From Pivot and Complement to the Feasibility Pump

Andrea Lodi
Canada Excellence Research Chair
École Polytechnique de Montréal, Québec, Canada
andrea.lodi@polymtl.ca

Discrepancy Theory & Integer Programming Workshop
June 11, 2018 @ CWI, Amsterdam (The Netherlands)

Setting

- We consider a general Mixed Integer Linear Programming problem (MILP) in the form

$$\max\{c^T x : Ax \leq b, x \geq 0,\ x_j \in \mathbb{Z}, \forall j \in I \subseteq \mathcal{N}\} \quad (1)$$

where matrix $A^{m\times n}$ does not have a special structure.

Setting

- We consider a general Mixed Integer Linear Programming problem (MILP) in the form

$$\max\{c^T x : Ax \leq b, x \geq 0, \; x_j \in \mathbb{Z}, \forall j \in I \subseteq \mathcal{N}\} \tag{1}$$

  where matrix $A^{m \times n}$ does not have a special structure.

- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

Setting

- We consider a general Mixed Integer Linear Programming problem (MILP) in the form

$$\max\{c^T x : Ax \leq b, x \geq 0, \; x_j \in \mathbb{Z}, \forall j \in I \subseteq \mathcal{N}\} \tag{1}$$

  where matrix $A^{m \times n}$ does not have a special structure.
- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

- The role of (primal) heuristics in MILP solvers is associated with three distinct aspects.

  1. Achieving Integer-Feasibility Quickly.
     Finding a first feasible solution is sometimes the main issue when solving a MILP. This is true theoretically because the feasibility problem for MILP is $\mathcal{NP}$-complete, but also from the user's perspective the solver needs to provide a feasible solution as quick as possible.

Setting

- We consider a general Mixed Integer Linear Programming problem (MILP) in the form

$$\max\{c^T x : Ax \leq b, x \geq 0,\ x_j \in \mathbb{Z}, \forall j \in I \subseteq \mathcal{N}\} \tag{1}$$

  where matrix $A^{m \times n}$ does not have a special structure.

- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

- The role of (primal) heuristics in MILP solvers is associated with three distinct aspects.

1. Achieving Integer-Feasibility Quickly.
   Finding a first feasible solution is sometimes the main issue when solving a MILP. This is true theoretically because the feasibility problem for MILP is $\mathcal{NP}$-complete, but also from the user's perspective the solver needs to provide a feasible solution as quick as possible.

2. Reaching (quasi-)Optimality Quickly.
   Of course, once a feasible solution is there, the challenge becomes getting better and better ones, and local search heuristics come into the play for that.

Setting

- We consider a general Mixed Integer Linear Programming problem (MILP) in the form

$$\max\{c^T x : Ax \leq b, x \geq 0,\ x_j \in \mathbb{Z}, \forall j \in I \subseteq \mathcal{N}\} \tag{1}$$

  where matrix $A^{m \times n}$ does not have a special structure.
- Thus, the problem is solved through branch-and-bound and the bounds are computed by iteratively solving the LP relaxations through a general-purpose LP solver.

- The role of (primal) heuristics in MILP solvers is associated with three distinct aspects.

  1. Achieving Integer-Feasibility Quickly.
     Finding a first feasible solution is sometimes the main issue when solving a MILP. This is true theoretically because the feasibility problem for MILP is $\mathcal{NP}$-complete, but also from the user's perspective the solver needs to provide a feasible solution as quick as possible.

  2. Reaching (quasi-)Optimality Quickly.
     Of course, once a feasible solution is there, the challenge becomes getting better and better ones, and local search heuristics come into the play for that.

  3. Analyzing and Repair Infeasible MILP solutions.

Achieving Integer-Feasibility Quickly

- Let us denote by $\mathcal{B}$ the set of binary variables among the integer-constrained ones in I, i.e., $\mathcal{B} \subseteq I$.

- We initially restrict our attention to MILPs in the form (1) in which the all integer-constrained variables are in fact binary, i.e., $I = \mathcal{B}$ and, in addition, to pure IPs, i.e., $I = \mathcal{B} = \mathcal{N}$.

- Let us also denote the continuous relaxation on (1) as

$$P := \{Ax \leq b, 0 \leq x \leq 1\}. \qquad (2)$$

Achieving Integer-Feasibility Quickly

- Let us denote by $\mathcal{B}$ the set of binary variables among the integer-constrained ones in I, i.e., $\mathcal{B} \subseteq I$.

- We initially restrict our attention to MILPs in the form (1) in which the all integer-constrained variables are in fact binary, i.e., $I = \mathcal{B}$ and, in addition, to pure IPs, i.e., $I = \mathcal{B} = \mathcal{N}$.

- Let us also denote the continuous relaxation on (1) as

$$P := \{Ax \leq b, 0 \leq x \leq 1\}. \tag{2}$$

- We look for an initial feasible solution by considering two algorithms. Namely,
  - the `Pivot and Complement` (`P&C`) [Balas & Martin, 1980], and
  - the `Feasibility Pump` (`FP`) [Fischetti, Glover & Lodi 2005].

Pivot & Complement: **basic idea**

- The basic idea of the heuristic is that problem (1) is equivalent to

$$\max\{c^T x : Ax + y = b, 0 \leq x \leq 1, y \geq 0, y_i \text{ basic}, \forall i = 1, \ldots, m\}.$$

Pivot & Complement: **basic idea**

- The basic idea of the heuristic is that problem (1) is equivalent to

$$\max\{c^T x : Ax + y = b, 0 \leq x \leq 1, y \geq 0, y_i \text{ basic}, \forall i = 1, \ldots, m\}. \qquad (3)$$

- In other words, if the bounded Simplex is used and all artificial slack variables are basic, then all original binary variables are nonbasic at their bound, i.e., integer and the solution is feasible.

Pivot & Complement: basic idea

- The basic idea of the heuristic is that problem (1) is equivalent to

$$\max\{c^T x : Ax + y = b, 0 \leq x \leq 1, y \geq 0, y_i \text{ basic}, \forall i = 1, \ldots, m\}. \qquad (3)$$

- In other words, if the bounded Simplex is used and all artificial slack variables are basic, then all original binary variables are nonbasic at their bound, i.e., integer and the solution is feasible.
- The P&C heuristic starts by solving the LP (2) reformulated through the slack variables.
- Then, to achieve the goal, as the name suggests, P&C performs two (types of) operations:
  - Pivoting operations to replace basic x variables by slack variables y, and

Pivot & Complement: **basic idea**

- The basic idea of the heuristic is that problem (1) is equivalent to

$$\max\{c^T x : Ax + y = b, 0 \leq x \leq 1, y \geq 0, y_i \text{ basic}, \forall i = 1, \ldots, m\}. \qquad (3)$$

- In other words, if the bounded Simplex is used and all artificial slack variables are basic, then all original binary variables are nonbasic at their bound, i.e., integer and the solution is feasible.

- The P&C heuristic starts by solving the LP (2) reformulated through the slack variables.

- Then, to achieve the goal, as the name suggests, P&C performs two (types of) operations:

  - Pivoting operations to replace basic x variables by slack variables y, and

  - Complementing operations to move towards primal feasibility if that is lost due to pivoting.

Pivot & Complement: **basic idea (cont.d)**

- Three types of pivoting are allowed:

  Type 1: primal feasible, exchanges one basic x_j with a y_i,

Pivot & Complement: basic idea (cont.d)

- Three types of pivoting are allowed:

  Type 1: primal feasible, exchanges one basic x_j with a y_i,

  Type 2: primal feasible, keeps the basic x_j's but reduces integer infeasibility

$$\sum_{j \in \mathcal{N}} \min\{x_j^*, 1 - x_j^*\}$$

Pivot & Complement: basic idea (cont.d)

- Three types of pivoting are allowed:

  Type 1: primal feasible, exchanges one basic x_j with a y_i,

  Type 2: primal feasible, keeps the basic x_j's but reduces integer infeasibility

$$\sum_{j \in \mathcal{N}} \min\{x_j^*, 1 - x_j^*\}$$

  Type 3: primal **in**feasible, exchanges one basic x_j with a y_i.

Pivot & Complement: basic idea (cont.d)

- Three types of pivoting are allowed:

  Type 1: primal feasible, exchanges one basic x_j with a y_i,

  Type 2: primal feasible, keeps the basic x_j's but reduces integer infeasibility

$$\sum_{j \in \mathcal{N}} \min\{x_j^*, 1 - x_j^*\}$$

  Type 3: primal **in**feasible, exchanges one basic x_j with a y_i.

- Complementation of one, two or three variables is performed to reduce infeasibility

$$\sum_{j \in \mathcal{N}} \max\{0, -x_j^*\} + \sum_{i=1}^{m} \max\{0, -y_i^*\}$$

Pivot & Complement: basic idea (cont.d)

- Three types of pivoting are allowed:

  Type 1: primal feasible, exchanges one basic x_j with a y_i,

  Type 2: primal feasible, keeps the basic x_j's but reduces integer infeasibility

$$\sum_{j \in \mathcal{N}} \min\{x_j^*, 1 - x_j^*\}$$

  Type 3: primal **in**feasible, exchanges one basic x_j with a y_i.

- Complementation of one, two or three variables is performed to reduce infeasibility

$$\sum_{j \in \mathcal{N}} \max\{0, -x_j^*\} + \sum_{i=1}^{m} \max\{0, -y_i^*\}$$

- The P&C heuristic in the 1980 proposed implementation runs in (high) polynomial time and makes extensive use of rounding and truncation to try to find solutions on the way.

- An addition improvement phase is performed still by using complementation in local search manner.

`Feasibility Pump`: **the basic scheme**

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.

`Feasibility Pump`: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.

`Feasibility Pump`: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.

- If x_j^* is integral $\forall j \in \mathcal{B}$, then x^* is a feasible MILP solution and we are done.

`Feasibility Pump`: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.

- If x_j^* is integral $\forall j \in \mathcal{B}$, then x^* is a feasible MILP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

Feasibility Pump: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.

- If x_j^* is integral $\forall j \in \mathcal{B}$, then x^* is a feasible MILP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

- From a geometric point of view, this simple heuristic generates two hopefully convergent trajectories of points x^* and $\widetilde{x}$ that satisfy feasibility in a partial but complementary way:

`Feasibility Pump`: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.

- If x_j^* is integral $\forall j \in \mathcal{B}$, then x^* is a feasible MILP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

- From a geometric point of view, this simple heuristic generates two hopefully convergent trajectories of points x^* and $\widetilde{x}$ that satisfy feasibility in a partial but complementary way:

  1. one, x^*, satisfies the linear constraints,

Feasibility Pump: the basic scheme

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ that is as close as possible to the current $\widetilde{x}$ by solving the problem

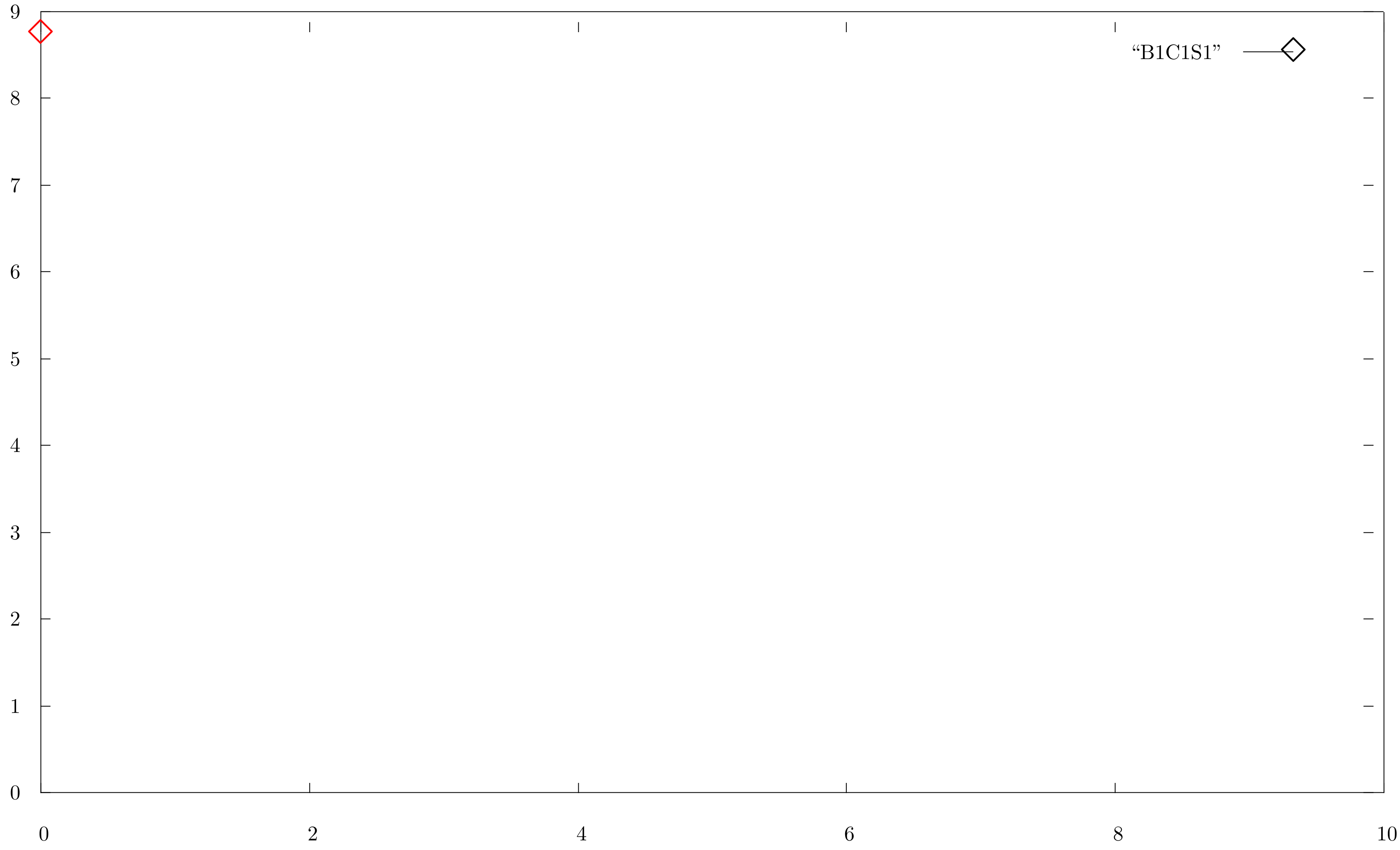
$$\min\{\Delta(x, \widetilde{x}) : x \in P\}$$

  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, it is an easily solvable LP problem.
- If x^*_j is integral $\forall j \in \mathcal{B}$, then x^* is a feasible MILP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

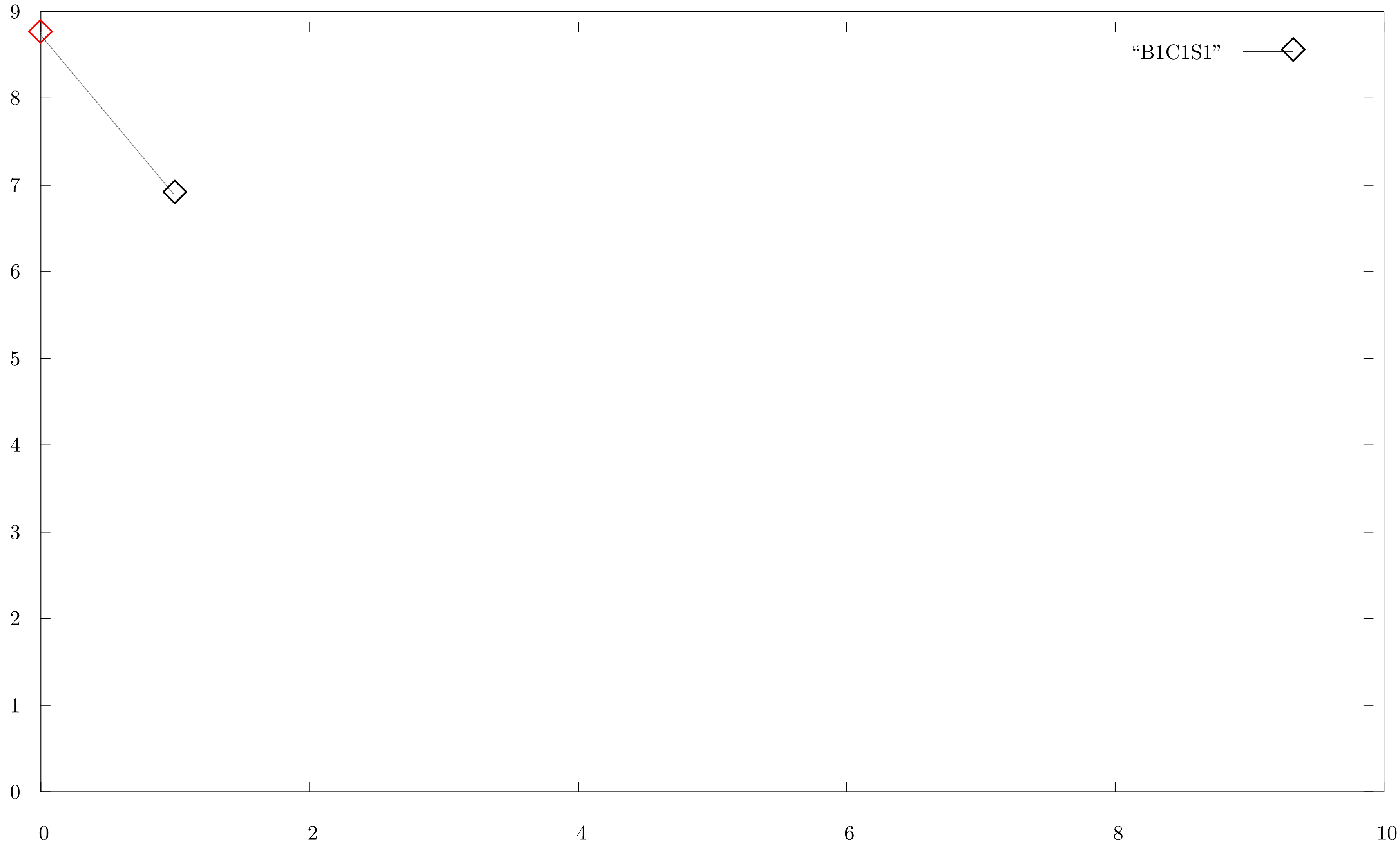
- From a geometric point of view, this simple heuristic generates two hopefully convergent trajectories of points x^* and $\widetilde{x}$ that satisfy feasibility in a partial but complementary way:

  1. one, x^*, satisfies the linear constraints,
  2. the other, $\widetilde{x}$, the integer requirement.

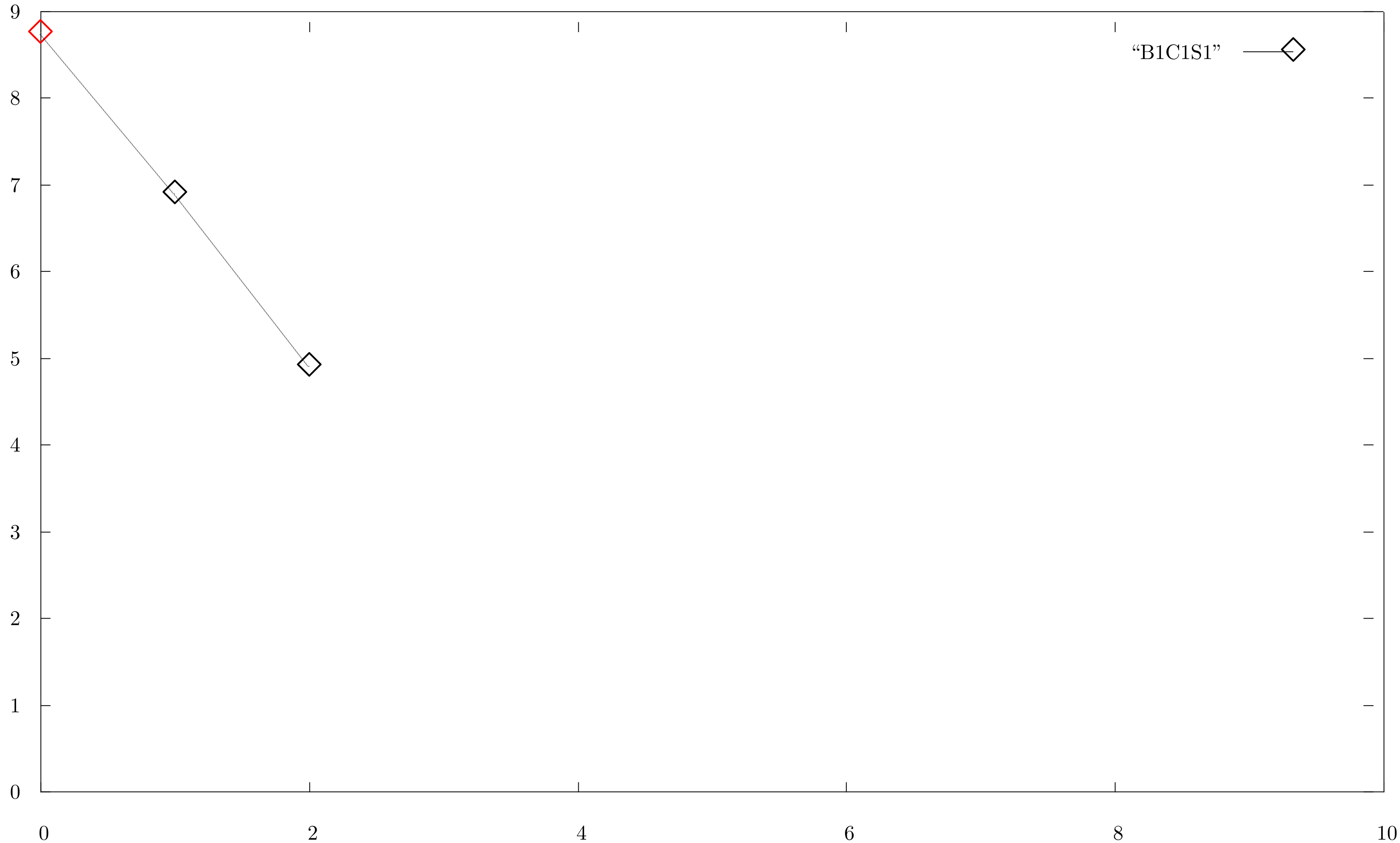
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



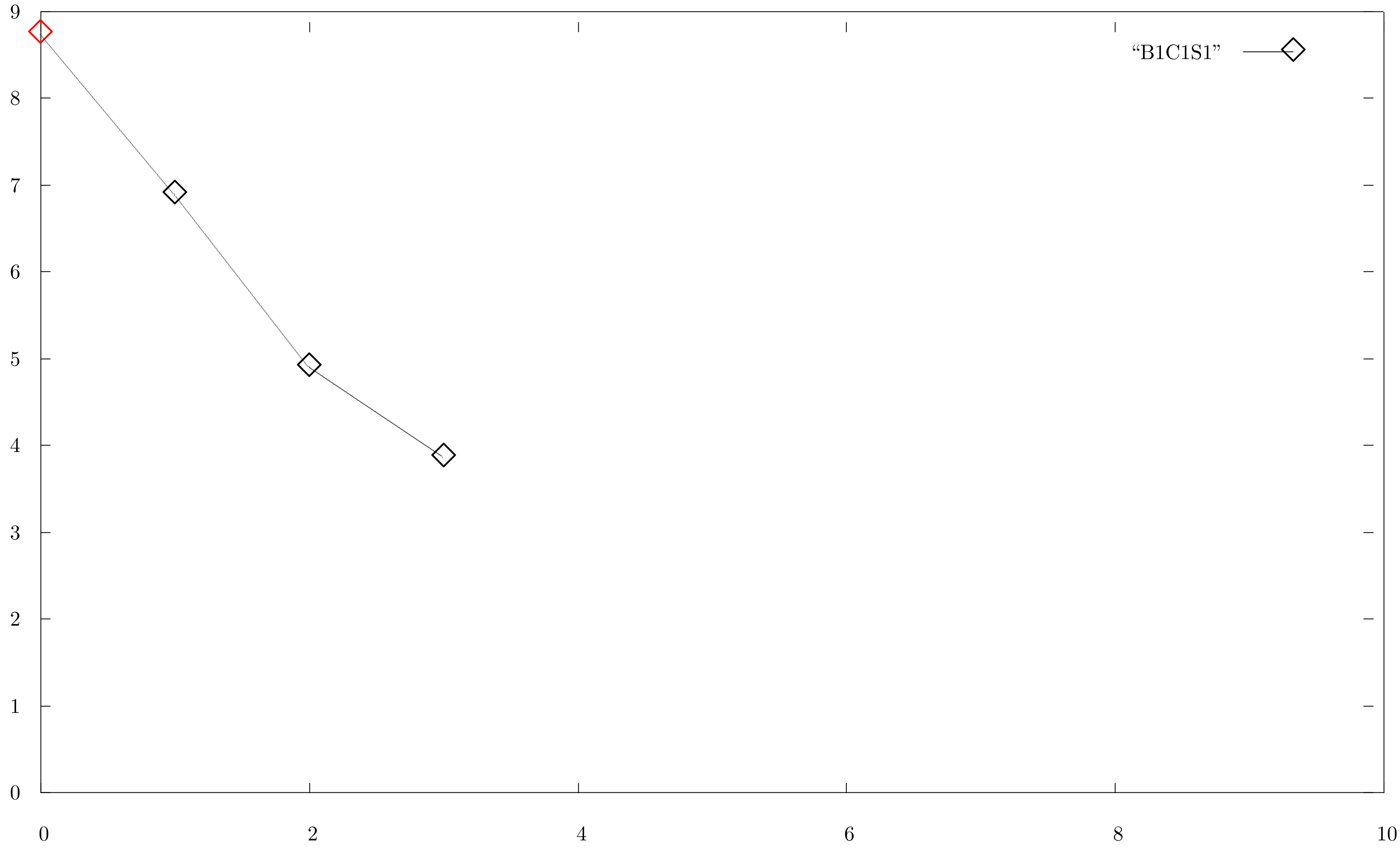
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



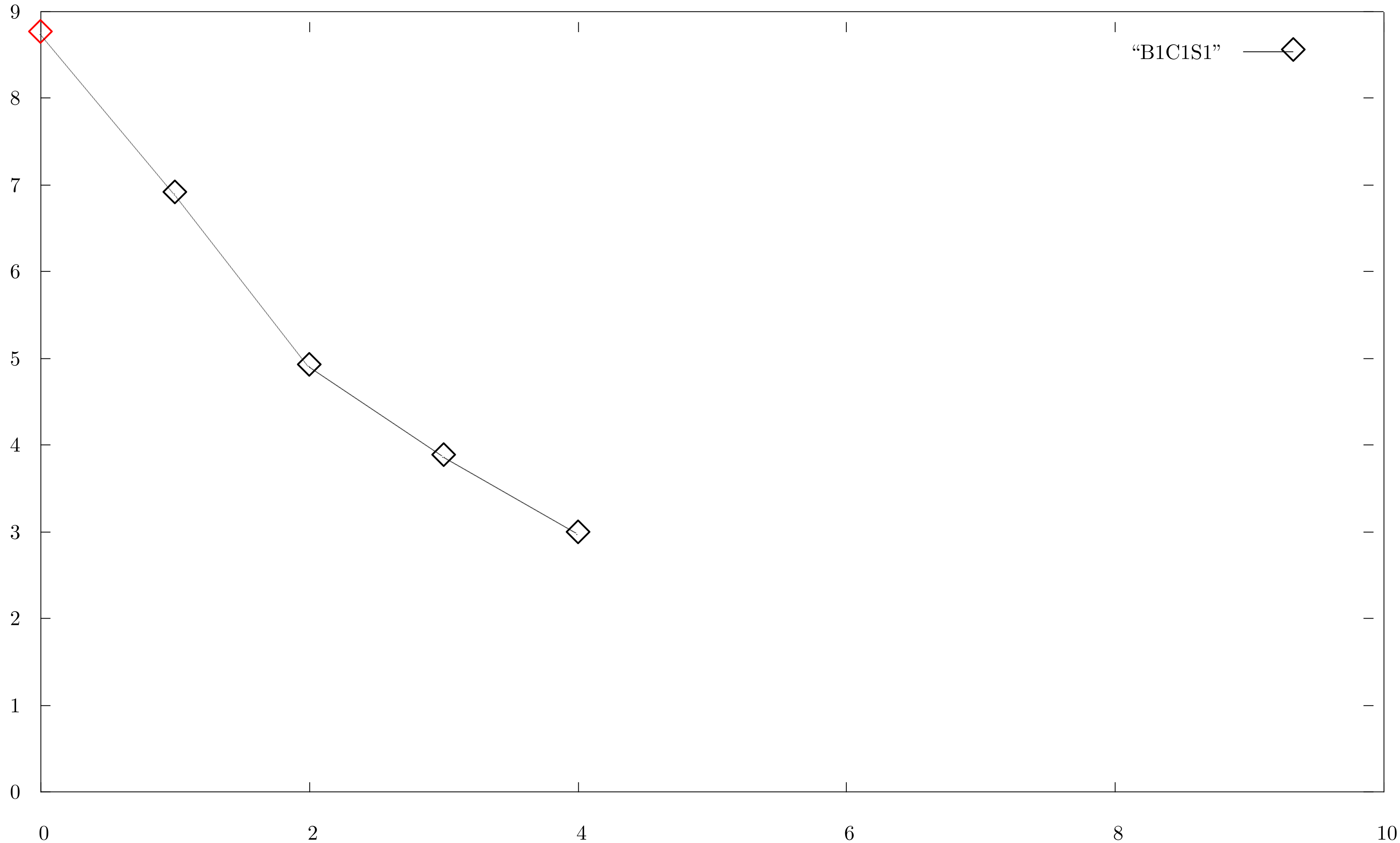
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



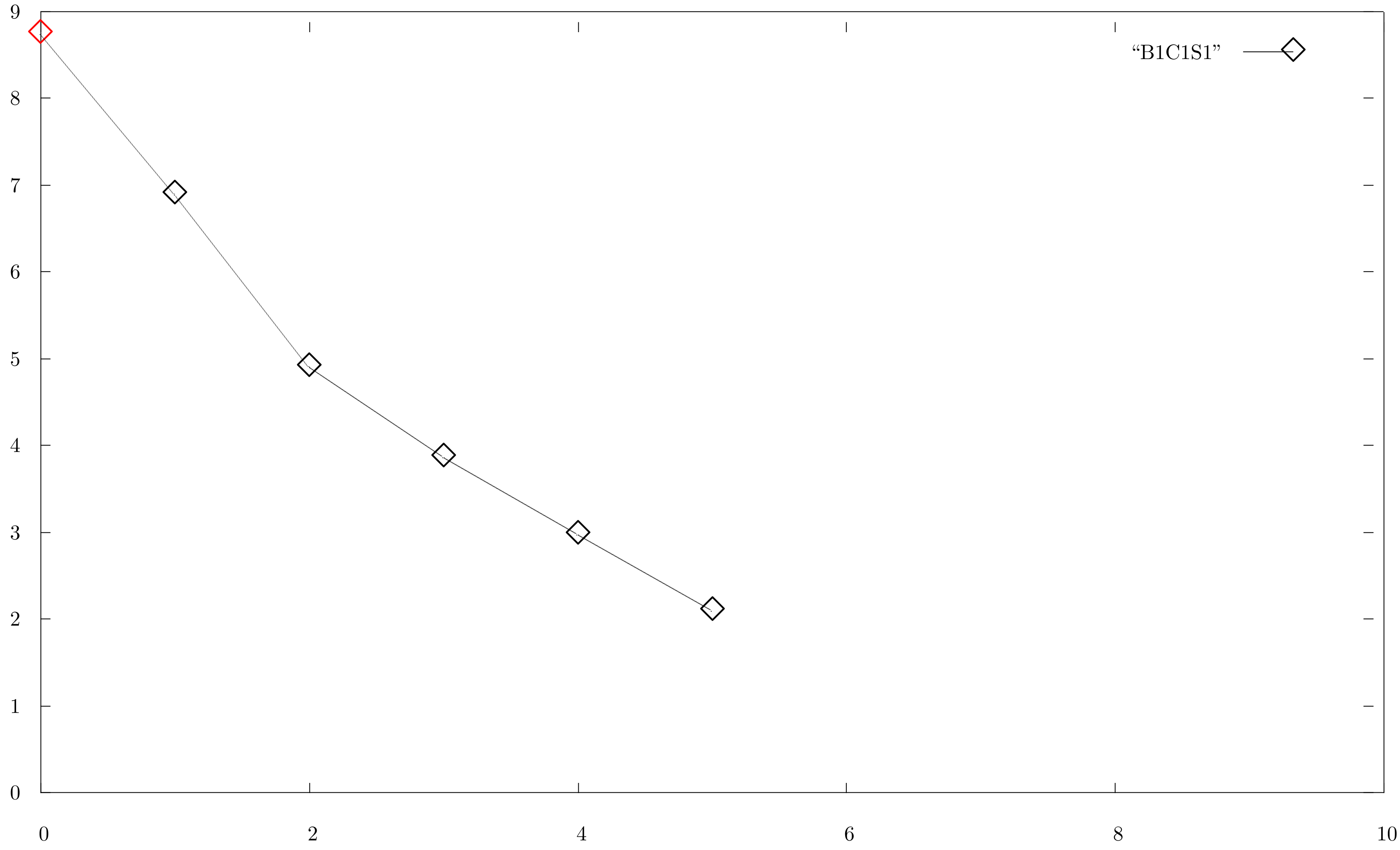
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



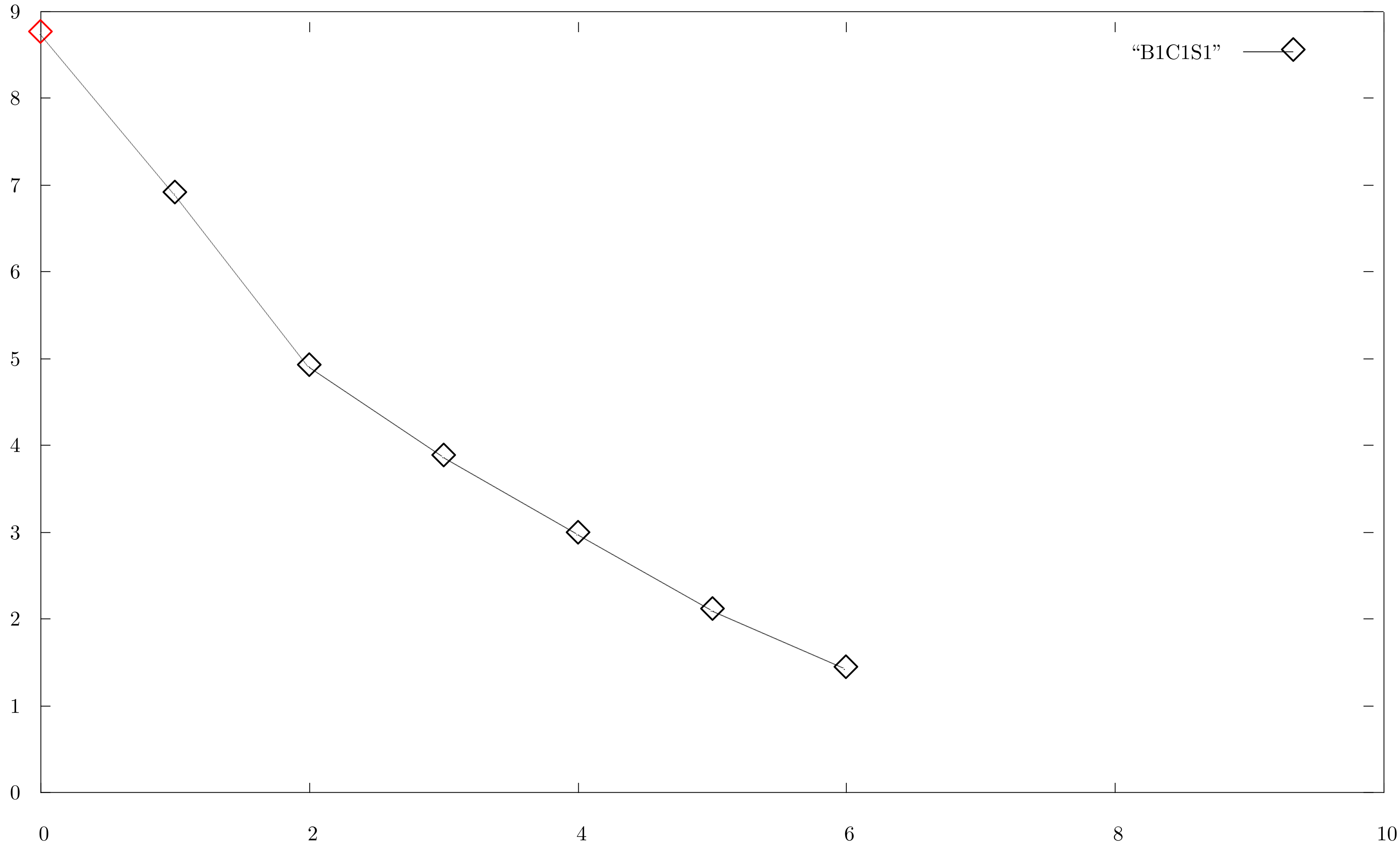
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



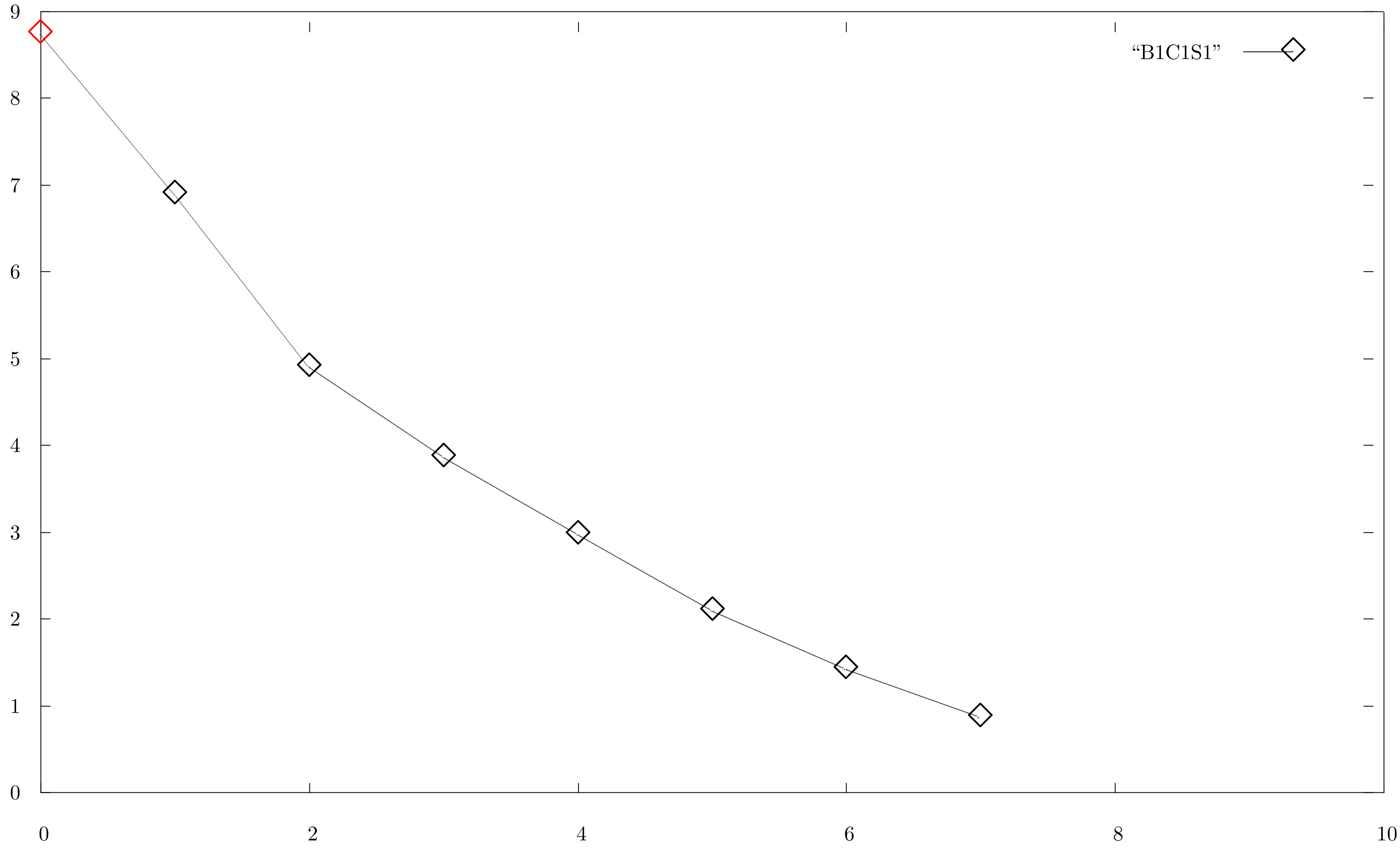
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



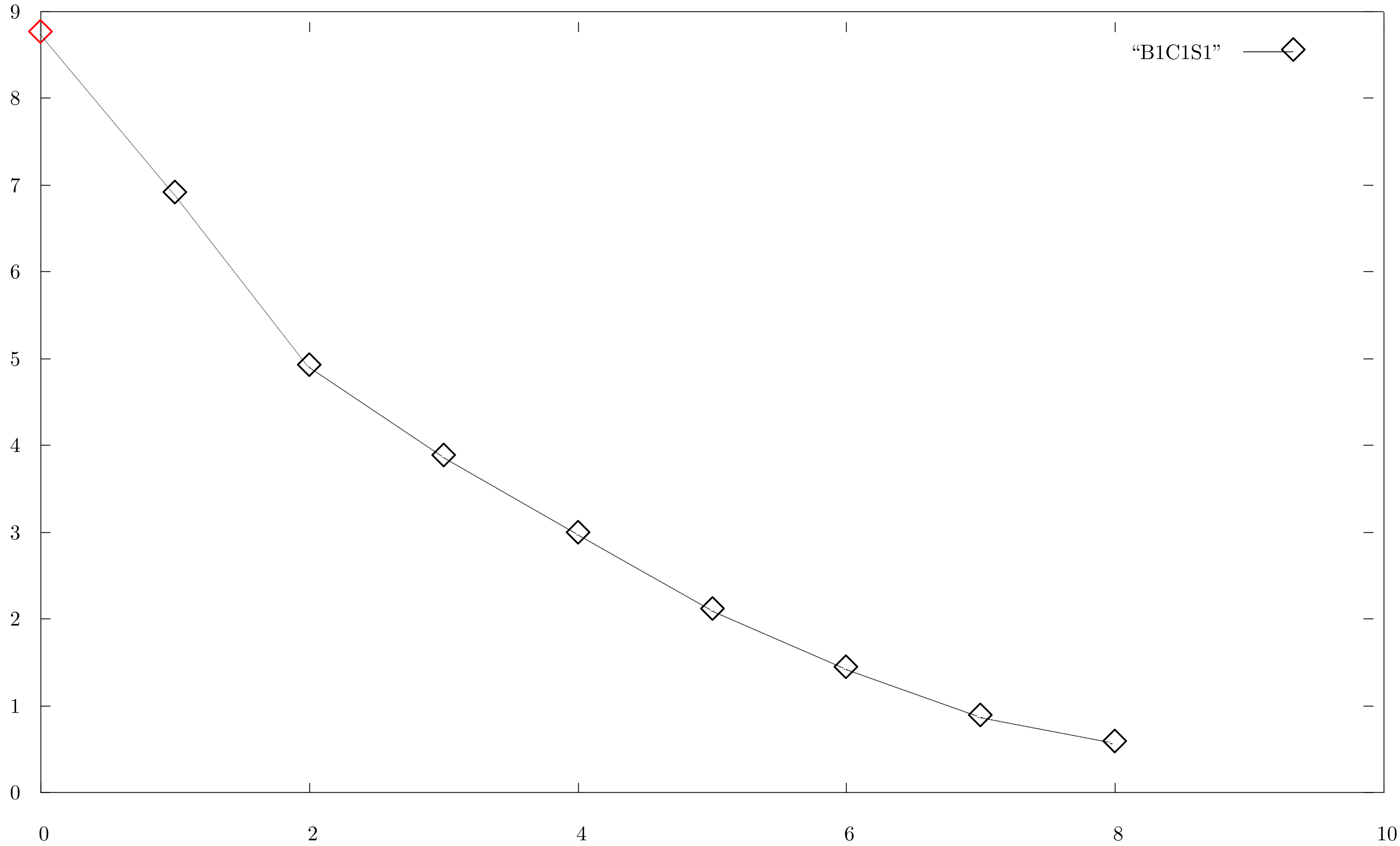
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



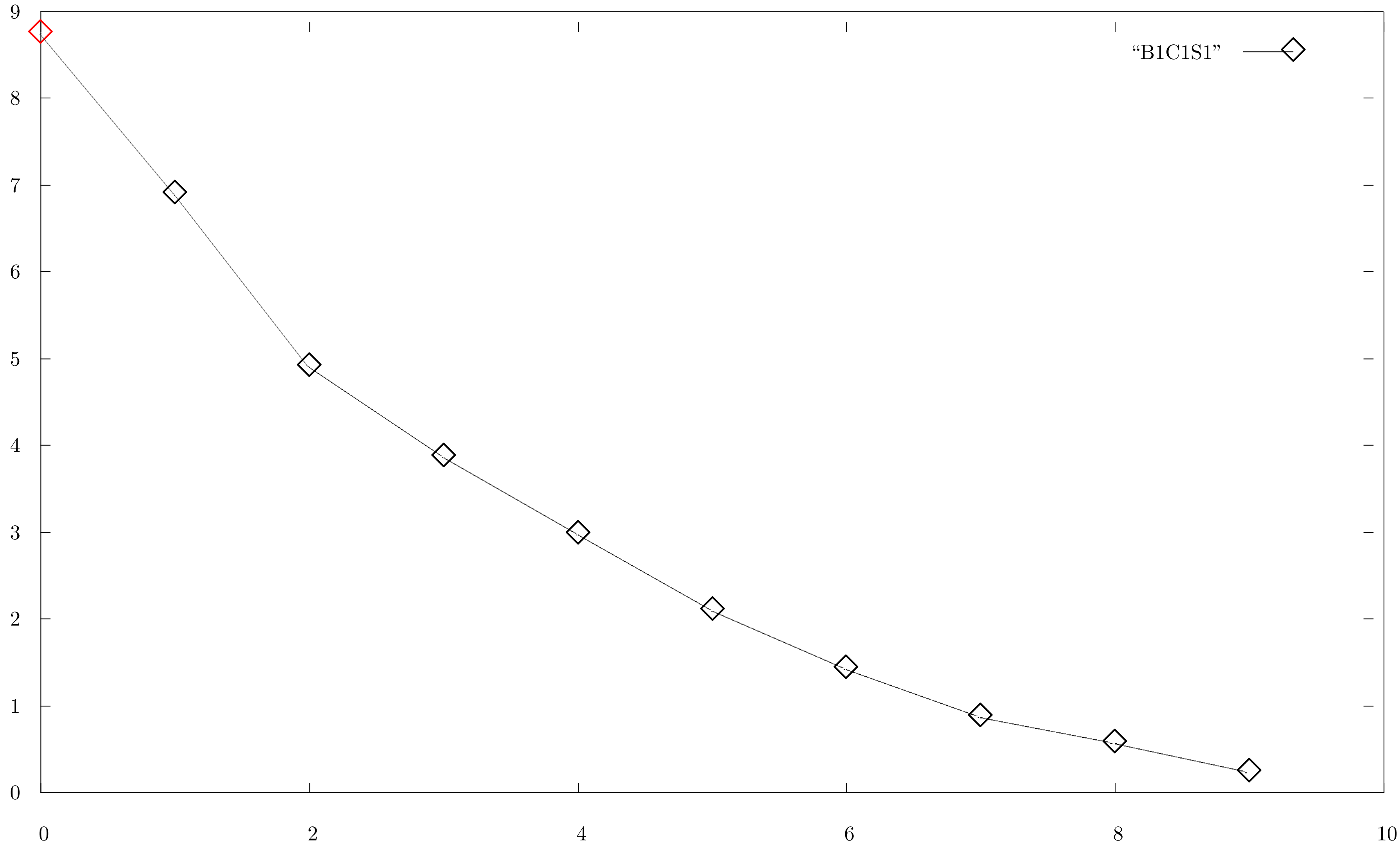
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



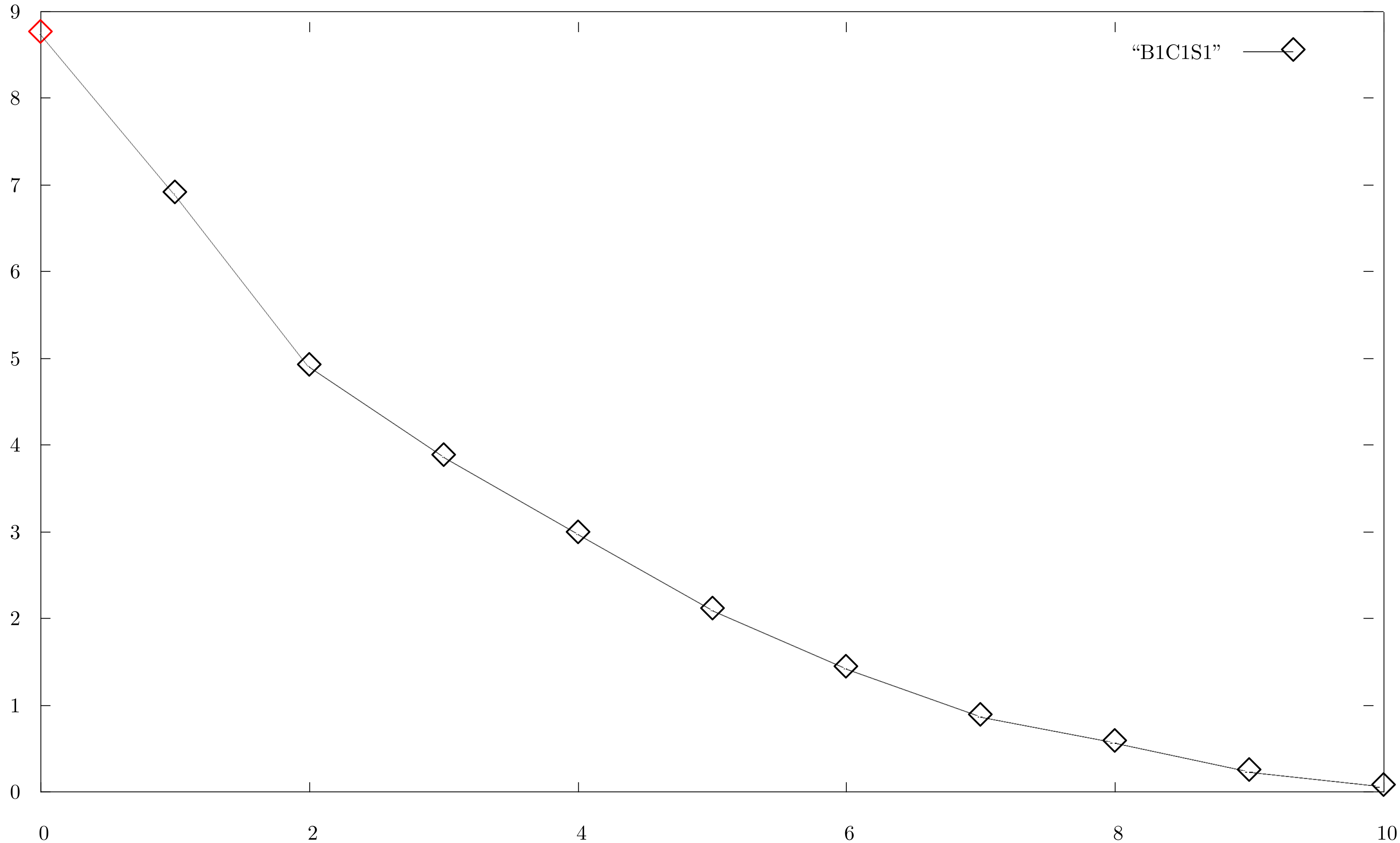
FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



FP: Definition of $\Delta(x^*, \widetilde{x})$

- We consider the L_1-norm distance between a generic point $x \in P$ and a given integer $\widetilde{x}$:

$$\Delta(x, \widetilde{x}) = \sum_{j \in \mathcal{B}} |x_j - \widetilde{x}_j|$$

The continuous variables x_j with $j \notin \mathcal{B}$, if any, do not contribute to this function.

FP: Definition of $\Delta(x^*, \widetilde{x})$

- We consider the L_1-norm distance between a generic point $x \in P$ and a given integer $\widetilde{x}$:

$$\Delta(x, \widetilde{x}) = \sum_{j \in \mathcal{B}} |x_j - \widetilde{x}_j|$$

The continuous variables x_j with $j \notin \mathcal{B}$, if any, do not contribute to this function.

- In the case of a binary MILP:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{B}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{B}: \widetilde{x}_j = 1} (1 - x_j) \tag{4}$$

FP: Definition of $\Delta(x^*, \widetilde{x})$

- We consider the L_1-norm distance between a generic point $x \in P$ and a given integer $\widetilde{x}$:

$$\Delta(x, \widetilde{x}) = \sum_{j \in \mathcal{B}} |x_j - \widetilde{x}_j|$$

The continuous variables x_j with $j \notin \mathcal{B}$, if any, do not contribute to this function.

- In the case of a binary MILP:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{B}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{B}: \widetilde{x}_j = 1} (1 - x_j) \tag{4}$$

- Given an integer $\widetilde{x}$, the closest point $x^* \in P$ can therefore be determined by solving the LP

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}. \tag{5}$$

FP: Definition of $\Delta(x^*, \widetilde{x})$

- We consider the L_1-norm distance between a generic point $x \in P$ and a given integer $\widetilde{x}$:

$$\Delta(x, \widetilde{x}) = \sum_{j \in \mathcal{B}} |x_j - \widetilde{x}_j|$$

  The continuous variables x_j with $j \notin \mathcal{B}$, if any, do not contribute to this function.

- In the case of a binary MILP:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{B}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{B}: \widetilde{x}_j = 1} (1 - x_j) \tag{4}$$

- Given an integer $\widetilde{x}$, the closest point $x^* \in P$ can therefore be determined by solving the LP

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}. \tag{5}$$

- We like to see such a distance as a difference of pressure between the two complementary infeasibility of x^* and $\widetilde{x}$, that we try to reduce by pumping the integrality of $\widetilde{x}$ into x^*.

FP: Definition of $\Delta(x^*, \widetilde{x})$

- We consider the L_1-norm distance between a generic point $x \in P$ and a given integer $\widetilde{x}$:

$$\Delta(x, \widetilde{x}) = \sum_{j \in \mathcal{B}} |x_j - \widetilde{x}_j|$$

The continuous variables x_j with $j \notin \mathcal{B}$, if any, do not contribute to this function.

- In the case of a binary MILP:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{B}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{B}: \widetilde{x}_j = 1} (1 - x_j) \tag{4}$$

- Given an integer $\widetilde{x}$, the closest point $x^* \in P$ can therefore be determined by solving the LP

$$\min\{\Delta(x, \widetilde{x}) : x \in P\}. \tag{5}$$

- We like to see such a distance as a difference of pressure between the two complementary infeasibility of x^* and $\widetilde{x}$, that we try to reduce by pumping the integrality of $\widetilde{x}$ into x^*.
- On the other hand, it is clearly a measure of vicinity, therefore a neighborhood.

FP: A first implementation

- MAIN PROBLEM, stalling issues:
  as soon as $\Delta(x^*, \widetilde{x})$ is not reduced when replacing $\widetilde{x}$ by x^*.

  If $\Delta(x^*, \widetilde{x}) > 0$ we still want to modify $\widetilde{x}$, even if this increases its distance from x^*.

  Hence, we reverse the rounding of some variables $x_j^*,\ j \in \mathcal{B}$ chosen so as to minimize the increase in the current value of $\Delta(x^*, \widetilde{x})$.

FP: A first implementation

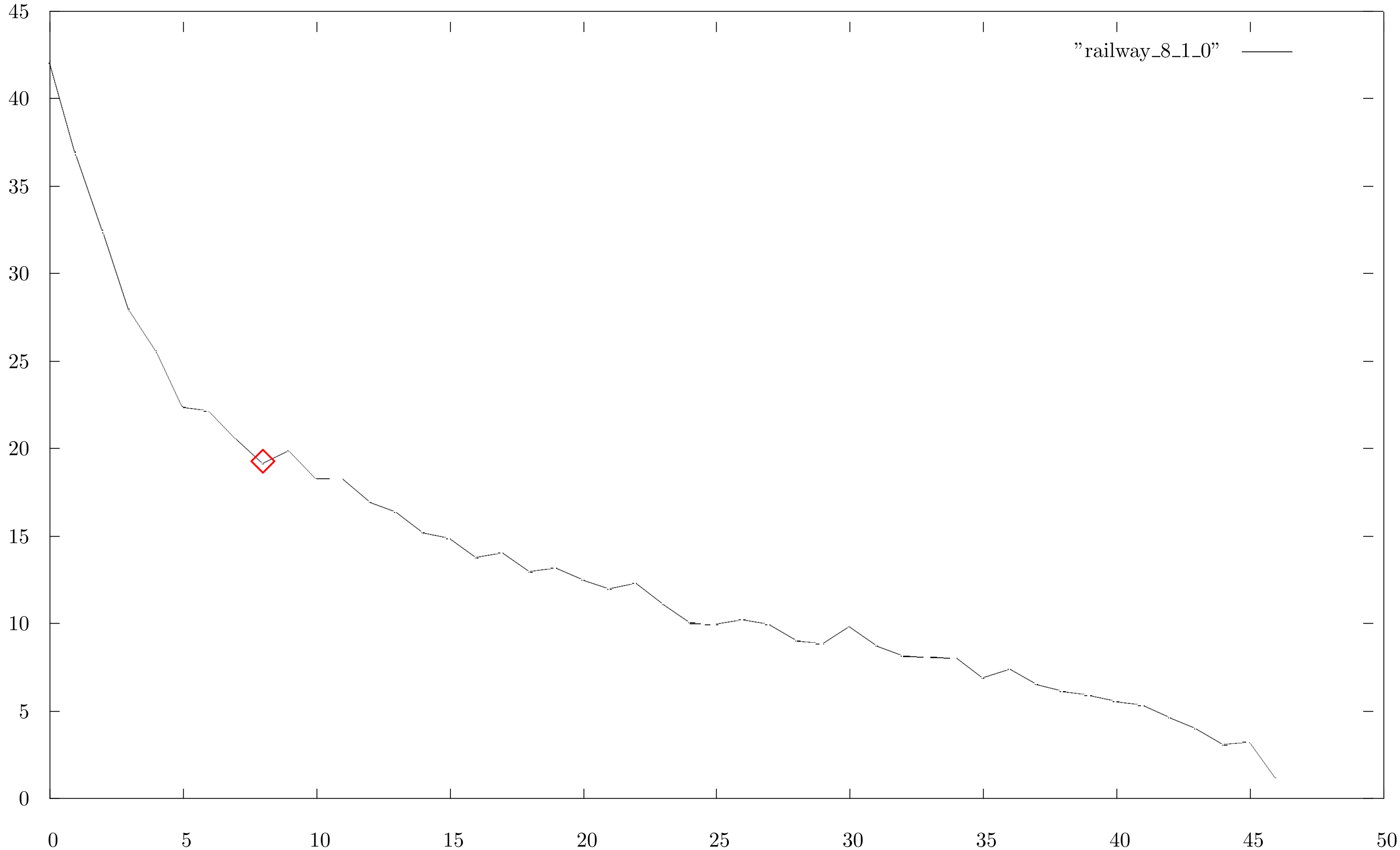
- MAIN PROBLEM, stalling issues:
  as soon as $\Delta(x^*, \widetilde{x})$ is not reduced when replacing $\widetilde{x}$ by x^*.

  If $\Delta(x^*, \widetilde{x}) > 0$ we still want to modify $\widetilde{x}$, even if this increases its distance from x^*.

  Hence, we reverse the rounding of some variables x^*_j, $j \in \mathcal{B}$ chosen so as to minimize the increase in the current value of $\Delta(x^*, \widetilde{x})$.

1. initialize nIT := 0 and $x^* :=$ argmax$\{c^T x : x \in P\}$;
2. if x^* is integer, return(x^*);
3. let $\widetilde{x} := [x^*]$ (= rounding of x^*);
4. while (time < TL) do
5. let nIT := nIT $+ 1$ and compute $x^* :=$ argmin$\{\Delta(x, \widetilde{x}) : x \in P\}$;
6. if x^* is integer, return(x^*);
7. if $\exists\, j \in \mathcal{B} : [x^*_j] \neq \widetilde{x}_j$ then
8. $\widetilde{x} := [x^*]$
   else
9. flip the TT = rand(T/2,3T/2) entries $\widetilde{x}_j$ $(j \in \mathcal{B})$ with highest $|x^*_j - \widetilde{x}_j|$
10. endif
11. enddo

FP: Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each pumping cycle



FP: better dealing with the objective function

- After the initialization of the algorithm in which the optimal solution x^* of the continuous relaxation is computed, the original objective function $c^T x$ is replaced by the distance function $\Delta(x, \widetilde{x})$.

- It is clear that if the number of FP iterations is large, especially if some random steps are applied, the trajectory can go very "far away" from the initial x^*, with potentially poor values of the original objective function.

- In other words, the search is not guided anymore by $c^T x$.

FP: better dealing with the objective function

- After the initialization of the algorithm in which the optimal solution x^* of the continuous relaxation is computed, the original objective function $c^T x$ is replaced by the distance function $\Delta(x, \widetilde{x})$.

- It is clear that if the number of FP iterations is large, especially if some random steps are applied, the trajectory can go very "far away" from the initial x^*, with potentially poor values of the original objective function.

- In other words, the search is not guided anymore by $c^T x$.

- The above issue can be partially corrected by taking a convex combination of the $\Delta(x, \widetilde{x})$ and $c^T x$ objective functions [Achterberg & Berthold 2007].
- Precisely, the combination used is

$$\Delta_\alpha(x, \widetilde{x}) := (1 - \alpha)\Delta(x, \widetilde{x}) - \alpha \frac{\sqrt{|\mathcal{B}|}}{||c||_2} c^T x \quad \text{with } \alpha \in [0, 1] \tag{6}$$

where α geometrically decreases at every iteration so as to give more and more emphasis on feasibility with respect to optimality.

FP: dealing with general integer variables

- The extension of the FP algorithm from 0-1 MILPs to general integers (variables in I) requires to redefine the objective function $\Delta(x, \widetilde{x})$ appropriately.

FP: dealing with general integer variables

- The extension of the FP algorithm from 0-1 MILPs to general integers (variables in I) requires to redefine the objective function $\Delta(x, \widetilde{x})$ appropriately. Indeed, for a variable, say x_j, which is not sit at a bound, say either ℓ_j or u_j, in an integer infeasible solution $\widetilde{x}$, it is less trivial to penalize its move.

FP: dealing with general integer variables

- The extension of the FP algorithm from 0-1 MILPs to general integers (variables in I) requires to redefine the objective function $\Delta(x, \widetilde{x})$ appropriately. Indeed, for a variable, say x_j, which is not sit at a bound, say either ℓ_j or u_j, in an integer infeasible solution $\widetilde{x}$, it is less trivial to penalize its move.
- A possible updated definition [Bertacco, Fischetti & Lodi 2007] is

$$\Delta(x, \widetilde{x}) := \sum_{j \in I : \widetilde{x}_j = \ell_j} (x_j - \ell_j) + \sum_{j \in I : \widetilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in I : \ell_j < \widetilde{x}_j < u_j} d_j \tag{7}$$

where the artificial variables $d_j (= |x_j - \widetilde{x}_j|)$ satisfy the additional constraints

$$d_j \geq x_j - \widetilde{x}_j \quad \text{and} \quad d_j \geq \widetilde{x}_j - x_j.$$

FP: dealing with general integer variables

- The extension of the FP algorithm from 0-1 MILPs to general integers (variables in I) requires to redefine the objective function $\Delta(x, \widetilde{x})$ appropriately. Indeed, for a variable, say x_j, which is not sit at a bound, say either ℓ_j or u_j, in an integer infeasible solution $\widetilde{x}$, it is less trivial to penalize its move.
- A possible updated definition [Bertacco, Fischetti & Lodi 2007] is

$$\Delta(x, \widetilde{x}) := \sum_{j \in I : \widetilde{x}_j = \ell_j} (x_j - \ell_j) + \sum_{j \in I : \widetilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in I : \ell_j < \widetilde{x}_j < u_j} d_j \qquad (7)$$

where the artificial variables $d_j (= |x_j - \widetilde{x}_j|)$ satisfy the additional constraints

$$d_j \geq x_j - \widetilde{x}_j \quad \text{and} \quad d_j \geq \widetilde{x}_j - x_j.$$

- In addition, the FP procedure is split into three parts:
  1. Binary first: the integrality requirement of the variables in $I \setminus \mathcal{B}$ is relaxed.
  2. Then, general integer: the requirement is reinstalled.
  3. A truncated enumeration phase is performed by solving the MILP with the original constraints and the objective function (7) which uses as $\widetilde{x}$ the "least infeasible" integer solution obtained so far.

The Mixed-Integer NON Linear case

- We consider here a Mixed Integer Non Linear Program of the form:

$$\text{MINLP}\begin{cases}\min f(x,y)\\ s.t.:\\ g(x,y)\leq b\\ x\in\mathbb{Z}^{n1}\\ y\in\mathbb{R}^{n2}\end{cases}$$

  where f is a function from $\mathbb{R}^{n1}\times\mathbb{R}^{n2}$ to $\mathbb{R}$ and g is a function from $\mathbb{R}^{n1}\times\mathbb{R}^{n2}$ to $\mathbb{R}^m$. We assume the feasible region $g(x,y)\leq b$ to be convex but we allow the single functions g_i to be nonconvex.

The Mixed-Integer NON Linear case

- We consider here a Mixed Integer Non Linear Program of the form:

$$\text{MINLP} \begin{cases} \min f(x,y) \\ s.t.: \\ g(x,y) \le b \\ x \in \mathbb{Z}^{n1} \\ y \in \mathbb{R}^{n2} \end{cases}$$

  where f is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}$ and g is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}^m$. We assume the feasible region $g(x,y) \le b$ to be convex but we allow the single functions g_i to be nonconvex.

- The Outer Approximation (`OA`) technique linearizes the constraints of the continuous relaxation of MINLP to build a mixed integer linear relaxation of MINLP [Duran & Grossmann, 1986].

The Mixed-Integer NON Linear case

- We consider here a Mixed Integer Non Linear Program of the form:

$$\text{MINLP} \begin{cases} \min f(x,y) \\ s.t.: \\ g(x,y) \leq b \\ x \in \mathbb{Z}^{n1} \\ y \in \mathbb{R}^{n2} \end{cases}$$

  where f is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}$ and g is a function from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}^m$. We assume the feasible region $g(x,y) \leq b$ to be convex but we allow the single functions g_i to be nonconvex.

- The Outer Approximation (`OA`) technique linearizes the constraints of the continuous relaxation of MINLP to build a mixed integer linear relaxation of MINLP [Duran & Grossmann, 1986].

- The idea [Bonami, Cornuéjols, Lodi & Margot 2009] is to combine such a linearization technique with a FP-type algorithm to obtain feasible solutions for MINLPs. Let us denote such an algorithm as `FP-NLP`.

FP-NLP: the basic scheme

- We start by any feasible solution of the continuous relaxation of MINLP, $(\overline{x}^0, \overline{y}^0)$.

`FP-NLP`: the basic scheme

- We start by any feasible solution of the continuous relaxation of MINLP, $(\overline{x}^0, \overline{y}^0)$.
- In each iteration $i \geq 1$, we compute $(\hat{x}^i, \hat{y}^i)$ by finding the closest point to $\overline{x}^{i-1}$ (using the L_1 norm) on the current outer approximation $(OA)^i$ of the problem

FP-NLP: the basic scheme

- We start by any feasible solution of the continuous relaxation of MINLP, $(\overline{x}^0, \overline{y}^0)$.
- In each iteration $i \geq 1$, we compute $(\hat{x}^i, \hat{y}^i)$ by finding the closest point to $\overline{x}^{i-1}$ (using the L_1 norm) on the current outer approximation $(OA)^i$ of the problem

$$
(\text{FOA})^i \quad \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ g(\overline{x}^k, \overline{y}^k) + J_g(\overline{x}^k, \overline{y}^k) \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b \quad \forall k = 0, \dots, i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}
$$

FP-NLP: the basic scheme

- We start by any feasible solution of the continuous relaxation of MINLP, $(\overline{x}^0, \overline{y}^0)$.
- In each iteration $i \geq 1$, we compute $(\hat{x}^i, \hat{y}^i)$ by finding the closest point to $\overline{x}^{i-1}$ (using the L_1 norm) on the current outer approximation $(OA)^i$ of the problem

$$
(\text{FOA})^i \quad \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ g(\overline{x}^k, \overline{y}^k) + J_g(\overline{x}^k, \overline{y}^k)\left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix}\right) \leq b \quad \forall k = 0, \ldots, i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}
$$

- We then compute the next $(\overline{x}^i, \overline{y}^i)$ by solving the NLP

FP-NLP: the basic scheme

- We start by any feasible solution of the continuous relaxation of MINLP, $(\overline{x}^0, \overline{y}^0)$.
- In each iteration $i \geq 1$, we compute $(\hat{x}^i, \hat{y}^i)$ by finding the closest point to $\overline{x}^{i-1}$ (using the L_1 norm) on the current outer approximation $(OA)^i$ of the problem

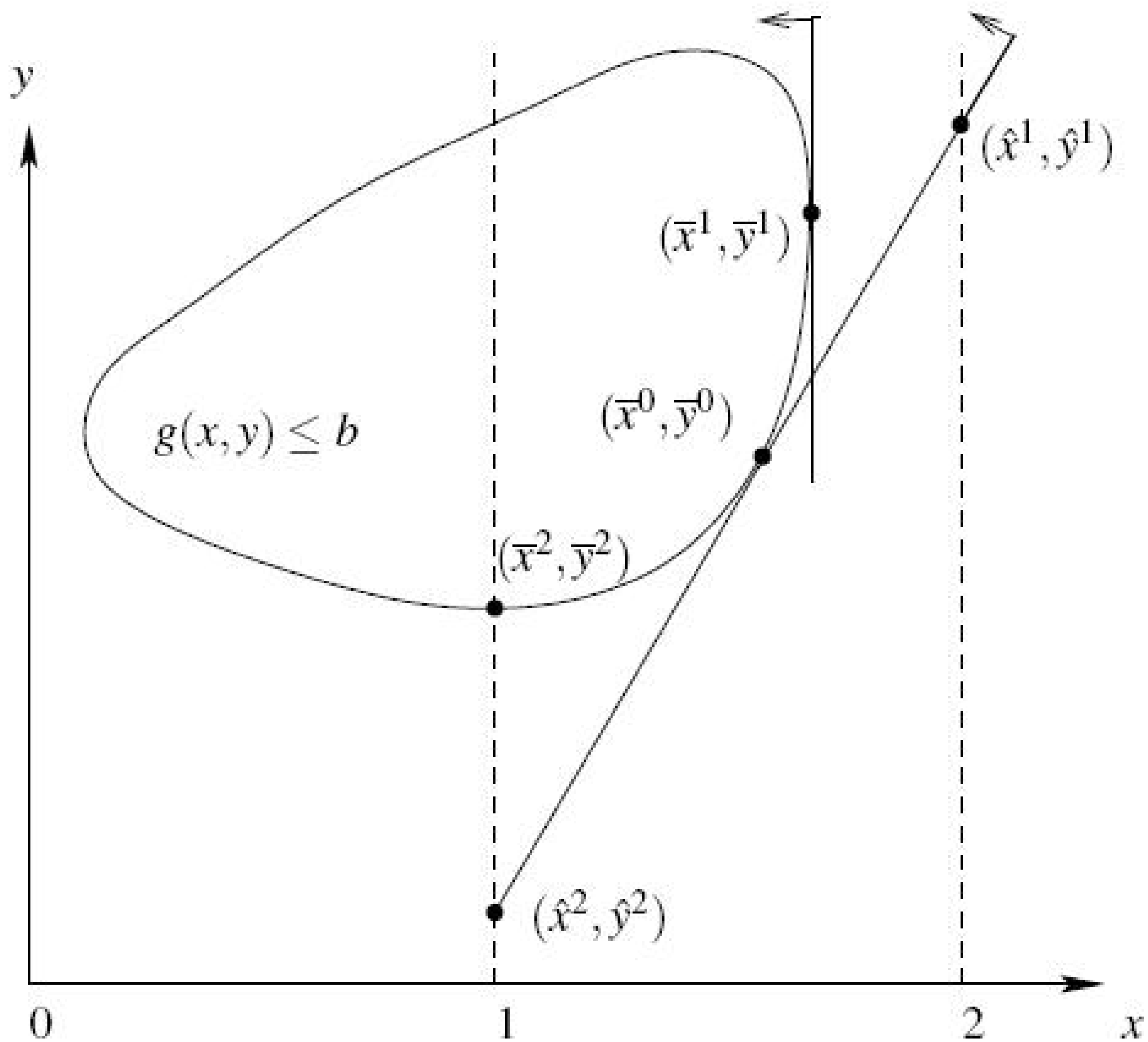
$$
(\text{FOA})^i \quad \begin{cases} \min \|x - \overline{x}^{i-1}\|_1 \\ g(\overline{x}^k, \overline{y}^k) + J_g(\overline{x}^k, \overline{y}^k) \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \overline{x}^k \\ \overline{y}^k \end{pmatrix} \right) \leq b \quad \forall k = 0, \dots, i-1 \\ x \in \mathbb{Z}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}
$$

- We then compute the next $(\overline{x}^i, \overline{y}^i)$ by solving the NLP

$$
(\text{FP-NLP})^i \quad \begin{cases} \min \|x - \hat{x}^i\|_2 \\ g(x, y) \leq b \\ x \in \mathbb{R}^{n_1} \\ y \in \mathbb{R}^{n_2}. \end{cases}
$$

- The procedure iterates between solving (FOA)^i and (FP-NLP)^i until either a feasible solution of MINLP is found or (FOA)^i becomes infeasible.

FP-NLP: a pictorial explanation



FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

  Thm: *In the basic FP, let* $(\hat{x}^i, \hat{y}^i)$ *be an optimal solution of (FOA)*^{i} *and* $(\overline{x}^i, \overline{y}^i)$ *an optimal solution of (FP-NLP)*^{i}*. If the constraint qualification for* $(FP - NLP)^i$ *holds at* $(\overline{x}^i, \overline{y}^i)$*, then* $\overline{x}^i \neq \overline{x}^k$ *for all* $k = 0, \ldots, i-1$*.*

FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

  Thm: *In the basic FP, let* $(\hat{x}^i, \hat{y}^i)$ *be an optimal solution of (FOA)*^{i} *and* $(\overline{x}^i, \overline{y}^i)$ *an optimal solution of (FP-NLP)*^{i}*. If the constraint qualification for* $(FP - NLP)^i$ *holds at* $(\overline{x}^i, \overline{y}^i)$*, then* $\overline{x}^i \neq \overline{x}^k$ *for all* $k = 0, \ldots, i-1$*.*

- Otherwise, when the point $(\overline{x}^i, \overline{y}^i)$ is not integer feasible and does not satisfy the constraint qualifications of (FP-NLP)^i, let $h(x) = ||x - \hat{x}^i||_2$.

FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

  Thm: *In the basic FP, let* $(\hat{x}^i, \hat{y}^i)$ *be an optimal solution of (FOA)*^{i} *and* $(\overline{x}^i, \overline{y}^i)$ *an optimal solution of (FP-NLP)*^{i}*. If the constraint qualification for* $(FP - NLP)^i$ *holds at* $(\overline{x}^i, \overline{y}^i)$*, then* $\overline{x}^i \neq \overline{x}^k$ *for all* $k = 0, \ldots, i-1$*.*

- Otherwise, when the point $(\overline{x}^i, \overline{y}^i)$ is not integer feasible and does not satisfy the constraint qualifications of (FP-NLP)^i, let $h(x) = ||x - \hat{x}^i||_2$.
- Then, we add, at iteration k, the following inequality to (FOA)^i:

$$(\overline{x}^k - \hat{x}^k)^T (x - \overline{x}^k) \geq 0 \tag{8}$$

FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

  Thm: *In the basic FP, let* $(\hat{x}^i, \hat{y}^i)$ *be an optimal solution of (FOA)*^{i} *and* $(\overline{x}^i, \overline{y}^i)$ *an optimal solution of (FP-NLP)*^{i}*. If the constraint qualification for* $(FP-NLP)^i$ *holds at* $(\overline{x}^i, \overline{y}^i)$*, then* $\overline{x}^i \neq \overline{x}^k$ *for all* $k = 0, \ldots, i-1$*.*

- Otherwise, when the point $(\overline{x}^i, \overline{y}^i)$ is not integer feasible and does not satisfy the constraint qualifications of (FP-NLP)^i, let $h(x) = ||x - \hat{x}^i||_2$.
- Then, we add, at iteration k, the following inequality to (FOA)^i:

$$(\overline{x}^k - \hat{x}^k)^T (x - \overline{x}^k) \geq 0 \tag{8}$$

- This inequality is valid for the continuous relaxation of MINLP and therefore can be added in the outer approximation constraints in (FOA)^i.

FP-NLP: theoretical results

- The next theorem shows that if constraint qualifications hold at each point $(\overline{x}^i, \overline{y}^i)$ the method cannot cycle:

  Thm: *In the basic FP, let* $(\hat{x}^i, \hat{y}^i)$ *be an optimal solution of (FOA)*^{i} *and* $(\overline{x}^i, \overline{y}^i)$ *an optimal solution of (FP-NLP)*^{i}*. If the constraint qualification for* $(FP-NLP)^i$ *holds at* $(\overline{x}^i, \overline{y}^i)$*, then* $\overline{x}^i \neq \overline{x}^k$ *for all* $k = 0, \ldots, i-1$*.*

- Otherwise, when the point $(\overline{x}^i, \overline{y}^i)$ is not integer feasible and does not satisfy the constraint qualifications of (FP-NLP)^i, let $h(x) = ||x - \hat{x}^i||_2$.
- Then, we add, at iteration k, the following inequality to (FOA)^i:

$$(\overline{x}^k - \hat{x}^k)^T (x - \overline{x}^k) \geq 0 \tag{8}$$

- This inequality is valid for the continuous relaxation of MINLP and therefore can be added in the outer approximation constraints in (FOA)^i.
- Then, we can prove that:

  Thm: *If the integer variables* x *are bounded, the algorithm enhanced by constraints (8) terminates in a finite number of iterations.*