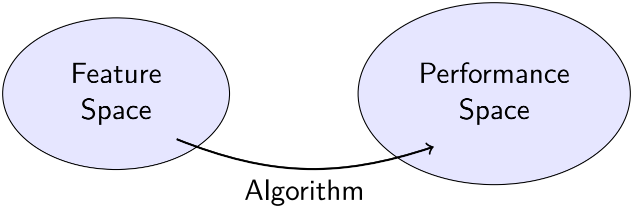
Generating mixed integer programming instances with challenging properties

Simon Bowly

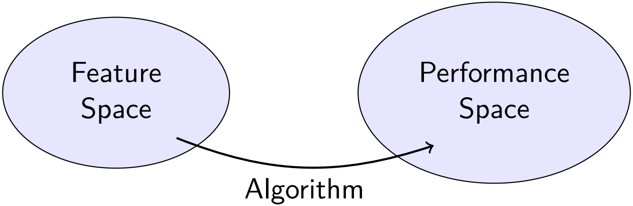
University of Melbourne

AMSI Optimise
June 20, 2019

Outline



Outline



Give an overview of...

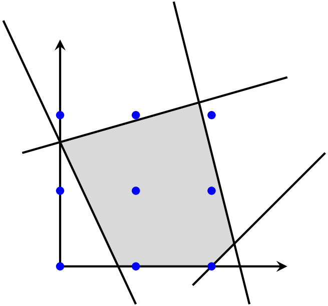
- MIP solvers (in particular branching strategies)
- Generating new instances using evolutionary algorithms
- Characteristics of the generated instances

MIP/Polyhedra

Mixed Integer Program

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

some or all x_i integral



MIP Solvers

Mixed Integer Program

$$
\begin{aligned}
&\text{maximize} && c^T x \\
&\text{subject to} && Ax \leq b \\
& && x \geq 0 \\
&\text{some or all } x_i \text{ integral}
\end{aligned}
$$

Many interacting components make up a MIP solver:

- Presolvers
- Cutting planes
- Primal heuristics
- Parallelisation
- LP solvers
- Branching rules
- Node selection rules
- Domain propagation

MIP Solvers

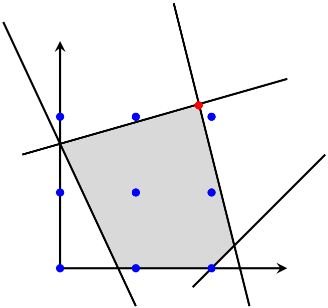
Mixed Integer Program

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \\ \text{some or all } x_i \text{ integral} \end{aligned}$$

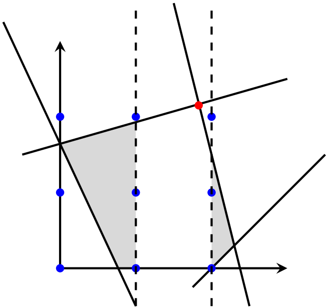
Many interacting components make up a MIP solver:

- Presolvers
- Cutting planes
- Primal heuristics
- Parallelisation
- LP solvers
- Branching rules
- Node selection rules
- Domain propagation

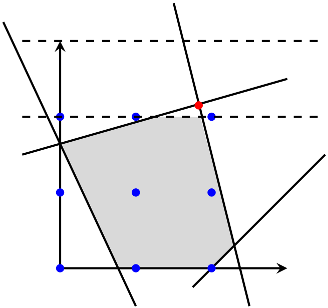
Branching Strategies



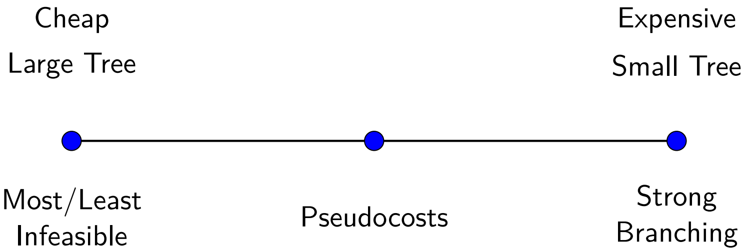
Branching Strategies



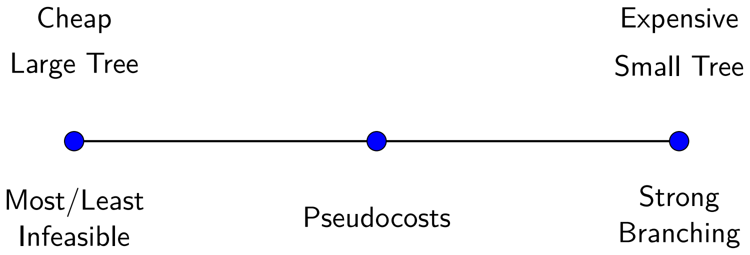
Branching Strategies



Branching Strategies



Branching Strategies



To conduct a branching study . . .

- Disable cuts beyond the root node
- Provide the optimal solution

(Linderoth and Savelsbergh, 1999)

Evolving Instances

Genetic algorithms

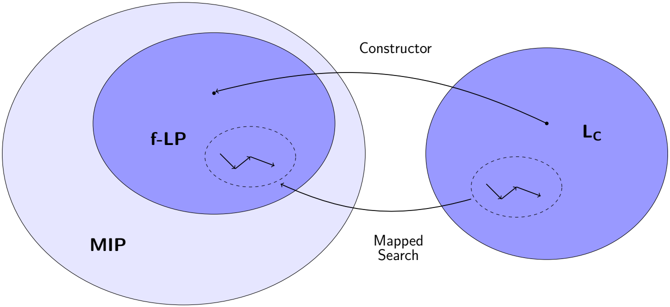
- Population based metaheuristic with fitness selection
- Crossover is the key search operator

Previous applications to instance generation

- Finding worst-case bounds for sorting algorithm performance (Cotta and Moscato, 2003).
- Exposing strengths and weaknesses of heuristics (van Hemert, 2006; Langdon and Poli, 2007).

Evolving Instances

How can we narrow the search space?



Restrict search to feasible, bounded problems.

Evolving Instances

Dual programs

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax + s = b \\ & x, s \geq 0 \end{aligned} \iff \begin{aligned} \min \quad & b^T y \\ \text{s.t.} \quad & A^T y - r = c \\ & y, r \geq 0 \end{aligned}$$

Optimality conditions

(x, s) and (y, r) feasible to primal and dual
$x_i r_i = 0 \quad \forall\, i$
$y_j s_j = 0 \quad \forall\, j$

Construction

$$\begin{aligned} b &= Ax + s \\ c &= A^T y - r \end{aligned}$$

Evolving Instances

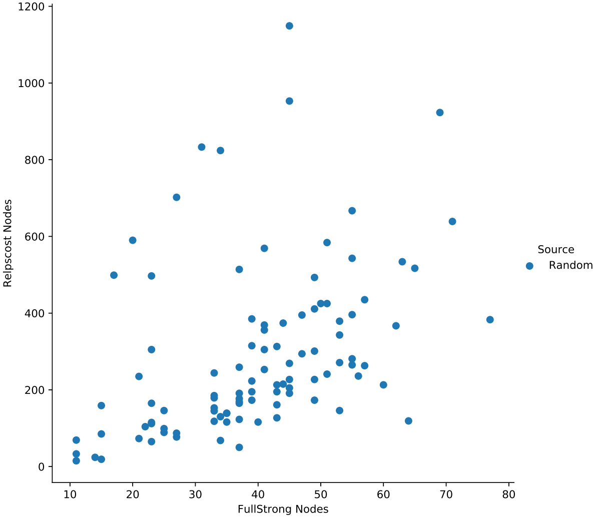
- Uniform row crossover:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ a_{31} & \dots & a_{3n} \end{bmatrix} \quad \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{2n} \\ b_{31} & \dots & b_{3n} \end{bmatrix}$$

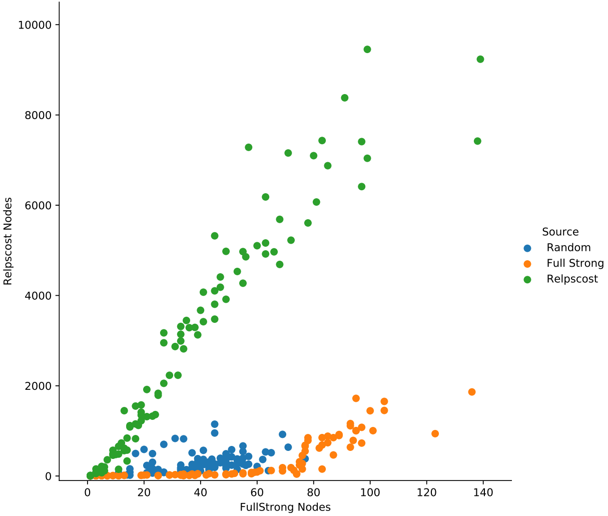
$$\begin{bmatrix} a_{21} & \dots & a_{2n} \\ b_{21} & \dots & b_{2n} \\ a_{31} & \dots & a_{3n} \end{bmatrix} \quad \begin{bmatrix} a_{11} & \dots & a_{1n} \\ b_{31} & \dots & b_{3n} \\ b_{11} & \dots & b_{1n} \end{bmatrix}$$

- Multi-objective NSGA-II algorithm (Deb et al., 2002)
- Small instances - 50 integer variables and 50 constraints
- *fullstrong* and *relpscost* branching strategies in SCIP 6.0.0

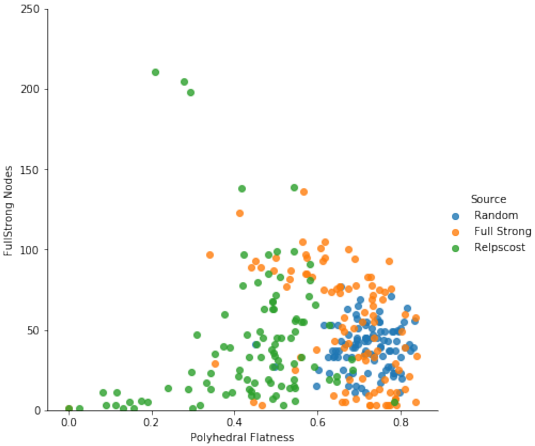
Performance Discriminating Instances



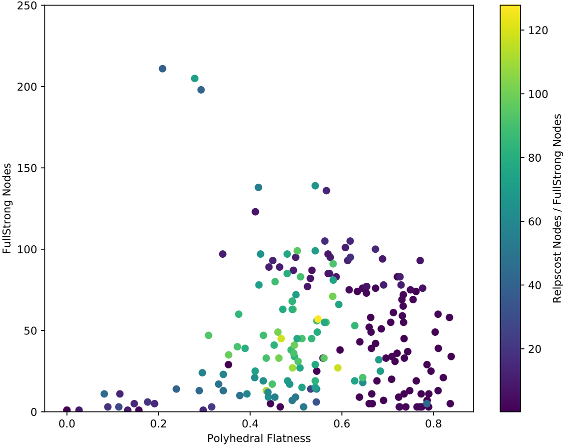
Performance Discriminating Instances



Properties of Generated Instances



Properties of Generated Instances



Takeaways

- Evolutionary generation allows us to search performance space by manipulating instances.
- Small instances can be generated which expose performance differences in fundamental components.
- Branching is just one facet of MIP solver performance - there's more of the performance space to explore.

References

Cotta, C. and Moscato, P. (2003). A mixed evolutionary-statistical analysis of an algorithm's complexity. *Applied Mathematics Letters*, 16(1):41–47.

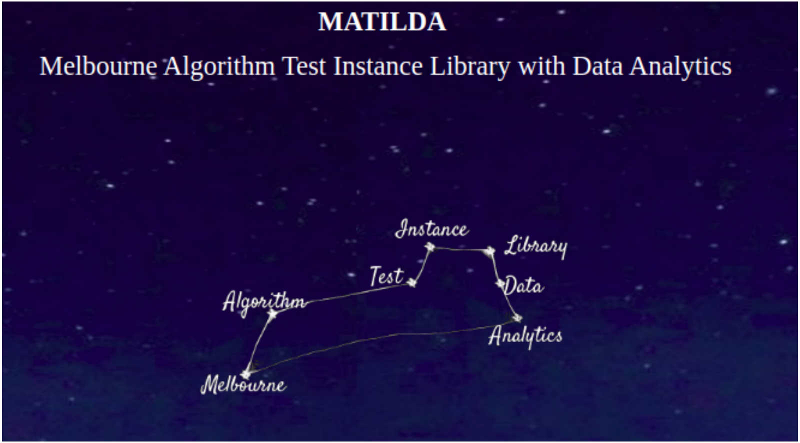
Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Langdon, W. and Poli, R. (2007). Evolving Problems to Learn About Particle Swarm Optimizers and Other Search Algorithms. *IEEE Transactions on Evolutionary Computation*, 11(5):561–578.

Linderoth, J. T. and Savelsbergh, M. W. P. (1999). A Computational Study of Search Strategies for Mixed Integer Programming. *INFORMS Journal on Computing*, 11(2):173–187.

van Hemert, J. (2006). Evolving Combinatorial Problem Instances That Are Difficult to Solve. *Evolutionary Computation*, 14(4):433–462.

MATILDA


MATILDA
Melbourne Algorithm Test Instance Library with Data Analytics
Instance
Library
Test
Data
Algorithm
Analytics
Melbourne

MATILDA

Professor Kate Smith-Miles

Project Lead

Kate Smith-Miles is an ARC Australian Laureate Fellow, and Professor in the School of Mathematics and Statistics, The University of Melbourne. Her areas of interest are Algorithm Performance Analysis, Machine Learning and Pattern Recognition, Mathematical Modelling and Optimisation. Find out more >

Dr Mario Andres Munoz Acosta

Research Fellow

Mario Andres received his Ph.D. degree in Engineering from The University of Melbourne and currently working as a Research Fellow at the School of Mathematics and Statistics, The University of Melbourne. Find out more >

DR Asghar Moeini Korbekandi

Research Fellow

Asghar Moeini completed his Ph.D. in operations research at Flinders University of Adelaide and Currently, is a Research Fellow at the School of Mathematics and Statistics, The University of Melbourne. Find out more >

Dr Sevvandi Kandanaarachchi

Research Fellow

Sevvandi did her PhD in Geometric Analysis, which is a branch of Pure Mathematics at Monash. She is currently working on anomaly detection and related research areas. Find out more >

Dr Davaatseren Baatar

Research Fellow

Jeffrey Christiansen

Research Assistant

Tao Yan

Research Assistant

Dr Neelofar

Systems Specialist & Web Developer

MATILDA

Footprint Analysis

GRAPH COLORING PROBLEM

INSTANCE SPACE 2D COORDINATES = PROJECTION MATRIX X FEATURES VECTOR

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} 0.559 & -0.702 \\ 0.614 & -0.007 \\ 0.557 & 0.712 \end{bmatrix}^T \begin{bmatrix} Density \\ AlgConnectivity \\ Energy \end{bmatrix}$$

Select a graph to visualize:

SVM Selection