POZNAN UNIVERSITY OF TECHNOLOGY

SYNTHESIS OF MATHEMATICAL PROGRAMMING MODELS FROM DATA

Tomasz P. Pawlak, Maciej Buzalski

Institute of Computing Science
Poznan University of Technology
Poznań, Poland

29.05.2018

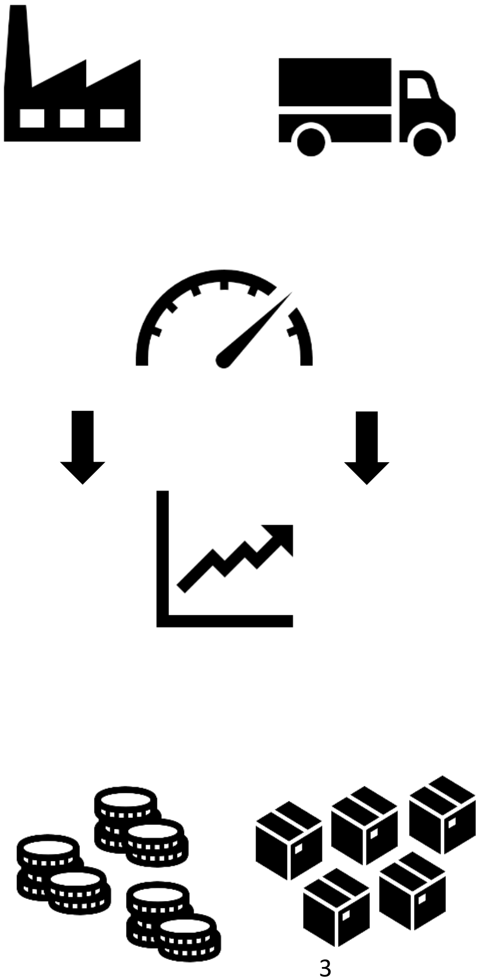
Parts of this work were supported by the Foundation for Polish Science and the National Science Centre, Poland, grant no. 2016/23/D/ST6/03735.

Outline

- Motivation
- Model Synthesis Problem
  - Decomposition
  - Types of problems
  - Challenges
- Model Synthesis using
  - Linear Programming
  - Genetic Programming
  - Decision trees
  - Evolutionary Strategy
  - Local Search

Motivation: a use case

- Consider a real-world business process, e.g.,
  - Manufacturing of a product
  - Delivery of goods
- Goal #1: Simulation, e.g.,
  - To estimate operating costs
  - To estimate production volume
- Goal #2: Optimization, e.g.,
  - To minimize operating costs
  - To maximize production volume
- To achieve these goals one needs a mathematical model of the business process



A Mathematical Programming (MP) Model

- Variables
  - Represent the input, output and parameters of the business process
  - Specified with domains: Real, Integer, Binary etc.
- Objective function
  - The goal to achieve
- Constraints
  - Specify relationships between variables, e.g., operating conditions
  - Manual construction is time-consuming:
    - Requires deep insight into the business process
    - Requires transformation to form accepted by a solver, e.g., linear
    - Humans are error-prone and errors in constraints are expensive
- Simulation and optimization achieved using solvers, e.g.,

GUROBI OPTIMIZATION G IBM CPLEX IBEX CHOCO

MODEL SYNTHESIS PROBLEM

General Model Synthesis Problem

- Input:
  - Examples of states (values of variables)
    - Acquired by e.g., recording operations of the process
  - A class of MP model to synthesize
    - E.g., Linear Programming, Quadratic Programming, etc.
- Output:
  - An objective function representing the outcome of the business process
  - A set of constraints comply with the examples

This is a regression problem

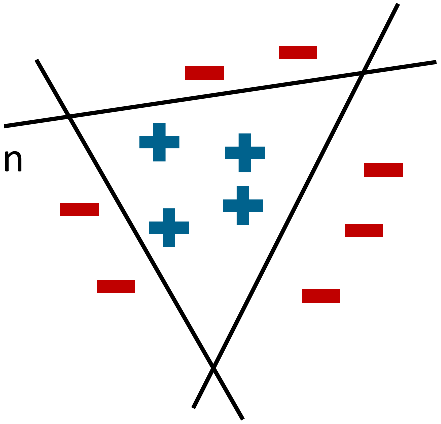
This is a classification problem

Problem decomposition

- The syntheses of an objective function and the constraints are largely independent
- The constraints define what is the feasible solution, and the objective function assesses the quality of this solution
- A single set of constraints can made up an MP model with an arbitrary objective function:
  - It does not matter for the constraints whether an objective function calculates a time cost, monetary profit, or waste of a material
- The synthesis of the objective function as a regression against a specific variable of the problem has many existing solutions
- The synthesis of the constraints surprisingly gained only a little attention in the state-of-the-art works

The remaining of this presentation is on constraint synthesis

Two-Class Constraint Synthesis Problem (2-CSP)



- Input:
  - Set X of examples **x** labeled as
    - Feasible – represent the states reached during normal execution
    - Infeasible – erroneous, faulty or undesired states
  - A class of constraints to synthesize
    - E.g., linear, quadratic, etc.
- Output:
  - A set C of constraints in the form of

$$p(\mathbf{x}) \leq a$$

  - Where $p(\mathbf{x})$ is a function of the given class and a is a constant
  - Such that
    - The number of feasible examples satisfying all constraints in C is maximized (true positives)
    - The number of infeasible examples violating at least one constraint in C is maximized (true negatives)

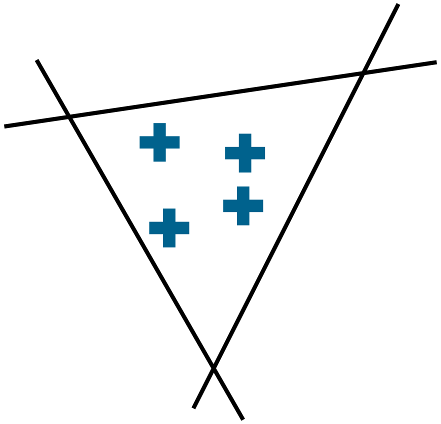
2-CSP is NP-hard

- The problem of determining whether two sets of examples are separable using a fixed number of $k \geq 2$ linear constraints is NP-complete [1].
- It is NP-complete even if k is not fixed but bounded by the square of the number of dimensions n, i.e., $2 \leq k \leq n^2$ [2].
- In consequence, synthesizing $k \geq 2$ linear constraints, where k is either fixed or bounded by n^2 , is NP-hard.
- The complexity of learning $k \geq 2$ non-linear constraints is an open question.

[1] Nimrod Megiddo, "On the complexity of polyhedral separability", Discrete & Computational Geometry 3, 4 (1988), pp. 325 – 337.

[2] Avrim L. Blum and Ronald L. Rivest, "Training a 3-node neural network is NP-complete", *Neural Networks* 5, 1 (1992), pp. 117 – 127.

One-Class Constraint Synthesis Problem (1-CSP)



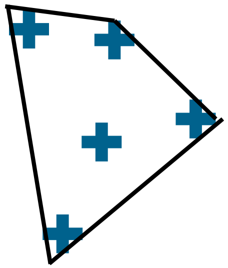
- Input:
  - Set X of examples **x**
    - No labels
    - The examples are assumed to represent feasible states
  - A class of constraints to synthesize
    - E.g., linear, quadratic, etc.
- Output:
  - A set C of constraints in the form of

$$p(\mathbf{x}) \leq a$$

  - Where $p(\mathbf{x})$ is a function of the given class and a is a constant
  - Such that
    - The number of feasible examples satisfying all constraints in C is maximized (true positives)
    - The margin of the constraints to the closest examples is minimized

Properties of 1-CSP

- This is a one-class classification problem
- Time-complexity of 1-CSP for k ≥ 2 constrains is an open question
- Assuming linear constraints, the convex hull co(X) is the optimal solution, because:
  - All examples from X are included in co(X),
  - The margin is 0 for all constraints, as the examples are the vertexes of co(X)



- The number of facets of a convex hull grows exponentially with dimensionality n, and so the time of calculating co(X)
  - Unfortunately, I have no formal proof for time-complexity w.r.t. n.

Convex hull is not the best solution when generalization is under consideration

- An optimal solution of an LP model made of
  - Convex hull-based constraints
  - Any linear objective function

  is an example from the training set X
- Hence:
  - Optimization of a convex hull-based LP model is futile
  - We cannot find a solution that we have not known before
  - Also, it is often more efficient to enumerate the known solutions from X than actually solving an LP model with a large number of constraints
- In general:
  - Too tight constraints limit generalization and may cause an optimal solution to be suboptimal in practice
  - Too loose constraints are detrimental too, as they may allow for solutions that happen inapplicable in practice

The curse of dimensionality

- Assume that:
  - The optimal solution is a constraint resembling an n-ball centered in **y** with radius r and volume $V \mu r^n$
  - An algorithm produces a constraint resembling an n-ball centered in **y**, but commits an error in the radius by extending it by infinitely small $\P r > 0$
- Then:
  - The volume of the synthesized n-ball is $V\square\!\propto (r+\partial r)^n$
  - It is exponentially with n greater than the volume of the optimal n-ball

$$\frac{V\square}{V} \propto \left(1+\frac{\partial r}{r}\right)^n \quad \text{and} \quad \lim_{n\to\square}\frac{V\square}{V}=\square$$

  - The volume of the margin between the n-balls grows exponentially with n
- In high dimensions, an apparently negligible error in a constraint may cause dramatic deterioration of virtually any data-set-backed performance measure for this constraint, as the margin between it and the optimal constraint may include exponential number of examples

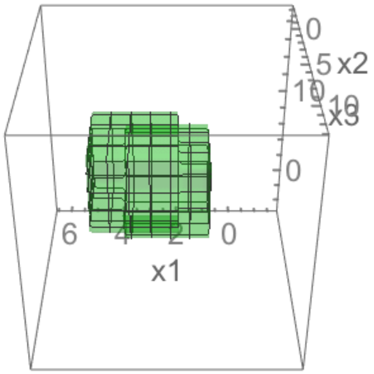
Synthetic performance assessment

- Synthetic benchmarks – MP models with parameters:
  - n – number of variables
  - k – number of alternative subsets of constraints
- Training sets
  - 1-CSP: Feasible examples uniformly sampled from feasible region of an MP model
  - 2-CSP: Feasible and infeasible examples uniformly sampled from the Cartesian product of domains of variables
- Test sets
  - Feasible and infeasible examples uniformly sampled from the Cartesian product of domains of variables
- Data-based measures of fidelity
  - E.g., accuracy, F_1-score,…
- Syntactic measures of fidelity
  - E.g., angles between the corresponding constraints in the synthesized and the benchmark MP models

Examples of benchmarks

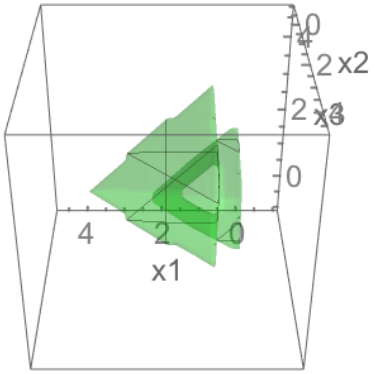
Cube$_n^k$:

$$\begin{aligned}
\forall_{j=1}^{k}\forall_{i=1}^{n} : x_i - Lb_j &\geq ij - L\\
\forall_{j=1}^{k}\forall_{i=1}^{n} : x_i + Lb_j &\leq ij + id + L\\
\textstyle\sum_{j=1}^{k} b_j &\geq 1\\
\forall_{i=1}^{n} : x_i \in &\ [i - ikd, i + 2ikd]\\
\forall_{j=1}^{k} : b_j \in &\ \{0, 1\}
\end{aligned}$$



Simplex$_n^k$:

$$\begin{aligned}
\forall_{j=1}^{k}\forall_{i=1}^{n}\forall_{l=i+1}^{n} : x_i \cot\tfrac{\pi}{12} - x_l \tan\tfrac{\pi}{12} - Lb_j &\geq 2j - 2 - L\\
\forall_{j=1}^{k}\forall_{i=1}^{n}\forall_{l=i+1}^{n} : x_l \cot\tfrac{\pi}{12} - x_i \tan\tfrac{\pi}{12} - Lb_j &\geq 2j - 2 - L\\
\forall_{j=1}^{k} : \textstyle\sum_{i=1}^{n} x_i + Lb_j &\leq jd + L\\
\textstyle\sum_{j=1}^{k} b_j &\geq 1\\
\forall_{i=1}^{n} : x_i \in &\ [-1, 2k + d]\\
\forall_{j=1}^{k} : b_j \in &\ \{0, 1\}
\end{aligned}$$



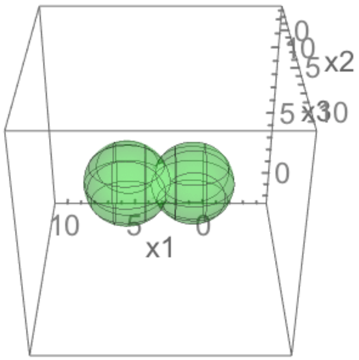
Ball$_n^k$:

$$\forall_{j=1}^{k} \sum_{i=1}^{n} \left(x_i - i - \frac{2\sqrt{6}(j-1)d}{i\pi}\right)^2 + Lb_j \leq d^2 + L$$

$$\sum_{j=1}^{k} b_j \geq 1$$

$$\forall_{i=1}^{n} : x_i \in \left[i - 2d, i + \frac{2\sqrt{6}(k-1)d}{\pi} + 2d\right]$$

$$\forall_{j=1}^{k} : b_j \in \{0, 1\}$$



where:
L is a big constant
d is a parameter = 2.7

Performance assessment on real-world problems

- Let there be a data set of states achieved by a business process
- General workflow consists of four steps:
  1. Synthesize a set of constraints from the data set
  2. Attach an objective function
     - A regression model calculated using the data set, or
     - A known objective function
  3. Optimize the MP model
  4. Validate the optimal solution with the data set
     - E.g., assess similarity to the one or more most similar examples

Challenges for assessing an MP model for a real-world business process

- 1-CSP:
  - The data set consists of feasible examples only
  - False positives and True negatives are 0 for any MP model
  - So, most of the performance measures typical to Machine Learning are biased or cannot be calculated
- 1-CSP & 2-CSP:
  - The optimal solution of an MP model usually lies on a constraint
    - In particular, this holds for all Linear Programming models
  - In statistical sense, the optimal solution is an outlier – it lies on the boundary of a distribution of feasible solutions
  - From the optimization point of view, there is no reason to reward the MP model for the interior of its feasible region; only the boundary matters
  - Hence, assessment of an MP model using a test set is futile:
    - The test set-based measures (usually) reward the MP model equally for each example
    - The test set is a sample of the space that may be far from the boundary of the feasible region
- 2-CSP:
  - The data set is often imbalanced, as infeasible states like errors and faults are avoided
- The actual MP model is unknown
  - We are unable to calculate syntactic measures of fidelity

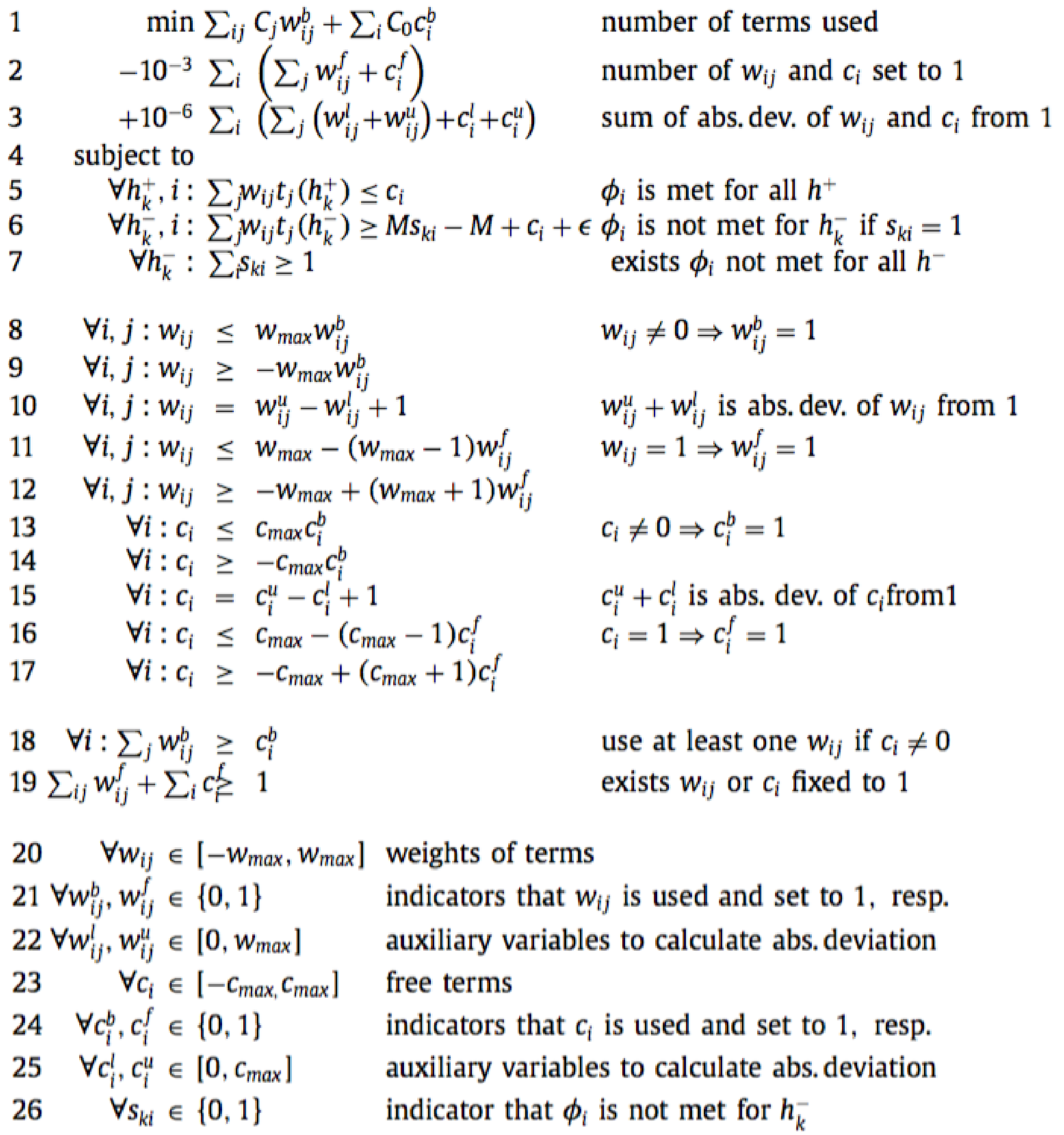
ALGORITHMS

Solving 2-CSP using Mixed-Integer Linear Programming (MILP)

- The general idea is to encode the 2-CSP using a MILP problem and solving optimally
- Ockham's razor:
  - The objective is to find the minimal set of constraints that separate the sets of feasible and infeasible examples
  - Misclassification is disallowed
- The algorithm synthesizes LP models and user-defined classes of NLP models

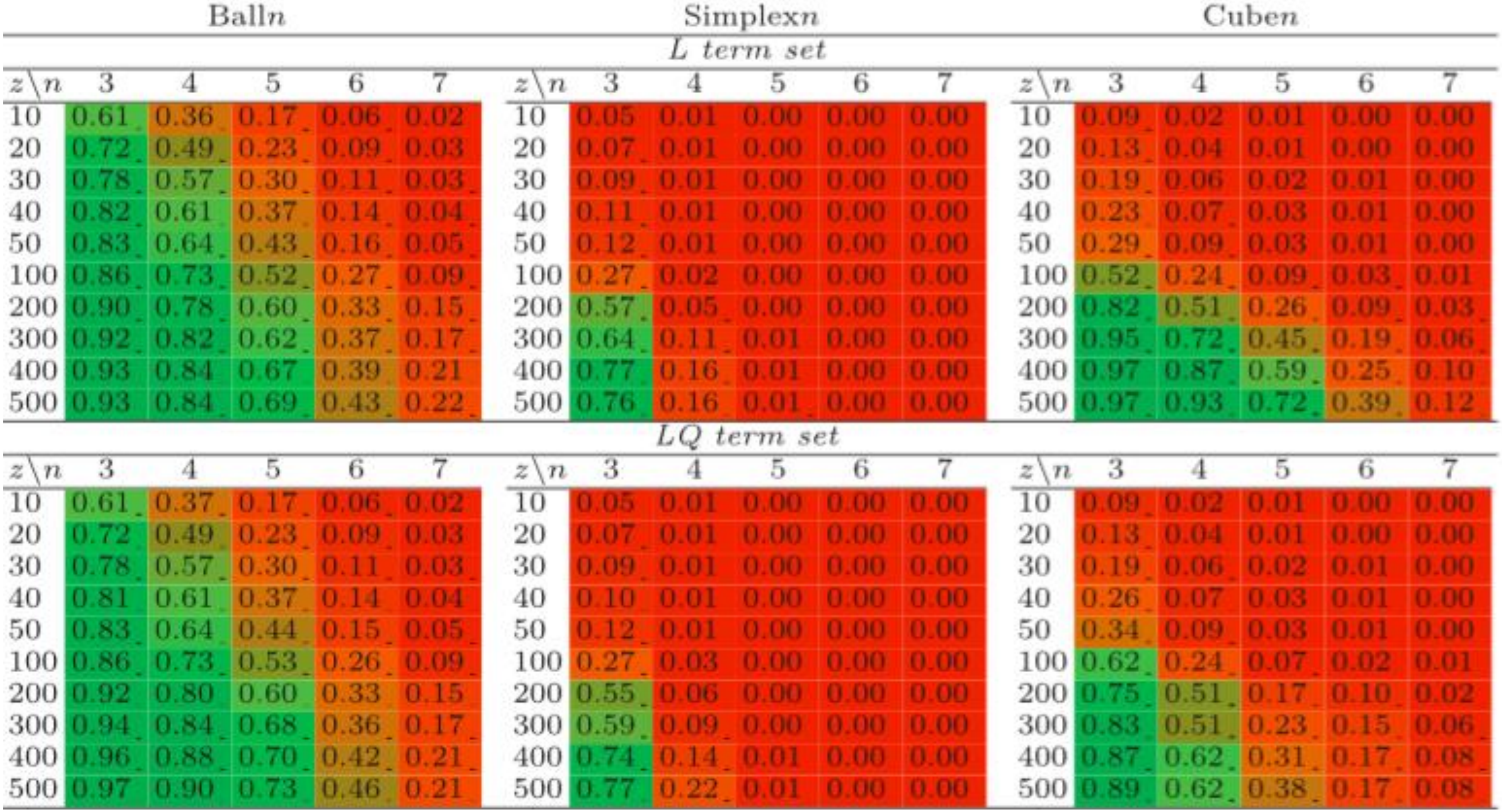
Tomasz P. Pawlak, Krzysztof Krawiec, Automatic synthesis of constraints from examples using mixed integer linear programming, European Journal of Operational Research 261 (2017) 1141-1157.
IF=3.297, 40p MNiSW

Encoding 2-CSP using MILP



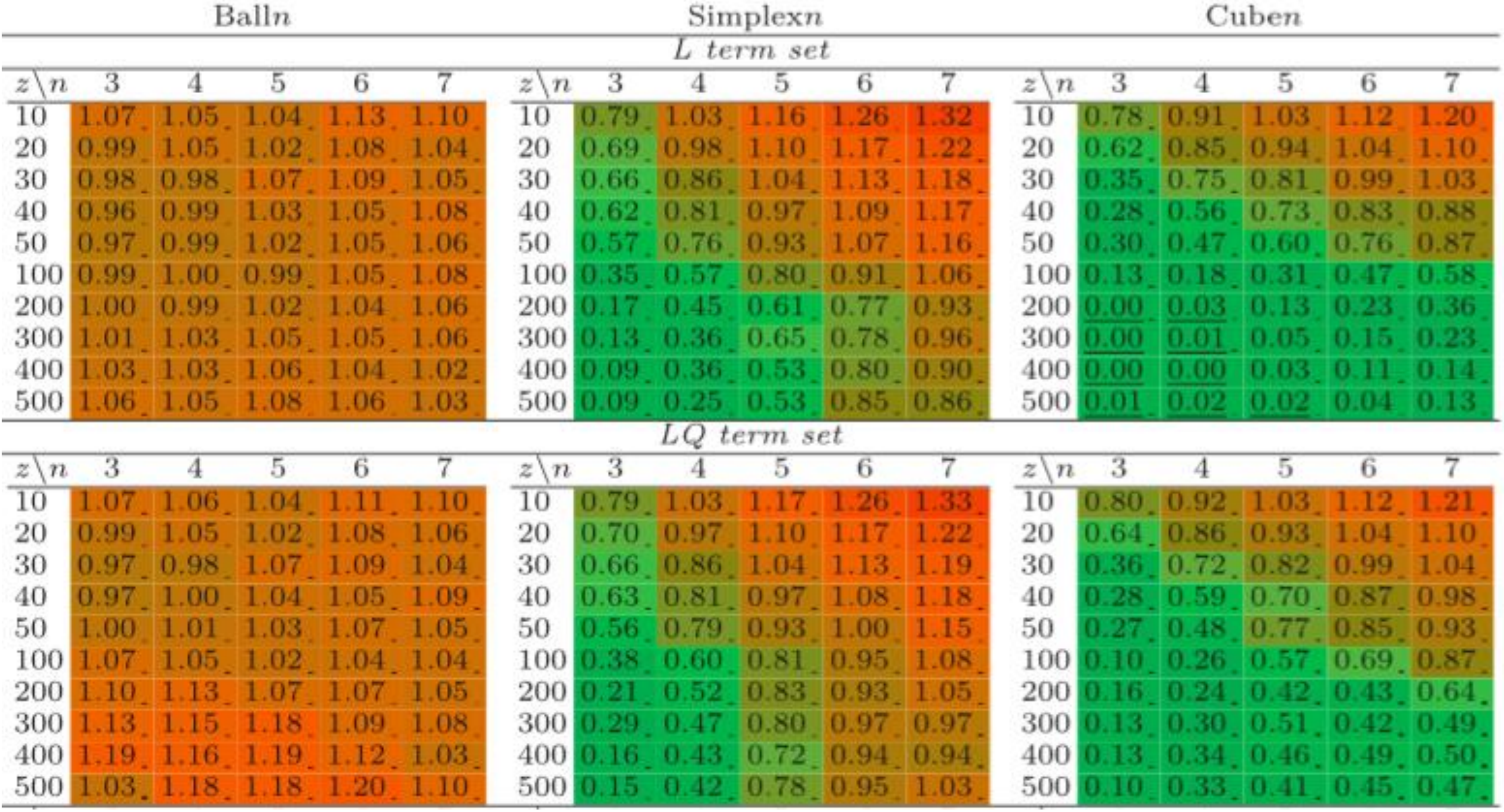
1	$\min \sum_{ij} C_j w^b_{ij} + \sum_i C_0 c^b_i$	number of terms used
2	$-10^{-3} \sum_i \left(\sum_j w^f_{ij} + c^f_i\right)$	number of w_{ij} and c_i set to 1
3	$+10^{-6} \sum_i \left(\sum_j (w^l_{ij} + w^u_{ij}) + c^l_i + c^u_i\right)$	sum of abs. dev. of w_{ij} and c_i from 1
4	subject to	
5	$\forall h^+_k, i : \sum_j w_{ij} t_j(h^+_k) \leq c_i$	ϕ_i is met for all h^+
6	$\forall h^-_k, i : \sum_j w_{ij} t_j(h^-_k) \geq M s_{ki} - M + c_i + \epsilon$	ϕ_i is not met for h^-_k if $s_{ki} = 1$
7	$\forall h^-_k : \sum_i s_{ki} \geq 1$	exists ϕ_i not met for all h^-
8	$\forall i, j : w_{ij} \leq w_{max} w^b_{ij}$	$w_{ij} \neq 0 \Rightarrow w^b_{ij} = 1$
9	$\forall i, j : w_{ij} \geq -w_{max} w^b_{ij}$	
10	$\forall i, j : w_{ij} = w^u_{ij} - w^l_{ij} + 1$	$w^u_{ij} + w^l_{ij}$ is abs. dev. of w_{ij} from 1
11	$\forall i, j : w_{ij} \leq w_{max} - (w_{max} - 1) w^f_{ij}$	$w_{ij} = 1 \Rightarrow w^f_{ij} = 1$
12	$\forall i, j : w_{ij} \geq -w_{max} + (w_{max} + 1) w^f_{ij}$	
13	$\forall i : c_i \leq c_{max} c^b_i$	$c_i \neq 0 \Rightarrow c^b_i = 1$
14	$\forall i : c_i \geq -c_{max} c^b_i$	
15	$\forall i : c_i = c^u_i - c^l_i + 1$	$c^u_i + c^l_i$ is abs. dev. of $c_i$from1
16	$\forall i : c_i \leq c_{max} - (c_{max} - 1) c^f_i$	$c_i = 1 \Rightarrow c^f_i = 1$
17	$\forall i : c_i \geq -c_{max} + (c_{max} + 1) c^f_i$	
18	$\forall i : \sum_j w^b_{ij} \geq c^b_i$	use at least one w_{ij} if $c_i \neq 0$
19	$\sum_{ij} w^f_{ij} + \sum_i c^f_i \geq 1$	exists w_{ij} or c_i fixed to 1
20	$\forall w_{ij} \in [-w_{max}, w_{max}]$	weights of terms
21	$\forall w^b_{ij}, w^f_{ij} \in \{0, 1\}$	indicators that w_{ij} is used and set to 1, resp.
22	$\forall w^l_{ij}, w^u_{ij} \in [0, w_{max}]$	auxiliary variables to calculate abs. deviation
23	$\forall c_i \in [-c_{max}, c_{max}]$	free terms
24	$\forall c^b_i, c^f_i \in \{0, 1\}$	indicators that c_i is used and set to 1, resp.
25	$\forall c^l_i, c^u_i \in [0, c_{max}]$	auxiliary variables to calculate abs. deviation
26	$\forall s_{ki} \in \{0, 1\}$	indicator that ϕ_i is not met for h^-_k

Jaccard indexes of the feasible regions of the synthesized and the actual MP models



	Ball$n$						Simplex$n$						Cube$n$				
							L term set										
$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7
10	0.61	0.36	0.17	0.06	0.02	10	0.05	0.01	0.00	0.00	0.00	10	0.09	0.02	0.01	0.00	0.00
20	0.72	0.49	0.23	0.09	0.03	20	0.07	0.01	0.00	0.00	0.00	20	0.13	0.04	0.01	0.00	0.00
30	0.78	0.57	0.30	0.11	0.03	30	0.09	0.01	0.00	0.00	0.00	30	0.19	0.06	0.02	0.01	0.00
40	0.82	0.61	0.37	0.14	0.04	40	0.11	0.01	0.00	0.00	0.00	40	0.23	0.07	0.03	0.01	0.00
50	0.83	0.64	0.43	0.16	0.05	50	0.12	0.01	0.00	0.00	0.00	50	0.29	0.09	0.03	0.01	0.00
100	0.86	0.73	0.52	0.27	0.09	100	0.27	0.02	0.00	0.00	0.00	100	0.52	0.24	0.09	0.03	0.01
200	0.90	0.78	0.60	0.33	0.15	200	0.57	0.05	0.00	0.00	0.00	200	0.82	0.51	0.26	0.09	0.03
300	0.92	0.82	0.62	0.37	0.17	300	0.64	0.11	0.01	0.00	0.00	300	0.95	0.72	0.45	0.19	0.06
400	0.93	0.84	0.67	0.39	0.21	400	0.77	0.16	0.01	0.00	0.00	400	0.97	0.87	0.59	0.25	0.10
500	0.93	0.84	0.69	0.43	0.22	500	0.76	0.16	0.01	0.00	0.00	500	0.97	0.93	0.72	0.39	0.12
							LQ term set										
$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7
10	0.61	0.37	0.17	0.06	0.02	10	0.05	0.01	0.00	0.00	0.00	10	0.09	0.02	0.01	0.00	0.00
20	0.72	0.49	0.23	0.09	0.03	20	0.07	0.01	0.00	0.00	0.00	20	0.13	0.04	0.01	0.00	0.00
30	0.78	0.57	0.30	0.11	0.03	30	0.09	0.01	0.00	0.00	0.00	30	0.19	0.06	0.02	0.01	0.00
40	0.81	0.61	0.37	0.14	0.04	40	0.10	0.01	0.00	0.00	0.00	40	0.26	0.07	0.03	0.01	0.00
50	0.83	0.64	0.44	0.15	0.05	50	0.12	0.01	0.00	0.00	0.00	50	0.34	0.09	0.03	0.01	0.00
100	0.86	0.73	0.53	0.26	0.09	100	0.27	0.03	0.00	0.00	0.00	100	0.62	0.24	0.07	0.02	0.01
200	0.92	0.80	0.60	0.33	0.15	200	0.55	0.06	0.00	0.00	0.00	200	0.75	0.51	0.17	0.10	0.02
300	0.94	0.84	0.68	0.36	0.17	300	0.59	0.09	0.00	0.00	0.00	300	0.83	0.51	0.23	0.15	0.06
400	0.96	0.88	0.70	0.42	0.21	400	0.74	0.14	0.01	0.00	0.00	400	0.87	0.62	0.31	0.17	0.08
500	0.97	0.90	0.73	0.46	0.21	500	0.77	0.22	0.01	0.00	0.00	500	0.89	0.62	0.38	0.17	0.08

Mean angles between the corresponding constraints in the synthesized and the actual MP models



	Balln						Simplexn						Cuben				
							L term set										
$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7
10	1.07	1.05	1.04	1.13	1.10	10	0.79	1.03	1.16	1.26	1.32	10	0.78	0.91	1.03	1.12	1.20
20	0.99	1.05	1.02	1.08	1.04	20	0.69	0.98	1.10	1.17	1.22	20	0.62	0.85	0.94	1.04	1.10
30	0.98	0.98	1.07	1.09	1.05	30	0.66	0.86	1.04	1.13	1.18	30	0.35	0.75	0.81	0.99	1.03
40	0.96	0.99	1.03	1.05	1.08	40	0.62	0.81	0.97	1.09	1.17	40	0.28	0.56	0.73	0.83	0.88
50	0.97	0.99	1.02	1.05	1.06	50	0.57	0.76	0.93	1.07	1.16	50	0.30	0.47	0.60	0.76	0.87
100	0.99	1.00	0.99	1.05	1.08	100	0.35	0.57	0.80	0.91	1.06	100	0.13	0.18	0.31	0.47	0.58
200	1.00	0.99	1.02	1.04	1.06	200	0.17	0.45	0.61	0.77	0.93	200	0.00	0.03	0.13	0.23	0.36
300	1.01	1.03	1.05	1.05	1.06	300	0.13	0.36	0.65	0.78	0.96	300	0.00	0.01	0.05	0.15	0.23
400	1.03	1.03	1.06	1.04	1.02	400	0.09	0.36	0.53	0.80	0.90	400	0.00	0.00	0.03	0.11	0.14
500	1.06	1.05	1.08	1.06	1.03	500	0.09	0.25	0.53	0.85	0.86	500	0.01	0.02	0.02	0.04	0.13
							LQ term set										
$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7	$z\backslash n$	3	4	5	6	7
10	1.07	1.06	1.04	1.11	1.10	10	0.79	1.03	1.17	1.26	1.33	10	0.80	0.92	1.03	1.12	1.21
20	0.99	1.05	1.02	1.08	1.06	20	0.70	0.97	1.10	1.17	1.22	20	0.64	0.86	0.93	1.04	1.10
30	0.97	0.98	1.07	1.09	1.04	30	0.66	0.86	1.04	1.13	1.19	30	0.36	0.72	0.82	0.99	1.04
40	0.97	1.00	1.04	1.05	1.09	40	0.63	0.81	0.97	1.08	1.18	40	0.28	0.59	0.70	0.87	0.98
50	1.00	1.01	1.03	1.07	1.05	50	0.56	0.79	0.93	1.00	1.15	50	0.27	0.48	0.77	0.85	0.93
100	1.07	1.05	1.02	1.04	1.04	100	0.38	0.60	0.81	0.95	1.08	100	0.10	0.26	0.57	0.69	0.87
200	1.10	1.13	1.07	1.07	1.05	200	0.21	0.52	0.83	0.93	1.05	200	0.16	0.24	0.42	0.43	0.64
300	1.13	1.15	1.18	1.09	1.08	300	0.29	0.47	0.80	0.97	0.97	300	0.13	0.30	0.51	0.42	0.49
400	1.19	1.16	1.19	1.12	1.03	400	0.16	0.43	0.72	0.94	0.94	400	0.13	0.34	0.46	0.49	0.50
500	1.03	1.18	1.18	1.20	1.10	500	0.15	0.42	0.78	0.95	1.03	500	0.10	0.33	0.41	0.45	0.47

Modeling of concrete

- Based on the slump test data-set [1]

https://en.wikipedia.org/wiki/Concrete_slump_test

[1] Yeh, I.-C. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites, 29*(6), 474–480.

QCQP models of concrete

Class S1

$$\begin{aligned} CoarseAggr(0.0073FlyAsh - 0.002893Cement) - 3.513Slag + \\ SP(Water - 1.044FlyAsh) \leq 0 \end{aligned}$$

Class S2

$$Water - 0.003818Slag * Cement \leq 0$$

Class S3

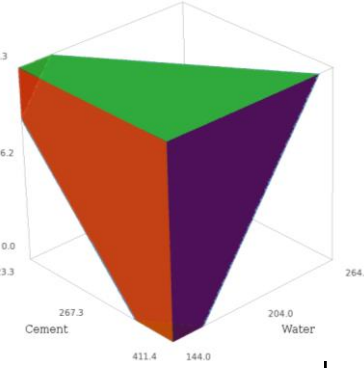
$$\begin{aligned} Water(37.68 - 0.02183CoarseAggr - 0.02857FineAggr - 0.02449Slag) + \\ CoarseAggr(0.003091FineAggr - 0.003236Cement) + \\ FlyAsh(0.005181FlyAsh + 0.01029Slag - 4.589) - 7.647SP \geq 1 \\ CoarseAggr(0.337 - 0.000835FineAggr) + Water \leq 0 \end{aligned}$$

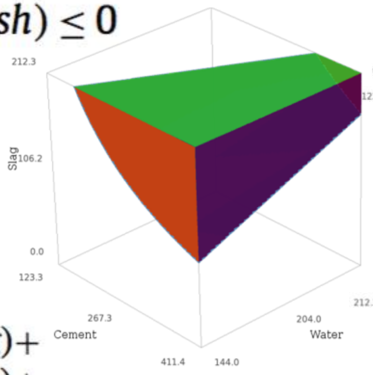
Class S4

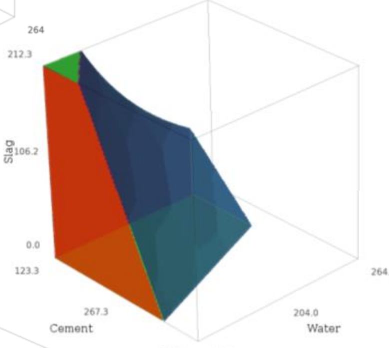
$$\begin{aligned} FlyAsh(18.69 - 0.01979FlyAsh) - 30.17Slag - 0.03053Water^2 + \\ FineAggr(0.03787Slag + 0.01417Water - 0.01539FlyAsh) + \\ CoarseAggr(0.005304Slag - 0.7613) - 0.6043Cement \leq 679.5 \\ 0.008107Water^2 - 0.6571Cement - SP - Slag \geq 1 \\ Cement - 0.002028Cement^2 + 0.04184Slag + 0.554Water \leq 235.1 \\ Cement \leq 354 \end{aligned}$$

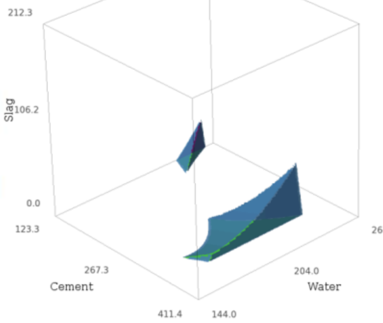
Class S5

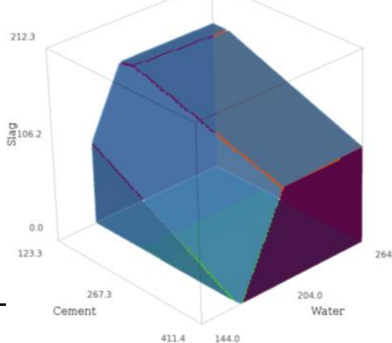
$$\begin{aligned} Slag(0.06552Cement + 0.2676Slag + SP - 0.08421FineAggr) + \\ FlyAsh(SP - 7.254) + Water(6.772 - 0.9654SP) \leq 0 \\ Slag(34.12 - 7.593SP) + 0.002812FineAggr * SP - 0.06889Cement \leq 0 \\ 0.0009766Cement^2 - 0.2784FlyAsh + CoarseAggr - 1.424FineAggr \leq 0 \\ Slag + SP + 0.2611CoarseAggr - 0.5486FineAggr \leq 1 \\ 3.301Water + 13.8SP - 0.3316Cement - Slag \geq 550 \\ Water \geq 171.3 \end{aligned}$$









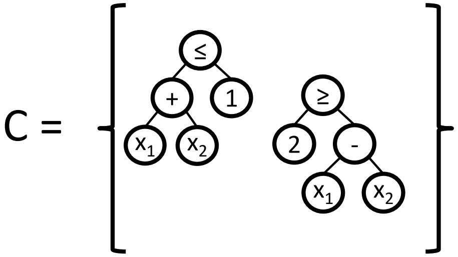


Conclusions

- Solving a MILP problem is NP-hard and so we had to terminate solver prematurely for large problem instances
- The algorithm overfits and is susceptible to noise
  - The MILP problem is aimed at minimizing model complexity while guarantying separation of the feasible and the infeasible examples
- The feasible region is often overestimated
  - The algorithm strives to use as simple constraints as possible and in the effect the feasible region features many outlying vertexes
- The training set is typically imbalanced with higher share of the feasible examples, while the algorithm requires higher share of the infeasible examples to avoid overestimating the feasible region

GENETICS: Genetic Programming Constraint Synthesis for 2-CSP

- Representation
  - An individual C is a set of constraints
  - A constraint is an Abstract Syntax Tree (AST)
- Strongly-typed Genetic Programming
  - Tree root is one of ≤, =, ≥
  - ≤, =, ≥ accept any other instructions as arguments
- Instruction sets
  - For Linear Programming models:
    - +, -, ×, x_i, 1, ERC
      where × is multiplication that accepts +, -, 1, ERC as its left-hand argument
  - For polynomial models:
    - +, -, ×, *, x^2, x_i, 1, ERC
      where * accepts all instructions as arguments



Tomasz P. Pawlak, Krzysztof Krawiec, Synthesis of Mathematical Programming Constraints with Genetic Programming, EuroGP 2017, Lecture Notes in Computer Science 10196:178-193, Springer, 2017.

GENETICS: Fitness function

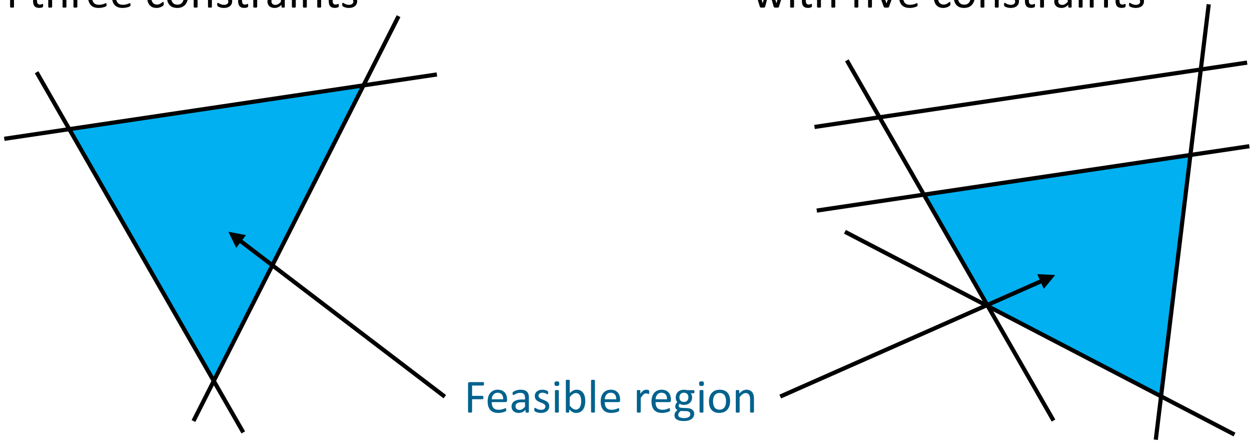
- Assessment of C on example **x** is:
  - For a feasible example **x**:
    - The number of constraints in C violated on **x**
  - For an infeasible example **x**:
    - 1 if all constraints in C are met for **x**, 0 otherwise
- Parsimony pressure tests
  - Minimize the number of constraints in C
  - Minimize the total number of nodes in constraints in C
- Assessment of C is a vector of assessments on examples and parsimony pressure tests
  - Lexicase selection runs on these vectors
- The best-of-run individual C minimizes a sum of vector elements

GENETICS: GP operators

- Ramped Half-and-Half (RHH) initialization
  - Grow and Full build individual constraints
  - Total number of constraints initialized in C is parameterized:
    - Grow draws it from the parameterized range
    - Full uses the maximum in that range

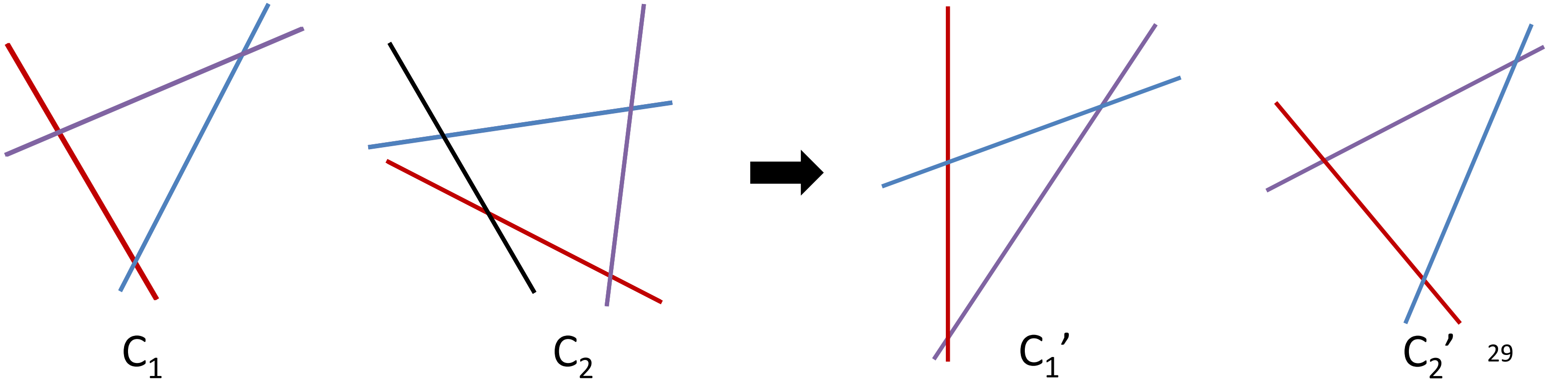
Grow-initialized individual with three constraints

Full-initialized individual with five constraints



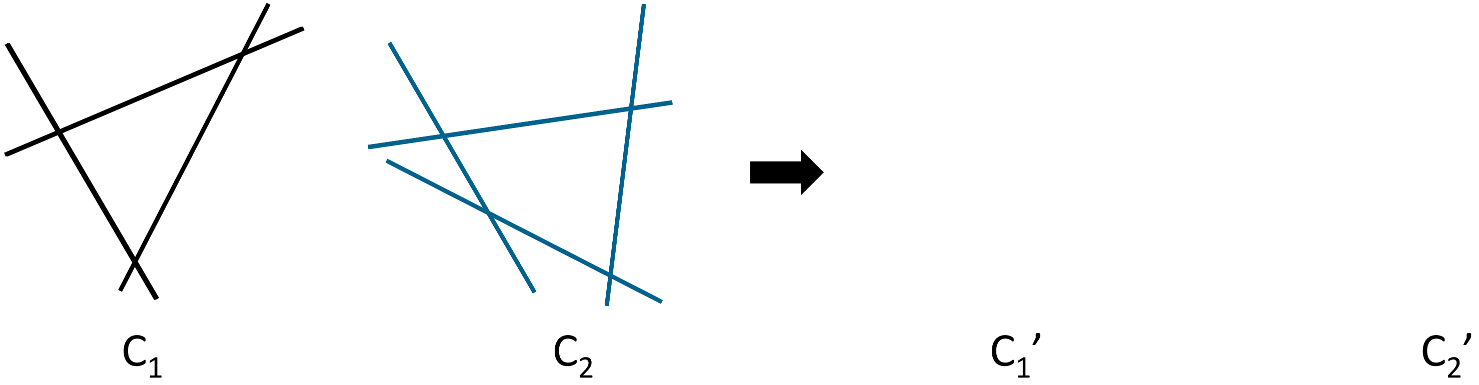
GenetiCS: GP operators

- Constraint Tree Crossover (CTX)
  - Draws one constraint from each parent C_1 and C_2
  - Runs Tree Swapping Crossover and inserts crossed-over constraints into corresponding offspring C_1' and C_2'
- Constraint Tree Mutation (CTM)
  - Initializes a random individual C_r using RHH
  - Runs CTX for the given parent C and C_r and returns one of resulting offspring



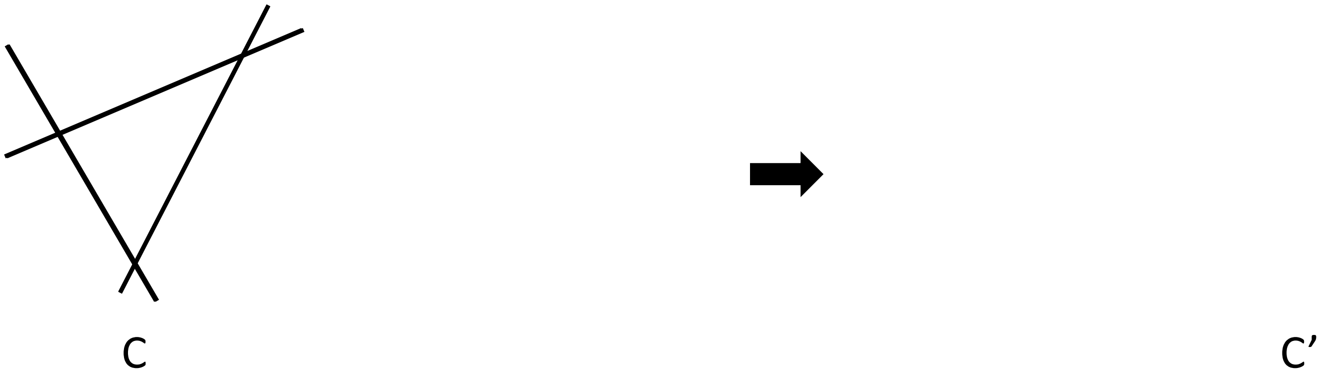
GENETICS: GP operators

- Constraint Swapping Crossover (CSX)
  - Given two parents C_1 and C_2, their constraints are randomly assigned to two offspring C_1' and C_2'
- Constraint Swapping Mutation (CSM)
  - Initializes a random individual C_r using RHH
  - Runs CSX for the given parent C and C_r and returns one of resulting offspring



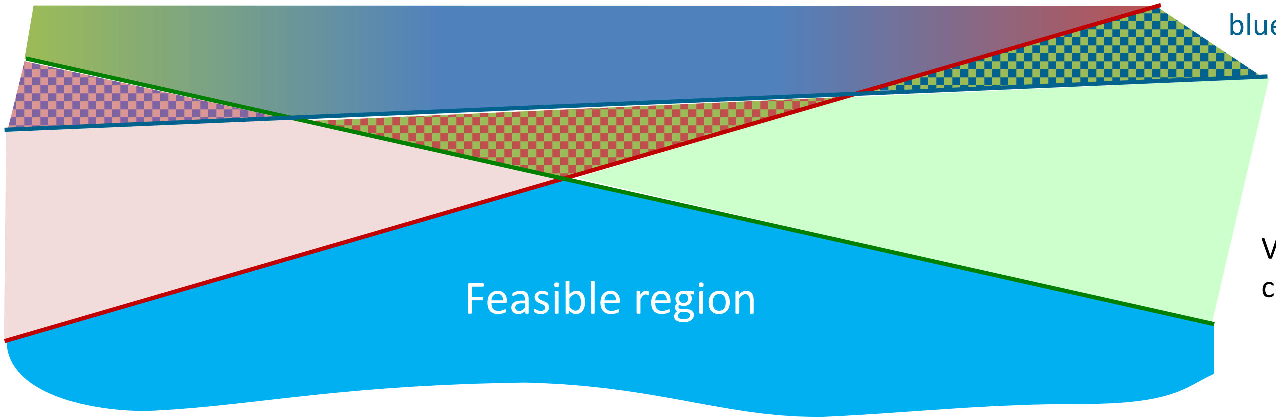
GENETICS: GP operators

- Gaussian Constant Mutation (GCM)
  - Draws a constant c from each constraint in the given parent C
  - Replaces c with c’~N(c, 1)



GENETICS: Post-processing

- Remove redundant constraints from the best-or-run individual:
  - A constraint is redundant if for all solutions it is either satisfied or another violated constraint exists



Violates green, blue and red constraints

Violates red and blue constraints

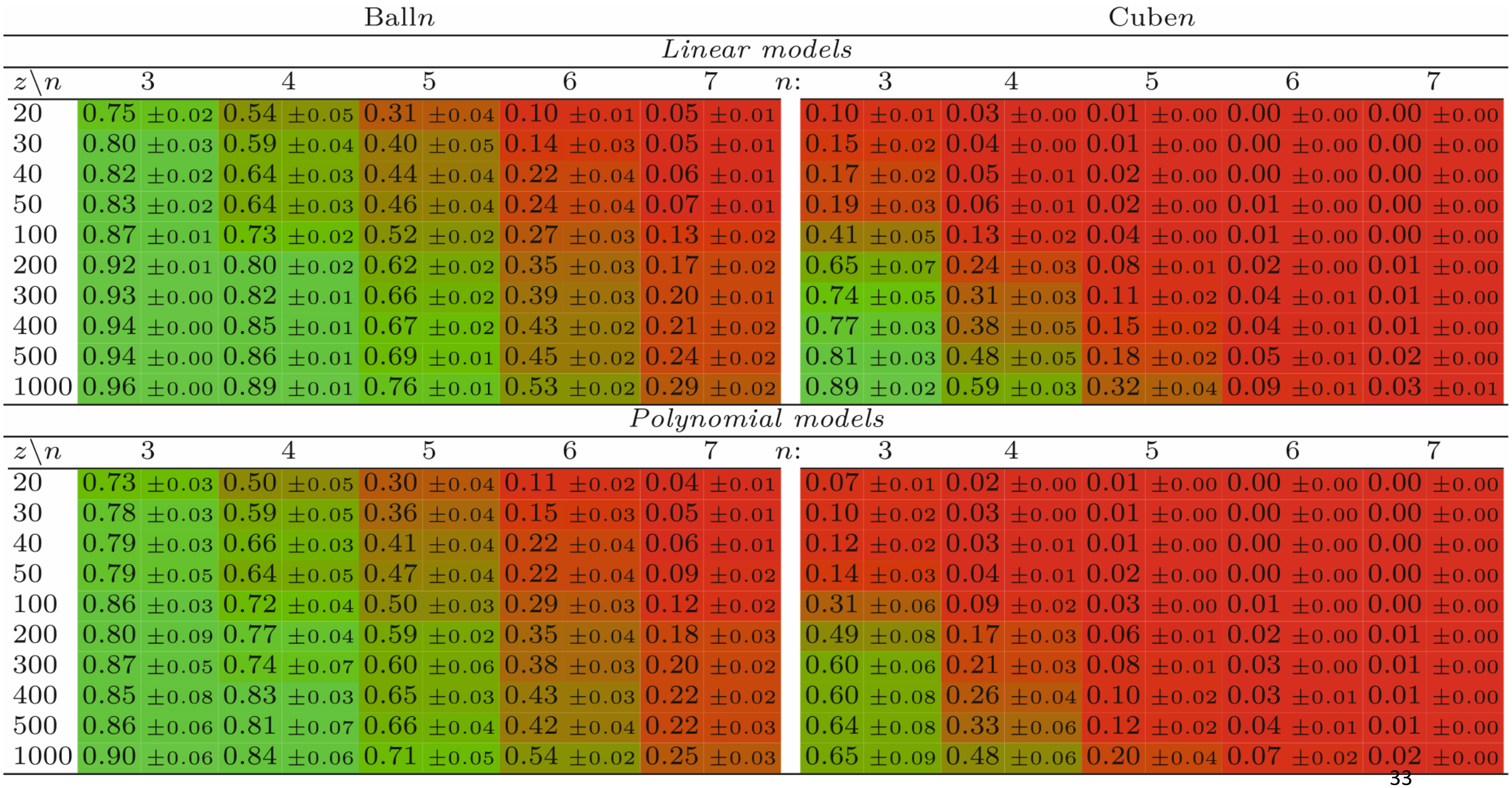
Violates green and blue constraints

Violates red constraint only

Violates green constraint only

No region contains solutions that violate blue constraint only: Blue constraint is redundant.

Mean Jaccard index of the feasible regions of the synthesized and the actual MP models



	Ball*n*						Cube*n*				
					Linear models						
z\n	3	4	5	6	7	*n*:	3	4	5	6	7
20	0.75 ±0.02	0.54 ±0.05	0.31 ±0.04	0.10 ±0.01	0.05 ±0.01		0.10 ±0.01	0.03 ±0.00	0.01 ±0.00	0.00 ±0.00	0.00 ±0.00
30	0.80 ±0.03	0.59 ±0.04	0.40 ±0.05	0.14 ±0.03	0.05 ±0.01		0.15 ±0.02	0.04 ±0.00	0.01 ±0.00	0.00 ±0.00	0.00 ±0.00
40	0.82 ±0.02	0.64 ±0.03	0.44 ±0.04	0.22 ±0.04	0.06 ±0.01		0.17 ±0.02	0.05 ±0.01	0.02 ±0.00	0.00 ±0.00	0.00 ±0.00
50	0.83 ±0.02	0.64 ±0.03	0.46 ±0.04	0.24 ±0.04	0.07 ±0.01		0.19 ±0.03	0.06 ±0.01	0.02 ±0.00	0.01 ±0.00	0.00 ±0.00
100	0.87 ±0.01	0.73 ±0.02	0.52 ±0.02	0.27 ±0.03	0.13 ±0.02		0.41 ±0.05	0.13 ±0.02	0.04 ±0.00	0.01 ±0.00	0.00 ±0.00
200	0.92 ±0.01	0.80 ±0.02	0.62 ±0.02	0.35 ±0.03	0.17 ±0.02		0.65 ±0.07	0.24 ±0.03	0.08 ±0.01	0.02 ±0.00	0.01 ±0.00
300	0.93 ±0.00	0.82 ±0.01	0.66 ±0.02	0.39 ±0.03	0.20 ±0.01		0.74 ±0.05	0.31 ±0.03	0.11 ±0.02	0.04 ±0.01	0.01 ±0.00
400	0.94 ±0.00	0.85 ±0.01	0.67 ±0.02	0.43 ±0.02	0.21 ±0.02		0.77 ±0.03	0.38 ±0.05	0.15 ±0.02	0.04 ±0.01	0.01 ±0.00
500	0.94 ±0.00	0.86 ±0.01	0.69 ±0.01	0.45 ±0.02	0.24 ±0.02		0.81 ±0.03	0.48 ±0.05	0.18 ±0.02	0.05 ±0.01	0.02 ±0.00
1000	0.96 ±0.00	0.89 ±0.01	0.76 ±0.01	0.53 ±0.02	0.29 ±0.02		0.89 ±0.02	0.59 ±0.03	0.32 ±0.04	0.09 ±0.01	0.03 ±0.01
					Polynomial models						
z\n	3	4	5	6	7	*n*:	3	4	5	6	7
20	0.73 ±0.03	0.50 ±0.05	0.30 ±0.04	0.11 ±0.02	0.04 ±0.01		0.07 ±0.01	0.02 ±0.00	0.01 ±0.00	0.00 ±0.00	0.00 ±0.00
30	0.78 ±0.03	0.59 ±0.05	0.36 ±0.04	0.15 ±0.03	0.05 ±0.01		0.10 ±0.02	0.03 ±0.00	0.01 ±0.00	0.00 ±0.00	0.00 ±0.00
40	0.79 ±0.03	0.66 ±0.03	0.41 ±0.04	0.22 ±0.04	0.06 ±0.01		0.12 ±0.02	0.03 ±0.01	0.01 ±0.00	0.00 ±0.00	0.00 ±0.00
50	0.79 ±0.05	0.64 ±0.05	0.47 ±0.04	0.22 ±0.04	0.09 ±0.02		0.14 ±0.03	0.04 ±0.01	0.02 ±0.00	0.00 ±0.00	0.00 ±0.00
100	0.86 ±0.03	0.72 ±0.04	0.50 ±0.03	0.29 ±0.03	0.12 ±0.02		0.31 ±0.06	0.09 ±0.02	0.03 ±0.00	0.01 ±0.00	0.00 ±0.00
200	0.80 ±0.09	0.77 ±0.04	0.59 ±0.02	0.35 ±0.04	0.18 ±0.03		0.49 ±0.08	0.17 ±0.03	0.06 ±0.01	0.02 ±0.00	0.01 ±0.00
300	0.87 ±0.05	0.74 ±0.07	0.60 ±0.06	0.38 ±0.03	0.20 ±0.02		0.60 ±0.06	0.21 ±0.03	0.08 ±0.01	0.03 ±0.00	0.01 ±0.00
400	0.85 ±0.08	0.83 ±0.03	0.65 ±0.03	0.43 ±0.03	0.22 ±0.02		0.60 ±0.08	0.26 ±0.04	0.10 ±0.02	0.03 ±0.01	0.01 ±0.00
500	0.86 ±0.06	0.81 ±0.07	0.66 ±0.04	0.42 ±0.04	0.22 ±0.03		0.64 ±0.08	0.33 ±0.06	0.12 ±0.02	0.04 ±0.01	0.01 ±0.00
1000	0.90 ±0.06	0.84 ±0.06	0.71 ±0.05	0.54 ±0.02	0.25 ±0.03		0.65 ±0.09	0.48 ±0.06	0.20 ±0.04	0.07 ±0.02	0.02 ±0.00

GENETICS: Conclusions

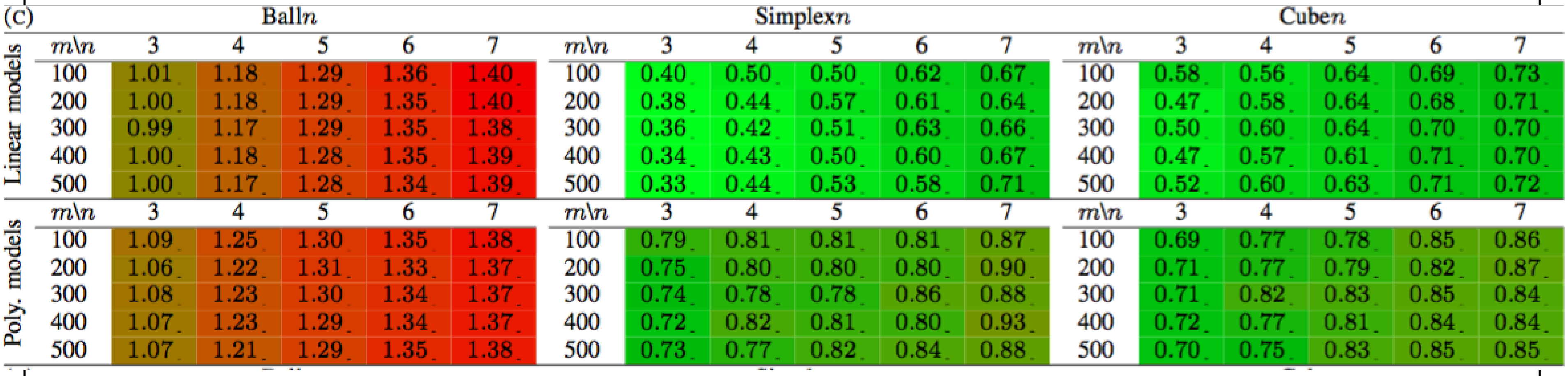
- The test-set results are better than for MILP-based method
- GENETICS still overestimates the feasible region
- Synthesized constraints have different syntax than the actual ones
  - GENETICS produces one out of many alternative models that fit training set
- Curse of dimensionality
  - When the number of variables becomes large GENETICS achieves worse results
  - Larger training set improves performance

Genetic One-Class Constraint Synthesis (GOCCS)

- An equivalent of GENETICS for 1-CSP
- Representation and operators are mostly the same
- The main differences consist of:
  - Validation set of unlabeled examples sampled prior to evolutionary run
    - The unlabeled examples are located from the known feasible examples
  - Two optimization criteria, both maximized:
    - The number of true positives
    - The number of true negatives
  - NSGA-II post-selection instead of Lexicase selection

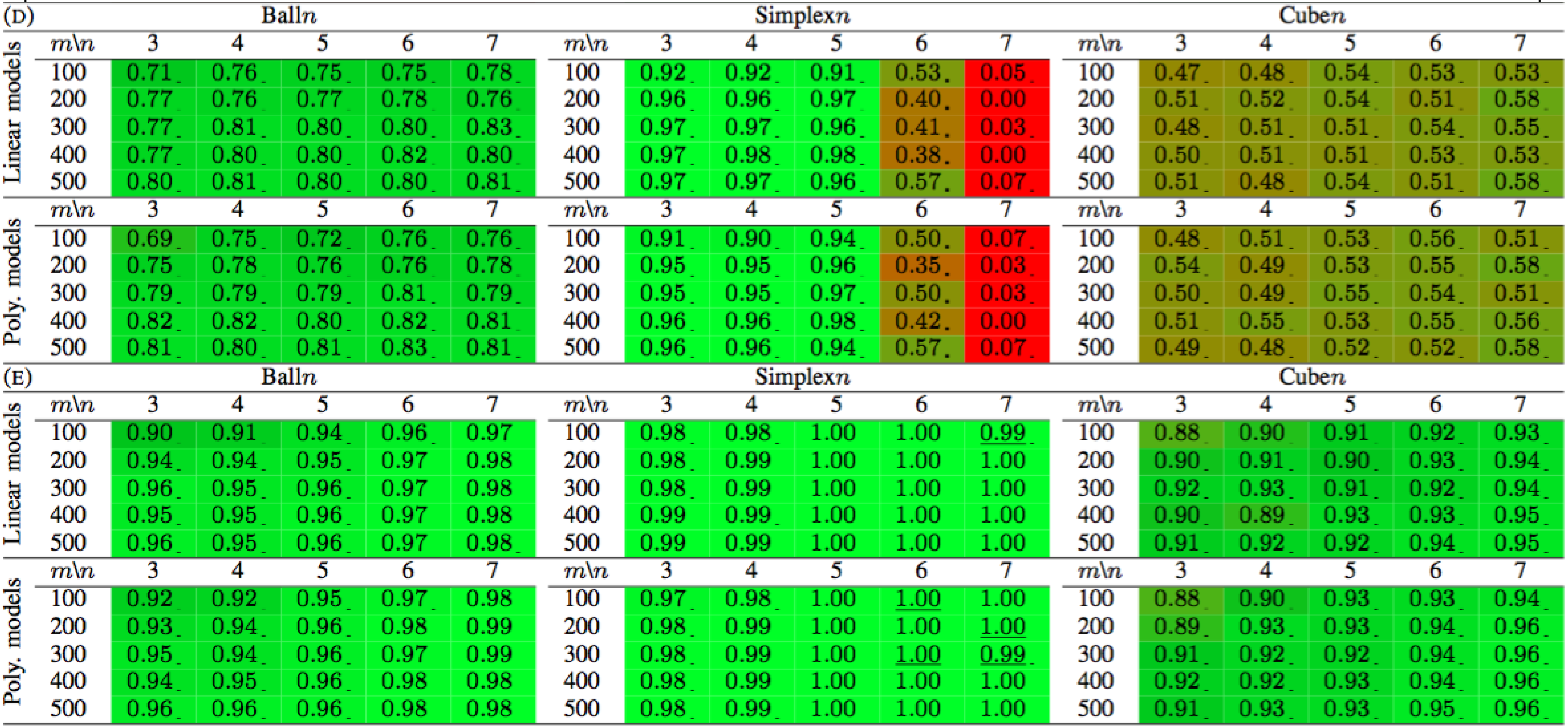
Tomasz P. Pawlak, Krzysztof Krawiec, Synthesis of Constraints for Mathematical Programming with One-Class Genetic Programming, IEEE Transactions on Evolutionary Computation, IEEE Press, 2018. IF=10.629, 50p MNiSW

Mean angle between the corresponding constraints in the synthesized and the actual MP models



(c)		Ball$n$						Simplex$n$						Cube$n$				
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Linear models	100	1.01	1.18	1.29	1.36	1.40	100	0.40	0.50	0.50	0.62	0.67	100	0.58	0.56	0.64	0.69	0.73
	200	1.00	1.18	1.29	1.35	1.40	200	0.38	0.44	0.57	0.61	0.64	200	0.47	0.58	0.64	0.68	0.71
	300	0.99	1.17	1.29	1.35	1.38	300	0.36	0.42	0.51	0.63	0.66	300	0.50	0.60	0.64	0.70	0.70
	400	1.00	1.18	1.28	1.35	1.39	400	0.34	0.43	0.50	0.60	0.67	400	0.47	0.57	0.61	0.71	0.70
	500	1.00	1.17	1.28	1.34	1.39	500	0.33	0.44	0.53	0.58	0.71	500	0.52	0.60	0.63	0.71	0.72
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Poly. models	100	1.09	1.25	1.30	1.35	1.38	100	0.79	0.81	0.81	0.81	0.87	100	0.69	0.77	0.78	0.85	0.86
	200	1.06	1.22	1.31	1.33	1.37	200	0.75	0.80	0.80	0.80	0.90	200	0.71	0.77	0.79	0.82	0.87
	300	1.08	1.23	1.30	1.34	1.37	300	0.74	0.78	0.78	0.86	0.88	300	0.71	0.82	0.83	0.85	0.84
	400	1.07	1.23	1.29	1.34	1.37	400	0.72	0.82	0.81	0.80	0.93	400	0.72	0.77	0.81	0.84	0.84
	500	1.07	1.21	1.29	1.35	1.38	500	0.73	0.77	0.82	0.84	0.88	500	0.70	0.75	0.83	0.85	0.85

Mean sensitivity (D) and specificity (E) on test-set

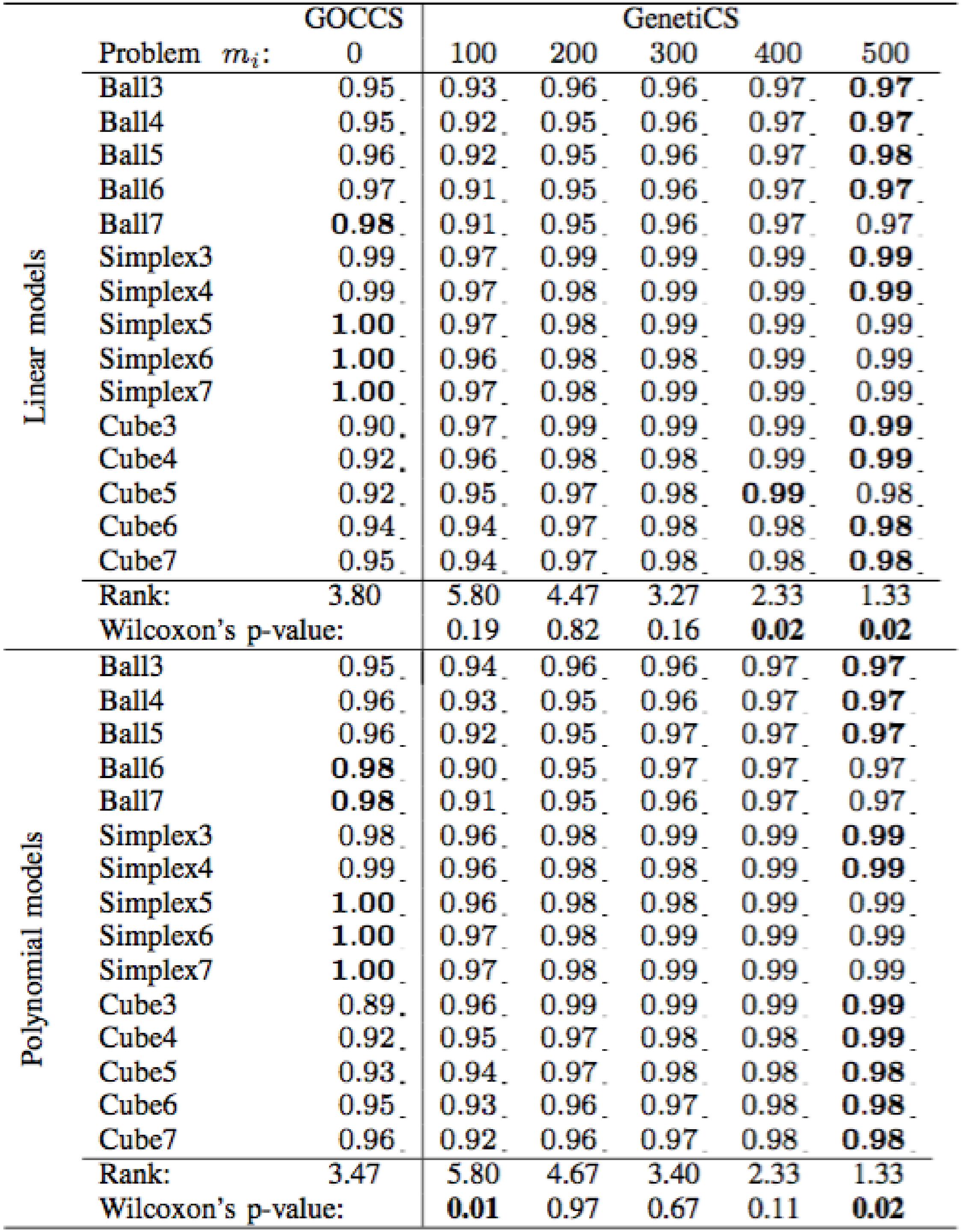


(D)		Ball$n$						Simplex$n$						Cube$n$				
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Linear models	100	0.71	0.76	0.75	0.75	0.78	100	0.92	0.92	0.91	0.53	0.05	100	0.47	0.48	0.54	0.53	0.53
	200	0.77	0.76	0.77	0.78	0.76	200	0.96	0.96	0.97	0.40	0.00	200	0.51	0.52	0.54	0.51	0.58
	300	0.77	0.81	0.80	0.80	0.83	300	0.97	0.97	0.96	0.41	0.03	300	0.48	0.51	0.51	0.54	0.55
	400	0.77	0.80	0.80	0.82	0.80	400	0.97	0.98	0.98	0.38	0.00	400	0.50	0.51	0.51	0.53	0.53
	500	0.80	0.81	0.80	0.80	0.81	500	0.97	0.97	0.96	0.57	0.07	500	0.51	0.48	0.54	0.51	0.58
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Poly. models	100	0.69	0.75	0.72	0.76	0.76	100	0.91	0.90	0.94	0.50	0.07	100	0.48	0.51	0.53	0.56	0.51
	200	0.75	0.78	0.76	0.76	0.78	200	0.95	0.95	0.96	0.35	0.03	200	0.54	0.49	0.53	0.55	0.58
	300	0.79	0.79	0.79	0.81	0.79	300	0.95	0.95	0.97	0.50	0.03	300	0.50	0.49	0.55	0.54	0.51
	400	0.82	0.82	0.80	0.82	0.81	400	0.96	0.96	0.98	0.42	0.00	400	0.51	0.55	0.53	0.55	0.56
	500	0.81	0.80	0.81	0.83	0.81	500	0.96	0.96	0.94	0.57	0.07	500	0.49	0.48	0.52	0.52	0.58

(E)		Ball$n$						Simplex$n$						Cube$n$				
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Linear models	100	0.90	0.91	0.94	0.96	0.97	100	0.98	0.98	1.00	1.00	0.99	100	0.88	0.90	0.91	0.92	0.93
	200	0.94	0.94	0.95	0.97	0.98	200	0.98	0.99	1.00	1.00	1.00	200	0.90	0.91	0.90	0.93	0.94
	300	0.96	0.95	0.96	0.97	0.98	300	0.98	0.99	1.00	1.00	1.00	300	0.92	0.93	0.91	0.92	0.94
	400	0.95	0.95	0.96	0.97	0.98	400	0.99	0.99	1.00	1.00	1.00	400	0.90	0.89	0.93	0.93	0.95
	500	0.96	0.95	0.96	0.97	0.98	500	0.99	0.99	1.00	1.00	1.00	500	0.91	0.92	0.92	0.94	0.95
	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7	$m\backslash n$	3	4	5	6	7
Poly. models	100	0.92	0.92	0.95	0.97	0.98	100	0.97	0.98	1.00	1.00	1.00	100	0.88	0.90	0.93	0.93	0.94
	200	0.93	0.94	0.96	0.98	0.99	200	0.98	0.99	1.00	1.00	1.00	200	0.89	0.93	0.93	0.94	0.96
	300	0.95	0.94	0.96	0.97	0.99	300	0.98	0.99	1.00	1.00	0.99	300	0.91	0.92	0.92	0.94	0.96
	400	0.94	0.95	0.96	0.98	0.98	400	0.98	0.99	1.00	1.00	1.00	400	0.92	0.92	0.93	0.94	0.96
	500	0.96	0.96	0.96	0.98	0.98	500	0.98	0.99	1.00	1.00	1.00	500	0.91	0.93	0.93	0.95	0.96

GOCCS vs GenetiCS: mean accuracy on test-set

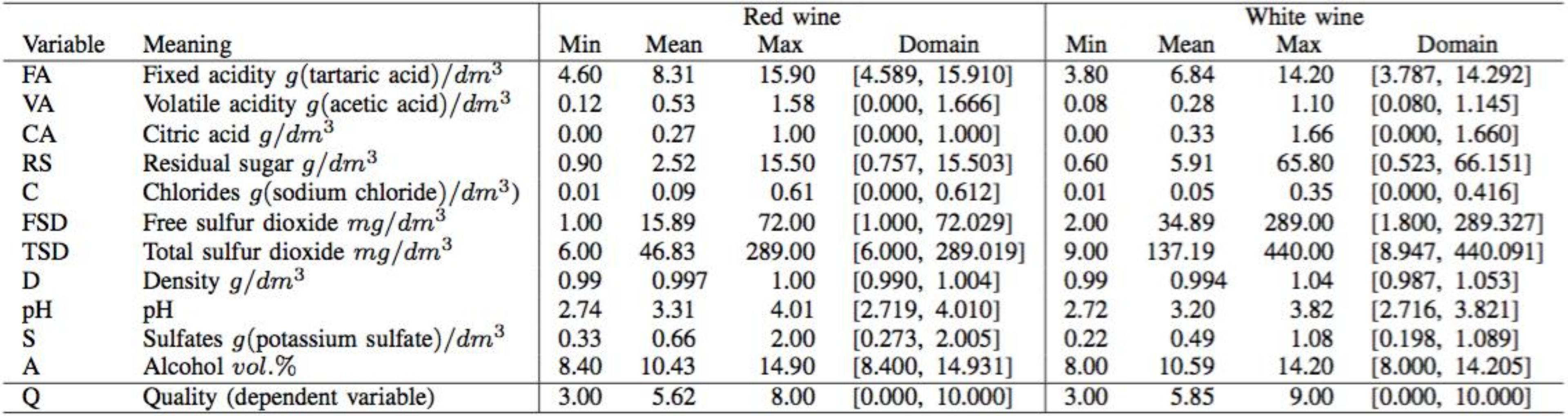
- Training set of m_f = 500 feasible examples
- m_i is the number of extra infeasible examples supplied to the training set for GenetiCS



		GOCCS	GenetiCS				
	Problem m_i:	0	100	200	300	400	500
Linear models	Ball3	0.95	0.93	0.96	0.96	0.97	**0.97**
	Ball4	0.95	0.92	0.95	0.96	0.97	**0.97**
	Ball5	0.96	0.92	0.95	0.96	0.97	**0.98**
	Ball6	0.97	0.91	0.95	0.96	0.97	**0.97**
	Ball7	**0.98**	0.91	0.95	0.96	0.97	0.97
	Simplex3	0.99	0.97	0.99	0.99	0.99	**0.99**
	Simplex4	0.99	0.97	0.98	0.99	0.99	**0.99**
	Simplex5	**1.00**	0.97	0.98	0.99	0.99	0.99
	Simplex6	**1.00**	0.96	0.98	0.98	0.99	0.99
	Simplex7	**1.00**	0.97	0.98	0.99	0.99	0.99
	Cube3	0.90	0.97	0.99	0.99	0.99	**0.99**
	Cube4	0.92	0.96	0.98	0.98	0.99	**0.99**
	Cube5	0.92	0.95	0.97	0.98	**0.99**	0.98
	Cube6	0.94	0.94	0.97	0.98	0.98	**0.98**
	Cube7	0.95	0.94	0.97	0.98	0.98	**0.98**
	Rank:	3.80	5.80	4.47	3.27	2.33	1.33
	Wilcoxon's p-value:		0.19	0.82	0.16	**0.02**	**0.02**
Polynomial models	Ball3	0.95	0.94	0.96	0.96	0.97	**0.97**
	Ball4	0.96	0.93	0.95	0.96	0.97	**0.97**
	Ball5	0.96	0.92	0.95	0.97	0.97	**0.97**
	Ball6	**0.98**	0.90	0.95	0.97	0.97	0.97
	Ball7	**0.98**	0.91	0.95	0.96	0.97	0.97
	Simplex3	0.98	0.96	0.98	0.99	0.99	**0.99**
	Simplex4	0.99	0.96	0.98	0.98	0.99	**0.99**
	Simplex5	**1.00**	0.96	0.98	0.98	0.99	0.99
	Simplex6	**1.00**	0.97	0.98	0.99	0.99	0.99
	Simplex7	**1.00**	0.97	0.98	0.99	0.99	0.99
	Cube3	0.89	0.96	0.99	0.99	0.99	**0.99**
	Cube4	0.92	0.95	0.97	0.98	0.98	**0.99**
	Cube5	0.93	0.94	0.97	0.98	0.98	**0.98**
	Cube6	0.95	0.93	0.96	0.97	0.98	**0.98**
	Cube7	0.96	0.92	0.96	0.97	0.98	**0.98**
	Rank:	3.47	5.80	4.67	3.40	2.33	1.33
	Wilcoxon's p-value:		**0.01**	0.97	0.67	0.11	**0.02**

Modeling of Wine Quality

- Wine Quality data set [1]
  - 11 physiochemical attributes of wine
  - A quality assessment [0-10] calculated as the median of the assessments made by at least three sensory assessors
  - 1599 red wine examples, 4898 white wine examples
  - 1-CSP



		Red wine				White wine			
Variable	Meaning	Min	Mean	Max	Domain	Min	Mean	Max	Domain
FA	Fixed acidity $g(\text{tartaric acid})/dm^3$	4.60	8.31	15.90	[4.589, 15.910]	3.80	6.84	14.20	[3.787, 14.292]
VA	Volatile acidity $g(\text{acetic acid})/dm^3$	0.12	0.53	1.58	[0.000, 1.666]	0.08	0.28	1.10	[0.080, 1.145]
CA	Citric acid g/dm^3	0.00	0.27	1.00	[0.000, 1.000]	0.00	0.33	1.66	[0.000, 1.660]
RS	Residual sugar g/dm^3	0.90	2.52	15.50	[0.757, 15.503]	0.60	5.91	65.80	[0.523, 66.151]
C	Chlorides $g(\text{sodium chloride})/dm^3)$	0.01	0.09	0.61	[0.000, 0.612]	0.01	0.05	0.35	[0.000, 0.416]
FSD	Free sulfur dioxide mg/dm^3	1.00	15.89	72.00	[1.000, 72.029]	2.00	34.89	289.00	[1.800, 289.327]
TSD	Total sulfur dioxide mg/dm^3	6.00	46.83	289.00	[6.000, 289.019]	9.00	137.19	440.00	[8.947, 440.091]
D	Density g/dm^3	0.99	0.997	1.00	[0.990, 1.004]	0.99	0.994	1.04	[0.987, 1.053]
pH	pH	2.74	3.31	4.01	[2.719, 4.010]	2.72	3.20	3.82	[2.716, 3.821]
S	Sulfates $g(\text{potassium sulfate})/dm^3$	0.33	0.66	2.00	[0.273, 2.005]	0.22	0.49	1.08	[0.198, 1.089]
A	Alcohol $vol.\%$	8.40	10.43	14.90	[8.400, 14.931]	8.00	10.59	14.20	[8.000, 14.205]
Q	Quality (dependent variable)	3.00	5.62	8.00	[0.000, 10.000]	3.00	5.85	9.00	[0.000, 10.000]

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, “Modeling wine preferences by data mining from physicochemical properties” *Decision Support Systems* 47(4):547-553, 2009.

Wine QP models

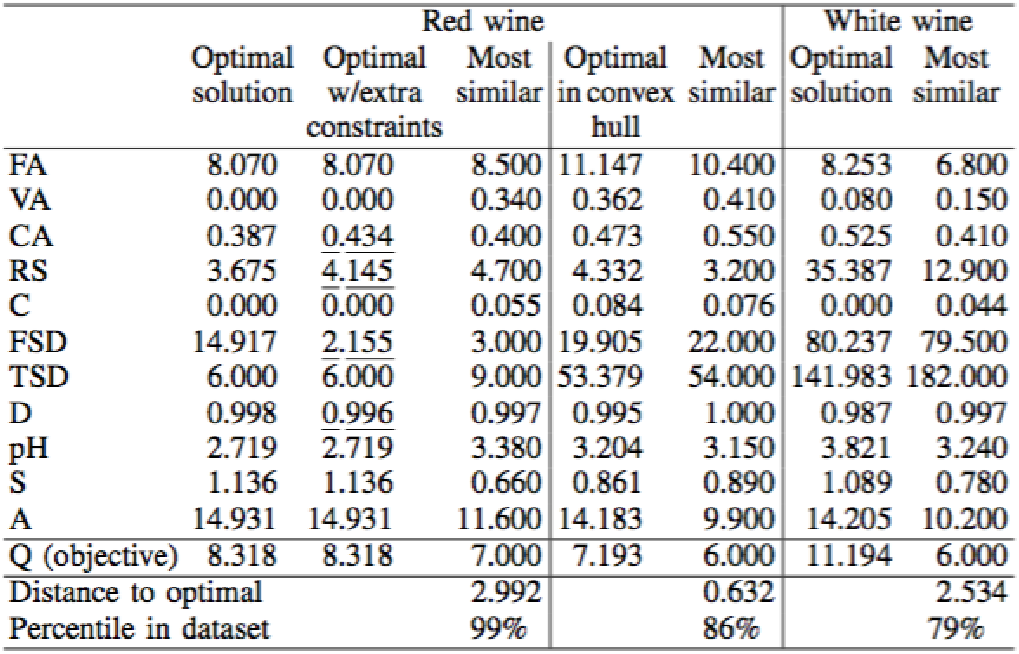
Red wine

$$
\begin{aligned}
\max \quad & -0.01015\mathrm{FA}^2-0.7481\mathrm{VA}^2-1.681\mathrm{S}^2+0.1639\mathrm{FA}+\\
& -1.836\mathrm{C}-0.00197\mathrm{TSD}-0.6832\mathrm{pH}+3.819\mathrm{S}+0.3003\mathrm{A}+2.873\\
\text{subject to} \quad & \\
& -\mathrm{VA}+\mathrm{C}\leq 0\\
& 1.669\mathrm{C}-\mathrm{S}\leq 0\\
& \mathrm{VA}+\mathrm{pH}+\mathrm{S}\leq 5.449\\
& \mathrm{RS}-3.455\mathrm{D}-\mathrm{pH}\leq 2.712\\
& 3.192\mathrm{VA}-\mathrm{RS}-\mathrm{FSD}\leq 0\\
& 2.384\mathrm{CA}-\mathrm{RS}-\mathrm{C}\leq 0\\
& \mathrm{VA}+2.384\mathrm{RS}+2.544\mathrm{C}+\mathrm{TSD}-5.717\mathrm{D}-16.3\mathrm{A}\leq 11.95\\
& 0.2764\mathrm{FA}+1.492\mathrm{VA}+0.3844\mathrm{RS}+\mathrm{C}+\mathrm{D}-\mathrm{pH}+4.712\mathrm{S}\leq 9.153\\
& 2\mathrm{FA}+\mathrm{CA}+1.686\mathrm{RS}+0.5031\mathrm{C}+1.384\mathrm{FSD}-\mathrm{TSD}-4.384\mathrm{pH}+\\
& \quad 2.384\mathrm{S}-2\mathrm{A}\leq 3.415\\
& \mathrm{FA}-10.17\mathrm{VA}+4.384\mathrm{CA}-2.384\mathrm{RS}-\mathrm{C}+2.384\mathrm{FSD}+\mathrm{TSD}+\\
& \quad 2.384\mathrm{pH}-2.384\mathrm{S}\geq 2.384\\
& 2.384\mathrm{FA}+0.4379\mathrm{VA}+\mathrm{CA}-0.3844\mathrm{RS}+6.123\mathrm{C}+4.136\mathrm{FSD}+\\
& \quad -0.3844\mathrm{TSD}-\mathrm{D}-2\mathrm{S}-0.6764\mathrm{A}\geq 0.1357
\end{aligned}
$$

White wine

$$
\begin{aligned}
\max \quad & -0.02784\mathrm{FA}^2-1.234\mathrm{CA}^2-0.00007981\mathrm{FSD}^2-0.00002581\mathrm{TSD}^2+\\
& 0.4588\mathrm{FA}-1.332\mathrm{VA}+1.259\mathrm{CA}+0.06963\mathrm{RS}+0.01358\mathrm{FSD}+\\
& 0.006554\mathrm{TSD}-144.9\mathrm{D}+0.8708\mathrm{pH}+0.6428\mathrm{S}+0.1996\mathrm{A}+141.8\\
\text{subject to} \quad & \\
& \mathrm{D}\leq 1.0052\\
& \mathrm{VA}-\mathrm{C}\geq 0\\
& \mathrm{CA}+\mathrm{C}-\mathrm{D}\leq 0\\
& -\mathrm{VA}+0.5283\mathrm{CA}-\mathrm{RS}-0.2167\mathrm{D}+2\mathrm{pH}+2\mathrm{A}\geq 0.6499\\
& -4.786\mathrm{FA}+4.893\mathrm{C}+0.107\mathrm{TSD}-2.893\mathrm{pH}+\mathrm{S}+\mathrm{A}\leq 2\\
& -\mathrm{FA}-\mathrm{VA}-0.9435\mathrm{CA}+2\mathrm{RS}+2\mathrm{C}+\mathrm{FSD}-\mathrm{TSD}+0.5283\mathrm{D}\leq 0.7209\\
& -2.8\mathrm{VA}-1.951\mathrm{CA}+0.051\mathrm{RS}+\mathrm{C}-0.0565\mathrm{TSD}+1.528\mathrm{D}+\\
& \quad 2.264\mathrm{pH}+0.04894\mathrm{S}+\mathrm{A}\geq 1.127\\
& 0.2791\mathrm{FA}-0.1927\mathrm{VA}-3\mathrm{CA}+0.2791\mathrm{RS}+\mathrm{C}+1.528\mathrm{FSD}-\mathrm{TSD}+\\
& \quad -1.528\mathrm{D}+0.5283\mathrm{S}-\mathrm{A}\leq 0.1316
\end{aligned}
$$

The optimal solutions to the wine models



	Red wine					White wine	
	Optimal solution	Optimal w/extra constraints	Most similar	Optimal in convex hull	Most similar	Optimal solution	Most similar
FA	8.070	8.070	8.500	11.147	10.400	8.253	6.800
VA	0.000	0.000	0.340	0.362	0.410	0.080	0.150
CA	0.387	0.434	0.400	0.473	0.550	0.525	0.410
RS	3.675	4.145	4.700	4.332	3.200	35.387	12.900
C	0.000	0.000	0.055	0.084	0.076	0.000	0.044
FSD	14.917	2.155	3.000	19.905	22.000	80.237	79.500
TSD	6.000	6.000	9.000	53.379	54.000	141.983	182.000
D	0.998	0.996	0.997	0.995	1.000	0.987	0.997
pH	2.719	2.719	3.380	3.204	3.150	3.821	3.240
S	1.136	1.136	0.660	0.861	0.890	1.089	0.780
A	14.931	14.931	11.600	14.183	9.900	14.205	10.200
Q (objective)	8.318	8.318	7.000	7.193	6.000	11.194	6.000
Distance to optimal			2.992		0.632		2.534
Percentile in dataset			99%		86%		79%

- Due to FSD > TSD in the optimal solution, we added extra constraints for the red wine model:

$$-\text{FSD} + \text{TSD} \geq 3$$

$$\text{FSD} - 0.8571\text{TSD} \leq 0$$

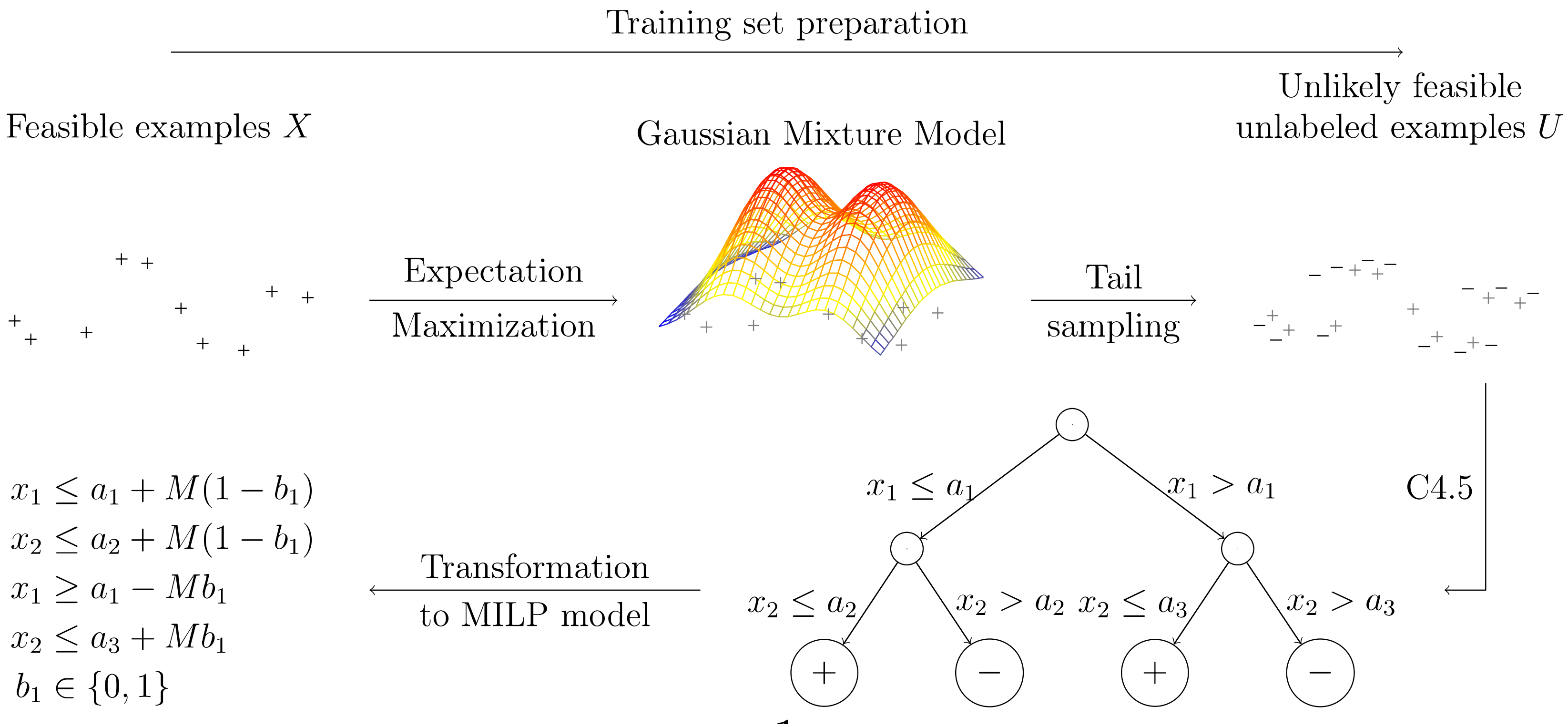
- They are based on the evidence in the data set that min TSD – FSD = 3 and max FSD/TSD = 0.8571

GOCCS: Conclusions

- GOCCS has good performance on low-dimensional 1-CSP
  - E.g., up to 6 – 7 variables
- GOCCS underestimates the feasible region
- GOCCS is susceptible to the curse of dimensionality
- The synthesized NLP models are oversize, however LP models have correct size

Constraint Synthesis with C4.5 (CSC4.5) for 1-CSP

- General idea: Build a decision tree and transform it to a MILP model



Training set preparation

Feasible examples X

Expectation Maximization

Gaussian Mixture Model

Tail sampling

Unlikely feasible unlabeled examples U

C4.5

$x_1 \leq a_1$

$x_1 > a_1$

$x_2 \leq a_2$

$x_2 > a_2$

$x_2 \leq a_3$

$x_2 > a_3$

$+$

$-$

$+$

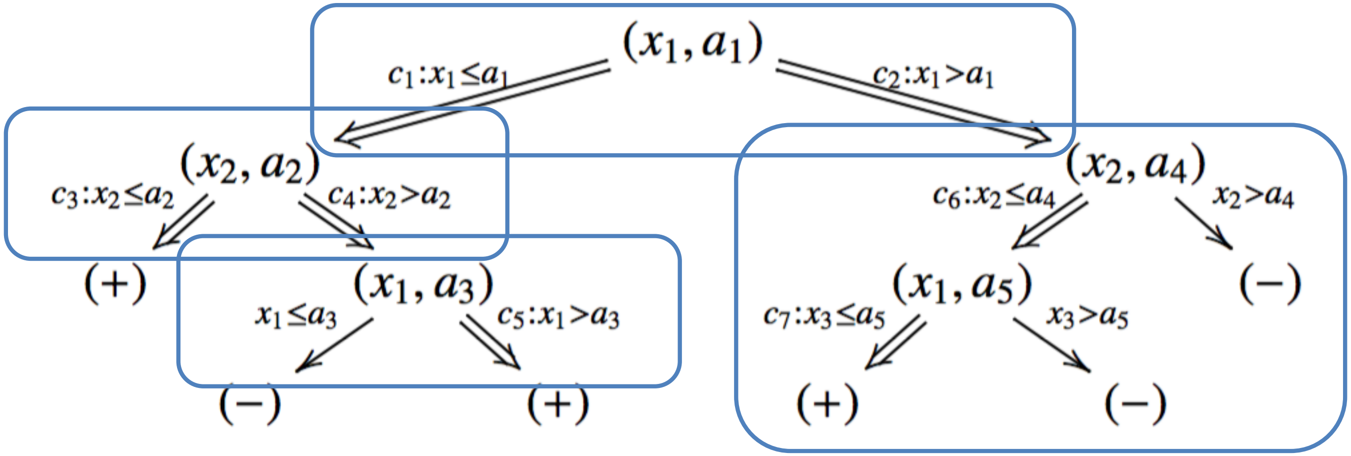
$-$

Transformation to MILP model

$$x_1 \leq a_1 + M(1 - b_1)$$
$$x_2 \leq a_2 + M(1 - b_1)$$
$$x_1 \geq a_1 - Mb_1$$
$$x_2 \leq a_3 + Mb_1$$
$$b_1 \in \{0, 1\}$$

- Patryk Kudła, Tomasz P. Pawlak, One-class synthesis of constraints for Mixed-Integer Linear Programming with C4.5 decision trees, Applied Soft Computing 68 (2018) 1-12. IF=3.541, 40p MNiSW

A decision tree is a MILP model



$$c_1 : x_1 \le a_1 \longrightarrow x_1 \le a_1 \underbrace{+M(1-b_1)}_{c_1 \text{ is active if } b_1=1}$$

$$c_2 : x_1 > a_1 \longrightarrow x_1 \ge a_1 \quad \underbrace{-Mb_1}_{c_2 \text{ is active if } b_1=0}$$

$$c_3 : x_2 \le a_2 \longrightarrow x_2 \le a_2 \underbrace{+M(1-b_1)}_{c_1 \text{ is active}} \underbrace{+M(1-b_2)}_{c_3 \text{ is active if } b_2=1}$$

$$c_4 : x_2 > a_2 \longrightarrow x_2 \ge a_2 \underbrace{-M(1-b_1)}_{c_1 \text{ is active}} \quad \underbrace{-Mb_2}_{c_4 \text{ is active if } b_2=0}$$

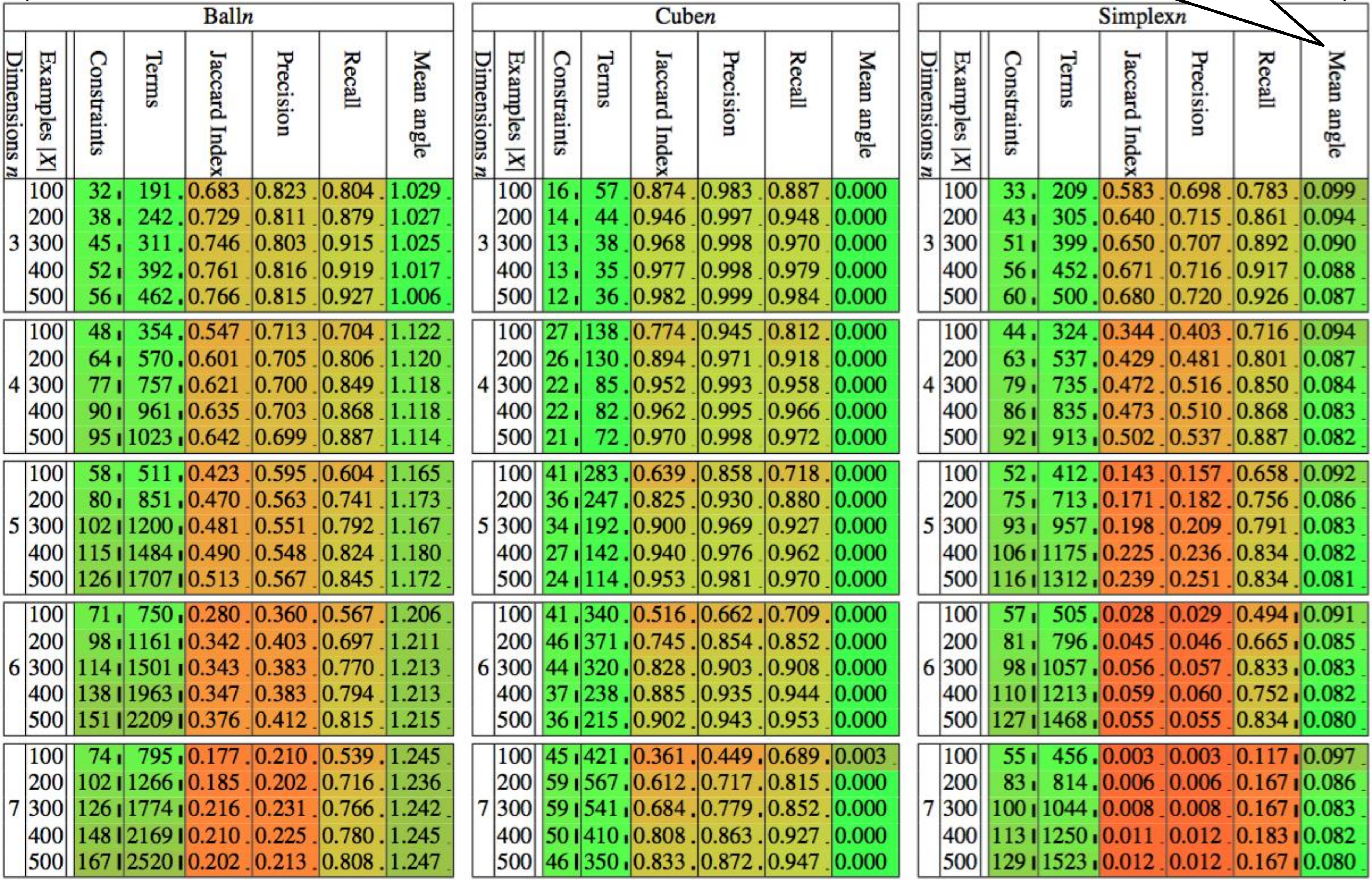
$$c_5 : x_1 > a_3 \longrightarrow x_1 \ge a_3 \underbrace{-M(1-b_1)}_{c_1 \text{ is active}} \underbrace{-Mb_2}_{c_4 \text{ is active}}$$

$$c_6 : x_2 \le a_4 \longrightarrow x_2 \le a_4 \underbrace{+Mb_1}_{c_2 \text{ is active}}$$

$$c_7 : x_3 \le a_5 \longrightarrow x_3 \le a_5 \underbrace{+Mb_1}_{c_2 \text{ is active}}$$

CSC4.5 Results

Theoretical minimum 0.072rad



Balln							
Dimensions n	Examples $\|X\|$	Constraints	Terms	Jaccard Index	Precision	Recall	Mean angle
3	100	32	191	0.683	0.823	0.804	1.029
	200	38	242	0.729	0.811	0.879	1.027
	300	45	311	0.746	0.803	0.915	1.025
	400	52	392	0.761	0.816	0.919	1.017
	500	56	462	0.766	0.815	0.927	1.006
4	100	48	354	0.547	0.713	0.704	1.122
	200	64	570	0.601	0.705	0.806	1.120
	300	77	757	0.621	0.700	0.849	1.118
	400	90	961	0.635	0.703	0.868	1.118
	500	95	1023	0.642	0.699	0.887	1.114
5	100	58	511	0.423	0.595	0.604	1.165
	200	80	851	0.470	0.563	0.741	1.173
	300	102	1200	0.481	0.551	0.792	1.167
	400	115	1484	0.490	0.548	0.824	1.180
	500	126	1707	0.513	0.567	0.845	1.172
6	100	71	750	0.280	0.360	0.567	1.206
	200	98	1161	0.342	0.403	0.697	1.211
	300	114	1501	0.343	0.383	0.770	1.213
	400	138	1963	0.347	0.383	0.794	1.213
	500	151	2209	0.376	0.412	0.815	1.215
7	100	74	795	0.177	0.210	0.539	1.245
	200	102	1266	0.185	0.202	0.716	1.236
	300	126	1774	0.216	0.231	0.766	1.242
	400	148	2169	0.210	0.225	0.780	1.245
	500	167	2520	0.202	0.213	0.808	1.247

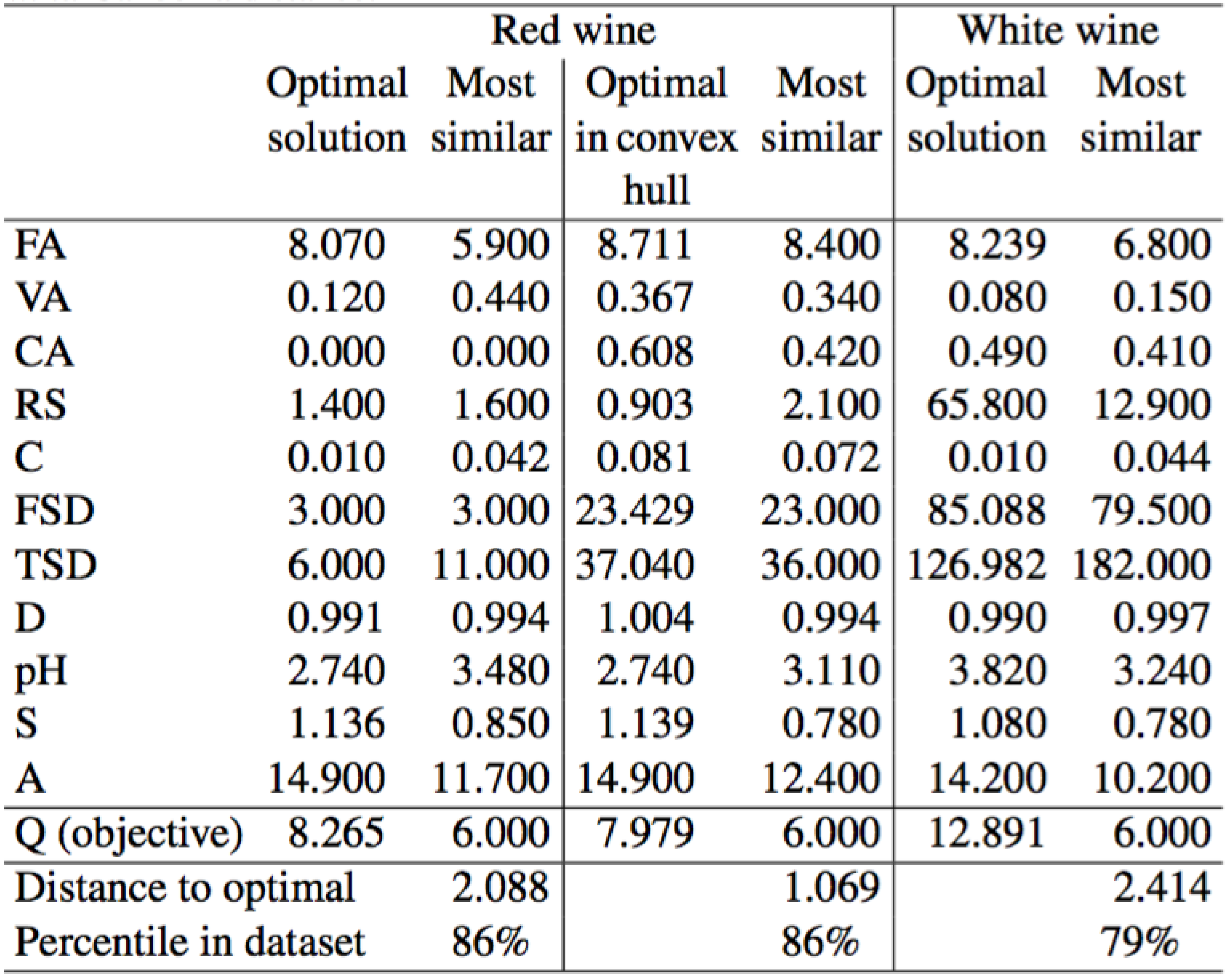
Cuben							
Dimensions n	Examples $\|X\|$	Constraints	Terms	Jaccard Index	Precision	Recall	Mean angle
3	100	16	57	0.874	0.983	0.887	0.000
	200	14	44	0.946	0.997	0.948	0.000
	300	13	38	0.968	0.998	0.970	0.000
	400	13	35	0.977	0.998	0.979	0.000
	500	12	36	0.982	0.999	0.984	0.000
4	100	27	138	0.774	0.945	0.812	0.000
	200	26	130	0.894	0.971	0.918	0.000
	300	22	85	0.952	0.993	0.958	0.000
	400	22	82	0.962	0.995	0.966	0.000
	500	21	72	0.970	0.998	0.972	0.000
5	100	41	283	0.639	0.858	0.718	0.000
	200	36	247	0.825	0.930	0.880	0.000
	300	34	192	0.900	0.969	0.927	0.000
	400	27	142	0.940	0.976	0.962	0.000
	500	24	114	0.953	0.981	0.970	0.000
6	100	41	340	0.516	0.662	0.709	0.000
	200	46	371	0.745	0.854	0.852	0.000
	300	44	320	0.828	0.903	0.908	0.000
	400	37	238	0.885	0.935	0.944	0.000
	500	36	215	0.902	0.943	0.953	0.000
7	100	45	421	0.361	0.449	0.689	0.003
	200	59	567	0.612	0.717	0.815	0.000
	300	59	541	0.684	0.779	0.852	0.000
	400	50	410	0.808	0.863	0.927	0.000
	500	46	350	0.833	0.872	0.947	0.000

Simplexn							
Dimensions n	Examples $\|X\|$	Constraints	Terms	Jaccard Index	Precision	Recall	Mean angle
3	100	33	209	0.583	0.698	0.783	0.099
	200	43	305	0.640	0.715	0.861	0.094
	300	51	399	0.650	0.707	0.892	0.090
	400	56	452	0.671	0.716	0.917	0.088
	500	60	500	0.680	0.720	0.926	0.087
4	100	44	324	0.344	0.403	0.716	0.094
	200	63	537	0.429	0.481	0.801	0.087
	300	79	735	0.472	0.516	0.850	0.084
	400	86	835	0.473	0.510	0.868	0.083
	500	92	913	0.502	0.537	0.887	0.082
5	100	52	412	0.143	0.157	0.658	0.092
	200	75	713	0.171	0.182	0.756	0.086
	300	93	957	0.198	0.209	0.791	0.083
	400	106	1175	0.225	0.236	0.834	0.082
	500	116	1312	0.239	0.251	0.834	0.081
6	100	57	505	0.028	0.029	0.494	0.091
	200	81	796	0.045	0.046	0.665	0.085
	300	98	1057	0.056	0.057	0.833	0.083
	400	110	1213	0.059	0.060	0.752	0.082
	500	127	1468	0.055	0.055	0.834	0.080
7	100	55	456	0.003	0.003	0.117	0.097
	200	83	814	0.006	0.006	0.167	0.086
	300	100	1044	0.008	0.008	0.167	0.083
	400	113	1250	0.011	0.012	0.183	0.082
	500	129	1523	0.012	0.012	0.167	0.080

Modeling of Wine Quality

- The same Wine Quality data set
  - Divided into training set and test set in roughly 50%:50%
  - Test set supplemented with unlabeled likely infeasible examples
- Red wine MIQP model
  - 517 constraints
  - 171 auxiliary binary variables
  - Jaccard index of feasible region calculated on test set: 0.90
- White wine MIQP model
  - 1373 constraints
  - 484 auxiliary binary variables
  - Jaccard index of feasible region calculated on test set: 0.91

The optimal solutions to the wine models



	Red wine				White wine	
	Optimal solution	Most similar	Optimal in convex hull	Most similar	Optimal solution	Most similar
FA	8.070	5.900	8.711	8.400	8.239	6.800
VA	0.120	0.440	0.367	0.340	0.080	0.150
CA	0.000	0.000	0.608	0.420	0.490	0.410
RS	1.400	1.600	0.903	2.100	65.800	12.900
C	0.010	0.042	0.081	0.072	0.010	0.044
FSD	3.000	3.000	23.429	23.000	85.088	79.500
TSD	6.000	11.000	37.040	36.000	126.982	182.000
D	0.991	0.994	1.004	0.994	0.990	0.997
pH	2.740	3.480	2.740	3.110	3.820	3.240
S	1.136	0.850	1.139	0.780	1.080	0.780
A	14.900	11.700	14.900	12.400	14.200	10.200
Q (objective)	8.265	6.000	7.979	6.000	12.891	6.000
Distance to optimal		2.088		1.069		2.414
Percentile in dataset		86%		86%		79%

CSC4.5: Conclusions

- The synthesized MILP models may be non-convex w.r.t. the input variables
- The synthesized MILP models are oversize
- CSC4.5 is unable to model interactions between variables
- CSC4.5 is susceptible to the curse of dimensionality

Evolutionary Strategy-based One-Class Constraint Synthesis (ESOCCS) for 1-CSP

- General idea:
  - Model the distribution of the feasible states using Gaussian Mixture Model and Expectation Maximization
  - Sample the tail of that distribution for unlabeled and likely infeasible examples
  - Evolve a population of constraints using (μ+λ)-Evolutionary Strategy
    - Single-population cooperative co-evolution
  - Select a minimal subset of constraints from the population to produce an MP model

Tomasz P. Pawlak, Synthesis of Mathematical Programming models with one-class evolutionary strategies, Swarm and Evolutionary Computation, Elsevier, 2018 (in press). IF=3.893, 50p MNiSW

(μ+λ)-Evolutionary Strategy in a single-population cooperative co-evolution mode

- Representation of a constraint:
  - Vector of weights w_i of predefined terms (e.g., variables)
  - A constant w_0
  - Vector of standard deviations σ_i corresponding to w_i
  - Vector of rotation angles α_{ij} corresponding to the pairs of w_i and w_j
- Population: a set of μ constraints
- Gaussian initialization $w_i \sim \mathrm{N}(0,1)$
- Correlated mutation

$$\forall_i : \sigma'_i \sim \sigma_i e^{\tau' N} e^{\tau N(0,1)} \qquad \text{Std. dev. mutation}$$

$$\forall_{ij} : \alpha'_{ij} \sim \beta \mathcal{N}(\alpha_{ij}, 1) \qquad \text{Angles mutation}$$

$$\Sigma_{ij} = \begin{cases} \sigma'^2_i & i = j \\ \frac{1}{2}\left(\sigma'^2_i - \sigma'^2_j\right)\tan(2\alpha'_{ij}) & i > j \\ \frac{1}{2}\left(\sigma'^2_j - \sigma'^2_i\right)\tan(2\alpha'_{ji}) & i < j \end{cases} \qquad \text{Element of covariance matrix}$$

$$\boldsymbol{w}' \sim \boldsymbol{\mathcal{N}}(\boldsymbol{w}, \boldsymbol{\Sigma}) \qquad \text{Weights mutation,}$$

- Hybrid recombination

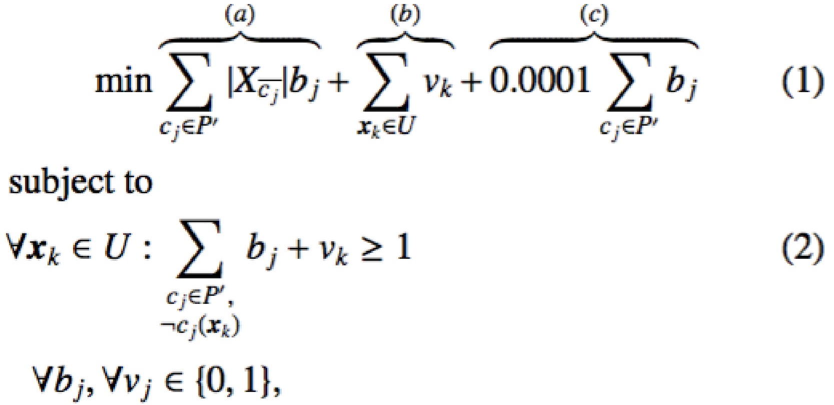
$$\forall_i : w'_i \sim \mathcal{U}\left(\{w_i^{(1)}, w_i^{(2)}\}\right) \qquad \text{Discrete recombination of } w_i$$

$$\forall_i : \sigma'_i = \frac{1}{2}\left(\sigma_i^{(1)} + \sigma_i^{(2)}\right) \qquad \text{Intermediate recombination of } \sigma_i$$

$$\forall_{ij} : \alpha'_{ij} = \frac{1}{2}\left(\alpha_{ij}^{(1)} + \alpha_{ij}^{(2)}\right) \qquad \text{Intermediate recombination of } \alpha_{ij},$$

Constraint selection and fitness assessment

- Let P’ be a constraint pool made of µ parent constraints and λ offspring constraints produced by the search operators
- Selection of the minimal subset of P’ that maximize the number of correctly classified examples is a generalized set cover problem



$$\min \overbrace{\sum_{c_j \in P'} |X_{\overline{c_j}}| b_j}^{(a)} + \overbrace{\sum_{x_k \in U} v_k}^{(b)} + \overbrace{0.0001 \sum_{c_j \in P'} b_j}^{(c)} \qquad (1)$$

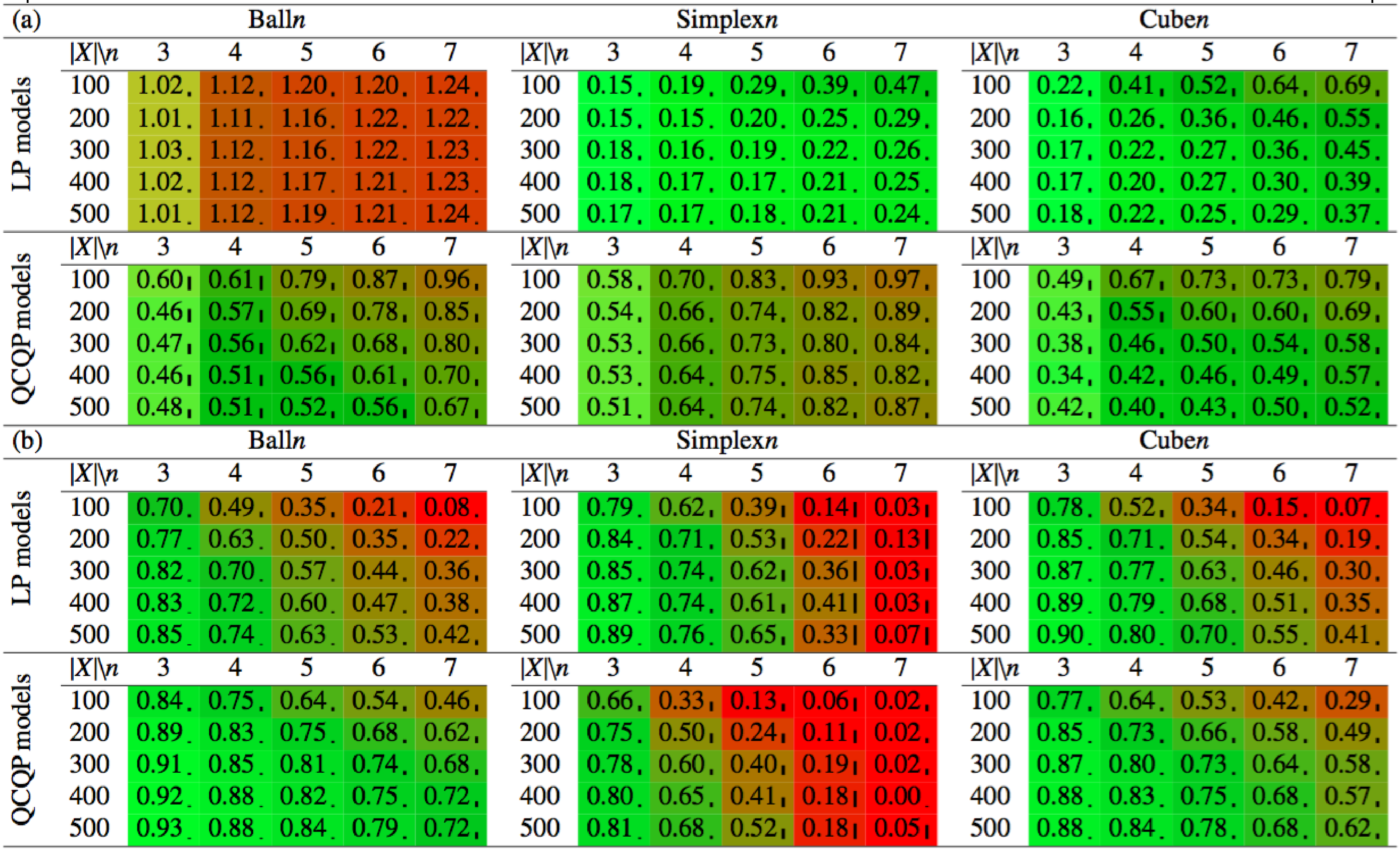
subject to

$$\forall \boldsymbol{x}_k \in U : \sum_{\substack{c_j \in P', \\ \neg c_j(\boldsymbol{x}_k)}} b_j + v_k \geq 1 \qquad (2)$$

$$\forall b_j, \forall v_j \in \{0, 1\},$$

- Where
  - (a) is the number of feasible examples violating the selected constraints
  - (b) is the number of unlabeled examples satisfying the selected constraints
  - (c) is the number of constraints
- This problem is solved optimally, thus the synthesized model is minimal
- The selected constraints are removed from P’ and advance to the next generation population
- Constraint selection repeats until the next generation population is full

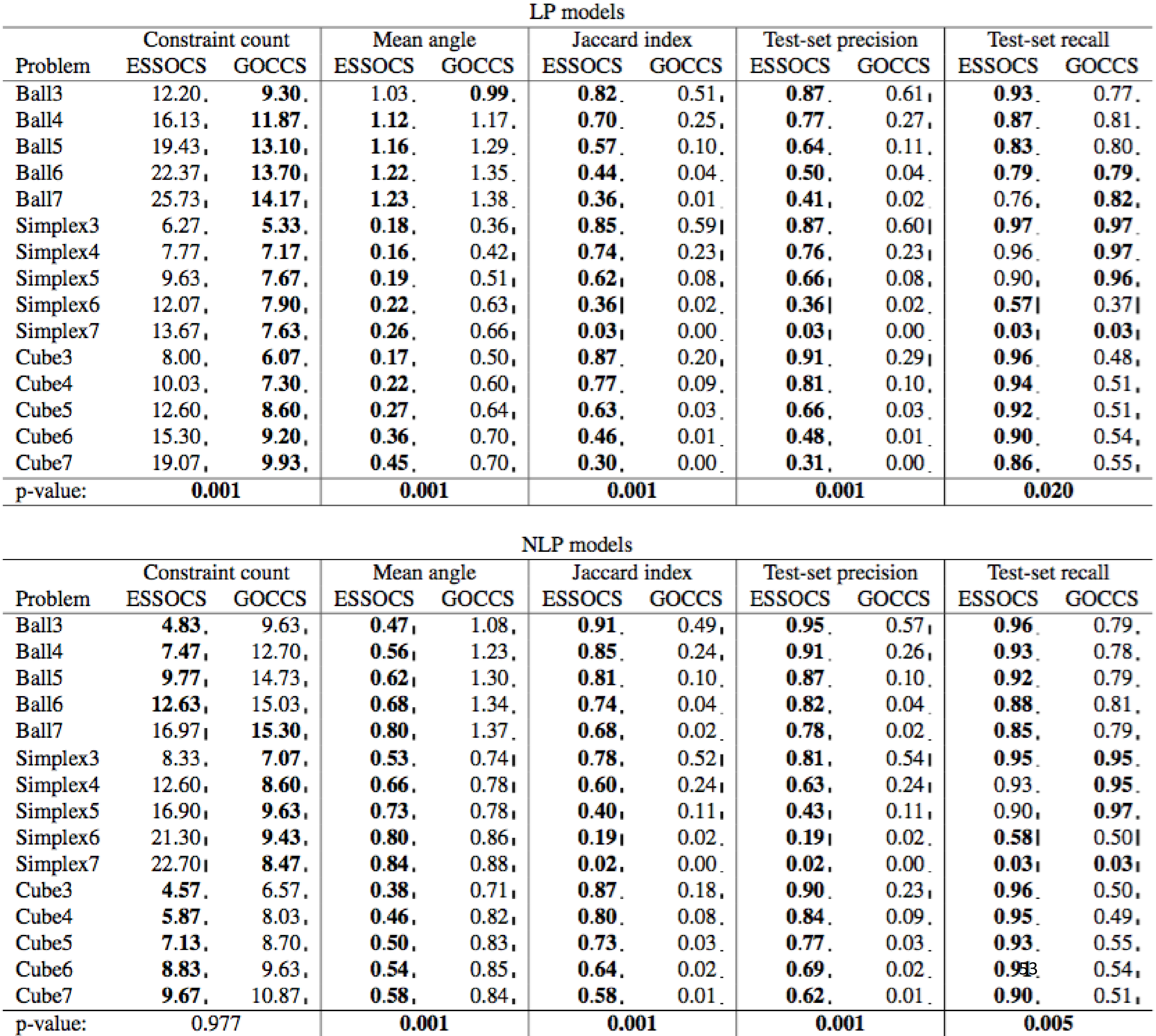
(a) Mean angle between the corresponding constraints
(b) Jaccard index of the feasible regions
of the synthesized and the actual MP models



(a)		Ball*n*						Simplex*n*						Cube*n*				
	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7
LP models	100	1.02	1.12	1.20	1.20	1.24	100	0.15	0.19	0.29	0.39	0.47	100	0.22	0.41	0.52	0.64	0.69
	200	1.01	1.11	1.16	1.22	1.22	200	0.15	0.15	0.20	0.25	0.29	200	0.16	0.26	0.36	0.46	0.55
	300	1.03	1.12	1.16	1.22	1.23	300	0.18	0.16	0.19	0.22	0.26	300	0.17	0.22	0.27	0.36	0.45
	400	1.02	1.12	1.17	1.21	1.23	400	0.18	0.17	0.17	0.21	0.25	400	0.17	0.20	0.27	0.30	0.39
	500	1.01	1.12	1.19	1.21	1.24	500	0.17	0.17	0.18	0.21	0.24	500	0.18	0.22	0.25	0.29	0.37
	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7
QCQP models	100	0.60	0.61	0.79	0.87	0.96	100	0.58	0.70	0.83	0.93	0.97	100	0.49	0.67	0.73	0.73	0.79
	200	0.46	0.57	0.69	0.78	0.85	200	0.54	0.66	0.74	0.82	0.89	200	0.43	0.55	0.60	0.60	0.69
	300	0.47	0.56	0.62	0.68	0.80	300	0.53	0.66	0.73	0.80	0.84	300	0.38	0.46	0.50	0.54	0.58
	400	0.46	0.51	0.56	0.61	0.70	400	0.53	0.64	0.75	0.85	0.82	400	0.34	0.42	0.46	0.49	0.57
	500	0.48	0.51	0.52	0.56	0.67	500	0.51	0.64	0.74	0.82	0.87	500	0.42	0.40	0.43	0.50	0.52

(b)		Ball*n*						Simplex*n*						Cube*n*				
	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7
LP models	100	0.70	0.49	0.35	0.21	0.08	100	0.79	0.62	0.39	0.14	0.03	100	0.78	0.52	0.34	0.15	0.07
	200	0.77	0.63	0.50	0.35	0.22	200	0.84	0.71	0.53	0.22	0.13	200	0.85	0.71	0.54	0.34	0.19
	300	0.82	0.70	0.57	0.44	0.36	300	0.85	0.74	0.62	0.36	0.03	300	0.87	0.77	0.63	0.46	0.30
	400	0.83	0.72	0.60	0.47	0.38	400	0.87	0.74	0.61	0.41	0.03	400	0.89	0.79	0.68	0.51	0.35
	500	0.85	0.74	0.63	0.53	0.42	500	0.89	0.76	0.65	0.33	0.07	500	0.90	0.80	0.70	0.55	0.41
	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7	|*X*|\n	3	4	5	6	7
QCQP models	100	0.84	0.75	0.64	0.54	0.46	100	0.66	0.33	0.13	0.06	0.02	100	0.77	0.64	0.53	0.42	0.29
	200	0.89	0.83	0.75	0.68	0.62	200	0.75	0.50	0.24	0.11	0.02	200	0.85	0.73	0.66	0.58	0.49
	300	0.91	0.85	0.81	0.74	0.68	300	0.78	0.60	0.40	0.19	0.02	300	0.87	0.80	0.73	0.64	0.58
	400	0.92	0.88	0.82	0.75	0.72	400	0.80	0.65	0.41	0.18	0.00	400	0.88	0.83	0.75	0.68	0.57
	500	0.93	0.88	0.84	0.79	0.72	500	0.81	0.68	0.52	0.18	0.05	500	0.88	0.84	0.78	0.68	0.62

ESOCCS vs GOCCS



LP models

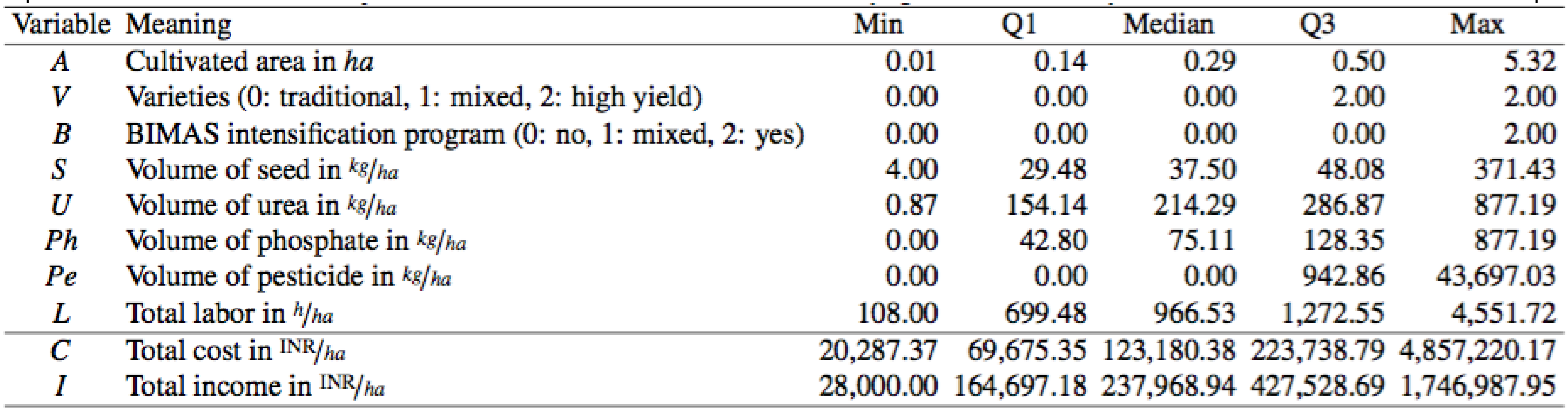
	Constraint count		Mean angle		Jaccard index		Test-set precision		Test-set recall	
Problem	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS
Ball3	12.20	**9.30**	1.03	**0.99**	**0.82**	0.51	**0.87**	0.61	**0.93**	0.77
Ball4	16.13	**11.87**	**1.12**	1.17	**0.70**	0.25	**0.77**	0.27	**0.87**	0.81
Ball5	19.43	**13.10**	**1.16**	1.29	**0.57**	0.10	**0.64**	0.11	**0.83**	0.80
Ball6	22.37	**13.70**	**1.22**	1.35	**0.44**	0.04	**0.50**	0.04	**0.79**	**0.79**
Ball7	25.73	**14.17**	**1.23**	1.38	**0.36**	0.01	**0.41**	0.02	0.76	**0.82**
Simplex3	6.27	**5.33**	**0.18**	0.36	**0.85**	0.59	**0.87**	0.60	**0.97**	**0.97**
Simplex4	7.77	**7.17**	**0.16**	0.42	**0.74**	0.23	**0.76**	0.23	0.96	**0.97**
Simplex5	9.63	**7.67**	**0.19**	0.51	**0.62**	0.08	**0.66**	0.08	0.90	**0.96**
Simplex6	12.07	**7.90**	**0.22**	0.63	**0.36**	0.02	**0.36**	0.02	**0.57**	0.37
Simplex7	13.67	**7.63**	**0.26**	0.66	**0.03**	0.00	**0.03**	0.00	**0.03**	**0.03**
Cube3	8.00	**6.07**	**0.17**	0.50	**0.87**	0.20	**0.91**	0.29	**0.96**	0.48
Cube4	10.03	**7.30**	**0.22**	0.60	**0.77**	0.09	**0.81**	0.10	**0.94**	0.51
Cube5	12.60	**8.60**	**0.27**	0.64	**0.63**	0.03	**0.66**	0.03	**0.92**	0.51
Cube6	15.30	**9.20**	**0.36**	0.70	**0.46**	0.01	**0.48**	0.01	**0.90**	0.54
Cube7	19.07	**9.93**	**0.45**	0.70	**0.30**	0.00	**0.31**	0.00	**0.86**	0.55
p-value:	**0.001**		**0.001**		**0.001**		**0.001**		**0.020**	

NLP models

	Constraint count		Mean angle		Jaccard index		Test-set precision		Test-set recall	
Problem	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS	ESSOCS	GOCCS
Ball3	**4.83**	9.63	**0.47**	1.08	**0.91**	0.49	**0.95**	0.57	**0.96**	0.79
Ball4	**7.47**	12.70	**0.56**	1.23	**0.85**	0.24	**0.91**	0.26	**0.93**	0.78
Ball5	**9.77**	14.73	**0.62**	1.30	**0.81**	0.10	**0.87**	0.10	**0.92**	0.79
Ball6	**12.63**	15.03	**0.68**	1.34	**0.74**	0.04	**0.82**	0.04	**0.88**	0.81
Ball7	16.97	**15.30**	**0.80**	1.37	**0.68**	0.02	**0.78**	0.02	**0.85**	0.79
Simplex3	8.33	**7.07**	**0.53**	0.74	**0.78**	0.52	**0.81**	0.54	**0.95**	**0.95**
Simplex4	12.60	**8.60**	**0.66**	0.78	**0.60**	0.24	**0.63**	0.24	0.93	**0.95**
Simplex5	16.90	**9.63**	**0.73**	0.78	**0.40**	0.11	**0.43**	0.11	0.90	**0.97**
Simplex6	21.30	**9.43**	**0.80**	0.86	**0.19**	0.02	**0.19**	0.02	**0.58**	0.50
Simplex7	22.70	**8.47**	**0.84**	0.88	**0.02**	0.00	**0.02**	0.00	**0.03**	**0.03**
Cube3	**4.57**	6.57	**0.38**	0.71	**0.87**	0.18	**0.90**	0.23	**0.96**	0.50
Cube4	**5.87**	8.03	**0.46**	0.82	**0.80**	0.08	**0.84**	0.09	**0.95**	0.49
Cube5	**7.13**	8.70	**0.50**	0.83	**0.73**	0.03	**0.77**	0.03	**0.93**	0.55
Cube6	**8.83**	9.63	**0.54**	0.85	**0.64**	0.02	**0.69**	0.02	**0.91**	0.54
Cube7	**9.67**	10.87	**0.58**	0.84	**0.58**	0.01	**0.62**	0.01	**0.90**	0.51
p-value:	0.977		**0.001**		**0.001**		**0.001**		**0.005**	

Modeling Rice Production

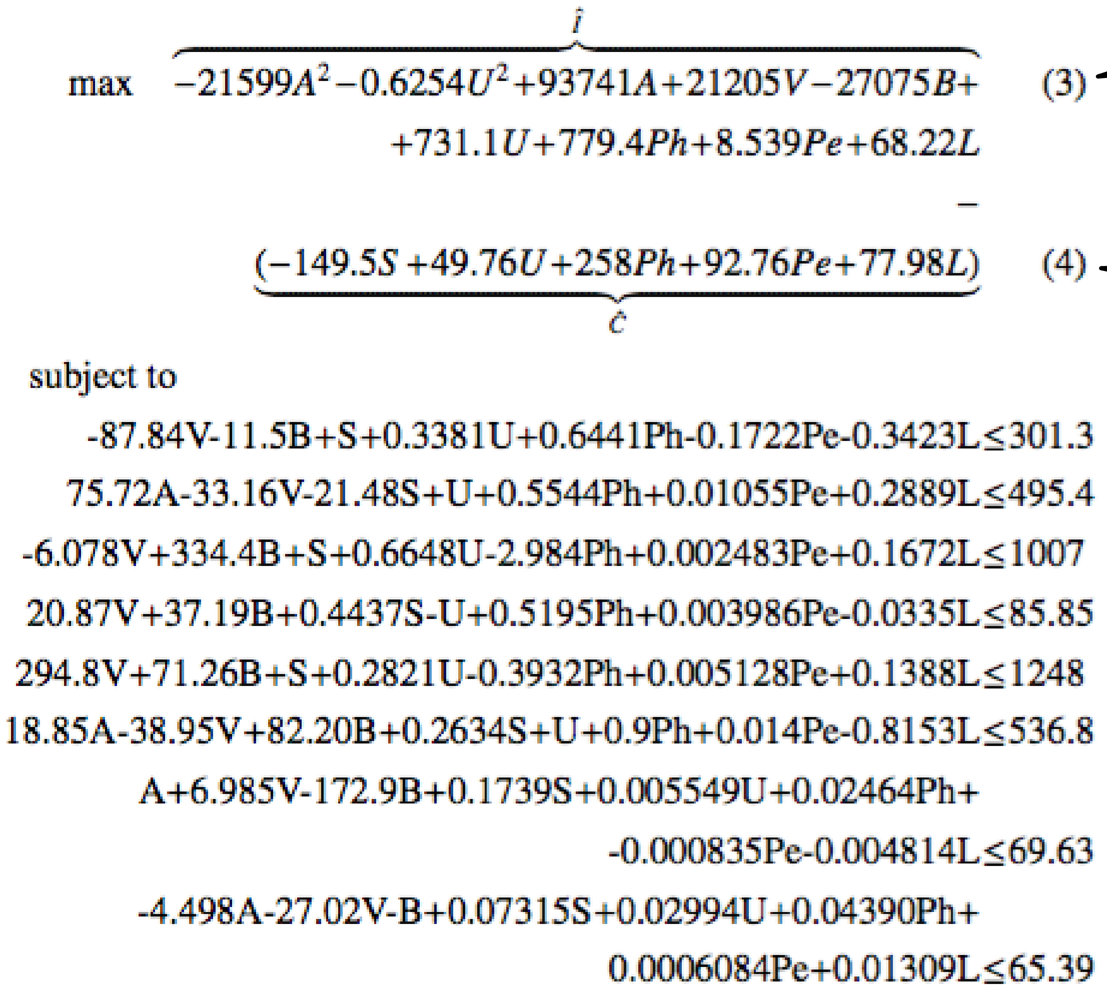
- Rice production in India data set [1]
  - Heavily preprocessed in this work
    - Variables in absolute units recalculated relatively to the farm area
    - Ordinal variables transformed into integers
    - Six output variables aggregated into two variables
  - 1022 examples of rice farms



Variable	Meaning	Min	Q1	Median	Q3	Max
A	Cultivated area in *ha*	0.01	0.14	0.29	0.50	5.32
V	Varieties (0: traditional, 1: mixed, 2: high yield)	0.00	0.00	0.00	2.00	2.00
B	BIMAS intensification program (0: no, 1: mixed, 2: yes)	0.00	0.00	0.00	0.00	2.00
S	Volume of seed in $^{kg}/_{ha}$	4.00	29.48	37.50	48.08	371.43
U	Volume of urea in $^{kg}/_{ha}$	0.87	154.14	214.29	286.87	877.19
Ph	Volume of phosphate in $^{kg}/_{ha}$	0.00	42.80	75.11	128.35	877.19
Pe	Volume of pesticide in $^{kg}/_{ha}$	0.00	0.00	0.00	942.86	43,697.03
L	Total labor in $^{h}/_{ha}$	108.00	699.48	966.53	1,272.55	4,551.72
C	Total cost in $^{INR}/_{ha}$	20,287.37	69,675.35	123,180.38	223,738.79	4,857,220.17
I	Total income in $^{INR}/_{ha}$	28,000.00	164,697.18	237,968.94	427,528.69	1,746,987.95

[1] Q.Feng, W.C.Horrace, Alternative technical efficiency measures: Skew, bias and scale, Journal of Applied Econometrics 27 (2) (2012) 253-268.

Rice production QP model



$$\max \quad \overbrace{-21599A^2-0.6254U^2+93741A+21205V-27075B+ \\ +731.1U+779.4Ph+8.539Pe+68.22L}^{i} \qquad (3)$$

$$-$$

$$\underbrace{(-149.5S+49.76U+258Ph+92.76Pe+77.98L)}_{\hat{c}} \qquad (4)$$

subject to

$$-87.84V-11.5B+S+0.3381U+0.6441Ph-0.1722Pe-0.3423L \le 301.3$$

$$75.72A-33.16V-21.48S+U+0.5544Ph+0.01055Pe+0.2889L \le 495.4$$

$$-6.078V+334.4B+S+0.6648U-2.984Ph+0.002483Pe+0.1672L \le 1007$$

$$20.87V+37.19B+0.4437S-U+0.5195Ph+0.003986Pe-0.0335L \le 85.85$$

$$294.8V+71.26B+S+0.2821U-0.3932Ph+0.005128Pe+0.1388L \le 1248$$

$$18.85A-38.95V+82.20B+0.2634S+U+0.9Ph+0.014Pe-0.8153L \le 536.8$$

$$A+6.985V-172.9B+0.1739S+0.005549U+0.02464Ph+ \\ -0.000835Pe-0.004814L \le 69.63$$

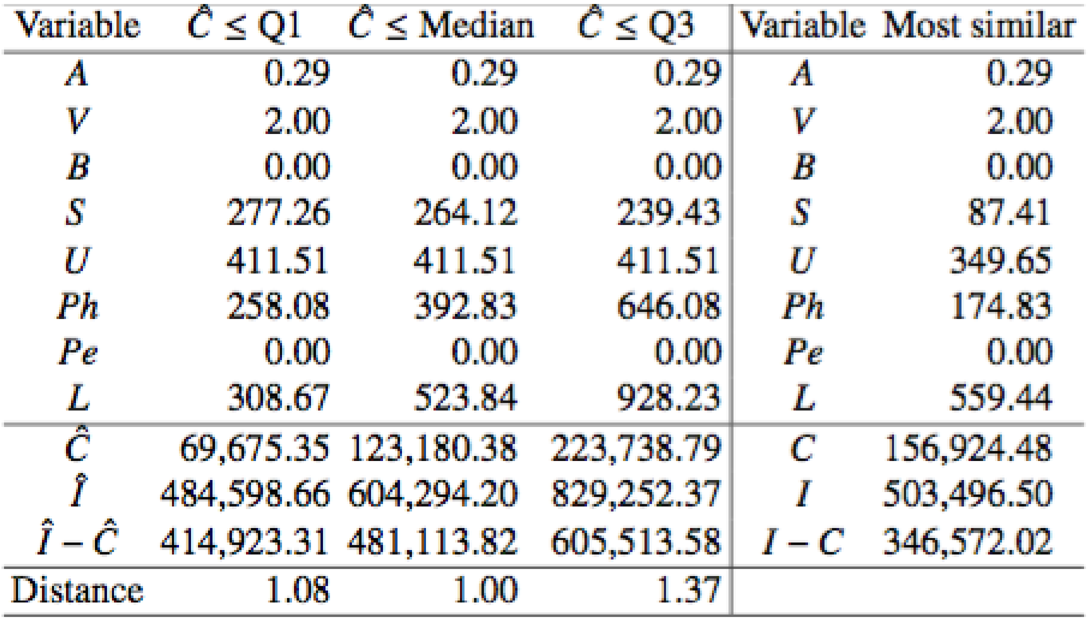
$$-4.498A-27.02V-B+0.07315S+0.02994U+0.04390Ph+ \\ 0.0006084Pe+0.01309L \le 65.39$$

Estimate of the total income

Estimate of the total cost

The optimal solutions to the rice production model(s)

- Three cases of farmer’s budget:
  - Q1, Median, Q3 cost in the data set
  - Modeled using an extra constraint on the cost estimate Ĉ



Variable	$\hat{C} \leq$ Q1	$\hat{C} \leq$ Median	$\hat{C} \leq$ Q3	Variable	Most similar
A	0.29	0.29	0.29	A	0.29
V	2.00	2.00	2.00	V	2.00
B	0.00	0.00	0.00	B	0.00
S	277.26	264.12	239.43	S	87.41
U	411.51	411.51	411.51	U	349.65
Ph	258.08	392.83	646.08	Ph	174.83
Pe	0.00	0.00	0.00	Pe	0.00
L	308.67	523.84	928.23	L	559.44
$\hat{C}$	69,675.35	123,180.38	223,738.79	C	156,924.48
$\hat{I}$	484,598.66	604,294.20	829,252.37	I	503,496.50
$\hat{I} - \hat{C}$	414,923.31	481,113.82	605,513.58	$I - C$	346,572.02
Distance	1.08	1.00	1.37		

ESOCCS^2 – five improvements to ESOCCS

- I1: Estimation of distribution of feasible states using Kernel Density Estimation (KDE) instead of Expectation Maximization (EM)
  - The assumption that the actual distribution is a Gaussian Mixture may be false
  - KDE is non-parametric estimation method
  - It difficult to estimate bandwidth matrix for KDE and many approaches exist
    - We used Silverman’s rule
- I2: Denser sampling of unlabeled examples
  - In ESOCCS the density of a sample of unlabeled examples decreases with dimensionality
  - ESOCCS^2 increases the sample size to reduce the decrease rate
- I3: Bounding-box initialization
  - The initial population is supplemented with bounding-box of the training set
- I4: Reuse of constraints in successively built models
  - The constraints are not removed from the pool, but use of some subsets of constraints together is forbidden
- I5: Prevention from degenerate empty models
  - Degenerated models are explicitly prohibited by new formulation of the generalized set cover problem

Tomasz P. Pawlak, Performance Improvements for Evolutionary Strategy-based One-Class Constraint Synthesis, GECCO‘18, ACM, 2018.

ESOCCS vs ESOCCS^2

- In total 23 combinations of I1 – I4 applied to ESOCCS are verified
- The p-values of the Wilcoxon signed rank test for significance of differences in Jaccard indexes of the feasible regions for each of I1 – I4:

	I1	I2	I3	I4
LP	**0.001**	**0.001**	0.135	**0.001**
NLP	**0.001**	**0.001**	0.074	**0.001**

In fact, this is significant deterioration

- ESOCCS with the combination of I2, I3, I4 is significantly better than bare ESOCCS and 16 other setups in LP models and 17 other setups in NLP models

ESOCCS: conclusions

- A fully configurable type of the synthesized model
- Good generalization performance and handling of noise
- Significant computation cost due to solving the set cover problem
- The curse of dimensionality is still an issue
- KDE (I1) failed probably due to inadequate algorithm for calculating bandwidth matrix

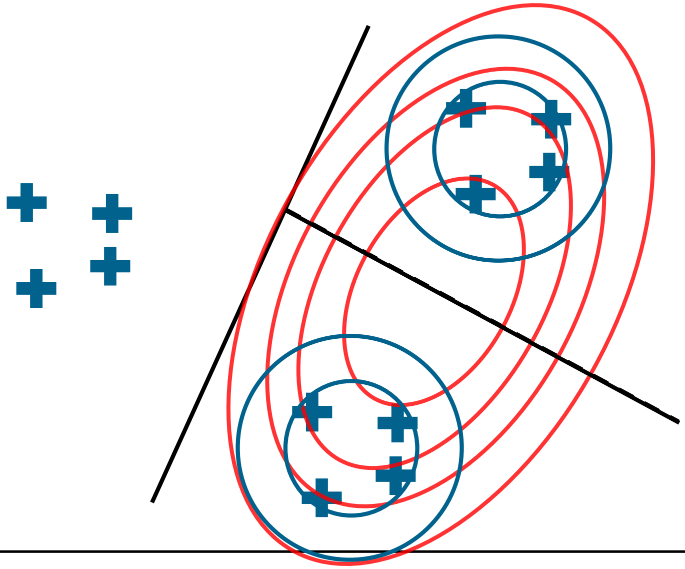
One-Class Constraint Acquisition with Local Search (OCCALS) for 1-CSP

- General idea:
  - Partition the training set using x-means
  - Find an LP model for each partition using local search
  - Remove redundant constraints in each LP model independently
  - Create a MILP model implementing alternative of the individual LP models

- Daniel Sroka, Tomasz P. Pawlak, One-Class Constraint Acquisition with Local Search, GECCO '18, ACM, 2018.

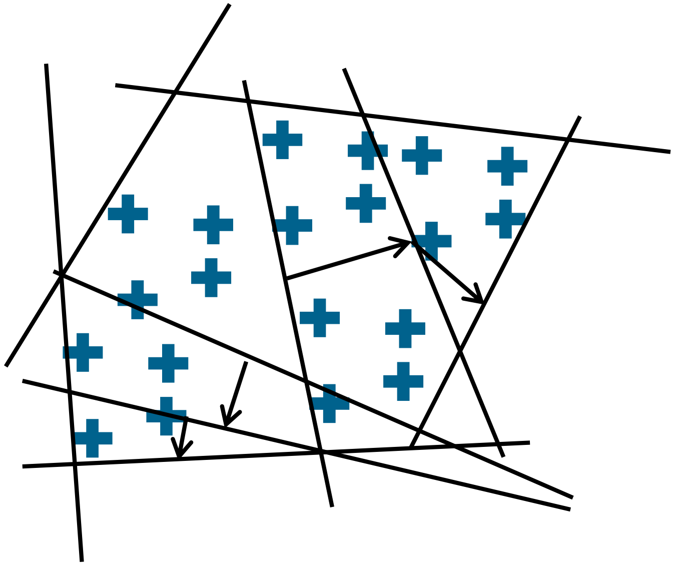
Training set partitioning using x-means

- Make an initial partitioning using k-means with a fixed k
  - E.g., k = 2
- Split each partition P_i further into P_i' and P_i'' using k-means with k=2
- If BIC(P_i', P_i'') > BIC(P_i) then
  - Replace P_i with P_i' and P_i''
  - Repeat these steps for P_i' and P_i''
- BIC is Bayesian Information Criterion under Gaussian Mixture



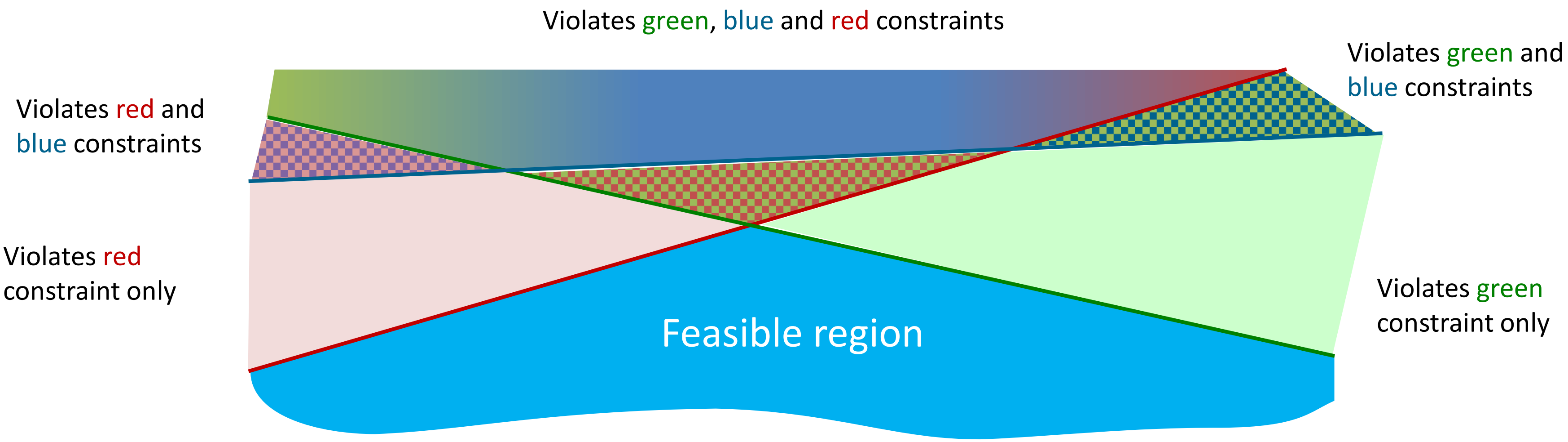
Local search of LP constraints

- For each partition c_{max} constraints are sought independently
- Local search finds one constraint at time
- The initial weights of a constraint are random
- Local search minimizes the number of false negatives
- Step size decreases with true positives



Removal of redundant constraints

- The same approach like in GENETICS is taken



Violates green, blue and red constraints

Violates red and blue constraints

Violates green and blue constraints

Violates red constraint only

Violates green constraint only

Feasible region

No region contains solutions that violate blue constraint only:
Blue constraint is redundant.

A MILP model implementing an alternative of LP models

Individual LP Models

$$\mathbf{w}_1\ \mathbf{x} \le a_1$$
$$\mathbf{w}_2\ \mathbf{x} \le a_2$$
$$\mathbf{w}_3\ \mathbf{x} \le a_3$$

$$\mathbf{v}_1\ \mathbf{x} \le b_1$$
$$\mathbf{v}_2\ \mathbf{x} \le b_2$$
$$\mathbf{v}_3\ \mathbf{x} \le b_3$$

A MILP model with
auxiliary binary variables d_i

$$\mathbf{w}_1\ \mathbf{x} + Ld_1 \le a_1 + L$$
$$\mathbf{w}_2\ \mathbf{x} + Ld_1 \le a_2 + L$$
$$\mathbf{w}_3\ \mathbf{x} + Ld_1 \le a_3 + L$$

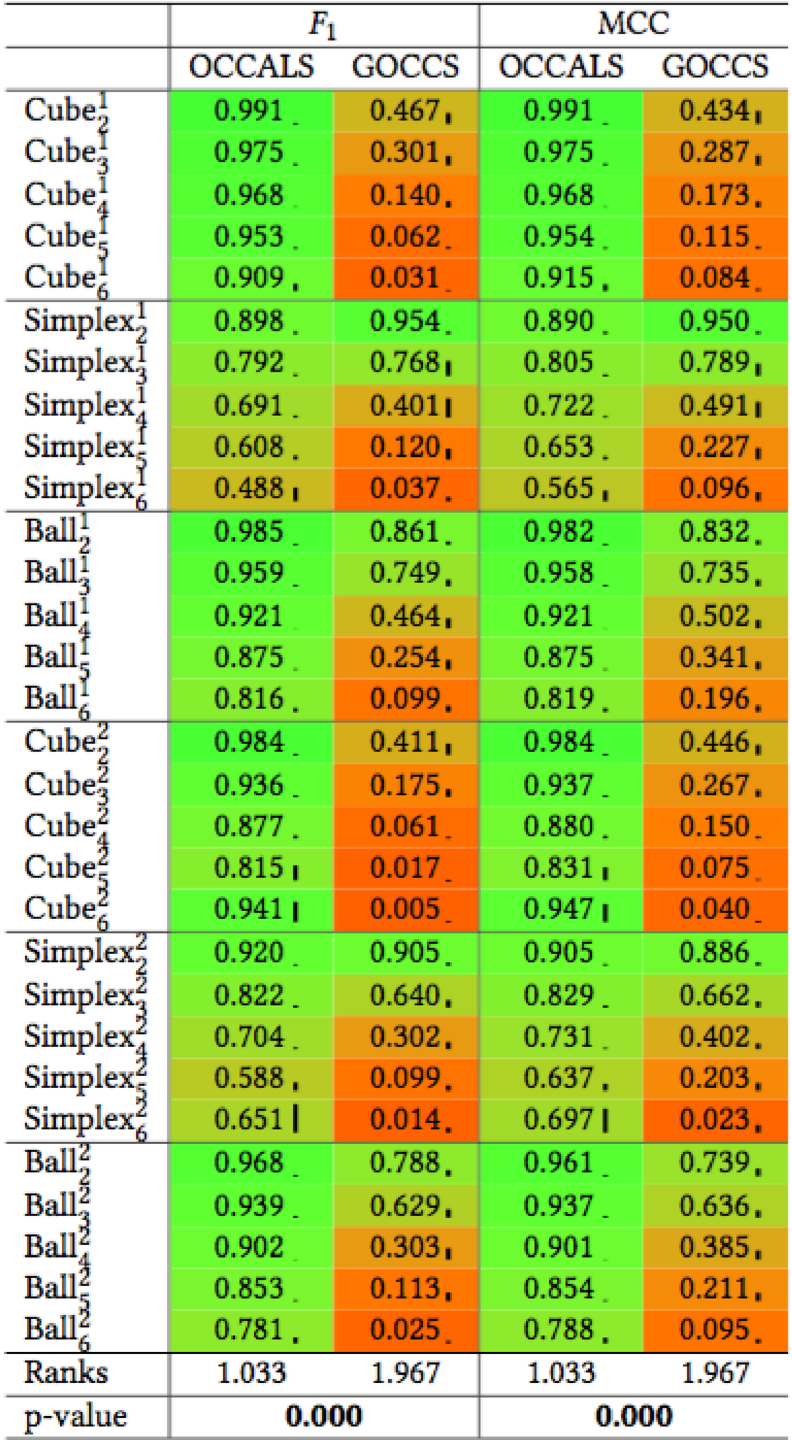
$$\mathbf{v}_1\ \mathbf{x} + Ld_2 \le b_1 + L$$
$$\mathbf{v}_2\ \mathbf{x} + Ld_2 \le b_2 + L$$
$$\mathbf{v}_3\ \mathbf{x} + Ld_2 \le b_3 + L$$

$$d_1 + d_2 \ge 1$$
$$d_1, d_2 \in \{0, 1\}$$

L is a big constant

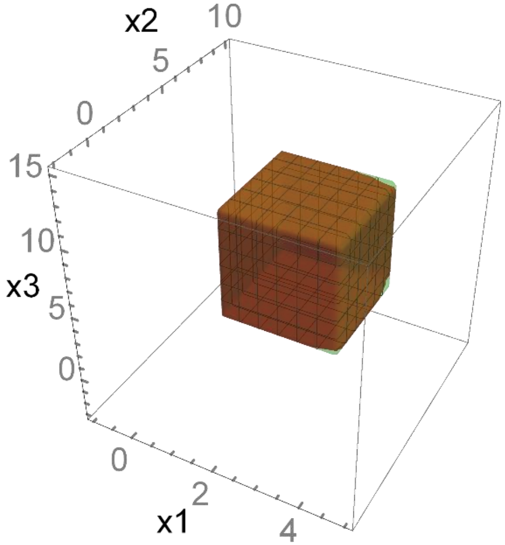
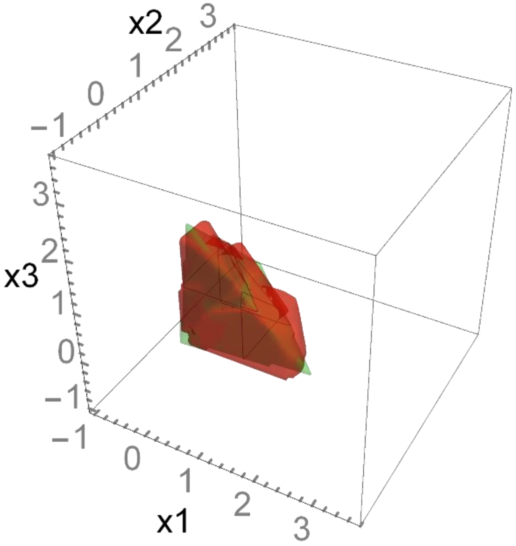
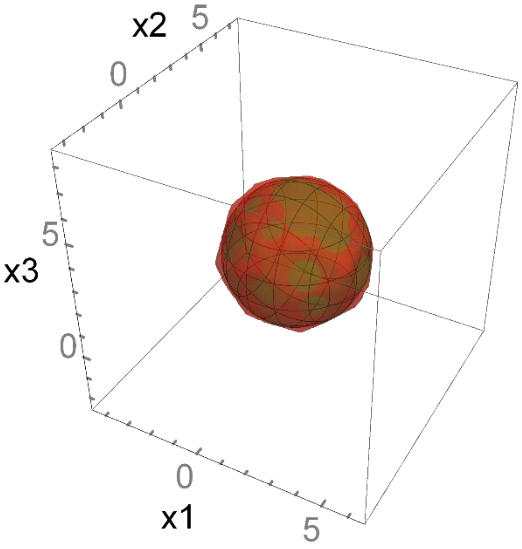
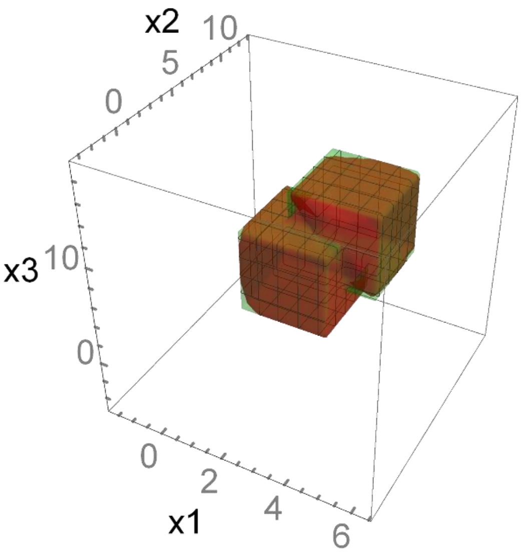
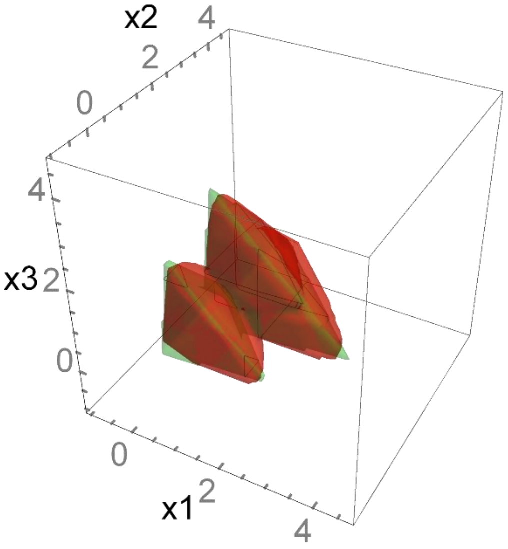
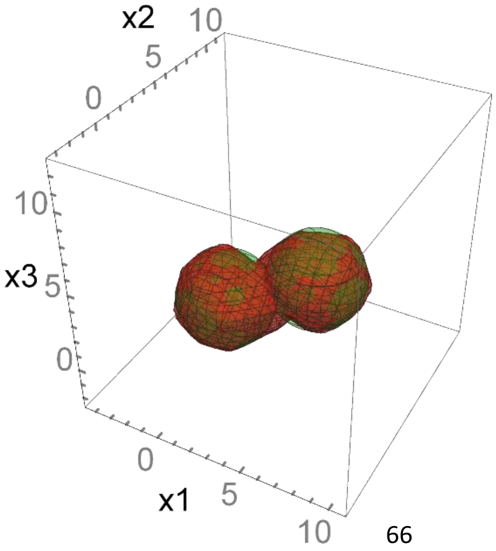
OCCALS vs GOCCS

- F_1 score on test set
- MCC – Matthew’s Correlation Coefficient on test set
  - 1 – ideal classification
  - 0 – random classification
  - -1 – anti-ideal classification
- The p-values for Wilcoxon signed-rank test for differences



	F_1		MCC	
	OCCALS	GOCCS	OCCALS	GOCCS
Cube_2^1	0.991	0.467	0.991	0.434
Cube_3^1	0.975	0.301	0.975	0.287
Cube_4^1	0.968	0.140	0.968	0.173
Cube_5^1	0.953	0.062	0.954	0.115
Cube_6^1	0.909	0.031	0.915	0.084
Simplex_2^1	0.898	0.954	0.890	0.950
Simplex_3^1	0.792	0.768	0.805	0.789
Simplex_4^1	0.691	0.401	0.722	0.491
Simplex_5^1	0.608	0.120	0.653	0.227
Simplex_6^1	0.488	0.037	0.565	0.096
Ball_2^1	0.985	0.861	0.982	0.832
Ball_3^1	0.959	0.749	0.958	0.735
Ball_4^1	0.921	0.464	0.921	0.502
Ball_5^1	0.875	0.254	0.875	0.341
Ball_6^1	0.816	0.099	0.819	0.196
Cube_2^2	0.984	0.411	0.984	0.446
Cube_3^2	0.936	0.175	0.937	0.267
Cube_4^2	0.877	0.061	0.880	0.150
Cube_5^2	0.815	0.017	0.831	0.075
Cube_6^2	0.941	0.005	0.947	0.040
Simplex_2^2	0.920	0.905	0.905	0.886
Simplex_3^2	0.822	0.640	0.829	0.662
Simplex_4^2	0.704	0.302	0.731	0.402
Simplex_5^2	0.588	0.099	0.637	0.203
Simplex_6^2	0.651	0.014	0.697	0.023
Ball_2^2	0.968	0.788	0.961	0.739
Ball_3^2	0.939	0.629	0.937	0.636
Ball_4^2	0.902	0.303	0.901	0.385
Ball_5^2	0.853	0.113	0.854	0.211
Ball_6^2	0.781	0.025	0.788	0.095
Ranks	1.033	1.967	1.033	1.967
p-value	**0.000**		**0.000**	

Visualization of the synthesized models

	Cube	Simplex	Ball
k=1			
k=2			

OCCALS: conclusions

- The most successful algorithm so far
- Quick and fast
- Good generalization performance and noise handling
- An improved version is under development

General conclusions

- Mathematical Programming (MP) is a common formalism for optimization methods, well-recognized both in business and academia
- Surprisingly, most of the MP models are handcrafted and the synthesis of MP models from data is underrepresented in literature
- We formally defined several variants of the synthesis problem, including 1-CSP and 2-CSP
- 2-CSP problem is known to be NP-hard
- 1-CSP is more practical due to no need for infeasible examples, but it is also more difficult to solve than 2-CSP
- We proposed several heuristic approaches to solve both 1-CSP and 2-CSP

Future research directions

- Drop requirement for second-class examples in some of the presented algorithms
- Design novel performance measures calculable using feasible examples only
- Design more objective experimental protocol for real-world data