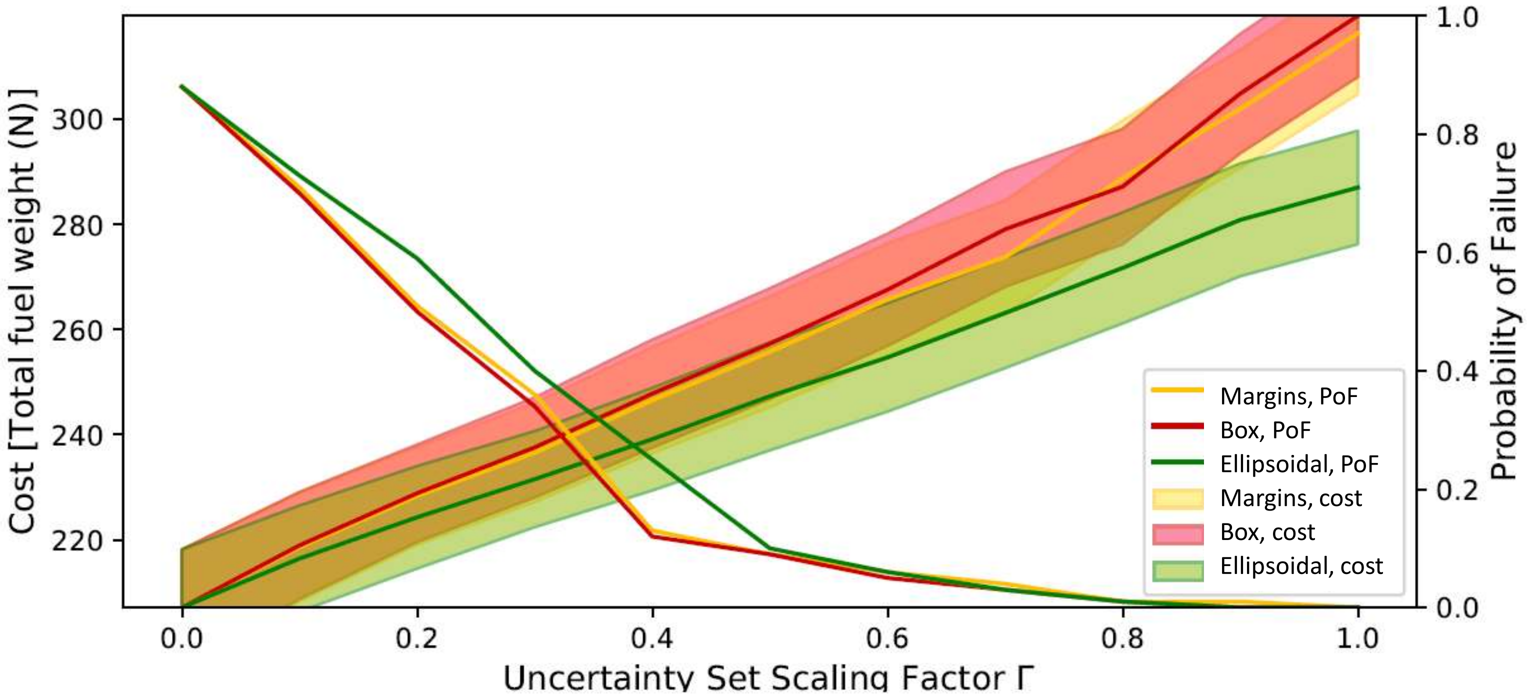


(a) Example L∞ or box sets.

(b) Example L2 or ellipsoidal sets.

Figure 3-5: Γ defines the overall size of norm uncertainty sets, while 3σ defines the relative size of the set in each uncertain parameter.

Primary result: RSP mitigates probability of constraint violation under uncertain outcomes,



and is less conservative than designs with margins.

robust is open source!

robust
latest
Search docs

- Robust optimization 101
- Installing robust
- Getting started
- More advanced commands
- Why robust optimization?
- Mathematical moves for robust GPs/SPs
- Approximations for tractable robust GPs
- Approaches to solving robust SPs
- Goal programming
- References

Welcome to robust's documentation!

robust is a framework for engineering system optimization under uncertainty using geometric and signomial programming.

Written in Python3, meshes with GPkit!

- Repository: https://github.com/convexengineering/robust
- Documentation: https://robust.readthedocs.io/

Contributions

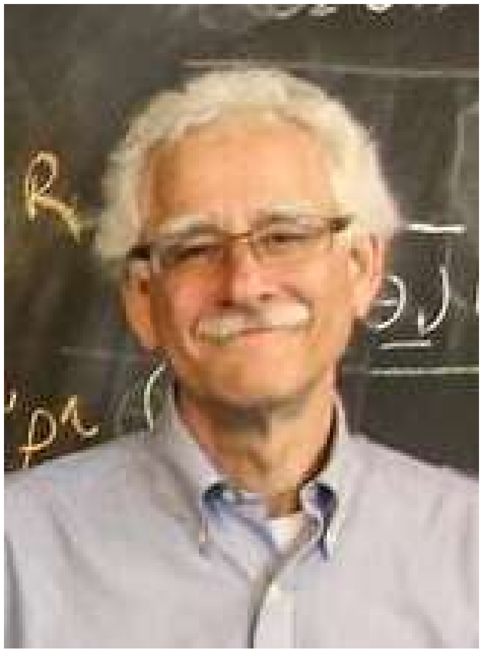
A **Global Optimization** method for aerospace design that is,

- General to explicit and inexplicit constraints with bounded decision variables,
- Compatible with mixed-integer linear optimization,
- Tractable and effective at addressing real world problems.

A **Robust Optimization** method for aerospace design under uncertainty that is,

- Sufficiently general to address aerospace design problems,
- Tractable and deterministic,
- Provides probabilistic guarantees of constraint satisfaction.

Acknowledgements



Dr. Bob Haimes

Prof. Mark Drela

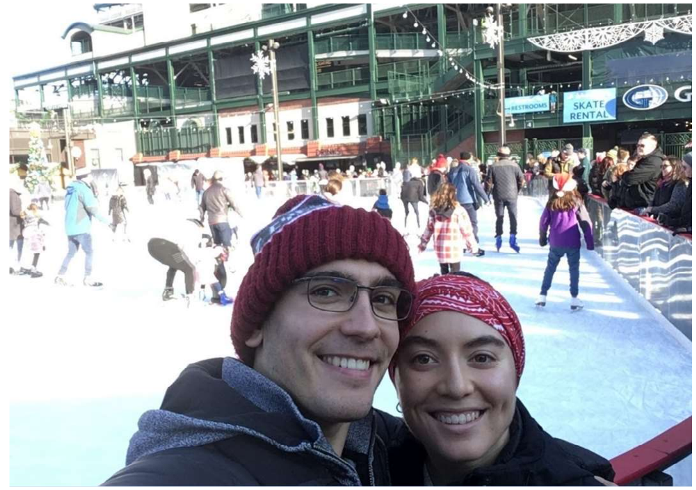
Prof. Dimitris Bertsimas

Prof. Oli de Weck

Dr. Jack Dunn

Ali Saab


B2C2


SKATE
RENTAL

Back-up slides

Requirements growth asks more and more from aircraft design, with diminishing marginal returns.

Wright Flyer (1903) requirements:
"Flight"!

DC3 (1936) requirements:
Wright Flyer reqs. +
Passengers and cargo +
1000mi range +
Basic reliability

B777 (2011) requirements:
DC3 reqs. +
Transonic flight +
Safety +
Emissions regulations +
ETOPS operations +
Fly-by-wire + …

MI optimization has experienced a dramatic improvement in solver efficiency.

1998 → 2003

Table 1 Mean performance improvement from CPLEX 5.0 to CPLEX 9.0

Model subset	Number of models	Mean speedup
All models	719	16X
CPLEX 5.0 more than 1 second	515	49X
CPLEX 5.0 more than 10 seconds	429	94X
CPLEX 5.0 more than 100 seconds	345	162X
CPLEX 5.0 more than 1000 seconds	268	344X

Larger problems have sped up more!

(This is <u>without</u> considering hardware improvements.)
MIO approximation approach is promising.

Bixby, R., & Rothberg, E. (2007). Progress in computational mixed integer programming - A look back from the other side of the tipping point. *Annals of Operations Research*, *149*(1), 37–41.

MIO approximations of difficult constraints

Inequality constraints can be represented as a union of polyhedra bounded by the splitting hyperplanes.

$$\{\mathbf{x} : g_i(\mathbf{x}) \geq 0\} \;\approx\; \mathbf{x} \in \bigcup_{l \in L_{i,1}} P_{i,l}$$

$$P_{i,l} = \{\mathbf{x} \in \mathbb{R}_n : \alpha_h \cdot \mathbf{x} \leq \beta_h,\ \forall\ h \in H_{i,l,-}\ ;\ \alpha_h \cdot \mathbf{x} \geq \beta_h,\ \forall\ h \in H_{i,l,+}\}$$

Equality constraints can be (a little less obviously) represented by the union of intersections of pairs of polyhedra.

$$\{\mathbf{x} : h_j(\mathbf{x}) = 0\} \;\approx\; \mathbf{x} \in \bigcup_{u \in L_{j,0},\ v \in L_{j,1}} \{P_{j,u} \cap P_{j,v}\}$$

The locally ideal MIO approximation is as follows:

$$\mathbf{x} \in \bigcup_{l \in L_{i,1}} \mathbf{P}_{i,l} \iff \begin{cases} \{\ \alpha_h^\top \mathbf{y}_l \leq \beta_h z_{i,l},\ \forall\ h \in H_{i,l,-}\ ; \\ \beta_h z_{i,l} \leq \alpha_h^\top \mathbf{y}_l,\ \forall\ h \in H_{i,l,+}\}\ \forall\ l \in L_{i,1}, \\ \mathbf{y}_l \in [\underline{\mathbf{x}} z_{i,l}, \overline{\mathbf{x}} z_{i,l}],\ l \in L_{i,1}, \\ \sum_{l \in L_{i,1}} \mathbf{y}_l = \mathbf{x}, \\ \sum_{l \in L_{i,1}} z_{i,l} = 1, \\ z_{i,l} \in \{0,1\},\ l \in L_{i,1}. \end{cases}$$

Where:

- $\mathbf{x}$ are the original variables
- $\boldsymbol{y}_l$ are the auxiliary variables in each leaf.
- $z_{i,l}$ are the binary variables in the feasible leaves.

The MIO approximation of the original problem is the following…

$$\begin{aligned}\min_x \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \geq 0,\ i \in I, \\ & h_j(\mathbf{x}) = 0,\ j \in J, \\ & \mathbf{Ax} \geq \mathbf{b},\ \mathbf{Cx} = \mathbf{d}, \\ & x_k \in [\underline{x}_k, \overline{x}_k],\ k \in [n].\end{aligned}$$

$\rightarrow$

$$\begin{aligned}\min_x \quad & f^* \\ \text{s.t.} \quad & f^*, \mathbf{x} \in \bigcup_{l \in L_f} \mathbf{P}_{i,l}, \\ & \mathbf{x} \in \bigcup_{l \in L_{i,1}} \mathbf{P}_{i,l},\ \forall\, i \in I, \\ & \mathbf{x} \in \bigcup_{l_0 \in L_{j,0},\ l_1 \in L_{j,1}} \{\mathbf{P}_{j,l_0} \cap \mathbf{P}_{j,l_1}\},\ \forall j \in J, \\ & \mathbf{Ax} \geq \mathbf{b},\ \mathbf{Cx} = \mathbf{d}, \\ & x_k \in [\underline{x}_k, \overline{x}_k],\ k \in [n] \\ & z_{i,l} \in \{0,1\},\ \forall l \in L_{i,1},\ i \in I, \\ & z_{j,l} \in \{0,1\},\ \forall l \in L_j,\ j \in J.\end{aligned}$$

…and its solution is a near-feasible, near-optimal **x***.

Solution check and repair using projected gradient descent.

- The MI approx. is replaced by an auto-differentiated local gradient.
- It takes steps (i.e. solve a quadratic optimization problem) that restore feasibility as well as descend the objective.
- Example projection step, with **d** step:

Δ objective + step penalty + infeasibility penalty

$$\begin{aligned} \min_{\mathbf{x},\mathbf{d},\lambda,\mu} \quad & \nabla f(\mathbf{x}^*)^\top \mathbf{d} + \beta \left|\left| \frac{\mathbf{d}}{\overline{\mathbf{x}} - \underline{\mathbf{x}}} \right|\right|_2^2 + \gamma(||\lambda||_2^2 + ||\mu||_2^2) \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{x}^* + \mathbf{d}, \\ & \vdots \end{aligned}$$

Gradient descent formulation

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{d},\lambda,\mu} \quad & \nabla f(\mathbf{x}^*)^\top \mathbf{d} + \gamma(||\lambda||_2^2 + ||\mu||_2^2) \\
\text{s.t.} \quad & \mathbf{x} = \mathbf{x}^* + \mathbf{d}, \\
& \left|\left| \frac{\mathbf{d}}{\overline{\mathbf{x}} - \underline{\mathbf{x}}} \right|\right|_2^2 \leq \alpha \exp\left(\frac{-\mathrm{rt}}{\mathrm{T}}\right), \\
& \left\{ \begin{array}{ll} \nabla g_i(\mathbf{x}^*)^\top \mathbf{d} + g_i(\mathbf{x}^*) \geq 0, & \text{if } g_i(\mathbf{x}^*) \geq 0 \\ \nabla g_i(\mathbf{x}^*)^\top \mathbf{d} + g_i(\mathbf{x}^*) + \lambda_i \geq 0, & \text{if } g_i(\mathbf{x}^*) \leq 0 \end{array} \right\}, \ \forall i \in I, \\
& \left\{ \begin{array}{l} \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} + h_j(\mathbf{x}^*) + \mu_j \geq 0, \\ \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} + h_j(\mathbf{x}^*) \leq \mu_j, \end{array} \right\}, \ \forall j \in J, \\
& \mathbf{A}\mathbf{x} \geq \mathbf{b}, \ \mathbf{C}\mathbf{x} = \mathbf{d}, \\
& x_k \in [\underline{x}_k, \overline{x}_k], \ \forall k \in [n] \\
& \left\{ \begin{array}{ll} \lambda_i = 0, & \text{if } g_i(\mathbf{x}^*) \geq 0 \\ \lambda_i \geq 0, & \text{if } g_i(\mathbf{x}^*) \leq 0 \end{array} \right\}, \ \forall i \in I, \\
& \mu_i \in \mathbb{R}_+, \ j \in J.
\end{aligned}
$$

Projected gradient descent formulation

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{d},\lambda,\mu} \quad & \nabla f(\mathbf{x}^*)^\top \mathbf{d} + \beta \left\| \frac{\mathbf{d}}{\overline{\mathbf{x}} - \underline{\mathbf{x}}} \right\|_2^2 + \gamma(||\lambda||_2^2 + ||\mu||_2^2) \\
\text{s.t.} \quad & \mathbf{x} = \mathbf{x}^* + \mathbf{d}, \\
& \left\{ \begin{array}{ll} \nabla g_i(\mathbf{x}^*)^\top \mathbf{d} + g_i(\mathbf{x}^*) \geq 0, & \text{if } g_i(\mathbf{x}^*) \geq 0 \\ \nabla g_i(\mathbf{x}^*)^\top \mathbf{d} + g_i(\mathbf{x}^*) + \lambda_i \geq 0, & \text{if } g_i(\mathbf{x}^*) \leq 0 \end{array} \right\}, \ \forall i \in I, \\
& \left\{ \begin{array}{l} \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} + h_j(\mathbf{x}^*) + \mu_j \geq 0, \\ \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} + h_j(\mathbf{x}^*) \leq \mu_j, \end{array} \right\}, \ \forall j \in J, \\
& \mathbf{A}\mathbf{x} \geq \mathbf{b}, \ \mathbf{C}\mathbf{x} = \mathbf{d}, \\
& x_k \in [\underline{x}_k, \overline{x}_k], \ \forall k \in [n] \\
& \left\{ \begin{array}{ll} \lambda_i = 0, & \text{if } g_i(\mathbf{x}^*) \geq 0 \\ \lambda_i \geq 0, & \text{if } g_i(\mathbf{x}^*) \leq 0 \end{array} \right\}, \ \forall i \in I, \\
& \mu_i \in \mathbb{R}_+, \ j \in J.
\end{aligned}
$$

Decision tree parameters in IAI

Parameter	OCT-H	ORT-H
Hyperplane sparsity	All	All
Regression sparsity	-	All
Max depth	5	5
Complexity factor	10^{-6}	10^{-6}
Minbucket	0.01	0.02
Random tree restarts	10	10
Hyperplane restarts	5	5

Table 2.1: Parameters for base decision trees in constraint learning.

OOS variables

The satellite OOS problem has the following decision variables and associated dimensions, where n_s is the number of client satellites.

$$
\begin{aligned}
\text{Satellite order variables} &: z_{i,j} \in \{0,1\}, && i,j \in [n_s], \\
\text{Orbit radii} &: r_{\text{orbit},i} \in [r_{\text{orbit,min}}, r_{\text{orbit,max}}], && i \in [n_s - 1], \\
\text{Orbital periods} &: T_{\text{orbit,i}} \in [T_{\text{orbit,min}}, T_{\text{orbit,max}}], && i \in [n_s - 1], \\
\text{Orbital period differences} &: \Delta T_{\text{orbit,i}} \in [\Delta T_{\text{min}}, \Delta T_{\text{max}}], && i \in [n_s - 1], \\
\text{True anomalies} &: \theta_i \in [-\pi, \pi], && i \in [n_s - 1], \\
\text{Transfer times} &: t_{\text{transfer,i}} \in [0, t_{\text{transfer,max}}], && i \in [n_s - 1], \\
\text{Maneuver times} &: t_{\text{maneuver,i}} \in \mathbb{R}^+, && i \in [n_s - 1], \\
\text{Orbital revolutions} &: N_{\text{orbit},i} \in [50, 500], && i \in [n_s - 1], \\
\text{Orbital entry mass ratios} &: f_{\text{entry,i}} \in [1, 1.0025], && i \in [n_s - 1], \\
\text{Orbital exit mass ratios} &: f_{\text{exit,i}} \in [1, 1.0025], && i \in [n_s - 1], \\
\text{Wet mass} &: m_{\text{wet}} \in [m_{\text{dry}}, 2000], && \\
\text{Intermediate masses} &: m_{i,j} \in [m_{\text{dry}}, 2000], && i \in [n_s - 1],\ j \in [5], \\
\text{Transferred fuel masses} &: m_{\text{fuel,i}} \in [m_{\text{fuel,min}}, m_{\text{fuel,max}}], && i \in [n_s].
\end{aligned}
$$

OOS linear constraints

The constraints are given below with brief descriptions.

$$\text{Each client visited once}: \sum_{i=1}^{n_s} z_{i,j} = 1, \qquad \forall j \in [n_s]$$

$$\text{One refuel per rendezvous}: \sum_{j=1}^{n_s} z_{i,j} = 1, \qquad \forall i \in [n_s]$$

$$\text{Fuel required for } i\text{th client}: m_{\text{fuel,i}} = \sum_{j=1}^{n_s} \Delta m_{\text{cf,j}} z_{i,j}, \qquad \forall i \in [n_s]$$

$$\text{True anomaly from client } i \text{ to } i+1: \ \theta_i = \sum_{j=1}^{n_s} \left((-\pi + 2\pi j / n_s)(z_{i+1,j} - z_{i,j}) \right), \qquad \forall i \in [n_s - 1]$$

$$\text{Wet mass}: m_{\text{wet}} = m_{1,1} + m_{\text{fuel,1}}$$

$$\text{Intermediate fuel transfers}: m_{i,5} = m_{i+1,1} + m_{\text{fuel,i+1}}, \qquad \forall i \in [n_s - 2]$$

$$\text{Dry mass}: m_{n_s-1,5} = m_{\text{dry}} + m_{\text{fuel,n}_\text{s}}$$

$$\text{Orbital period difference}: \Delta T_{\text{orbit,i}} = T_{\text{orbit,i}} - T_{\text{client}}, \qquad \forall i \in [n_s - 1]$$

$$\text{Total maneuver time}: \sum_{i=1}^{n_s-1} t_{\text{maneuver,i}} \leq t_{\max}$$

OOS nonlinear constraints (1)

- Transfer orbit entry burn ($n_s - 1$ constraints): Describes mass ratio (entry mass over exit mass) of the satellite during transfer orbit entry.

$$f_{\text{entry,i}} = \max\left[\exp\left(\frac{1}{gI_{\text{sp}}}\sqrt{\frac{\mu}{r_{\text{orbit,i}}}}\left(\sqrt{\frac{2r_{\text{client}}}{r_{\text{client}}+r_{\text{orbit,i}}}}-1\right)\right),\right.$$
$$\left.\exp\left(\frac{1}{gI_{\text{sp}}}\sqrt{\frac{\mu}{r_{\text{client}}}}\left(\sqrt{\frac{2r_{\text{orbit,i}}}{r_{\text{client}}+r_{\text{orbit,i}}}}-1\right)\right)\right],\ i \in [n_s - 1].$$

- Transfer orbit exit burn ($n_s - 1$ constraints): Describes the mass ratio (entry mass over exit mass) of the satellite during transfer orbit exit.

$$f_{\text{exit,i}} = \max\left[\exp\left(\frac{1}{gI_{\text{sp}}}\sqrt{\frac{\mu}{r_{\text{client}}}}\left(1-\sqrt{\frac{2r_{\text{orbit,i}}}{r_{\text{client}}+r_{\text{orbit,i}}}}\right)\right),\right.$$
$$\left.\exp\left(\frac{1}{gI_{\text{sp}}}\sqrt{\frac{\mu}{r_{\text{orbit,i}}}}\left(1-\sqrt{\frac{2r_{\text{client}}}{r_{\text{client}}+r_{\text{orbit,i}}}}\right)\right)\right],\ i \in [n_s - 1].$$

- Mass conservation ($4(n_s - 1)$ constraints): Couples the fractional change in mass of the satellite to the absolute change in mass during each burn phase.

$$\begin{aligned}
m_{i,1} &= f_{\text{entry,i}} m_{i,2}, && i \in [n_s - 1],\\
m_{i,2} &= f_{\text{exit,i}} m_{i,3}, && i \in [n_s - 1],\\
m_{i,3} &= f_{\text{exit,i}} m_{i,4}, && i \in [n_s - 1],\\
m_{i,4} &= f_{\text{entry,i}} m_{i,5}, && i \in [n_s - 1].
\end{aligned}$$

OOS nonlinear constraints (2)

- Phasing orbit period ($n_s - 1$ constraints): Describes the period of the phasing orbit.

$$T_{\text{orbit,i}} = 2\pi\sqrt{\frac{r^3_{\text{orbit,i}}}{\mu}},\ i \in [n_s - 1].$$

- Transfer time ($n_s - 1$ constraints): Describes the Hohmann transfer time from the client to phasing orbit.

$$t_{\text{transfer,i}} = 2\pi\sqrt{\frac{(r_{\text{client}} + r_{\text{orbit,i}})^3}{8\mu}},\ i \in [n_s - 1].$$

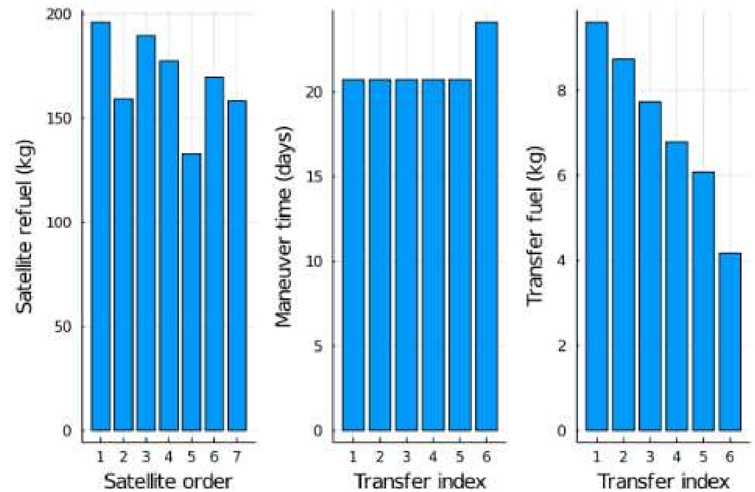
- Number of transfer orbit revolutions ($n_s - 1$ constraints): Describes the number of revolutions in phasing orbit.

$$N_{\text{orbit,i}}\Delta T_{\text{orbit,i}} = T_{\text{client,i}}\theta_i,\ i \in [n_s - 1].$$

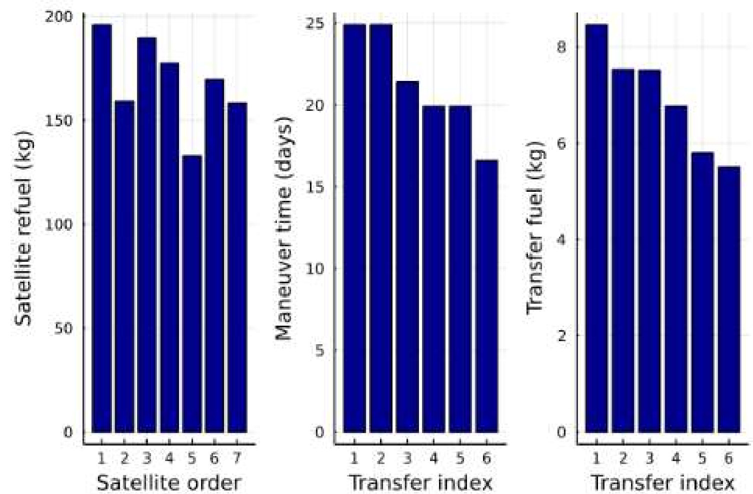
- Maneuver time ($n_s - 1$ constraints): Describes the maneuver time (transfer and phasing time) between clients.

$$t_{\text{maneuver,i}} = t_{\text{transfer,i}} + N_{\text{orbit,i}}T_{\text{orbit,i}},\ i \in [n_s - 1].$$

OOS solution



(a) The OCT-H solution.



(b) The MI-bilinear solution.

Considerations for constraint generation:

- Whether we have data or functions?
- Does the function have an accessible gradient / is it auto-differentiable?
- Is the function/data convex and can we detect its convexity?
- Are the functions expensive?

Future with for OCT-HaGOn

- Improving OCT-H training and accuracy
  - Dynamic sampling/re-training/re-optimization
- MI-convex formulation
  - Can already embed convex constraints directly,
  - But is there something to gain by changing the type of in-leaf approximation?
- Random restarts of the tree approximators
- Integration of other MIO-compatible ML models

Aerospace Design via Robust Optimization

Mathematical moves to obtain RSPs

- LPs have tractable robust counterparts.
- Two-term posynomials are LP-approximable.
- All posynomials are LP-approximable.
- GPs have robust formulations.
- RSPs can be represented as sequential RGPs.

LPs have tractable robust counterparts.

$$\begin{aligned} &\textit{minimize} \quad \boldsymbol{c}^\top \boldsymbol{x} \\ &\textit{subject to} \quad \boldsymbol{a}_i \boldsymbol{x} \leq b_i, \quad \forall \boldsymbol{a}_i \in \mathcal{U}_i, \quad \forall i = 1, \ldots, m, \end{aligned}$$

$$\mathcal{U} = \{(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m) : \boldsymbol{a}_i = \boldsymbol{a}_i^0 + \Delta_i u_i, \; i = 1, \ldots, m, \quad ||u||_2 \leq \rho\},$$

Robust counterpart

I tip my hat to the editor!

$$\begin{aligned} &\textit{mininize} \quad \boldsymbol{c}^\top \boldsymbol{x} \\ &\textit{subject to} \quad \boldsymbol{a}_i^0 \boldsymbol{x} \leq b_i - \rho ||\Delta_i \boldsymbol{x}||_2, \quad \forall i = 1, \ldots, m. \end{aligned}$$

A tractable SOCP!

A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

Two-term posynomials are LP-approximable.

Corollary 1 *For $r \geq 3$, the unique best r-term PWL convex lower approximation $\underline{h}_r\colon \mathbf{R}^2 \to \mathbf{R}$ of the two-term log-sum-exp function is*

$$\underline{h}_r(y_1, y_2) = \max\{y_1, \underline{a}^\star_{r-2} y_1 + \underline{a}^\star_1 y_2 + \underline{b}^\star_1, \underline{a}^\star_{r-3} y_1 + \underline{a}^\star_2 y_2 + \underline{b}^\star_2, \ldots, \underline{a}^\star_1 y_1 + \underline{a}^\star_{r-2} y_2 + \underline{b}^\star_{r-2}, y_2\} \quad (24)$$

and the unique best r-term PWL convex upper approximation $\overline{h}_r : \mathbf{R}^2 \to \mathbf{R}$ is

$$\overline{h}_r(y_1, y_2) = \underline{h}_r(y_1, y_2) + \underline{\epsilon}_\phi(r), \quad (25)$$

where $a^\star_i, b^\star_i, i = 1, \ldots, r-2$ are the coefficients of the segments of $\underline{\phi}_r$ defined in (23).

Approximation error vs. degree of PWL approximation r.

Hsiung, K. L., Kim, S. J., and Boyd, S., "Tractable approximate robust geometric programming," *Optimization and Engineering*, vol. 9, 2008, pp. 95–118.

All posynomials must then be LP-approximable.

The recipe:

$$
\begin{aligned}
\min \quad & f_0(\boldsymbol{x}) & & \\
\text{s.t.} \quad & \max_{\zeta \in \mathcal{Z}} \left\{ e^{\boldsymbol{a_{i1}}(\zeta)\boldsymbol{x} + b_{i1}(\zeta)} + e^{t_1} \right\} & \leq 1 \quad & \forall i : K_i \geq 3 \\
& \max_{\zeta \in \mathcal{Z}} \left\{ e^{\boldsymbol{a_{ik}}(\zeta)\boldsymbol{x} + b_{ik}(\zeta)} + e^{t_k} \right\} & \leq e^{t_{k-1}} \quad & \forall i : K_i \geq 4 \\
& & & \forall k \in 2, ..., K_i - 2 \\
& \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=K_i-1}^{K_i} e^{\boldsymbol{a_{ik}}(\zeta)\boldsymbol{x} + b_{ik}(\zeta)} \right\} & \leq e^{t_{K_i-2}} \quad & \forall i : K_i \geq 3 \\
& \max_{\zeta \in \mathcal{Z}} \left\{ \sum_{k=1}^{K_i} e^{\boldsymbol{a_{ik}}(\zeta)\boldsymbol{x} + b_{ik}(\zeta)} \right\} & \leq 1 \quad & \forall i : K_i \leq 2
\end{aligned}
$$

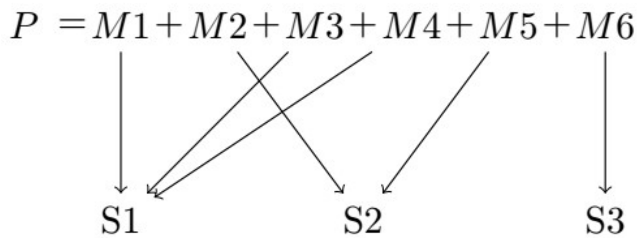
Simple example:

$$
\begin{aligned}
\min \quad & f & \\
\text{s.t.} \quad & \max \{M_1 + M_2 + M_3 + M_4\} & \leq 1 \\
& \max \{M_5 + M_6\} & \leq 1
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & f & \\
\text{s.t.} \quad & \max \left\{M_1 + e^{t_1}\right\} & \leq 1 \\
& \max \left\{M_2 + e^{t_2}\right\} & \leq e^{t_1} \\
& \max \{M_3 + M_4\} & \leq e^{t_2} \\
& \max \{M_5 + M_6\} & \leq 1
\end{aligned}
$$

Saab, A., Burnell, E., and Hoburg, W. W., "Robust Designs Via Geometric Programming." 2018. ArXiv:1808.07192

Uncoupled posynomials are robustified separately



$$
max\{P\} \leq 1 \iff \begin{array}{rl} t_1 + t_2 + t3 & \leq 1 \\ max\{S_1\} = max\{M_1 + M_3 + M_4\} & \leq t_1 \\ max\{S_2\} = max\{M_2 + M_5\} & \leq t_2 \\ max\{S_3\} = max\{M_6\} & \leq t_3 \end{array}
$$

Figure 2: Partitioning of a large posynomial into smaller posynomials requires the addition of auxiliary variables. S_i are posynomials with independent sets of variables.

Three approximations exist for RGP.

Increasingly conservative

- Simple conservative
  - Maximizes each monomial term separately
- Linearized perturbations
  - Separates large posynomial into decoupled posynomials
  - Robustifies smaller posy's using RLO techniques

Uncertain coefficients only

- Best pairs
  - Separates large posynomial into decoupled posynomials
  - Finds least conservative combination of monomial pairs

Uncertain coefficients and exponents

Saab, A., Burnell, E., and Hoburg, W. W., "Robust Designs Via Geometric Programming." 2018. ArXiv:1808.07192

We augment the SP solution heuristic.

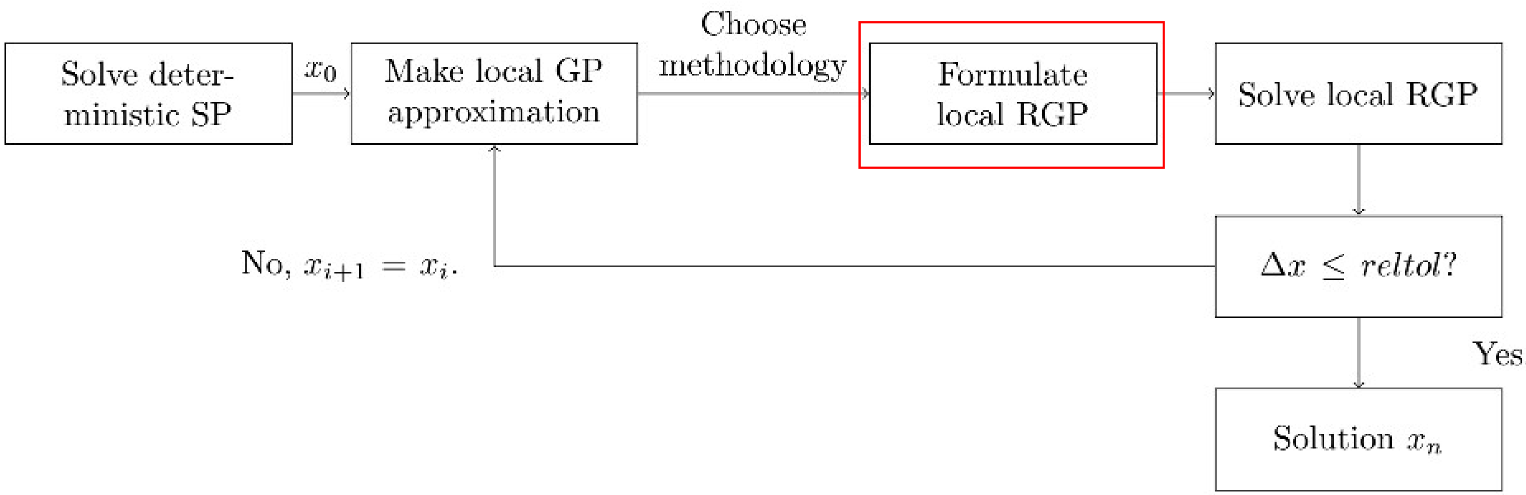
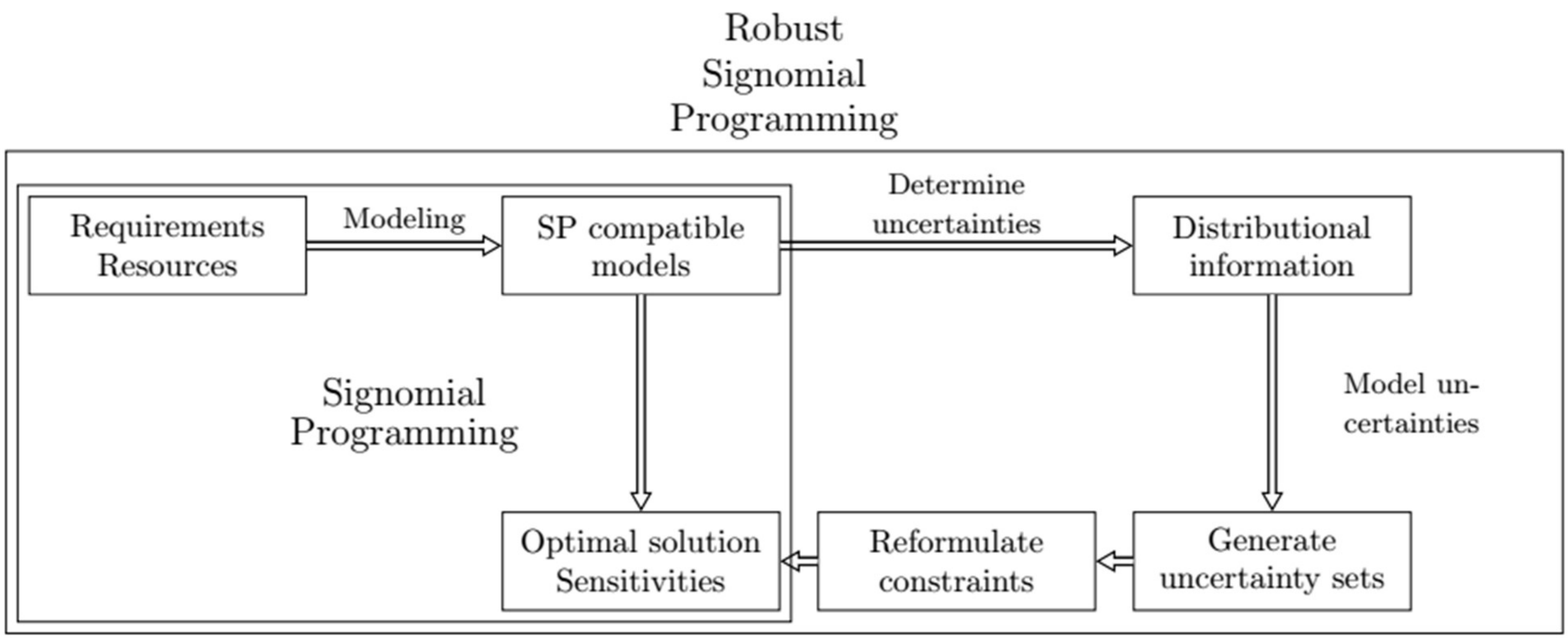


Figure 3: A block diagram showing the steps of solving an RSP.

RSP formulations exist for all SP-compatible pro

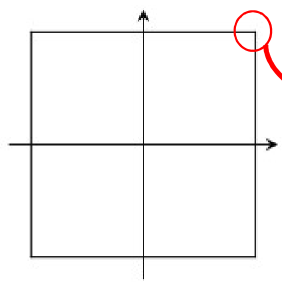


Uncertainty sets considered

Box (L-∞ norm)

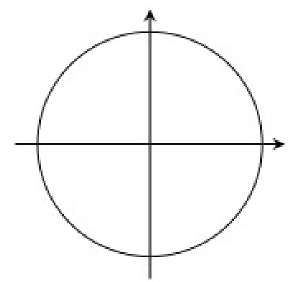
Elliptical (L-2 norm)

More coı n margins. A less coı ıdidate!



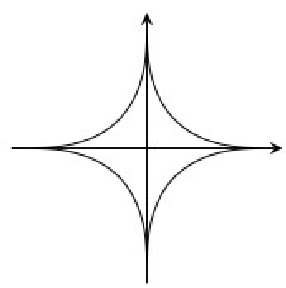
$p = \infty$

Margins optimize on a
corner of the hypercube!

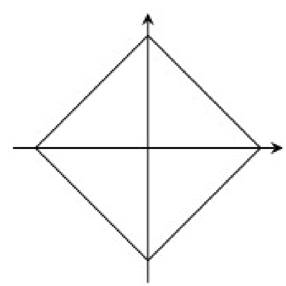


$p = 2$

Other norms also valid.



$p = \frac{1}{2}$



$p = 1$

Goal programming: risk is a global design objective.

Standard RO form

$$\min\ f_0(x)$$
$$\text{s.t.}\ \max_u\ f_i(x,u) \le 0,\ \ i = 1,\dots,n$$
$$\|u\| \le \Gamma$$

Goal programming form

$$\max\ \Gamma$$
$$\text{s.t.}\ \max_u\ f_i(x,u) \le 0, i = 1,\dots,n$$
$$\|u\| \le \Gamma$$
$$f_0(x) \le (1+\delta) f_0^*,\ \ \delta \ge 0$$

RO form	Γ	δ	PoF	Goal form	δ	Γ	PoF
	0.00	2.5×10^{-4}	0.94		-	-	-
	0.10	0.057	0.87		0.057	0.10	0.87
	0.20	0.118	0.76		0.118	0.20	0.76
	0.30	0.183	0.60		0.183	0.30	0.60
	0.40	0.252	0.38		0.252	0.40	0.38
	0.50	0.326	0.20		0.326	0.50	0.21
	0.60	0.406	0.10		0.406	0.60	0.10
	0.70	0.492	0.07		0.492	0.70	0.07
	0.80	0.583	0.04		0.583	0.80	0.04
	0.90	0.681	0.01		0.681	0.90	0.01
	1.00	0.787	0.00		0.787	1.00	0.00

Suggests a good formulation for multi-objective design space exploration:

$$f_{0,j}(x) \le (1+\delta_j) f_{0,j}^*,\ \ \delta_j \ge 0,\ \ j = 1,\dots,m$$

Applications

Effective integration of simulations and data into aerospace design

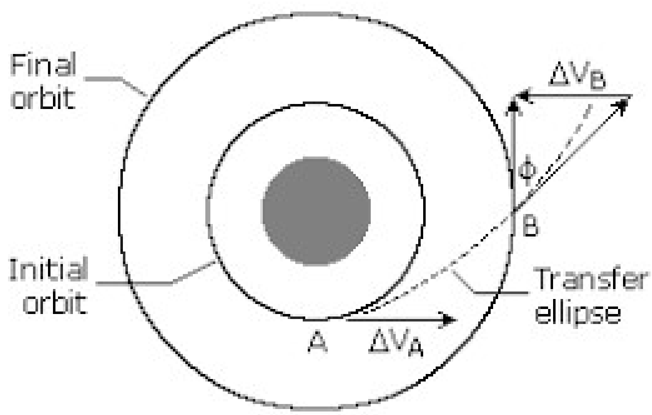
Aurora D8

Virgin Hyperloop

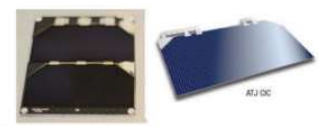
Electra.aero

Efficient discrete decisions in design

Scheduling



Component selection



$\eta_A(-)$	0.305	0.275
$\rho_A(kg/m^2)$	5.35	0.84

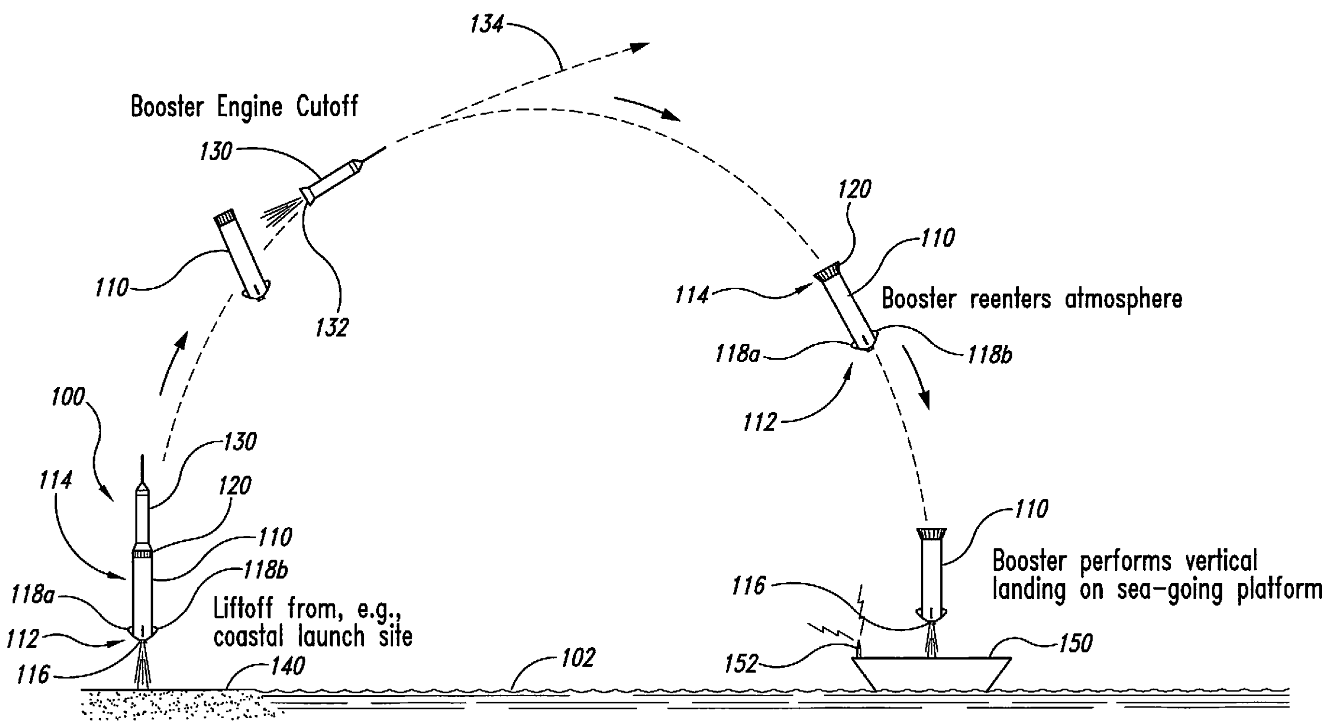
Figure 1. Solar panel catalog

$E_B(kJ)$	138.6	144	144	165.6	1,607.04
$m_B(kg)$	0.253	0.310	0.355	0.710	3.95

Figure 2. Battery catalog

Norheim, J. (2020). Satellite Component Selection with Mixed Integer Nonlinear Programming. *IEEE Aerospace Conference Proceedings*.

Nonlinear dynamics and control (1)

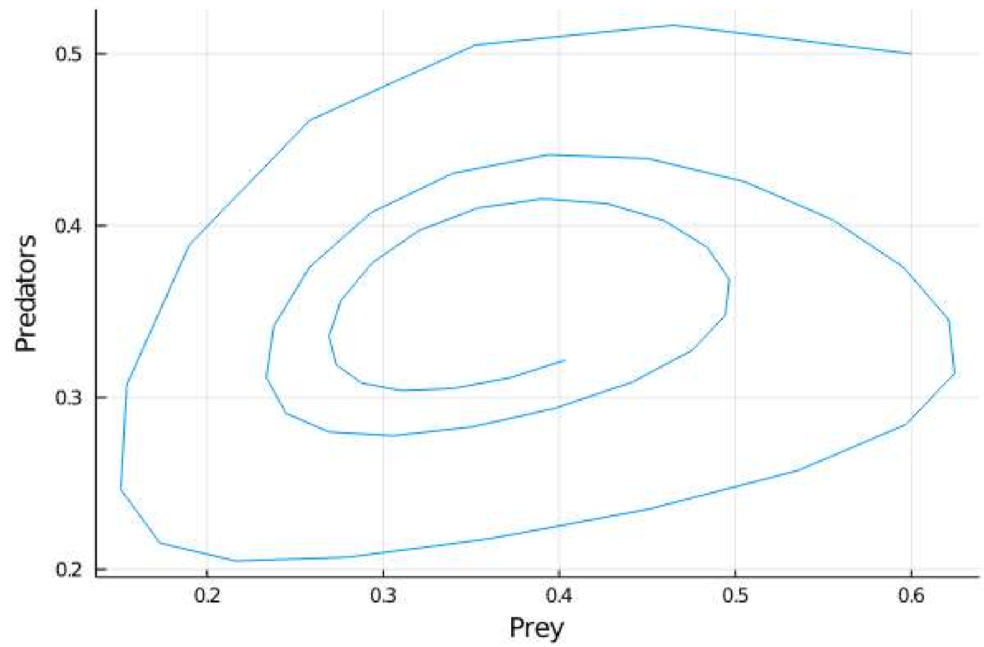


Bezos, J. P., Lai, G., & Findlay, S. R. (2014). *Sea Landing of Space Launch Vehicles and Associated Systems and Methods* (Patent No. US 8,678,321 B2).

Nonlinear dynamics and control (2)

Example Lotka-Volterra population model.

$$\frac{dx}{dt} = x(a - by)$$
$$\frac{dy}{dt} = y(-c + dx)$$



Contributions

A **Global Optimization** method to optimize over objectives and constraints that is

- General to addressing explicit, inexplicit and data-driven constraints with bounded decision variables,
- Compatible with mixed-integer convex optimization,
- Tractable and effective at addressing real world problems.