Turbo-charging the Feasibility Pump

Natashia L. Boland[§], Andrew C. Eberhard[†], ***Faramroze G. Engineer***[§], Matteo Fischetti[‡], and Martin W.P. Savelsbergh [¶]

[§]School of Mathematical and Physical Sciences, University of Newcastle
[†]School of Mathematical and Geospatial Sciences, RMIT
[‡]Dipartimento di Ingegneria dell'Informazione, University of Padova
[¶]Mathematics Informatics and Statistics, CSIRO

Integer Programming Down Under, 7th July 2011
Newcastle, NSW, Australia

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Outline

- The Feasibility Pump (FP)
- A Line Search procedure within FP
  - Efficient characterization of points of interest
  - Choosing start and end points
  - Extending the line search
  - Propagation within the line search
- Computational results

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

The Feasibility Pump (FP)

$x \in R^n$ and $J \subseteq \{1,\ldots,n\}$

$$
\begin{aligned}
\min \quad & cx \\
\text{s.t.} \quad & Ax \le b \\
& x_j \text{ is integer } \forall j \in J
\end{aligned}
$$

Feasibility Pump[1]: The general idea

Start with **LP feasible** ***x***

z $\leftarrow$ **closest integer** point to ***x*** $\leftarrow$ **rounding** ***x***, ***i***.e. [***x***]

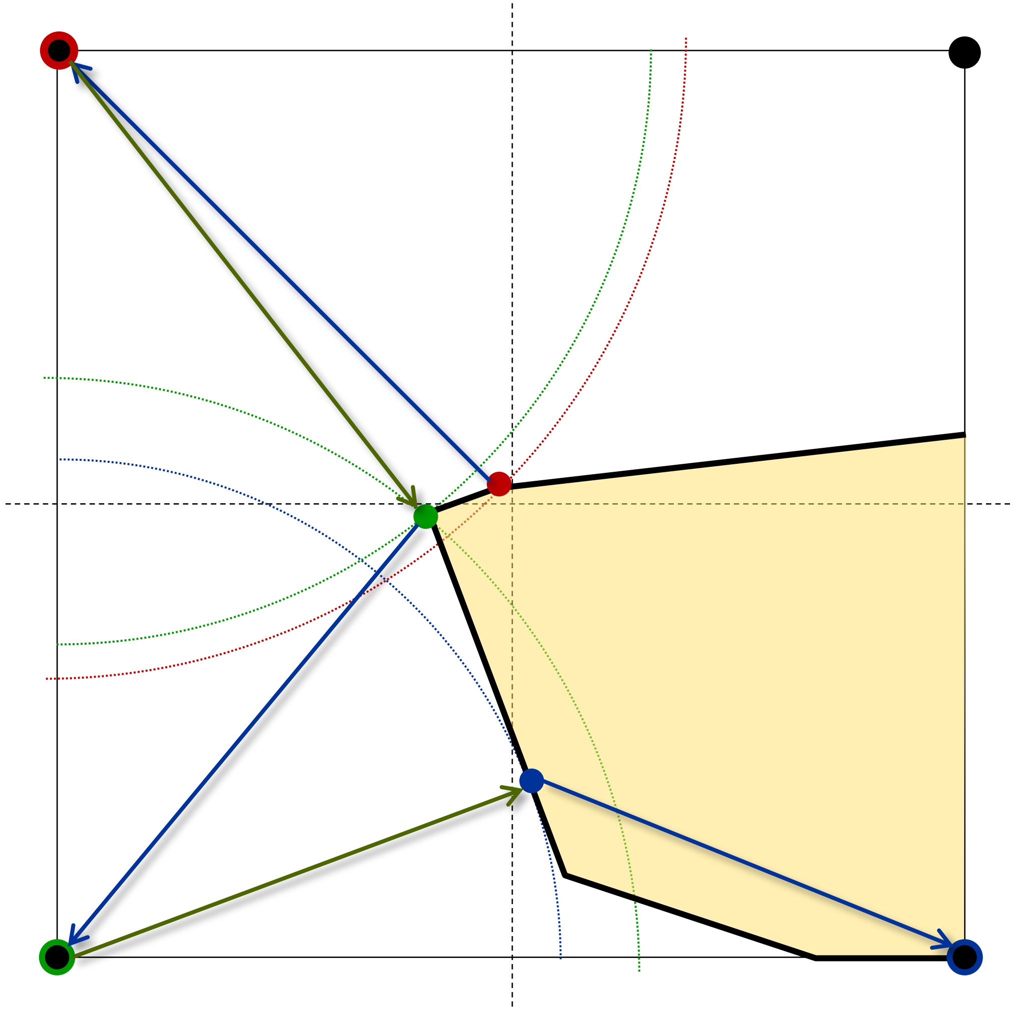
x $\leftarrow$ **closest LP feasible** point to ***z*** $\leftarrow$ **projecting** ***z*** onto LP feasible region

Repeat until ***z*** is **feasible**

[1] Fischetti, M., F. Glover, A. Lodi. 2005. The feasibility pump. *Mathematical Programming* **104** 91-104.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

The Feasibility Pump (FP)



Feasibility Pump: The general idea

Start with **LP feasible** ***x***
z ← **closest integer** point to ***x***
x ← **closest LP feasible** point to ***z***
Repeat until ***z*** is **feasible**

$\boldsymbol{x}_1$ $\boldsymbol{z}_1$

$\boldsymbol{x}_2$ $\boldsymbol{z}_2$

$\boldsymbol{x}_3$ $\boldsymbol{z}_3$

Two scenarios:

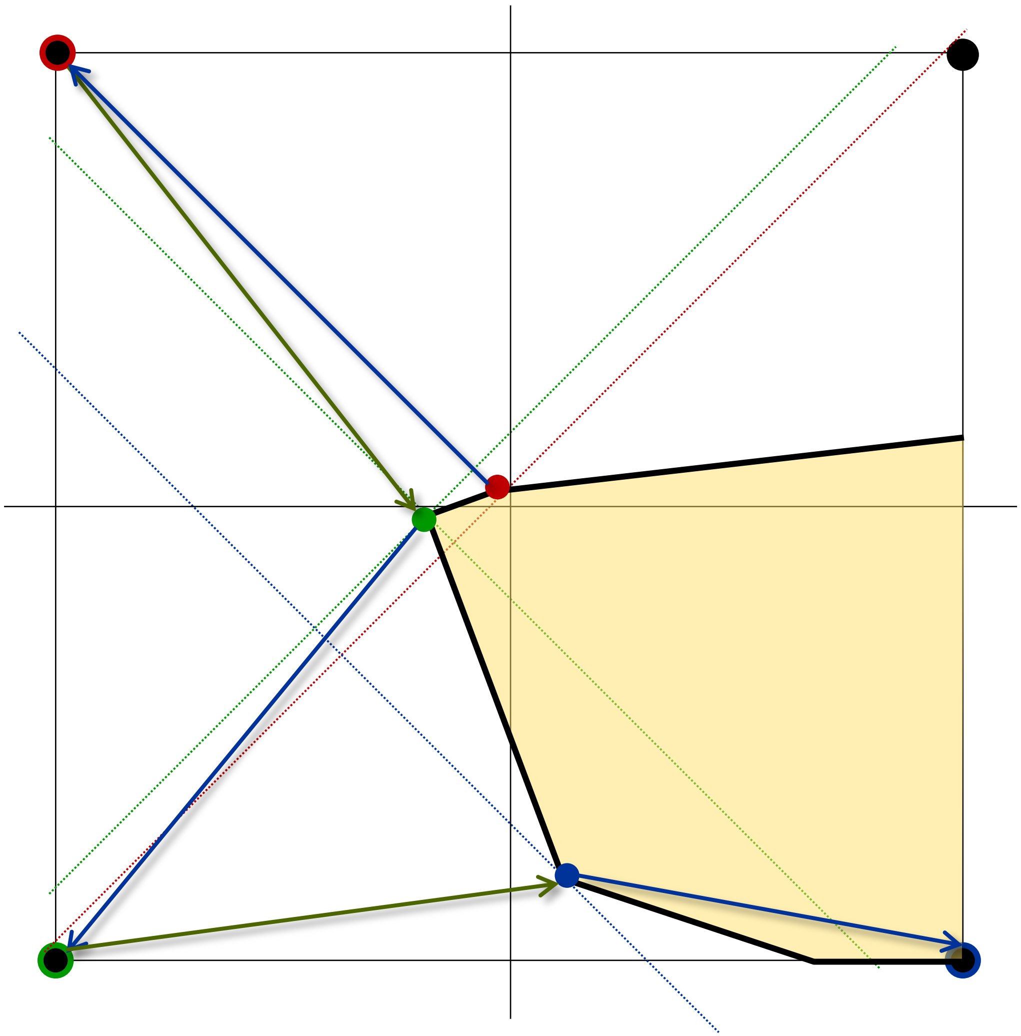
1. Feasible ***z***
2. Cycling (i.e., $[\boldsymbol{x}] = \boldsymbol{z}$ and $\mathtt{proj}_{LP}(\boldsymbol{z}) = \boldsymbol{x}$)

$$d(\boldsymbol{x}_1,\boldsymbol{z}_1) \le d(\boldsymbol{z}_1,\boldsymbol{x}_2) \le d(\boldsymbol{x}_2,\boldsymbol{z}_2) \le d(\boldsymbol{z}_2,\boldsymbol{x}_3) \le d(\boldsymbol{x}_3,\boldsymbol{z}_3)$$

$\boldsymbol{x} \longrightarrow$ integer $\qquad \boldsymbol{z} \longrightarrow$ feasible

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

The Feasibility Pump (FP)



Feasibility Pump: The general idea

Start with **LP feasible** ***x***
z ← **closest integer** point to ***x***
x ← **closest LP feasible** point to ***z***
Repeat until ***z*** is **feasible**

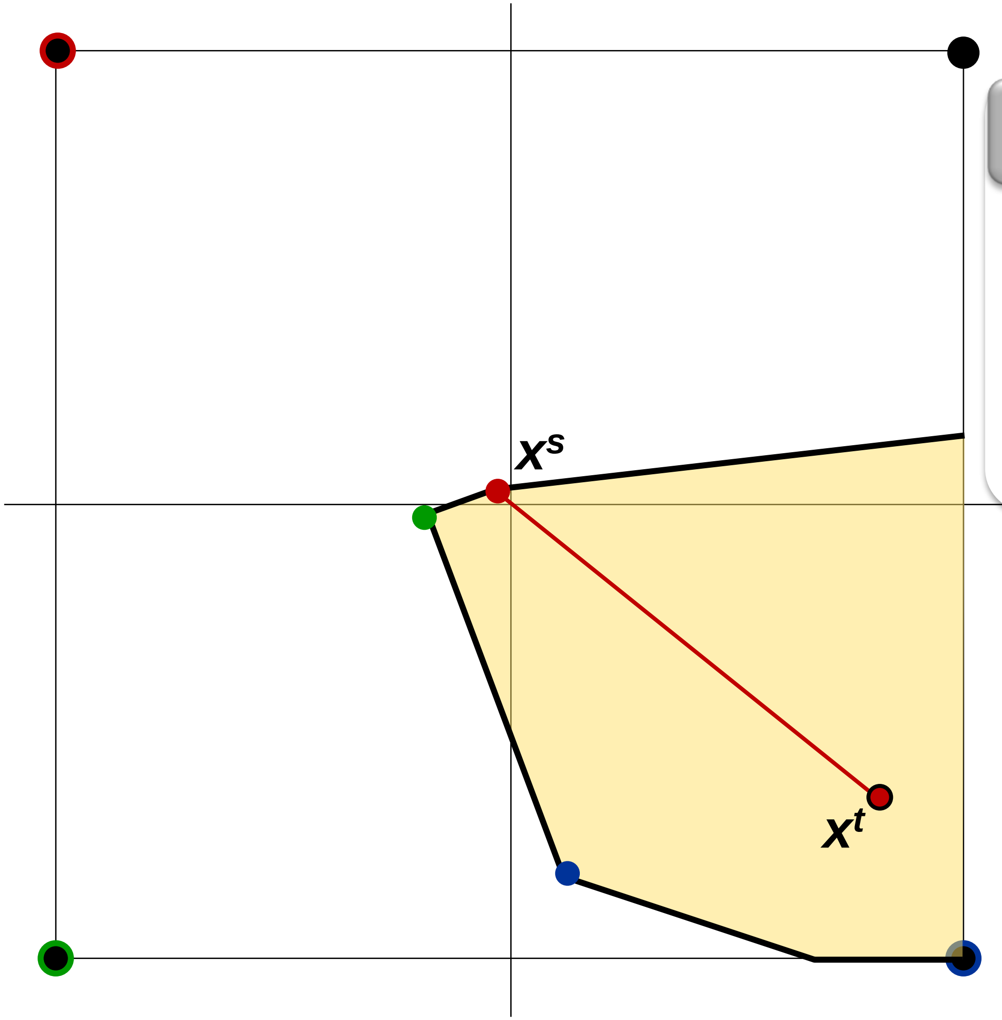
Spends most time in projection procedure:

- May overlook good integer solutions close to ***x***

Fix:

- Spend more time around FP iterates ***x*** to find feasible integer solutions rather than relying on naïve rounding
- Make search more balanced

Line Search within FP



A new substitute for rounding

For each FP iterate $\boldsymbol{x}$, round all points along a line segment passing through $\boldsymbol{x}$ and a point ***deep within the feasible region***.

Q. How to find suitable $\boldsymbol{x}^t$?

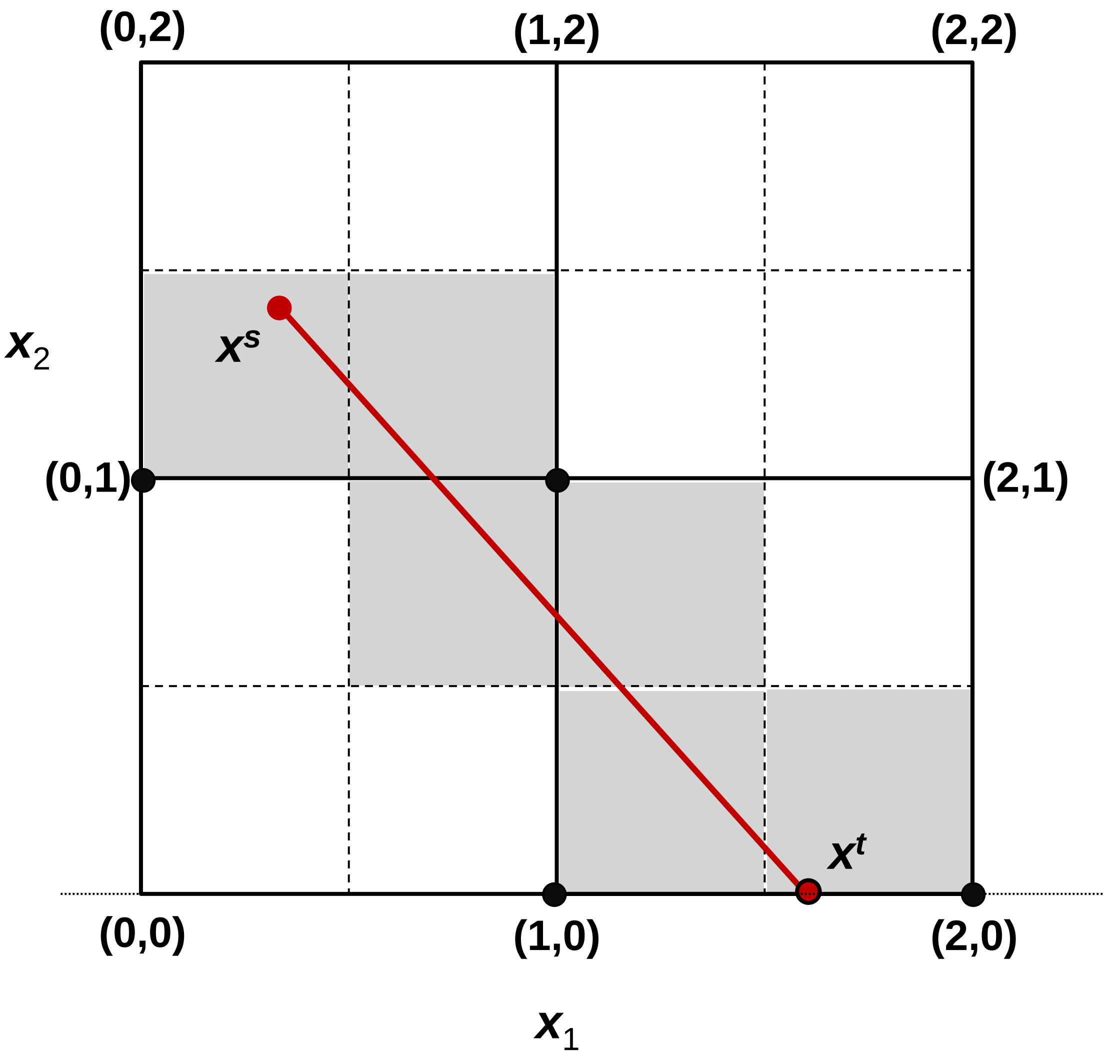
Q. How to find all rounded solutions along the shooting line ***efficiently***?

The chosen line segment is called the shooting line with starting point $\boldsymbol{x}^s$ and end point $\boldsymbol{x}^t$.

[2] Hillier, F.S. 1969. Efficient Heuristic Procedures for Integer Linear Programming with an Interior. *Operations Research* **17**(4) 600-637.

Line Search within FP

Finding all rounded points along the shooting line



x^s = (0.3, 1.4) and x^t = (1.6, 0)

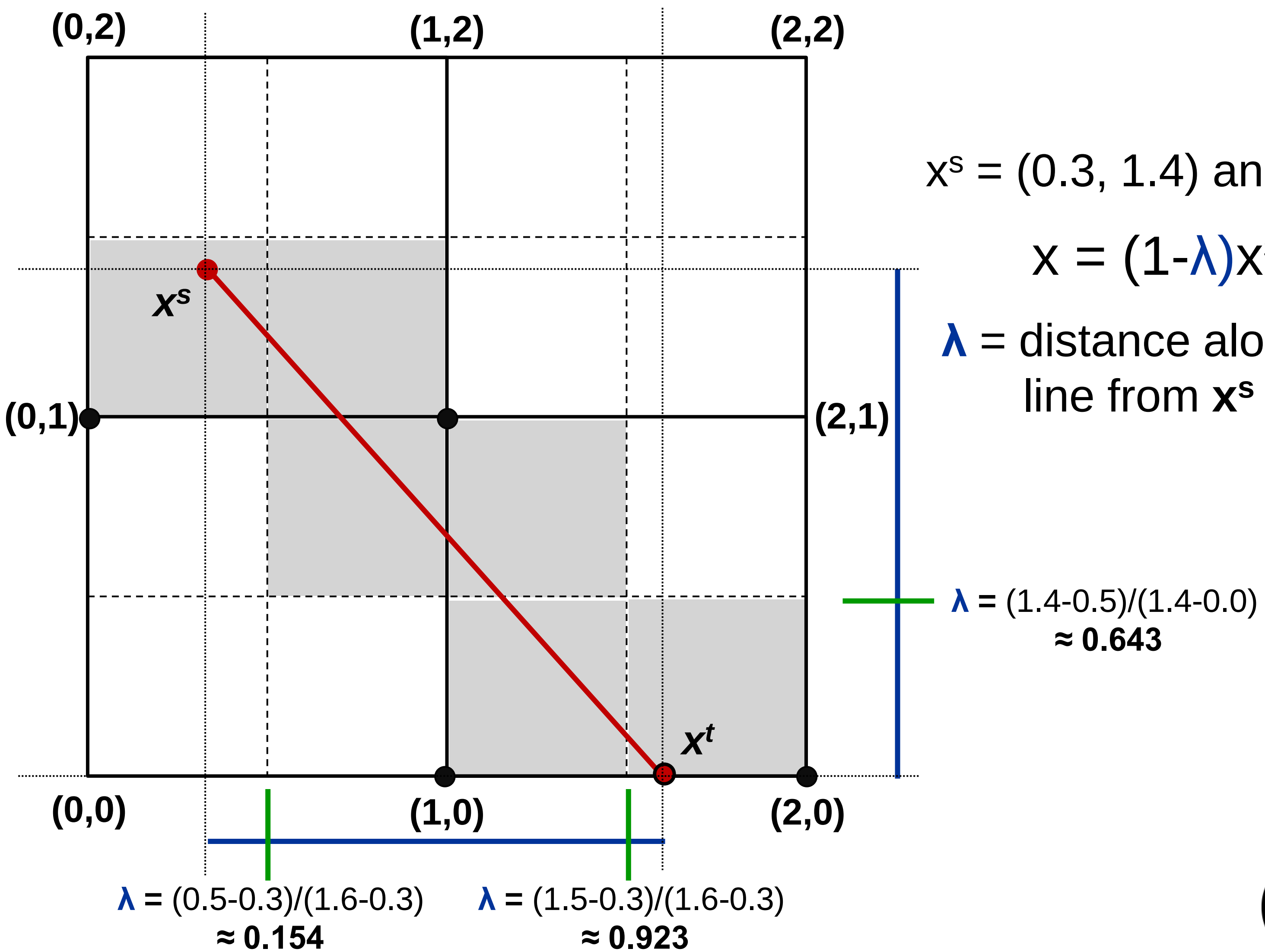
$$x = (1-\lambda)x^s + \lambda x^t$$

λ = distance along shooting line from $\mathbf{x^s}$

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Line Search within FP

Finding all rounded points along the shooting line



$x^s = (0.3, 1.4)$ and $x^t = (1.6, 0)$

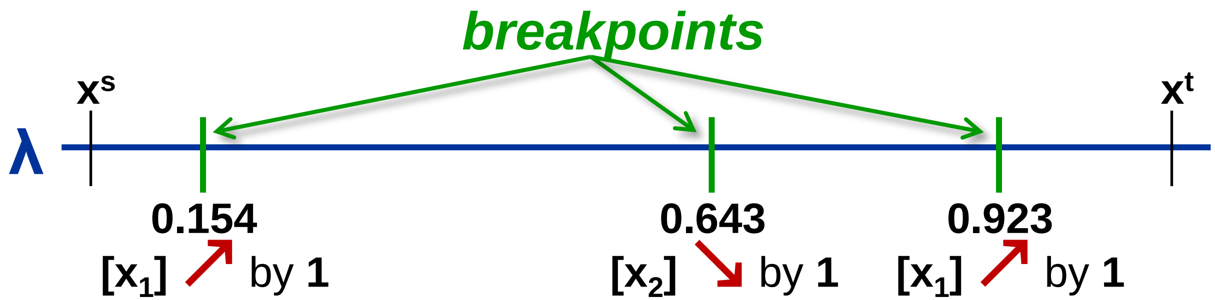
$$x = (1-\lambda)x^s + \lambda x^t$$

λ = distance along shooting line from $\mathbf{x^s}$

Line Search within FP

Finding all rounded points along the shooting line

$x^s = (0.3, 1.4)$ and $x^t = (1.6, 0)$



$$
\left.\begin{array}{ll}
[x^s] & = (0, 1) \text{ when } \lambda = 0.0 \\
[(1-\lambda)x^s + \lambda x^t] & = (1, 1) \text{ when } \lambda \approx 0.154 \\
 & = (1, 0) \text{ when } \lambda \approx 0.643 \\
 & = (2, 0) \text{ when } \lambda \approx 0.923 \\
[x^t] & = (2, 0) \text{ when } \lambda = 1.0
\end{array}\right\}
$$

All integer points obtained from rounding points along the shooting line

Observation

Rounded value along consecutive breakpoints is different by a unit value for one integer variable.

Line Search within FP

Characterizing the breakpoints

Set of all breakpoints for variable $j \in J$:

$$\Lambda(x_j^s, x_j^t) = \{0 \le \lambda \le 1 : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha\}$$

Computed in $O(|[x_j^s - x_j^t]|)$

Set of all tuples consisting of variable index, breakpoint, and indication of change in value ordered by distance from x^s:

$$\Psi(x^s, x^t) = \left\{ (j_k, \lambda_k, d_k) : \begin{array}{l} \text{(i) } \lambda_k \in \Lambda(x^s_{j_k}, x^t_{j_k}) \\ \text{(ii) } d_k = \begin{cases} +1, \text{ if } x^s_{j_k} < x^t_{j_k} \\ -1, \text{ otherwise, and} \end{cases} \\ \text{(iii) } \lambda_k \le \lambda_{k+1} \end{array} \right\}$$

Line Search within FP

The shooting procedure (pure integer case)

Compute $\Psi(\boldsymbol{x}^s, \boldsymbol{x}^t) = \{(\boldsymbol{j}_k,\boldsymbol{\lambda}_k,\boldsymbol{d}_k)_{k=1,..,K}\}$

$\boldsymbol{x} \leftarrow [\boldsymbol{x}^s]$

forall $k = 1,\ldots,K$

  $\boldsymbol{x}_{j_k} \leftarrow \boldsymbol{x}_{j_k} + \boldsymbol{d}_{j_k}$

  if $\boldsymbol{x}$ is an *incumbent solution*

    record $\boldsymbol{x}$

  end

end

No. of operations is $O(\Sigma_j|[x^s_j-x^t_j]|)$ = total number of breakpoints

Vs

At least $O(n(\Sigma_j|[x^s_j-x^t_j]|))$ if doing rounding to find all integer solutions along shooting line

Line Search within FP

The shooting procedure (pure integer case)

Compute $\Psi(\boldsymbol{x}^s, \boldsymbol{x}^t) = \{(\boldsymbol{j}_k, \boldsymbol{\lambda}_k, \boldsymbol{d}_k)_{k=1,..,K}\}$
$\boldsymbol{x} \leftarrow [\boldsymbol{x}^s]$
forall $k = 1,\dots,K$
  $\boldsymbol{x}_{jk} \leftarrow \boldsymbol{x}_{jk} + \boldsymbol{d}_{jk}$
  if $\boldsymbol{x}$ is an *incumbent solution*
    record $\boldsymbol{x}$
  end
end

Dealing with continuous variables

Option 1: For continuous variables simply use values at the breakpoints. Efficient but may overlook feasible solutions.

Option 2: Fix integer values and solve LP for continuous variables after each change in some integer value. Less efficient but can take advantage of ***warm starts***.

$$\begin{aligned} \min \quad & \sum_{j \notin J} c_j x_j \\ \text{s.t.} \quad & \sum_{j \notin J} A_j x_j \le b - \sum_{j \in J} A_j \bar{x}_j \end{aligned}$$

Fixed integer values

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Line Search within FP

Finding suitable end points $\boldsymbol{x}^t$

Ideally, $\boldsymbol{x}^t$ would be in the convex hull of feasible solutions. In practice, we settle for a point $\boldsymbol{x}^t$ that leads towards LP feasibility.

Option 1: $\boldsymbol{x}^t$ = ***Analytic Centre*** of LP region:

$$\begin{array}{ll} \max & \sum_{i \in Q} \ln(b_i - a_i^T x) \\ \text{s.t.} & Ax \leq b \end{array}$$

where $Q = \{i : \exists\, x \text{ s.t. } a_i^T x < b_i\}$, i.e. constraints defining relative interior

Line Search within FP

Finding suitable end points $\boldsymbol{x}^t$

Ideally, $\boldsymbol{x}^t$ would be in the convex hull of feasible solutions. In practice, we settle for a point $\boldsymbol{x}^t$ that leads towards LP feasibility.

Option 2: $\boldsymbol{x}^t$ = analytic centre of LP region with cuts that eliminate the infeasible rounded FP iterates found so far. Same as finding analytic centre but giving more weight to constraints violated by rounded solutions.

If $[\boldsymbol{x}^s]$ violates constraint i, i.e.,

$$\sum_{j\in J} a^i_j [x^s_j] + \sum_{j\notin J} a^i_j x^s_j > b_i$$

then add cut of the form (in terms of integer variables):

$$\sum_{j\in J} a^i_j x_j \le b_i - \sum_{j\notin J} a^i_j x^s_j$$

[3] Naoum-Sawaya, J., S. Elhedhli. 2011. An interior point cutting plane heuristic for mixed integer programming. *Computers and Operations Research* **38**(9) 1335-1341.

Line Search within FP

Finding suitable end points $\boldsymbol{x}^t$

Ideally, $\boldsymbol{x}^t$ would be in the convex hull of feasible solutions. In practice, we settle for a point $\boldsymbol{x}^t$ that leads towards LP feasibility.

Option 3: $\boldsymbol{x}^t = \boldsymbol{x}^s + \boldsymbol{d}$ where $\boldsymbol{d}$ is a conic combination of constraints violated by $[\boldsymbol{x}^s]$ weighted based on relative degree of violation, i.e.,

$$d = \sum_{i \in Q} \left(\frac{b_i - {a^i}^T [x^s]}{\|a^i\|_2} \right) a^i$$

where $Q = \{i : a^i[x^s] > b_i\}$.

Line Search within FP

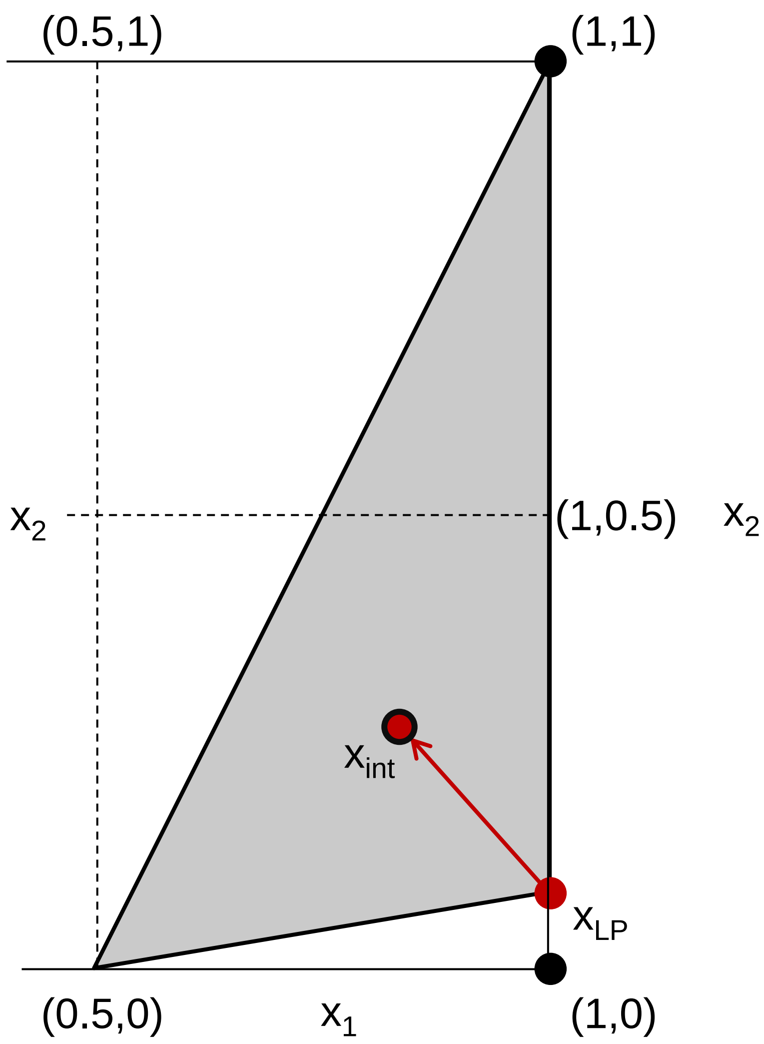
Finding suitable end points x^t

Description	Advantages	Disadvantages
Analytic Centre (AC)	• Compute x^t **only once** • Guarantees x^t is **LP feasible**	• **Expensive** (but can use path-following trajectory) • Does not consider **integer violation** of $[x^s]$
Analytic Centre Cutting Plane Method (ACCPM)	• Guarantees x^t is **LP feasible** • Greater **emphasis** to constraints violated by $[x^s]$	• Compute a new x^t for **each line search** • **Expensive** (but can use path-following trajectory)
Conic Direction (D)	• **Inexpensive** to compute • Greater **emphasis** to constraints violated by $[x^s]$	• Compute a new x^t for **each line search** • Does not guarantee x^t is **LP feasible**

Q. Is feasibility of x^t important?

Line Search within FP

Q. Is feasibility of $\boldsymbol{x^t}$ important?

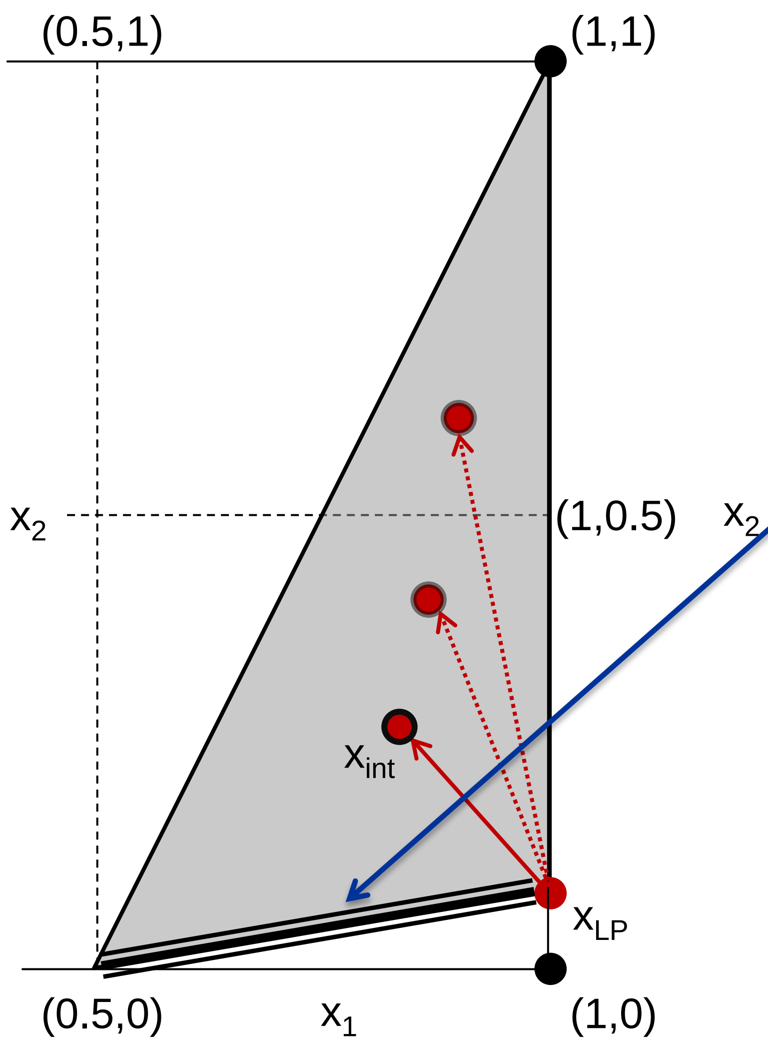


$$\begin{aligned} \min \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & 2x_1 - x_2 \geq 1 \\ & 2x_1 - 10x_2 \leq 1 \\ & x_1, x_2 \in \{0,1\} \end{aligned}$$

$x_{LP} = (1.0,0.1)$
$x_{int} = (0.8,0.25)$

Line Search within FP

Q. Is feasibility of $\boldsymbol{x}^t$ important?



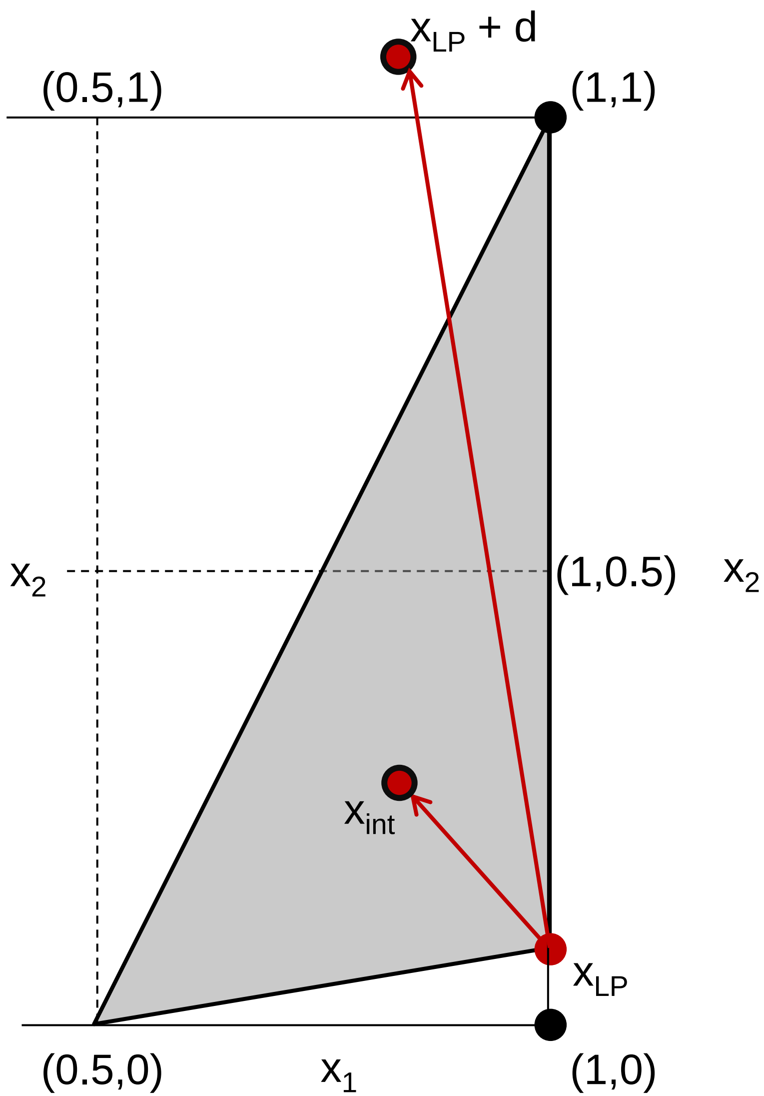
$$
\begin{array}{ll}
\min & x_1 + 2x_2 \\
\text{s.t.} & 2x_1 - x_2 \geq 1 \\
 & \alpha(2x_1 - 10x_2 \leq 1) \\
 & x_1, x_2 \in \{0,1\}
\end{array}
$$

$x_{LP} = (1.0, 0.1)$

$x_{int} = (0.8, 0.25)$

Line Search within FP

Q. Is feasibility of $\boldsymbol{x^t}$ important?



$$\begin{aligned} \min \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & 2x_1 - x_2 \geq 1 \\ & 2x_1 - 10x_2 \leq 1 \\ & x_1, x_2 \in \{0,1\} \end{aligned}$$

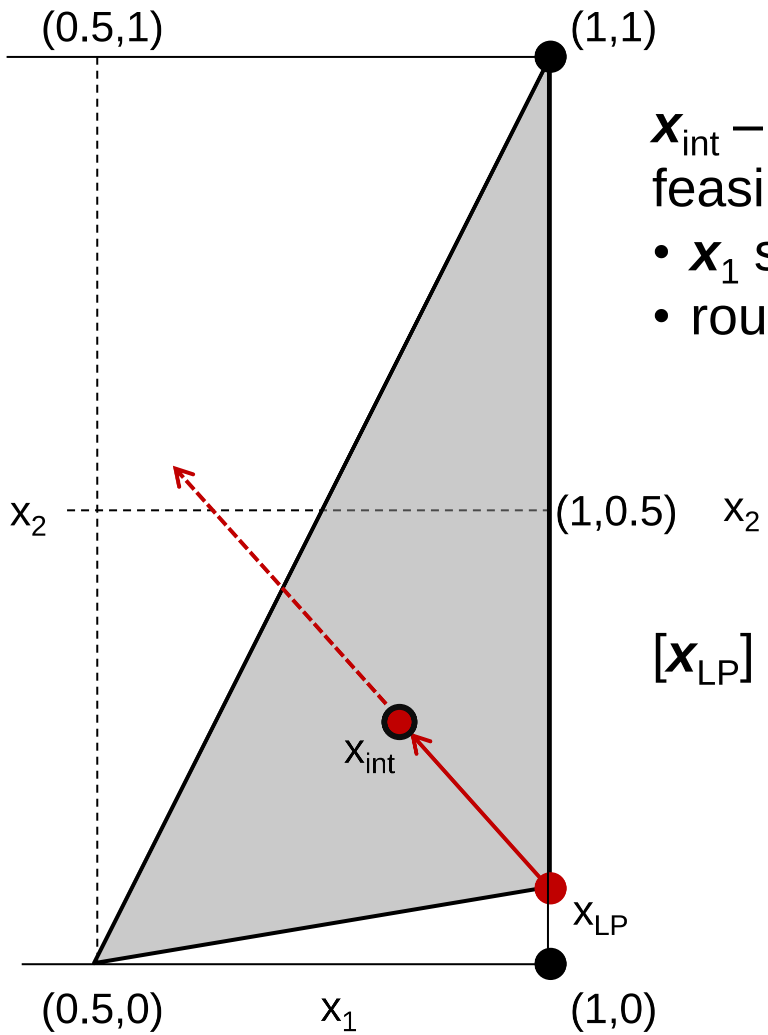
$x_{LP} = (1.0,0.1)$
$x_{int} = (0.8,0.25)$

$$d = \frac{1}{\sqrt{104}} \begin{bmatrix} -2 \\ 10 \end{bmatrix}$$

$x_{LP} + d \approx (0.804,\ 1.080)$

Line Search within FP

Q. Is feasibility of $\boldsymbol{x}^t$ important?



$\boldsymbol{x}_{\text{int}} - \boldsymbol{x}_{\text{LP}}$ is a ***direction*** towards LP feasibility. Hence, to ***recover feasibility***:

- $\boldsymbol{x}_1$ should decrease and $\boldsymbol{x}_2$ should increase
- rounded value of $\boldsymbol{x}_2$ changes before $\boldsymbol{x}_1$

$x_{LP} = (1.0,0.1)$
$x_{int} = (0.8,0.25)$

$[\boldsymbol{x}_{LP}]$ = (1,0)
= (1,1) after incrementing $\boldsymbol{x}_2$ ← **feasible**
= (0,1) after decrementing $\boldsymbol{x}_1$

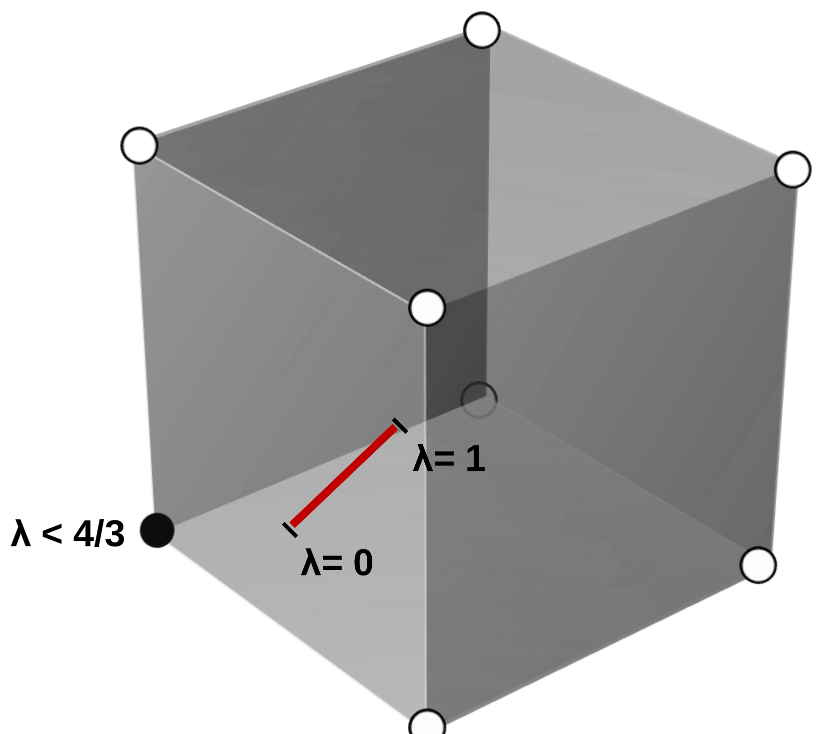
Line Search within FP

Extending the Line Search

Set of all breakpoints for variable $j \in J$:

Convex combination

$$\Lambda(x_j^s, x_j^t) = \{\boxed{0 \le \lambda \le 1} : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha\}$$



Extending the Line Search

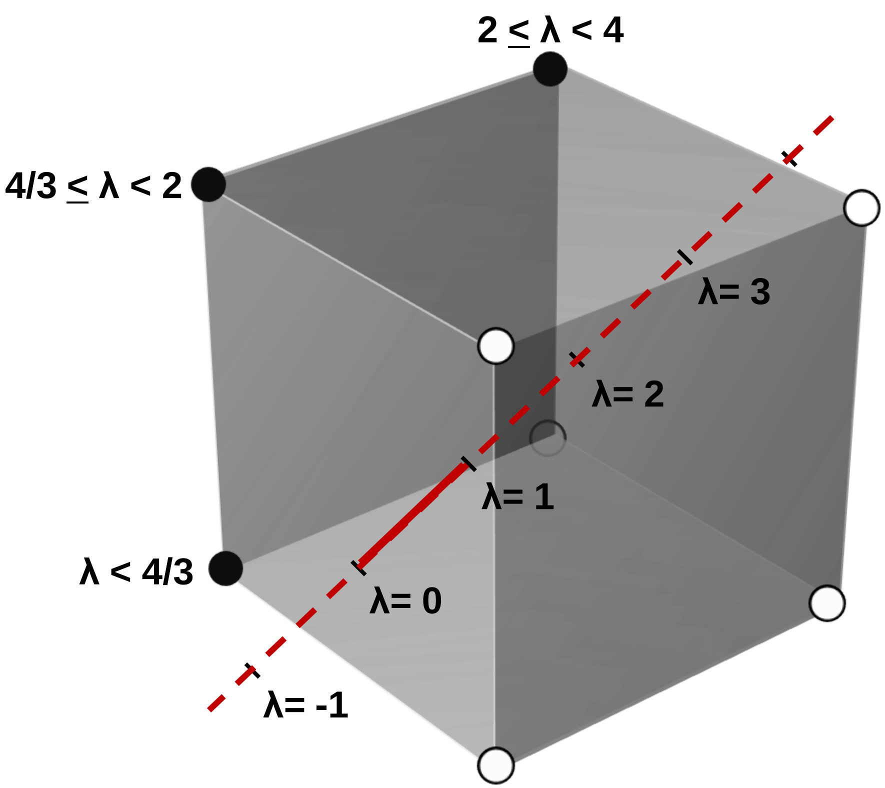
Set of all breakpoints for variable $j \in J$:

Convex combination

$$\Lambda(x_j^s, x_j^t) = \{\boxed{0 \le \lambda \le 1} : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha\}$$

Linear combination within variable bounds

$$\overline{\Lambda}(x_j^s, x_j^t) = \{\lambda : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha \text{ s.t. } \boxed{l_j \le \alpha \le u_j}\}$$



Line Search within FP

Extending the Line Search

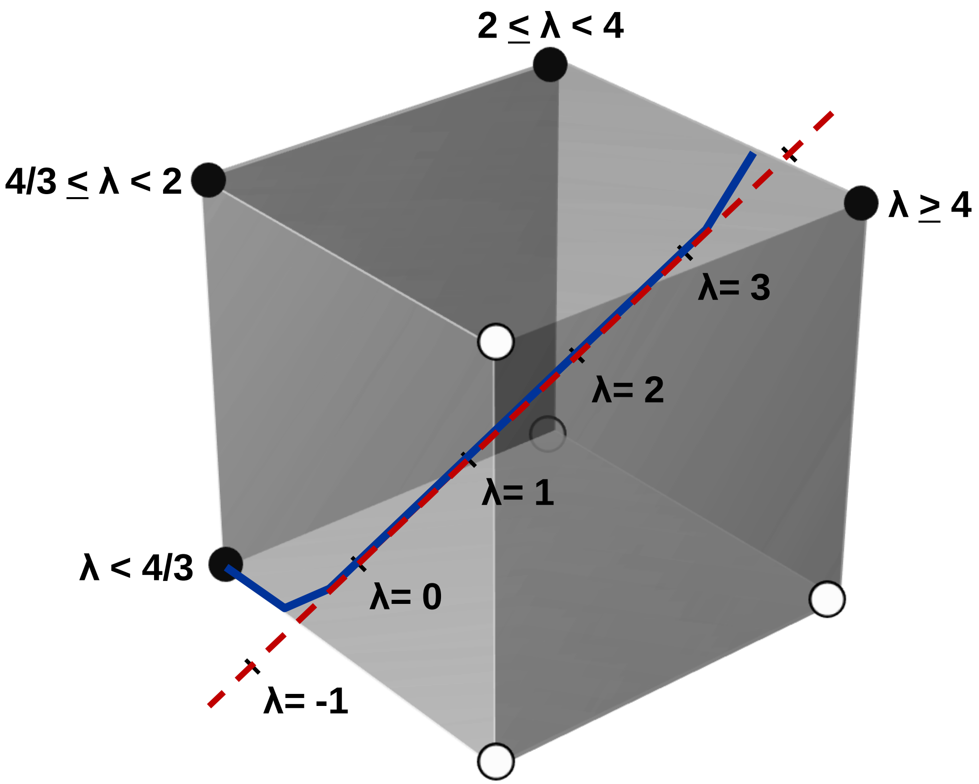
Set of all breakpoints for variable $j \in J$:

Convex combination

$$\Lambda(x_j^s, x_j^t) = \{\boxed{0 \le \lambda \le 1} : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha\}$$

Linear combination within variable bounds

$$\overline{\Lambda}(x_j^s, x_j^t) = \{\lambda : (1-\lambda)x_j^s + \lambda x_j^t = \alpha + 0.5 \text{ for some integer } \alpha \text{ s.t. } \boxed{l_j \le \alpha \le u_j}\}$$

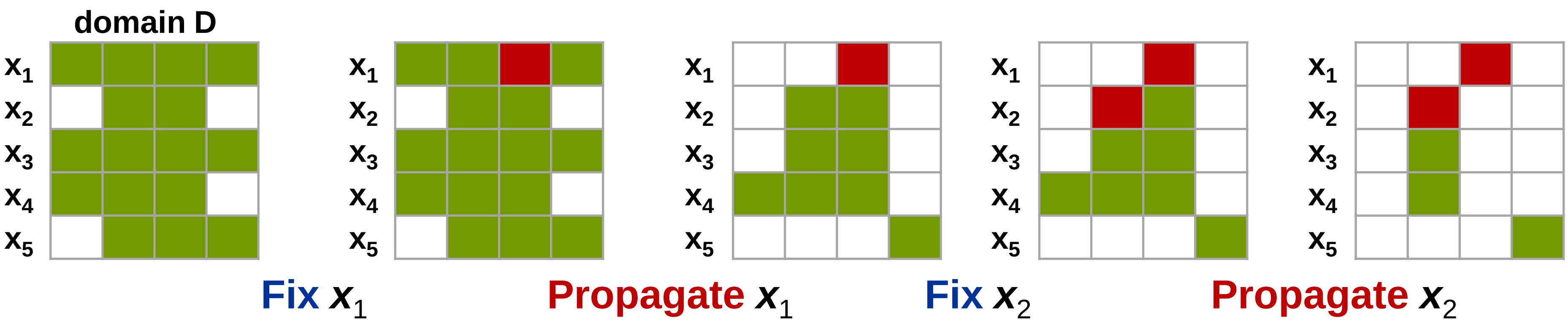


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Line Search within FP

Propagation in MIP[4,5]

When fixing a variable, the feasible domain of other variables can be shrunk (or the problem can be declared infeasible) based on logic propagation.



- Knapsack propagation
- Linear propagation
- Cover propagation
- Partition propagation
- Cardinality propagation
- …

[4] Savelsbergh, M. W. P. 1994. Preprocessing and probing techniques for mixed integer programming problems. *Informs Journal on Computing* **6**(4) 445-454.
[5] Achterberg, T. 2007. Constraint integer programming. Ph.D. thesis, TU Berlin.

Line Search within FP

$$\mathbf{D} = (\boldsymbol{l}',\boldsymbol{u}') \subseteq (\boldsymbol{l}, \boldsymbol{u})$$

$$[\boldsymbol{x}_j]^{\mathbf{D}} = \text{argmin}_{\mathbf{z}} \{|\boldsymbol{z} - \boldsymbol{x}_j| : \boldsymbol{l}'_{\text{j}} \le \boldsymbol{z}_j \le \boldsymbol{u}'_{\text{j}} \text{ and } \boldsymbol{z} \text{ integer}\}$$

z is the closest integer value to $\boldsymbol{x}_j$ within the allowable domain

Propagation in rounding[6]

propRound(x)

$\mathbf{D} = (\boldsymbol{l}',\boldsymbol{u}') \leftarrow (\boldsymbol{l}, \boldsymbol{u})$
rank($\boldsymbol{x}$) = $\{j_1, j_2, \ldots, j_{|J|}\}$
forall $k = 1,\ldots,|J|$
  $\boldsymbol{x}_{jk} \leftarrow [\boldsymbol{x}_{jk}]^{D}$
  $\mathbf{D} \leftarrow$ **propagate**($\boldsymbol{x}_{jk}$)
end
if ***x*** is an *incumbent solution*
  record ***x***
end

[6] Fischetti, M., D. Salvagnin. 2009. Feasibility pump 2.0. *Mathematical Programming Computation* **1** 201-222.

Line Search within FP

$$\mathbf{D} = (\boldsymbol{l}',\boldsymbol{u}') \subseteq (\boldsymbol{l}, \boldsymbol{u})$$

$$[\boldsymbol{x}_j]^{\mathrm{D}} = \mathrm{argmin}_{\boldsymbol{z}} \{|\boldsymbol{z} - \boldsymbol{x}_j| : \boldsymbol{l}'_{\mathrm{j}} \leq \boldsymbol{z}_j \leq \boldsymbol{u}'_{\mathrm{j}} \text{ and } \boldsymbol{z} \text{ integer}\}$$

Propagation in rounding

```
propRound(x)
------------------------------
D = (l',u') ← (l, u)
rank(x) = {j_1, j_2, …, j_|J|}
forall k = 1,…,|J|
|   x_jk ← [x_jk]^D
|   D ← propagate(x_jk)
end
if x is an incumbent solution
      record x
end
```

Propagation in line search

```
Compute Ψ(x^s, x^t) = {(j_k,λ_k,d_k)_k=1,..,K}
x ← propRound(x^s)
D = (l',u') ← (l, u)
forall k = 1,…,K
|   x_jk ← [x_jk + d_jk]^D
|   if x is an incumbent solution
|   |  record x
|   end
|   if no more changes for j_k
|   |  D ← propagate(x_jk)
|   end
end
```

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Computational Results

Test-bed: 1,200+ MIP instances from **`miplib3`**[7], **`miplib2010`**[8], **`cor@l`**[9], and **`orlib`**[10].

Experimental setup:

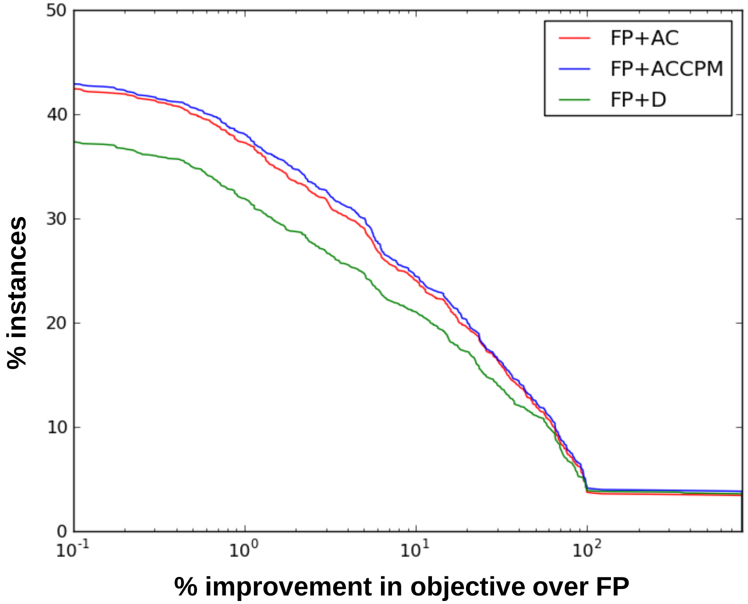
1. Run with ***default FP*** and in addition to simple rounding, use each new FP iterate as the starting point of the line search. Three ***line searches*** are conducted using each of the three proposed schemes for the endpoint.
2. Same as 1 except line search is ***extended*** for $-1 \leq \lambda \leq 2$ and ***projected*** back onto hypercube defined by variable bounds.
3. Same as 2 except ***propagation*** techniques used with rounding and line search.

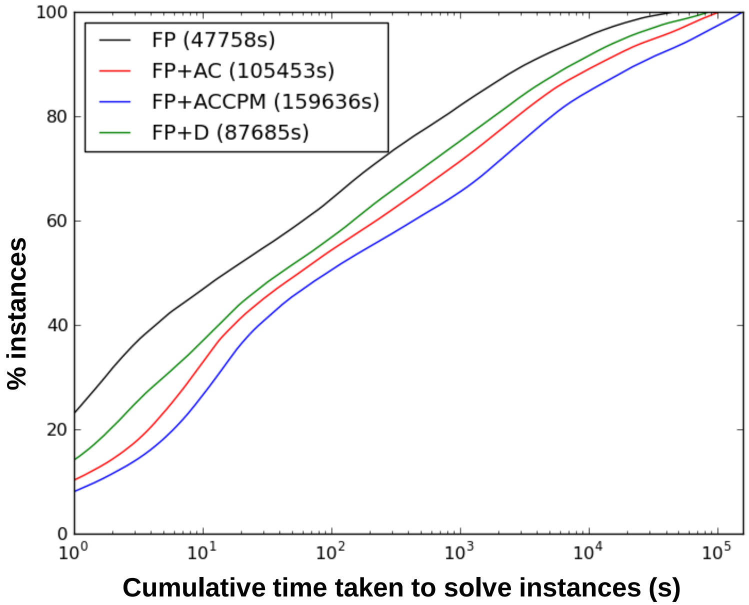
---

[7] http://miplib.zib.de/miplib2003/
[8] http://miplib.zib.de/miplib2010/
[9] http://coral.ie.lehigh.edu/~mip-instances/instances/
[10] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mipinfo.html

Computational Results

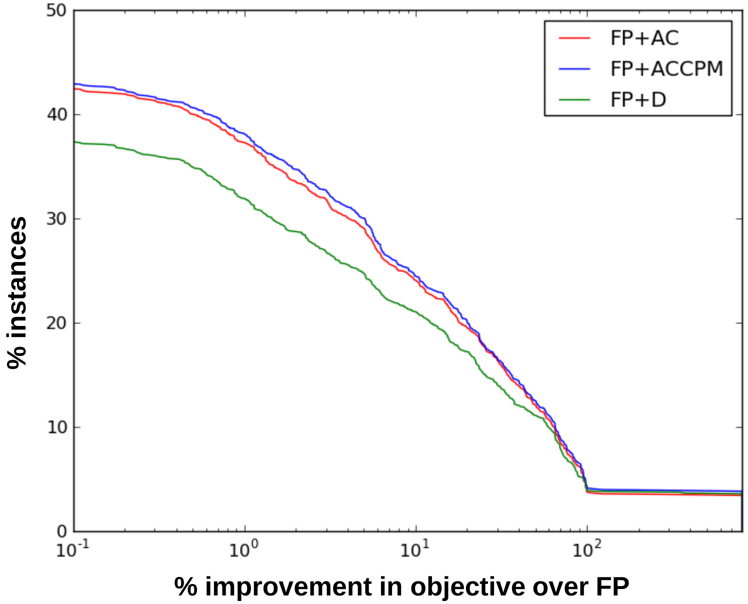
Experiment 1

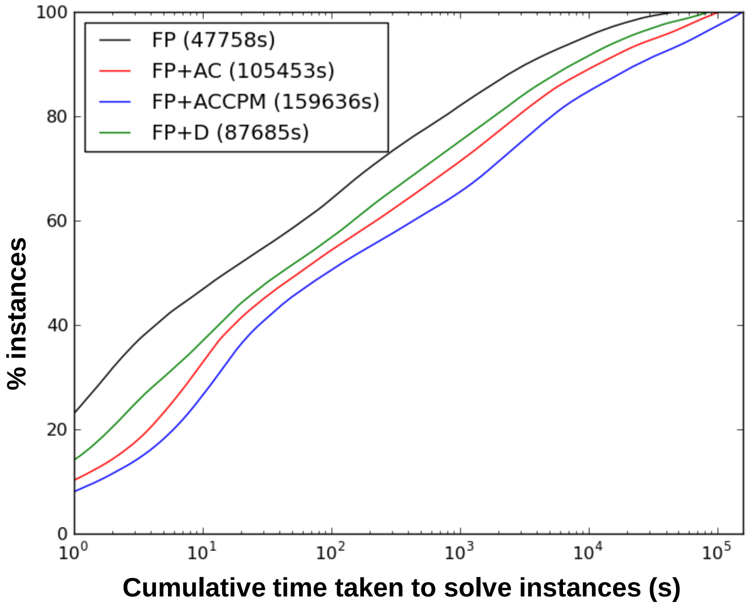




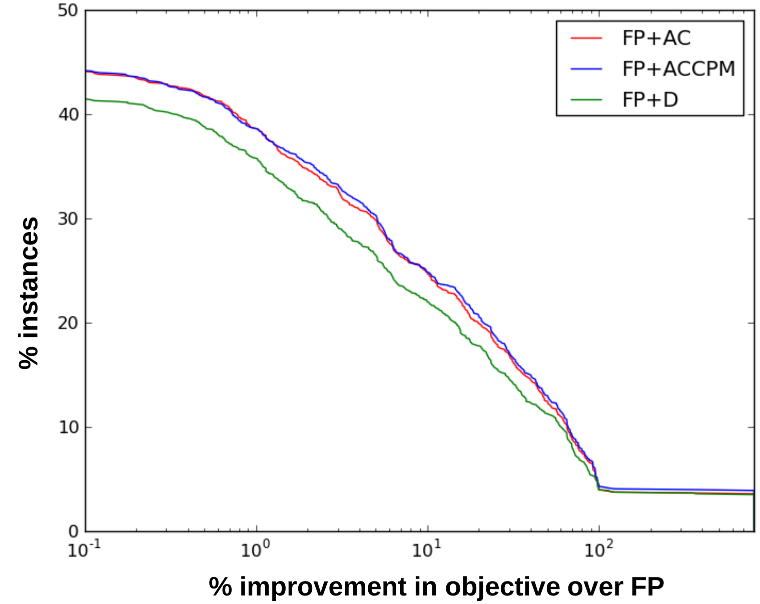
Computational Results

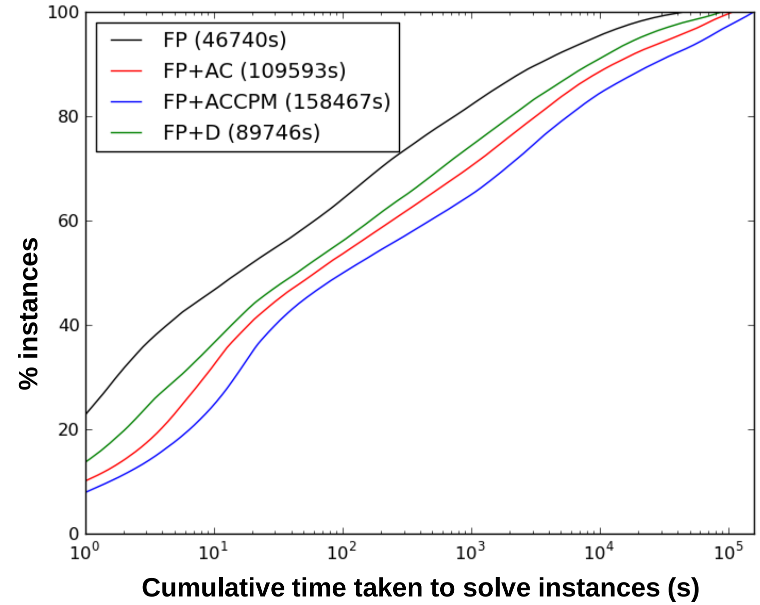
Experiment 1





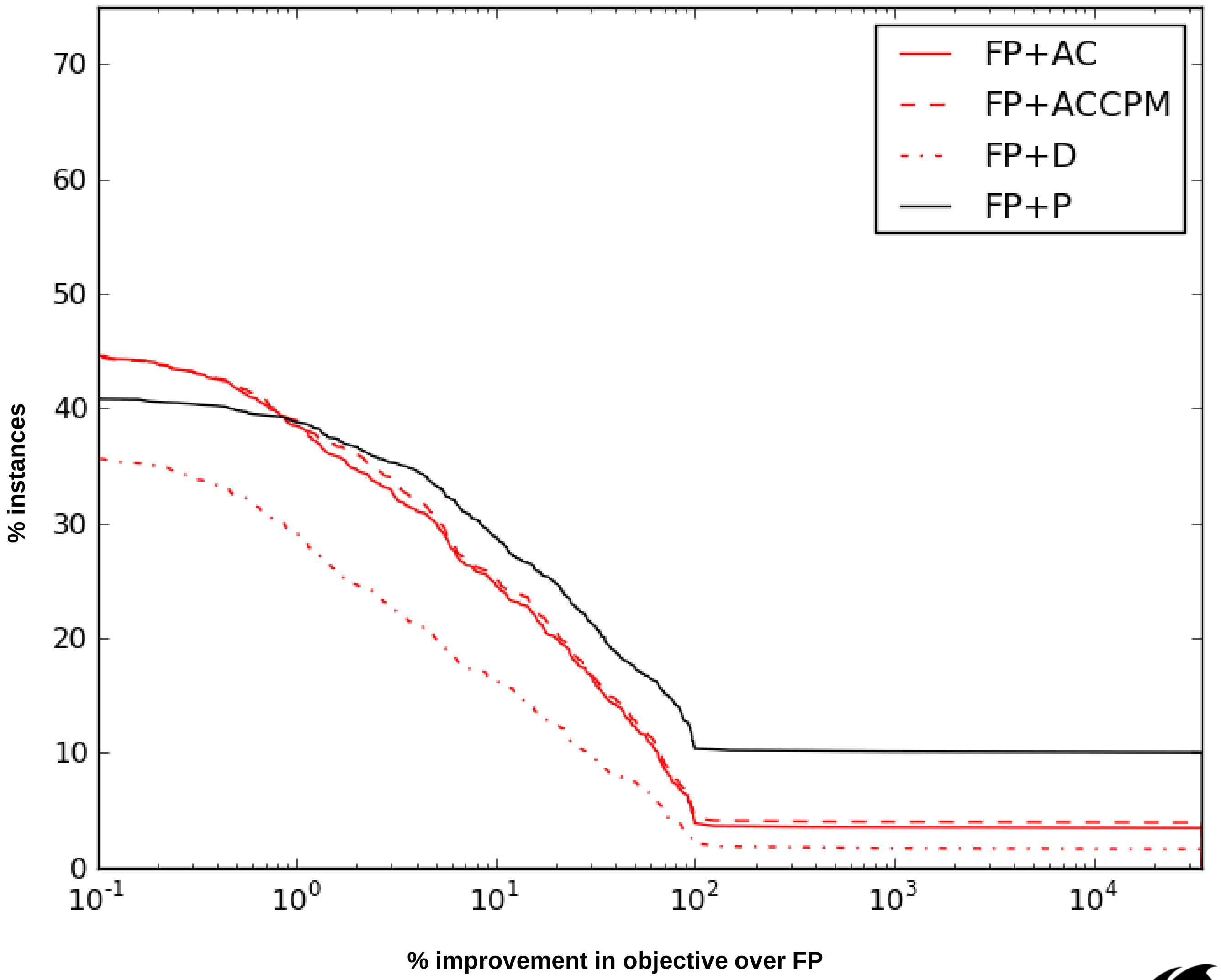
Experiment 2





Computational Results

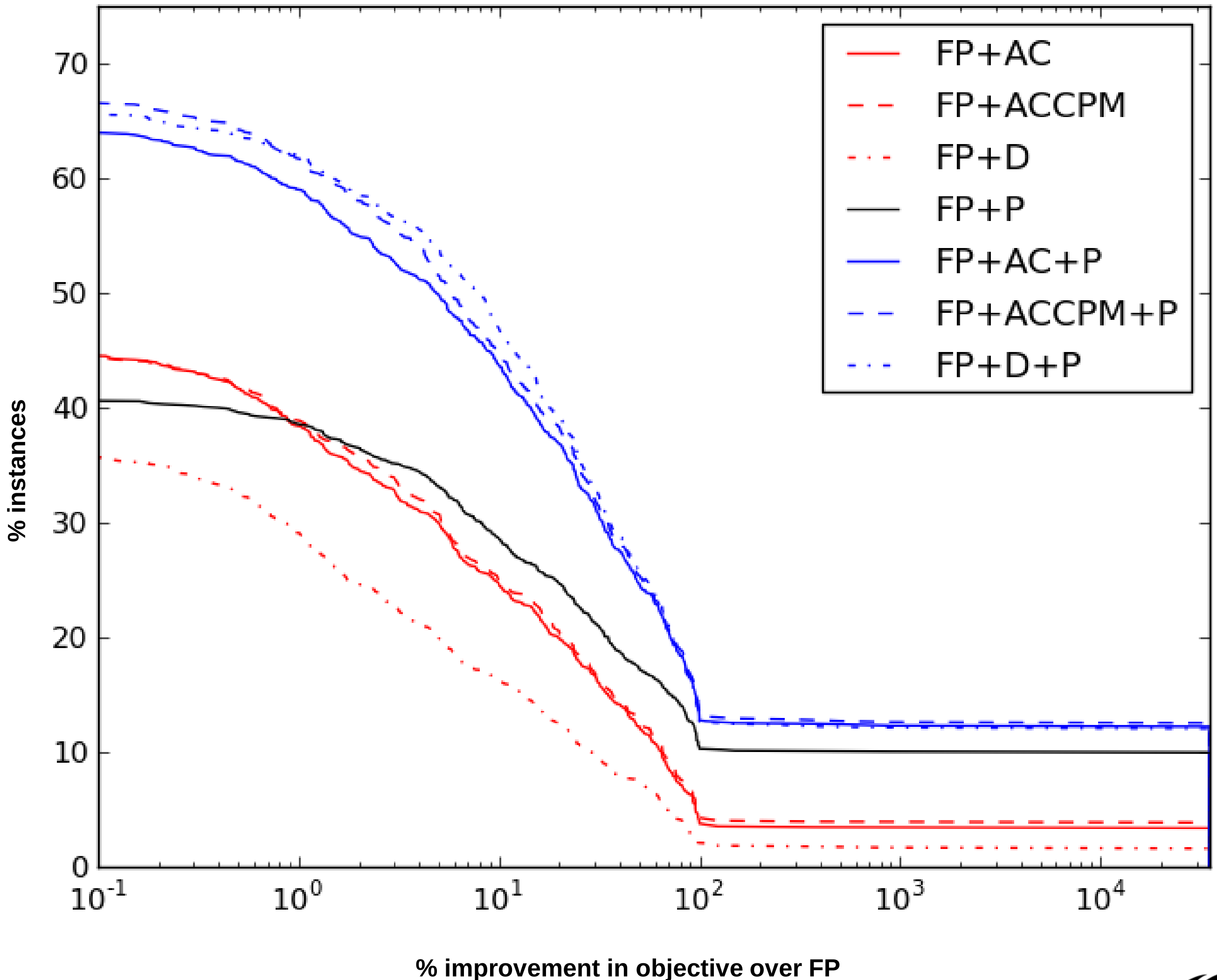
FP Vs FP+P



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Computational Results

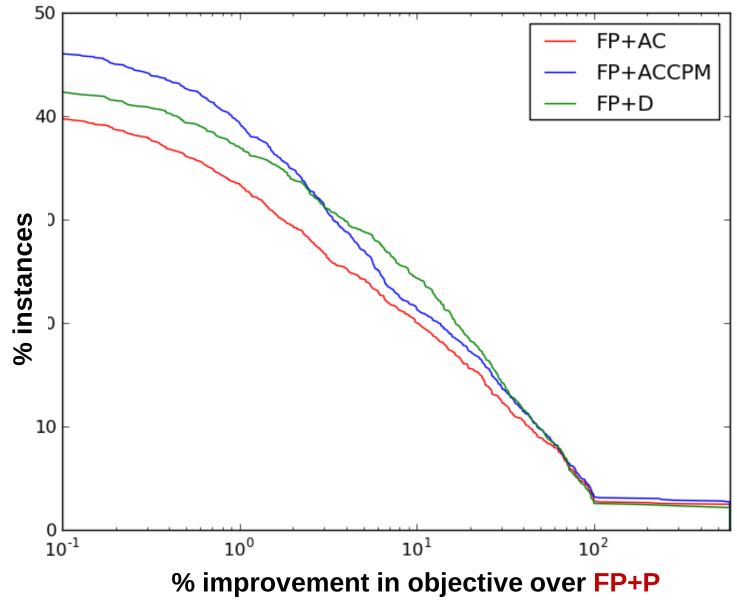
FP Vs FP+P

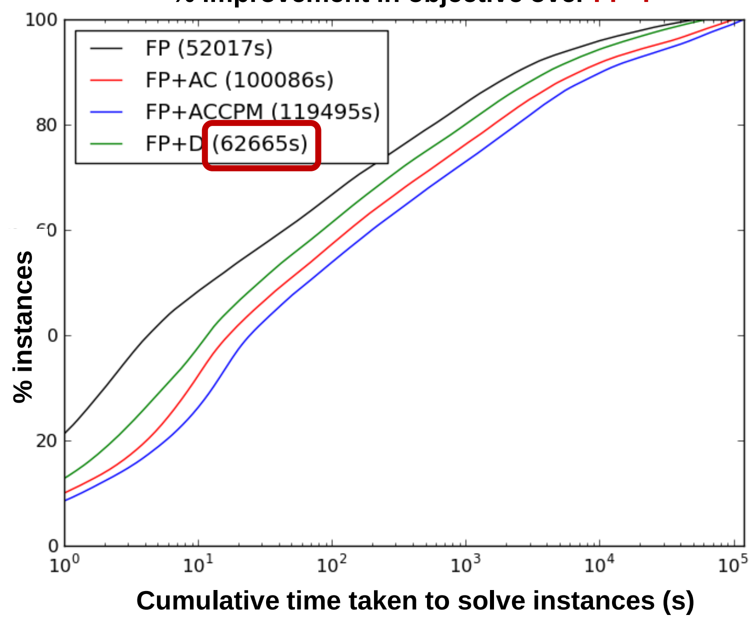


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Computational Results

Experiment 3





THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Conclusions

- FP can be improved by investing effort into the ***transformation procedure*** and making the search more ***balanced***
- “Exotic” end points are ***unwarranted***
- Line search improves upon FP in ***40+%*** of cases with ***negligible*** deterioration in time
- Line search is ***complementary*** to propagation