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 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 Feasibility Pump (FP)

\[ x \in \mathbb{R}^n \text{ and } J \subseteq \{1, \ldots, n\} \]

\[
\begin{align*}
\min & \quad cx \\
\text{s.t.} & \quad Ax \leq b \\
& \quad x_j \text{ is integer } \forall j \in J
\end{align*}
\]

Feasibility Pump\(^{[1]}\): The general idea

Start with \textbf{LP feasible} \( x \)

\[ z \leftarrow \text{closest integer point to } x \]

\[ x \leftarrow \text{closest LP feasible point to } z \]

Repeat until \( z \) is \textbf{feasible}

The Feasibility Pump (FP)

Start with LP feasible \( x \)

\[ z \leftarrow \text{closest integer point to } x \]

\[ x \leftarrow \text{closest LP feasible point to } z \]

Repeat until \( z \) is feasible

**Feasibility Pump: The general idea**

Two scenarios:

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

\[ d(x_1,z_1) \leq d(z_1,x_2) \leq d(x_2,z_2) \leq d(z_2,x_3) \leq d(x_3,z_3) \]

\( x \rightarrow \text{integer} \quad z \rightarrow \text{feasible} \)
The Feasibility Pump (FP)

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

Feasibility Pump: The general idea

Start with LP feasible $x$
$z \leftarrow$ closest integer point to $x$
$x \leftarrow$ closest LP feasible point to $z$
Repeat until $z$ is feasible
The chosen line segment is called the shooting line with starting point \( x^s \) and end point \( x^t \).

Q. How to find suitable \( x^t \)?

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

A new substitute for rounding

For each FP iterate \( x \), round all points along a line segment passing through \( x \) and a point deep within the feasible region.

Finding all rounded points along the shooting line

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

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

$\lambda = \text{distance along shooting line from } x^s$
Line Search within FP

Finding all rounded points along the shooting line

\[ x^s = (0.3, 1.4) \text{ and } x^t = (1.6, 0) \]

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

\[ \lambda = \text{distance along shooting line from } x^s \]

\[ \lambda = \frac{(1.4 - 0.5)}{(1.4 - 0.0)} \approx 0.643 \]
Line Search within FP

Finding all rounded points along the shooting line

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

\[ (1 - \lambda)x^s + \lambda x^t \]

- \( (0, 1) \) when \( \lambda = 0.0 \)
- \( (1, 0) \) when \( \lambda \approx 0.643 \)
- \( (2, 0) \) when \( \lambda \approx 0.923 \)
- \( (1, 1) \) when \( \lambda \approx 0.154 \)

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 $j \in J$:

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

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 $x^s$:

$$\Psi(x^s, x^t) = \begin{cases} (j_k, \lambda_k, d_k) : & (i) \ \lambda_k \in \Lambda(x_{j_k}^s, x_{j_k}^t) \\ & (ii) \ d_k = \begin{cases} +1, \text{ if } x_{j_k}^s < x_{j_k}^t \\ -1, \text{ otherwise, and} \end{cases} \\ & (iii) \ \lambda_k \leq \lambda_{k+1} \end{cases}$$
Line Search within FP

The shooting procedure (pure integer case)

Compute \( \Psi(x^s, x^t) = \{(j_k, \lambda_k, d_k)_{k=1,...,K}\} \)

\[ x \leftarrow [x^s] \]

\[
\text{forall} \quad k = 1,\ldots,K \\
\]

\[ x_{jk} \leftarrow x_{jk} + d_{jk} \]

\[ \text{if } x \text{ is an incumbent solution} \]

\[ \text{record } x \]

\[
\text{end}
\]

\[
\text{end}
\]

No. of operations is \( O(\sum_j|\|x^s_j - x^t_j\|)| \) = total number of breakpoints

Vs

At least \( O(n(\sum_j|\|x^s_j - x^t_j\|)|) \) if doing rounding to find all integer solutions along shooting line
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*.

\[
\begin{align*}
\text{min} & \quad \sum_{j \not\in J} c_j x_j \\
\text{s.t.} & \quad \sum_{j \not\in J} A_j x_j \leq b - \sum_{j \in J} A_j \bar{x}_j
\end{align*}
\]
Line Search within FP

Finding suitable end points $x^t$

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

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

$$\max \sum_{i \in Q} \ln(b_i - a_i^T x)$$

s.t. $Ax \leq b$

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

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

**Option 2:** $x^t =$ 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 $[x^s]$ violates constraint $i$, i.e.,

$$\sum_{j \in J} a_j^i [x_j^s] + \sum_{j \notin J} a_j^i x_j^s > b_i$$

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

$$\sum_{j \in J} a_j^i x_j \leq b_i - \sum_{j \notin J} a_j^i x_j^s$$

---

Line Search within FP

Finding suitable end points $x^t$

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

**Option 3:** $x^t = x^s + d$ where $d$ is a conic combination of constraints violated by $[x^s]$ 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 $x^t$

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

Q. Is feasibility of $x^t$ important?
Q. Is feasibility of $x^t$ important?

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

$x_{LP} = (1.0,0.1)$
$x_{int} = (0.8,0.25)$
Q. Is feasibility of $x^t$ important?

$\min \ x_1 + 2x_2$

subject to  
$2x_1 - x_2 \geq 1$

$\alpha(2x_1 - 10x_2 \leq 1)$

$x_1, x_2 \in \{0,1\}$

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

$x_{int} = (0.8, 0.25)$
Q. Is feasibility of $x^t$ important?

min $x_1 + 2x_2$

s.t. $2x_1 - x_2 \geq 1$
     $2x_1 - 10x_2 \leq 1$
     $x_1, x_2 \in \{0, 1\}$

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

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

$x_{LP} + d \approx (0.804, 1.080)$
Line Search within FP

Q. Is feasibility of $x^t$ important?

$x_{\text{int}} - x_{\text{LP}}$ is a direction towards LP feasibility. Hence, to recover feasibility:

- $x_1$ should decrease and $x_2$ should increase
- rounded value of $x_2$ changes before $x_1$

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

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

$[x_{\text{LP}}] = (1, 0)$

$= (1, 1)$ after incrementing $x_2$  
$= (0, 1)$ after decrementing $x_1$
Extending the Line Search

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

$$\Lambda(x^s_j, x^t_j) = \{0 \leq \lambda \leq 1: (1 - \lambda)x^s_j + \lambda x^t_j = \alpha + 0.5\text{ for some integer } \alpha\}$$
Line Search within FP

Extending the Line Search

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

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

Convex combination

Linear combination within variable bounds

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

Convex combination

Linear combination within variable bounds
Line Search within FP

Extending the Line Search

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

$\Lambda(x^s_j, x^t_j) = \left\{ 0 \leq \lambda \leq 1 : (1 - \lambda)x^s_j + \lambda x^t_j = \alpha + 0.5 \text{ for some integer } \alpha \right\}$

**Convex combination**

Linear combination within variable bounds

$\overline{\Lambda}(x^s_j, x^t_j) = \left\{ \lambda : (1 - \lambda)x^s_j + \lambda x^t_j = \alpha + 0.5 \text{ for some integer } \alpha \text{ s.t. } l_j \leq \alpha \leq u_j \right\}$

![Diagram showing line search with lambda values and variable bounds]
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

---


Line Search within FP

\[ D = (l', u') \subseteq (l, u) \]
\[ [x_j]^D = \arg\min_z \{ |z - x_j| : l'_j \leq z_j \leq u'_j \text{ and } z \text{ integer} \} \]

Propagation in rounding\(^6\)

```python
propRound(x)

D = (l', u') ← (l, u)
rank(x) = {j_1, j_2, ..., j_{|J|}}
forall k = 1, ..., |J|:
    \begin{align*}
    & x_{jk} ← [x_{jk}]^D \\
    & D ← propagate(x_{jk})
    \end{align*}
end
if x is an incumbent solution:
    record x
end
```

Note: 
- \( z \) is the closest integer value to \( x_j \) within the allowable domain

---

Line Search within FP

\[ D = (l', u') \subseteq (l, u) \]

\[ [x_j]^D = \arg\min_z \{ |z - x_j| : l'_j \leq z_j \leq u'_j \text{ and } z \text{ integer} \} \]

**Propagation in rounding**

\[
\text{propRound}(x)
\]

\[
D = (l', u') \leftarrow (l, u) \\
\text{rank}(x) = \{j_1, j_2, \ldots, j_{\lvert J \rvert}\} \\
\text{forall } k = 1, \ldots, \lvert J \rvert \\
\quad \text{\quad } x_{jk} \leftarrow [x_{jk}]^D \\
\quad \text{\quad } D \leftarrow \text{propagate}(x_{jk}) \\
\text{end} \\
\text{if } x \text{ is an incumbent solution} \\
\quad \text{record } x \\
\text{end}
\]

**Propagation in line search**

\[
\text{Compute } \Xi(x^s, x^t) = \{(j_k, \lambda_k, d_k)_{k=1,\ldots,K}\}
\]

\[
x \leftarrow \text{propRound}(x^s)
\]

\[
D = (l', u') \leftarrow (l, u) \\
\text{forall } k = 1, \ldots, K \\
\text{forall } j_{jk} \\
\quad \text{\quad } x_{jk} \leftarrow [x_{jk} + d_{jk}]^D \\
\quad \text{\quad } \text{if } x \text{ is an incumbent solution} \\
\quad \text{\quad } \quad \text{record } x \\
\quad \text{\quad } \text{end} \\
\quad \text{\quad } \text{if no more changes for } j_k \\
\quad \text{\quad } \quad D \leftarrow \text{propagate}(x_{jk}) \\
\quad \text{\quad } \text{end} \\
\text{end}
Computational Results

Test-bed: 1,200+ MIP instances from miplib3\textsuperscript{[7]}, miplib2010\textsuperscript{[8]}, cor@l\textsuperscript{[9]}, and orlib\textsuperscript{[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.

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

**Experiment 1**

- **% improvement in objective over FP**
- **Cumulative time taken to solve instances (s)**
Computational Results

**Experiment 1**

- % improvement in objective over FP
- Cumulative time taken to solve instances (s)

**Experiment 2**

- % improvement in objective over FP
- Cumulative time taken to solve instances (s)
Computational Results

FP Vs FP+P

% improvement in objective over FP

% instances

FP+AC
FP+ACCPM
FP+D
FP+P
Computational Results

**FP Vs FP+P**

% improvement in objective over FP

% instances

% improvement in objective over FP
Computational Results

Experiment 3

- % improvement in objective over FP+P
- Cumulative time taken to solve instances (s)
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