Integer and Combinatorial Optimization: Finding Feasible Solutions in Branch and Bound

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

March 2019
Feasible solutions

We consider the problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0 \\
& \quad x_j \text{ binary, } j=1,\ldots,p
\end{align*}
\]

where \(1 \leq p \leq n\).

Good feasible solutions give good upper bounds \(z^u\).

The feasibility pump uses rounding, diving successively fixes variables, and local branching solves a small version of the integer program.
Properties of a feasible solution

A feasible iterate satisfies two properties:
- feasible in the LP relaxation, so $Ax = b$, $x \geq 0$.
- integral.

The feasibility pump alternates between these two requirements.
Properties of a feasible solution

A feasible iterate satisfies two properties:
- feasible in the LP relaxation, so $Ax = b$, $x \geq 0$.
- integral.

The feasibility pump alternates between these two requirements.
The feasibility pump

1. Let $x^{LP}$ be feasible in the LP relaxation.
2. Round the first $p$ components of $x^{LP}$ to get a solution $x^I$ satisfying the integrality restriction.
3. If $x^I$ is feasible in the integer program, STOP.
4. Else, solve the following LP to find a feasible solution to the LP relaxation that is close to $x^I$:

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^{p} |x_j - x^I_j|$$

subject to

$$Ax = b$$
$$x \geq 0$$

5. Update $x^{LP}$ and return to Step 2.
The feasibility pump

1. Let $x^{LP}$ be feasible in the LP relaxation.
2. Round the first $p$ components of $x^{LP}$ to get a solution $x^I$ satisfying the integrality restriction.
3. If $x^I$ is feasible in the integer program, STOP.
4. Else, solve the following LP to find a feasible solution to the LP relaxation that is close to $x^I$:

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^p |x_j - x^I_j|$$
subject to

$$Ax = b$$
$$x \geq 0$$

5. Update $x^{LP}$ and return to Step 2.
The feasibility pump

1. Let $x^{LP}$ be feasible in the LP relaxation.
2. Round the first $p$ components of $x^{LP}$ to get a solution $x^I$ satisfying the integrality restriction.
3. If $x^I$ is feasible in the integer program, STOP.
4. Else, solve the following LP to find a feasible solution to the LP relaxation that is close to $x^I$:

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^{p} |x_j - x^I_j|$$
subject to  \[ Ax = b \]
\[ x \geq 0 \]

5. Update $x^{LP}$ and return to Step 2.
The feasibility pump

1. Let $x^{LP}$ be feasible in the LP relaxation.
2. Round the first $p$ components of $x^{LP}$ to get a solution $x^I$ satisfying the integrality restriction.
3. If $x^I$ is feasible in the integer program, STOP.
4. Else, solve the following LP to find a feasible solution to the LP relaxation that is close to $x^I$:

   $$\min_{x \in \mathbb{R}^n} \sum_{j=1}^{p} |x_j - x_j^I|$$

   subject to
   
   $$Ax = b$$
   $$x \geq 0$$

5. Update $x^{LP}$ and return to Step 2.
The feasibility pump

1. Let $x^{LP}$ be feasible in the LP relaxation.
2. Round the first $p$ components of $x^{LP}$ to get a solution $x^I$ satisfying the integrality restriction.
3. If $x^I$ is feasible in the integer program, STOP.
4. Else, solve the following LP to find a feasible solution to the LP relaxation that is close to $x^I$:

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^{p} |x_j - x^I_j|$$
subject to

$$Ax = b$$
$$x \geq 0$$

5. Update $x^{LP}$ and return to Step 2.
Comments on the feasibility pump

There is no guarantee that this procedure will converge.
If it cycles, can use a random perturbation to restart.

The use of the $L_1$-norm encourages sparsity.
In this context, that means it encourages $x_j = x_j^I$ for $j = 1, \ldots, p$, so it has the effect of encouraging integrality.
An example

\[
\begin{align*}
\min_{x \in \mathbb{R}^2} & \quad x_1 + 2x_2 \\
\text{subject to} & \quad -3x_1 + 5x_2 \geq 0 \\
& \quad 3x_1 + 5x_2 \geq 4 \\
& \quad x \geq 0 \\
& \quad x_j \text{ binary, } j=1,2
\end{align*}
\]
The solution to the LP relaxation is \( x^{LP} = \left( \frac{2}{3}, \frac{2}{5} \right) \).

Rounding gives \( x^I = (1, 0) \).

Closest point in \( L_1 \)-norm to \( x^I \) is the updated \( x^{LP} = (1, \frac{3}{5}) \).

Rounding gives \( x^I = (1, 1) \); the algorithm terminates.

Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is \( x^* = (0, 1) \).
The example

The solution to the LP relaxation is $x^{LP} = \left( \frac{2}{3}, \frac{2}{5} \right)$. Rounding gives $x^I = (1, 0)$.

Closest point in $L_1$-norm to $x^I$ is the updated $x^{LP} = (1, \frac{3}{5})$. Rounding gives $x^I = (1, 1)$; the algorithm terminates.

Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is $x^* = (0, 1)$. 
The example

The solution to the LP relaxation is $x^{LP} = \left( \frac{2}{3}, \frac{2}{5} \right)$.

Rounding gives $x^I = (1, 0)$.

Closest point in $L_1$-norm to $x^I$ is the updated $x^{LP} = (1, \frac{3}{5})$.

Rounding gives $x^I = (1, 1)$; the algorithm terminates.

Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is $x^* = (0, 1)$. 

The feasibility pump
The example

The solution to the LP relaxation is \( x^{LP} = \left( \frac{2}{3}, \frac{2}{5} \right) \).
Rounding gives \( x^I = (1, 0) \).
Closest point in \( L_1 \)-norm to \( x^I \) is the updated \( x^{LP} = \left( 1, \frac{3}{5} \right) \).
Rounding gives \( x^I = (1, 1) \); the algorithm terminates.
Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is \( x^* = (0, 1) \).
The example

The solution to the LP relaxation is $x^{LP} = \left( \frac{2}{3}, \frac{2}{5} \right)$. Rounding gives $x^I = (1, 0)$. Closest point in $L_1$-norm to $x^I$ is the updated $x^{LP} = \left( 1, \frac{3}{5} \right)$. Rounding gives $x^I = (1, 1)$; the algorithm terminates. Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is $x^* = (0, 1)$. 
The example

The solution to the LP relaxation is $x^{LP} = \left(\frac{2}{3}, \frac{2}{5}\right)$. Rounding gives $x^I = (1, 0)$. Closest point in $L_1$-norm to $x^I$ is the updated $x^{LP} = \left(1, \frac{3}{5}\right)$. Rounding gives $x^I = (1, 1)$; the algorithm terminates. Note that the solution returned by the algorithm is not optimal to the integer program: the optimal point is $x^* = (0, 1)$. 
Refinements

The method can be refined to try to improve the quality of the final solution.

For example, the subproblem for updating $x^{LP}$ can include the original objective function $c^T x$:

$$\min_{x \in \mathbb{R}^n} \quad \alpha \sum_{j=1}^{p} |x_{j} - x_{j}^l| + (1 - \alpha)c^T x$$

subject to

$$Ax = b$$

$$x \geq 0$$

where $0 \leq \alpha \leq 1$.

The parameter $\alpha$ can be gradually increased or decreased if desired.
Diving

Let $x^{LP}$ be a feasible solution to the LP relaxation.

We can fix one of the variables to a binary value and solve the LP relaxation again.

We can repeat this process until either we find an integer solution or the LP becomes infeasible.

This is like depth first search branching, where we push one branch of the tree out as far as possible.

Various rules can be used to determine which variable to fix.

For example, we can choose to fix the variable with the smallest value of $\min\{x_j, 1 - x_j\}$ to its closest bound.
Local branching

Once we have a feasible integer solution $x^I$, we can look for better feasible solutions in its neighborhood.

One way to define the neighborhood is to limit the number of integer variables that can be changed from their values in $x^I$.

This can be accomplished by solving the integer program

$$\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x \\
{\text{subject to}} & \quad Ax = b \\
& \quad x \geq 0 \\
& \quad ||x - x^I||_1 \leq k \\
& \quad x_j \text{ binary, j=1,…,p}
\end{align*}$$

for a positive integer parameter $k$, for example $k = 20$. 
Writing the constraint as a linear constraint

The one-norm constraint is equivalent to the linear constraint

\[ \sum_{j: x_j^l = 0} x_j + \sum_{j: x_j^l = 1} (1 - x_j) \leq k. \]

The local branching integer program should give a far smaller enumeration tree than the original problem.

If a better solution is found then the process can be repeated, after updating \( x^l \).