The feasibility pump

Matteo Fischetti
University of Padova, Italy
matteo.fischetti@unipd.it

Fred W. Glover
University of Colorado at Boulder, USA
fred.glover@colorado.edu

Andrea Lodi
University of Bologna, Italy
alodi@deis.unibo.it

Aussois, January 2004

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.
- However, the exact solution of the resulting models often cannot be carried out for the problem sizes of interest in real-world applications, hence one is interested in effective heuristic methods.

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.
- However, the exact solution of the resulting models often cannot be carried out for the problem sizes of interest in real-world applications, hence one is interested in effective heuristic methods.
- Moreover, in some important practical cases, state-of-the-art MIP solvers may spend a very large computational effort before initializing their incumbent solution.

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.
- However, the exact solution of the resulting models often cannot be carried out for the problem sizes of interest in real-world applications, hence one is interested in effective heuristic methods.
- Moreover, in some important practical cases, state-of-the-art MIP solvers may spend a very large computational effort before initializing their incumbent solution.
- We concentrate on heuristic methods to find a feasible solution for hard MIPs which are of paramount important in practice.

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.
- However, the exact solution of the resulting models often cannot be carried out for the problem sizes of interest in real-world applications, hence one is interested in effective heuristic methods.
- Moreover, in some important practical cases, state-of-the-art MIP solvers may spend a very large computational effort before initializing their incumbent solution.
- We concentrate on heuristic methods to find a feasible solution for hard MIPs which are of paramount important in practice.
- This issue became even more important in the recent years, due to the success of local-search approaches for general MIPs such as *local branching* [Fischetti & Lodi, 2002]

  and *RINS* and *guided dives* [Danna, Rothberg, Le Pape, 2003]

  Indeed, these methods can only be applied if an initial feasible solution is known.

Motivation

- Mixed-integer linear programming plays a central role in modeling difficult-to-solve (NP-hard) combinatorial problems.
- However, the exact solution of the resulting models often cannot be carried out for the problem sizes of interest in real-world applications, hence one is interested in effective heuristic methods.
- Moreover, in some important practical cases, state-of-the-art MIP solvers may spend a very large computational effort before initializing their incumbent solution.
- We concentrate on heuristic methods to find a feasible solution for hard MIPs which are of paramount important in practice.
- This issue became even more important in the recent years, due to the success of local-search approaches for general MIPs such as *local branching* [Fischetti & Lodi, 2002]

  and *RINS* and *guided dives* [Danna, Rothberg, Le Pape, 2003]

  Indeed, these methods can only be applied if an initial feasible solution is known.

Hence: the earlier a feasible solution is found, the better!

The basic scheme

- How do you define feasibility for a MIP problem of the form:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}\} \quad ?$$

The basic scheme

- How do you define feasibility for a MIP problem of the form:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}\} \quad ?$$

- We propose the following definition:

The basic scheme

- How do you define feasibility for a MIP problem of the form:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}\} \quad ?$$

- We propose the following definition:

  a feasible solution is a point $x^* \in P := \{x : Ax \geq b\}$ s.t. is coincident with its rounding $\widetilde{x}$

The basic scheme

- How do you define feasibility for a MIP problem of the form:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}\} \quad ?$$

- We propose the following definition:

  a feasible solution is a point $x^* \in P := \{x : Ax \geq b\}$ s.t. is coincident with its rounding $\widetilde{x}$

  where:

  1. $[\cdot]$ represents scalar rounding to the nearest integer;
  2. $\widetilde{x}_j := [x^*_j]$ if $j \in \mathcal{I}$; and
  3. $\widetilde{x}_j := x^*_j$ otherwise.

The basic scheme

- How do you define feasibility for a MIP problem of the form:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}\} \quad ?$$

- We propose the following definition:

  a feasible solution is a point $x^* \in P := \{x : Ax \geq b\}$ s.t. is coincident with its rounding $\widetilde{x}$

  where:

  1. $[\cdot]$ represents scalar rounding to the nearest integer;
  2. $\widetilde{x}_j := [x_j^*]$ if $j \in \mathcal{I}$; and
  3. $\widetilde{x}_j := x_j^*$ otherwise.

- Replacing coincident with as close as possible relatively to a suitable distance function $\Delta(x^*, \widetilde{x})$ suggests an iterative heuristic for finding a feasible solution of a given MIP.

The basic scheme (cont.d)

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

The basic scheme (cont.d)

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ which 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, is an easily solvable LP problem.

The basic scheme (cont.d)

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

- At each iteration we look for a point $x^* \in P$ which 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, is an easily solvable LP problem.

- If $\Delta(x^*, \widetilde{x}) = 0$, then x^* is a feasible MIP solution and we are done.

The basic scheme (cont.d)

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ which 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, is an easily solvable LP problem.

- If $\Delta(x^*, \widetilde{x}) = 0$, then x^* is a feasible MIP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

The basic scheme (cont.d)

- We start from any $x^* \in P$, and define its rounding $\widetilde{x}$.
- At each iteration we look for a point $x^* \in P$ which 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, is an easily solvable LP problem.

- If $\Delta(x^*, \widetilde{x}) = 0$, then x^* is a feasible MIP 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 $\widetilde{x}$ which satisfy feasibility in a complementary but partial way:

The basic scheme (cont.d)

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

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

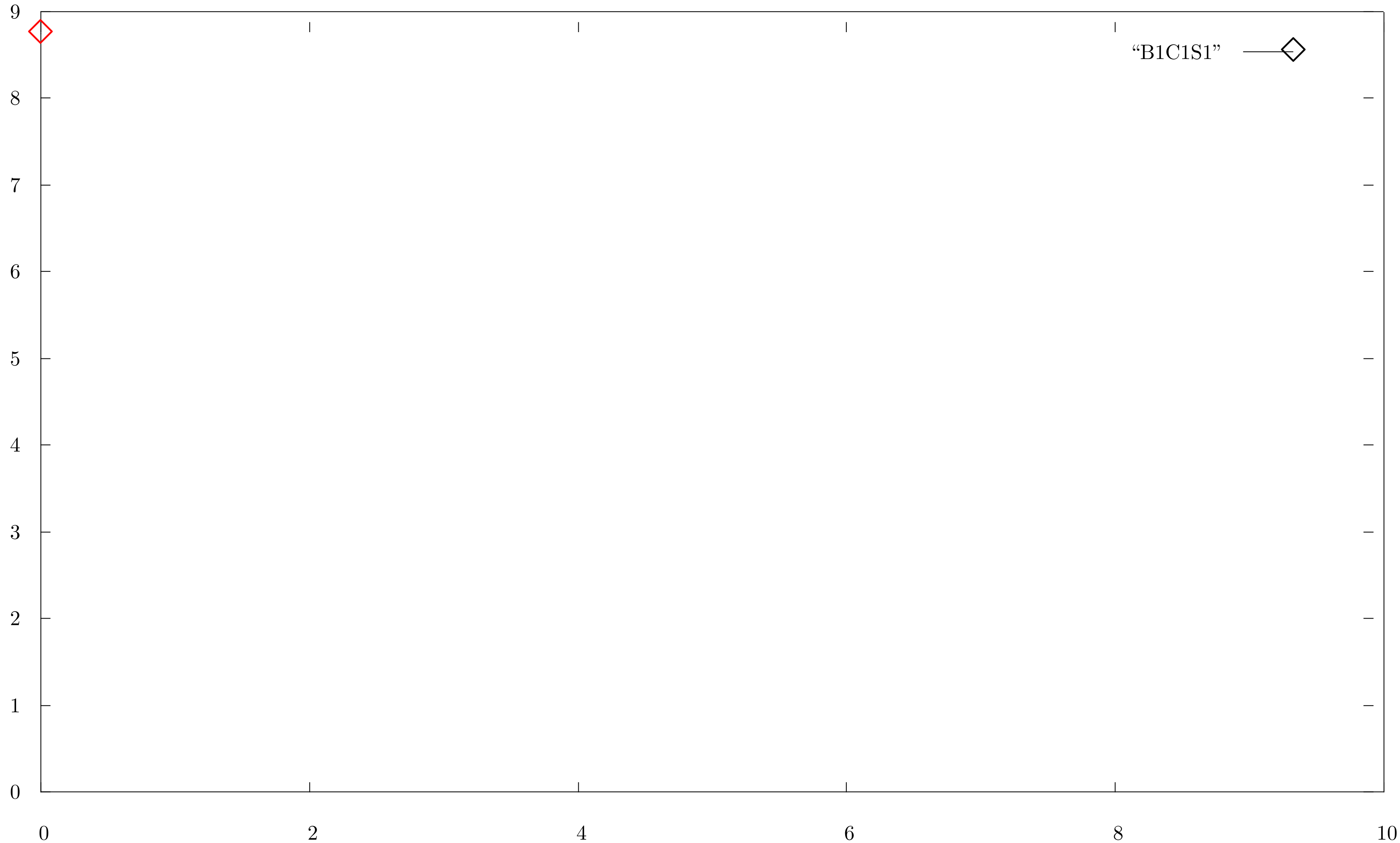
  Assuming $\Delta(x, \widetilde{x})$ is chosen appropriately, is an easily solvable LP problem.

- If $\Delta(x^*, \widetilde{x}) = 0$, then x^* is a feasible MIP solution and we are done.
- Otherwise, we replace $\widetilde{x}$ by the rounding of x^*, and repeat.

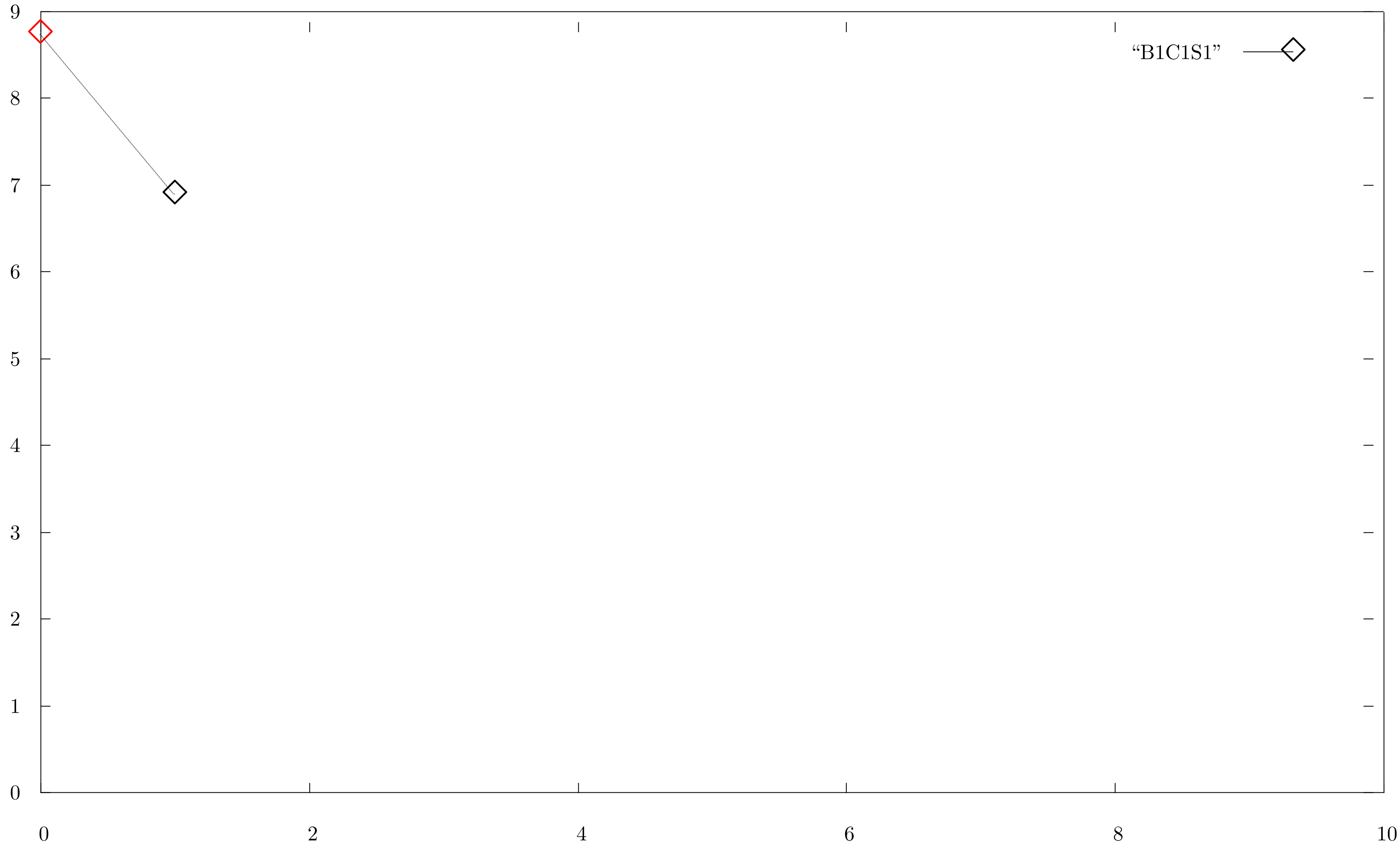
- From a geometric point of view, this simple heuristic generates two hopefully convergent trajectories of points x^* and $\widetilde{x}$ which satisfy feasibility in a complementary but partial way:

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

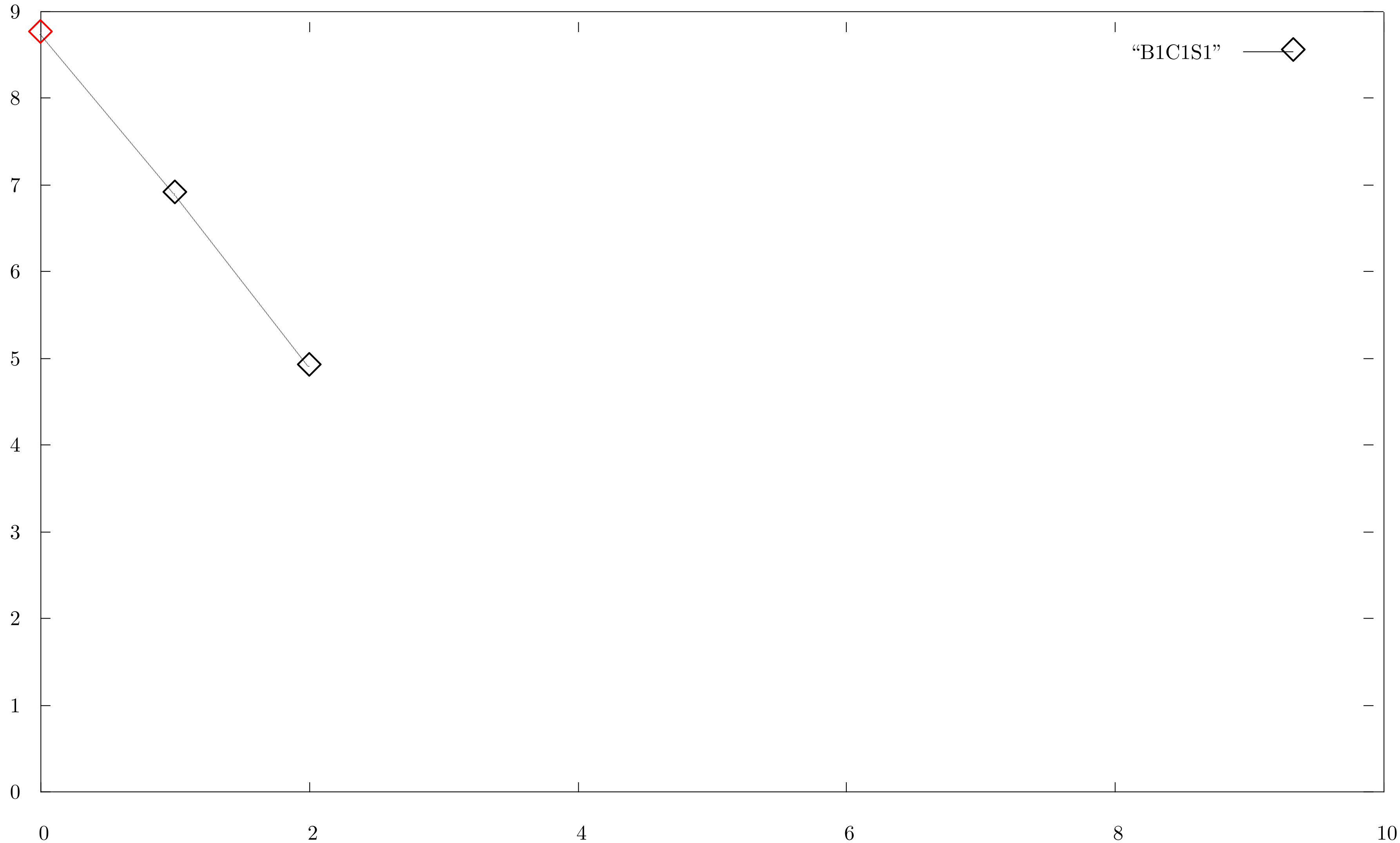
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



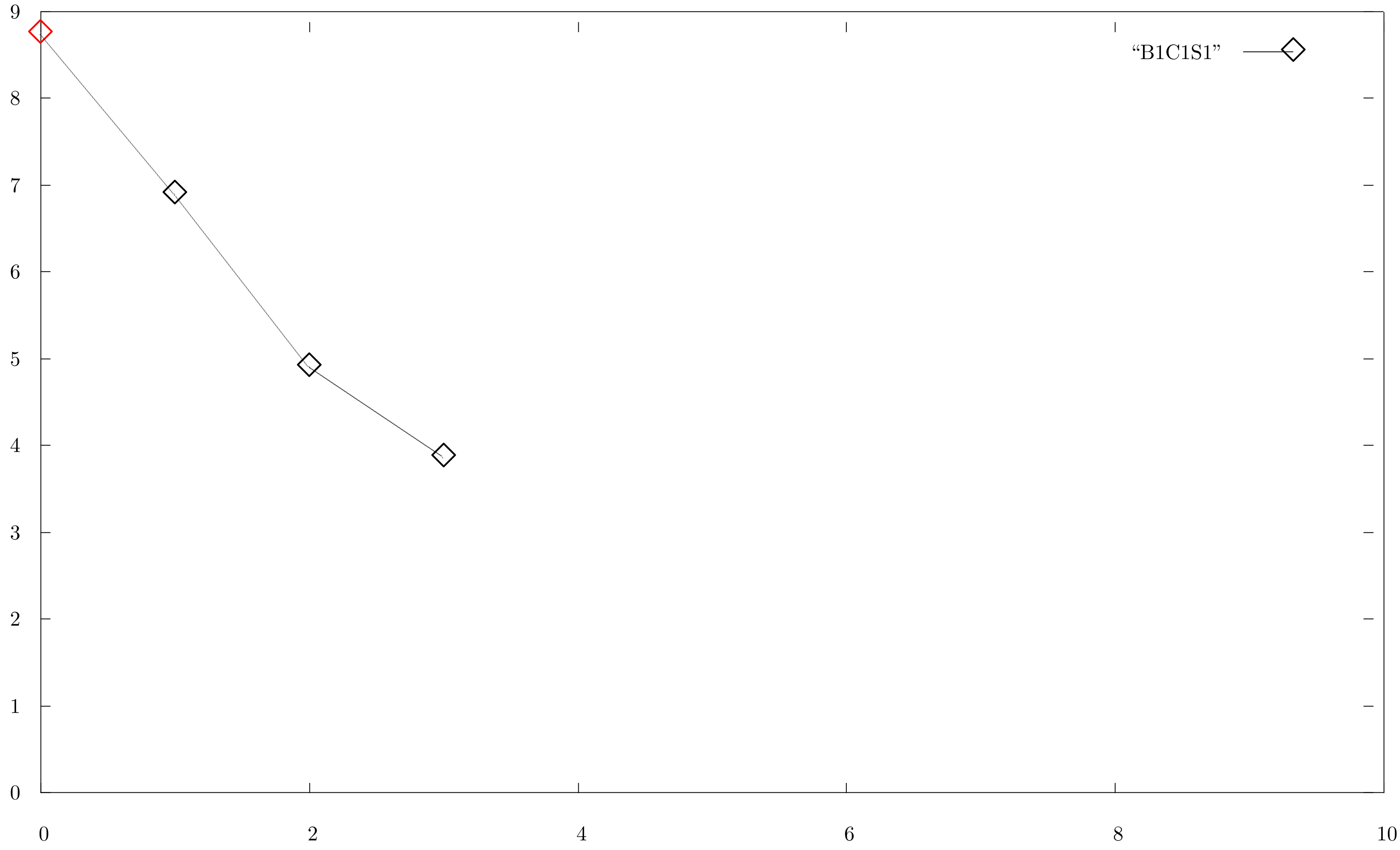
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



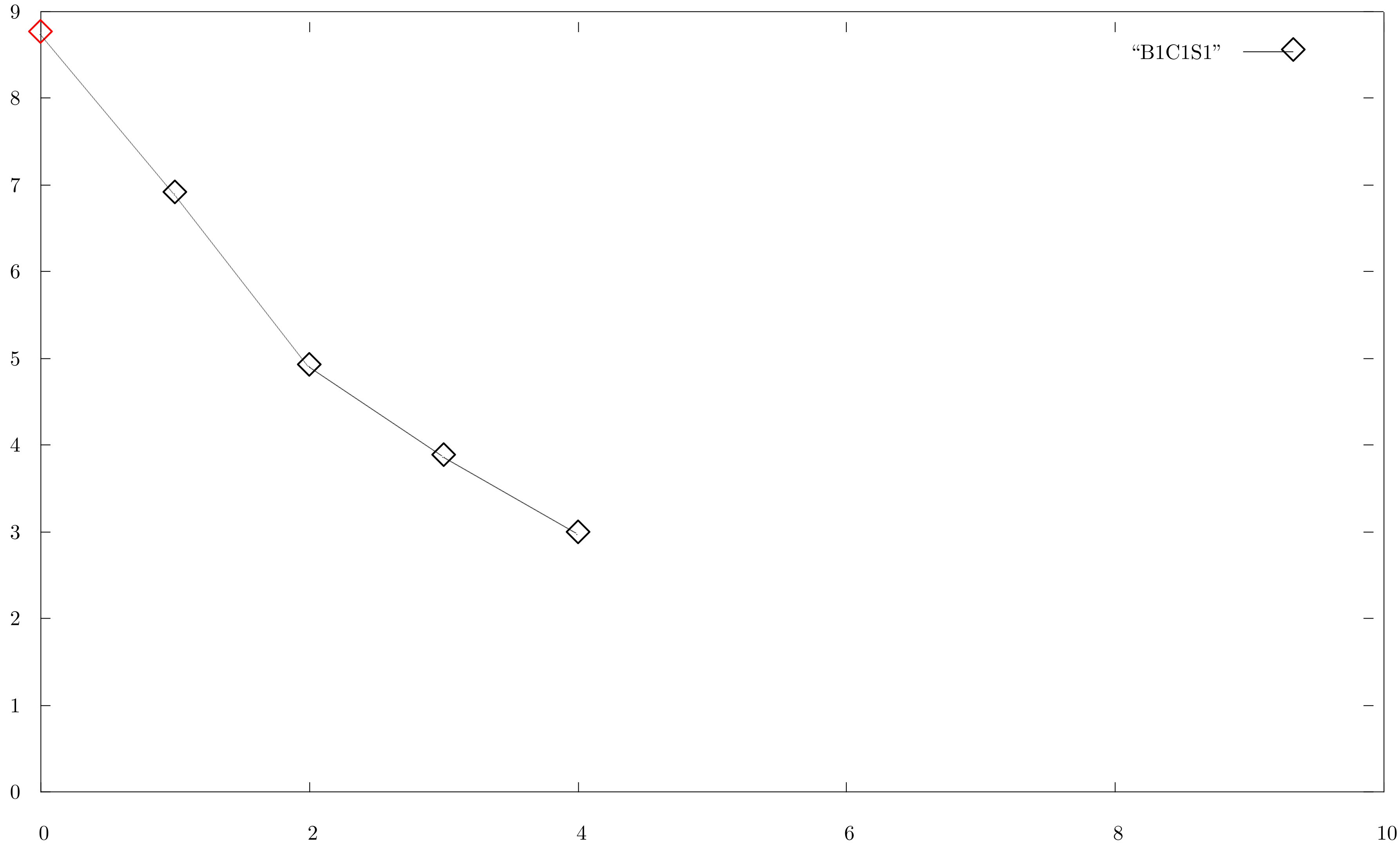
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



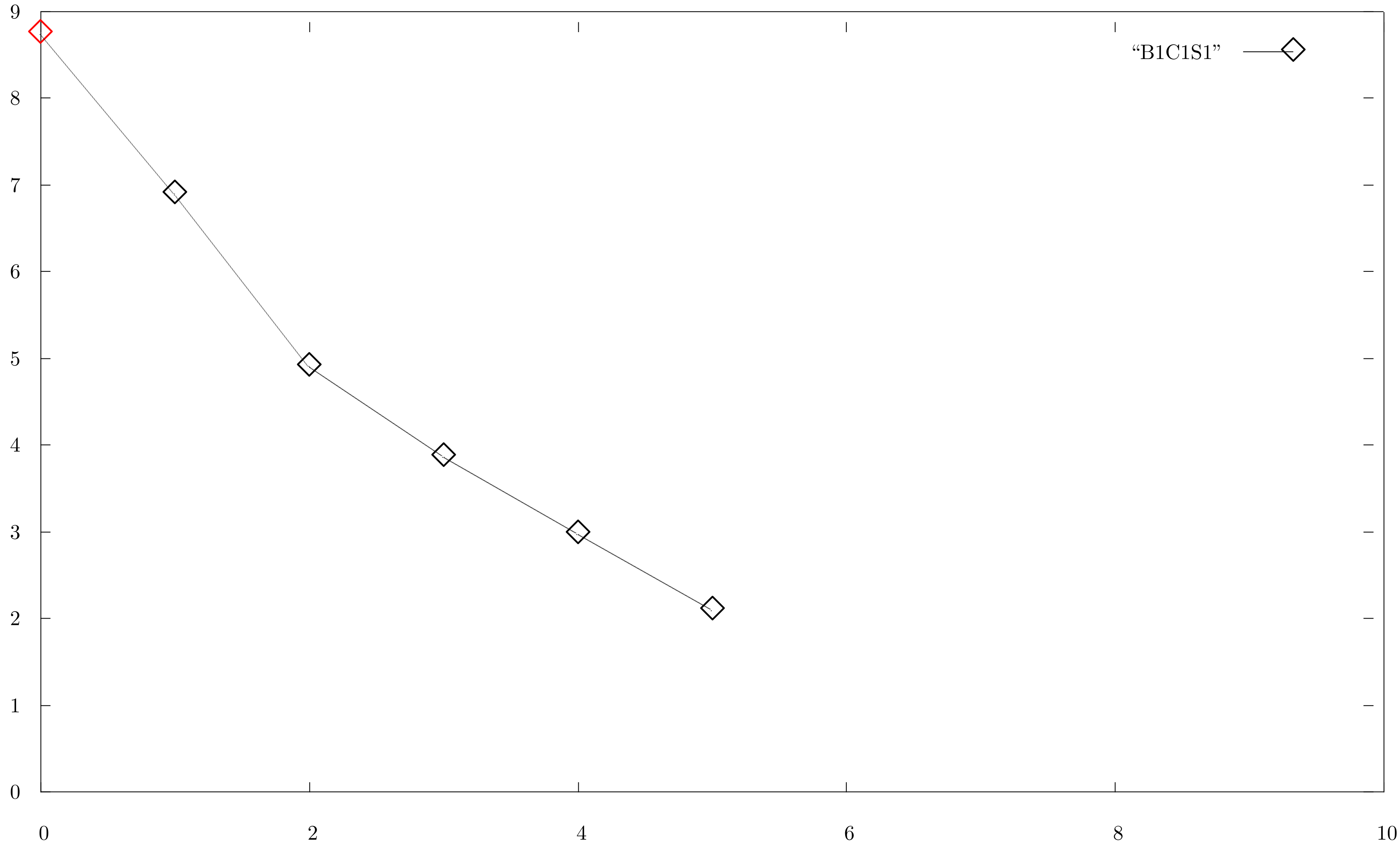
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



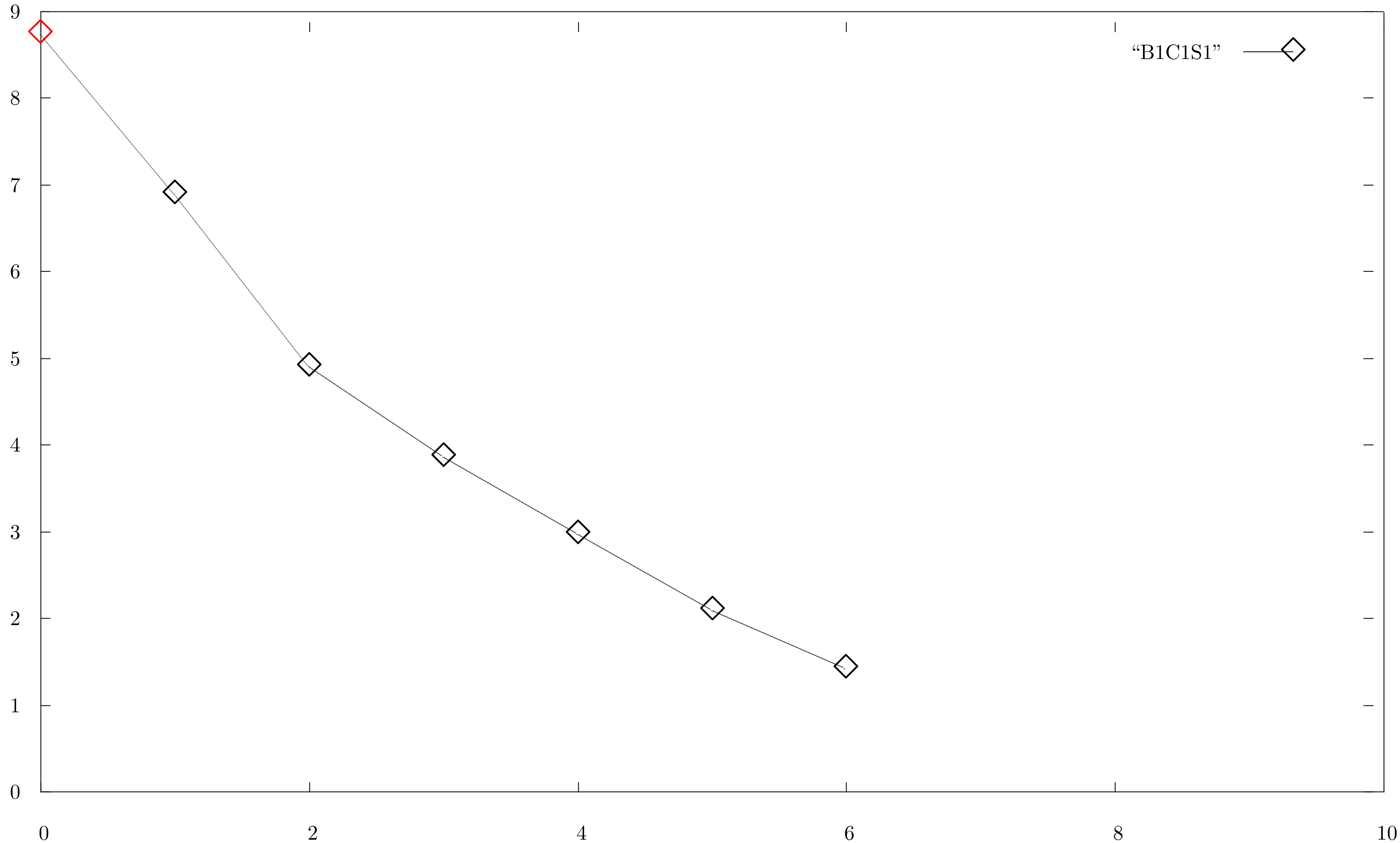
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



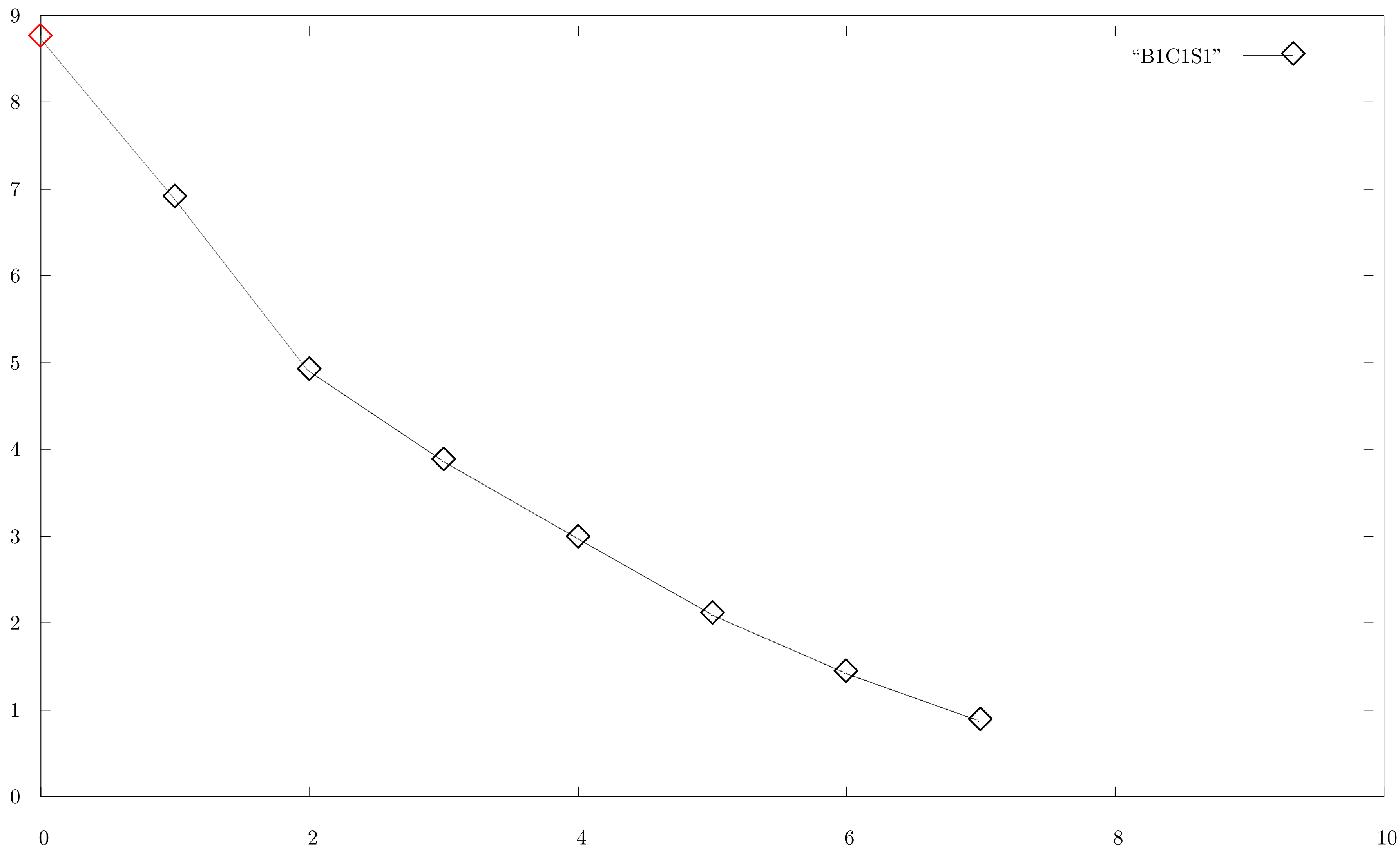
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



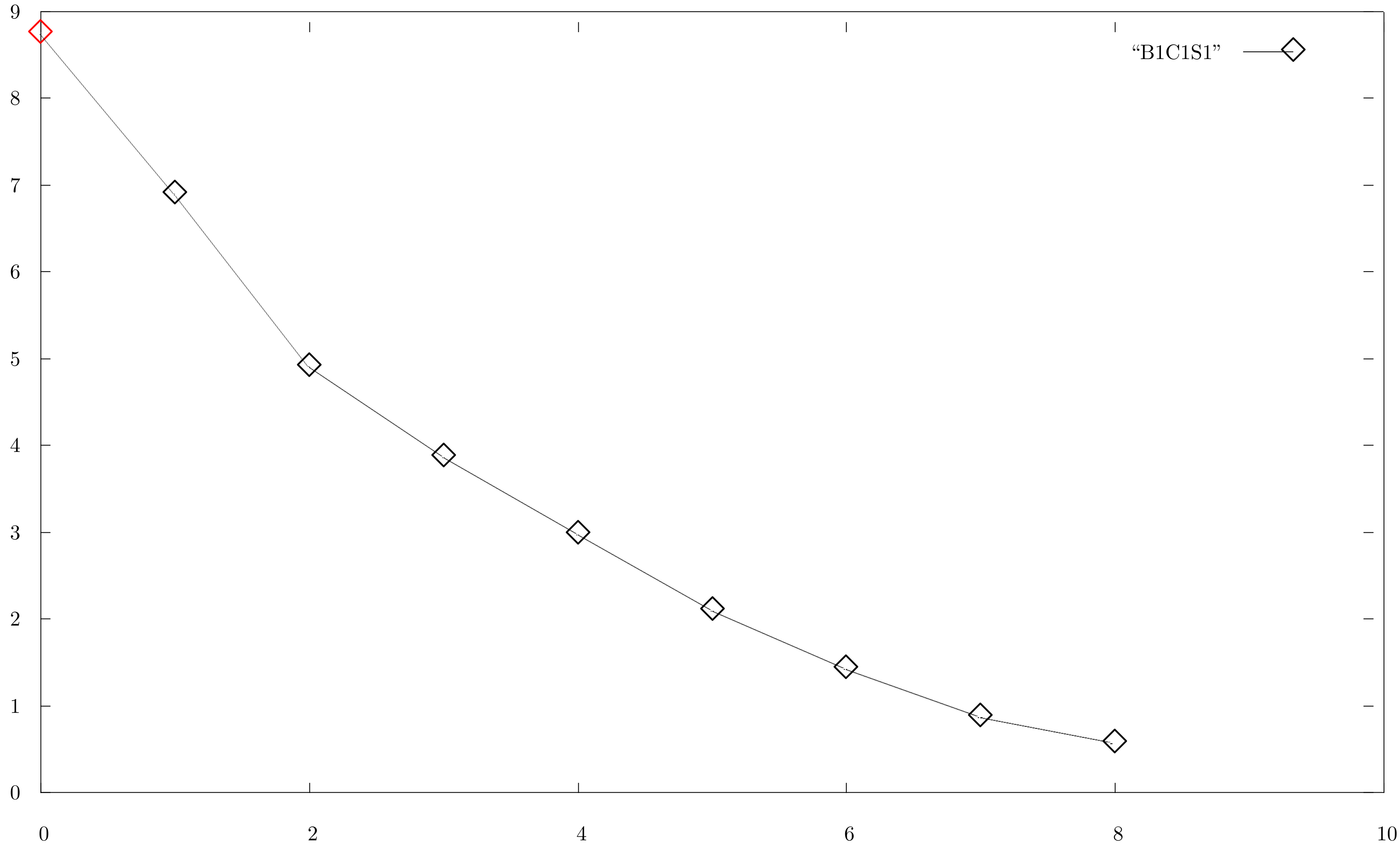
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



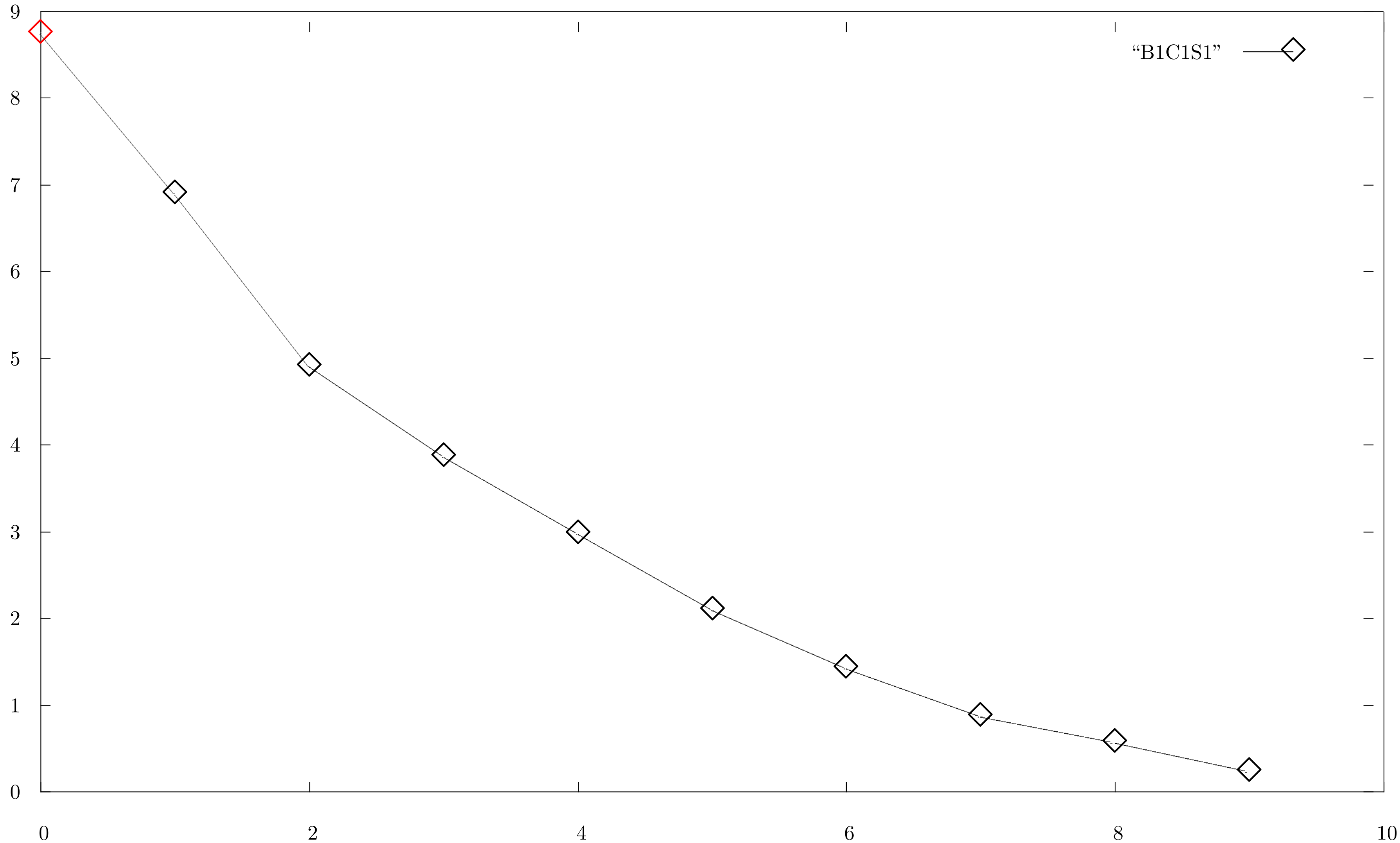
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



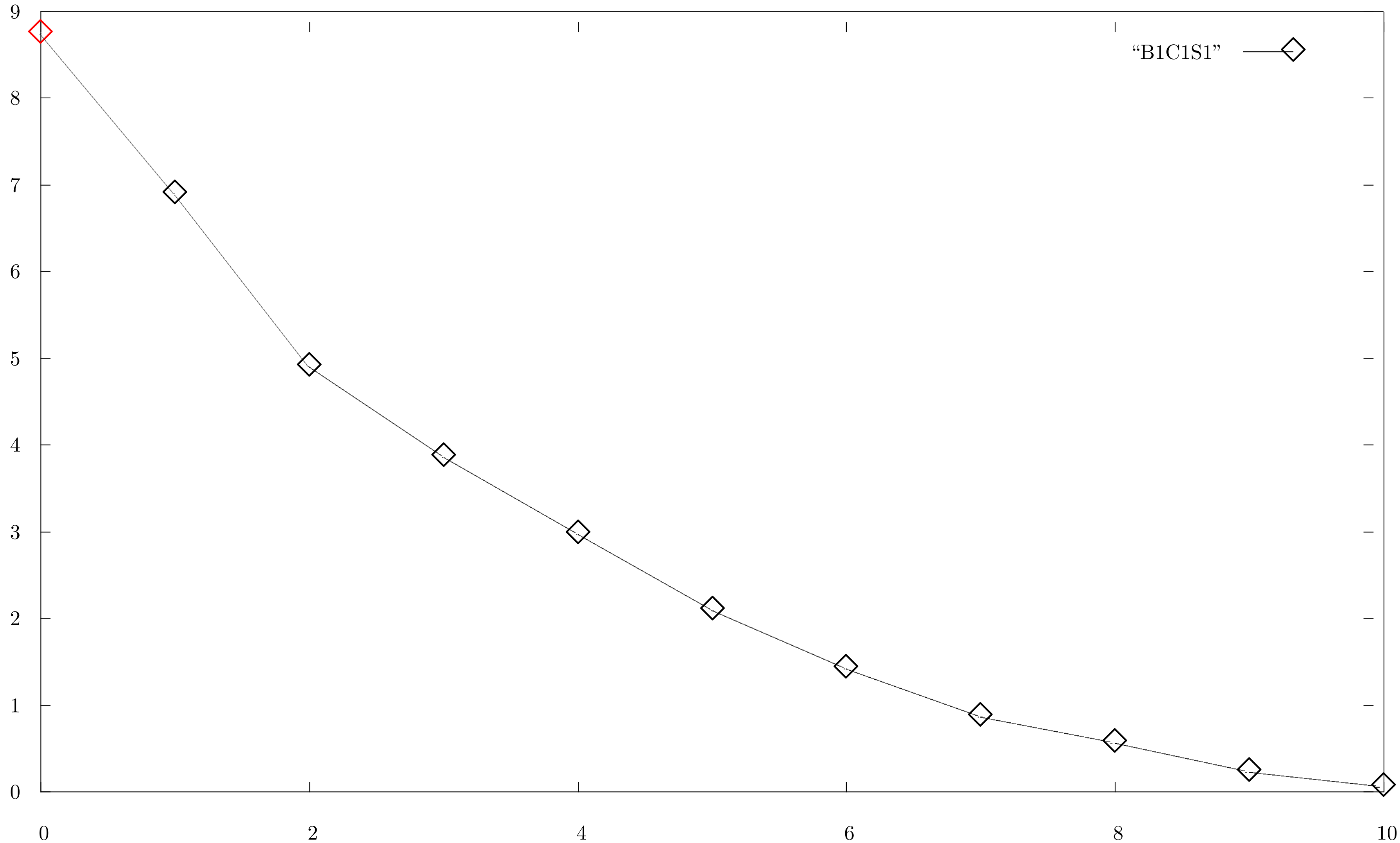
Plot of the infeasibility measure $\Delta(x^*, \widetilde{x})$ at each iteration



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



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



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

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

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

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

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

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

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

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

- If w.l.o.g. MIP constraints include the bounds $l_j \leq x_j \leq u_j, \ \forall \ j \in \mathcal{I}$, we can write:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I}: \widetilde{x}_j = l_j} (x_j - l_j) + \sum_{j \in \mathcal{I}: \widetilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in \mathcal{I}: l_j < \widetilde{x}_j < u_j} (x_j^+ + x_j^-)$$

  where the additional variables x_j^+ and x_j^- require the additional constraints:

$$x_j = \widetilde{x}_j + x_j^+ - x_j^-, \quad x_j^+ \geq 0, \ x_j^- \geq 0, \quad \forall j \in \mathcal{I} : l_j < \widetilde{x}_j < u_j \tag{1}$$

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

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

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

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

- If w.l.o.g. MIP constraints include the bounds $l_j \le x_j \le u_j, \ \forall \ j \in \mathcal{I}$, we can write:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I}: \widetilde{x}_j = l_j} (x_j - l_j) + \sum_{j \in \mathcal{I}: \widetilde{x}_j = u_j} (u_j - x_j) + \sum_{j \in \mathcal{I}: l_j < \widetilde{x}_j < u_j} (x_j^+ + x_j^-)$$

  where the additional variables x_j^+ and x_j^- require the additional constraints:

$$x_j = \widetilde{x}_j + x_j^+ - x_j^-, \quad x_j^+ \ge 0, \ x_j^- \ge 0, \quad \forall j \in \mathcal{I} : l_j < \widetilde{x}_j < u_j \qquad (1)$$

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

$$\min\{\Delta(x, \widetilde{x}) : Ax \ge b, \ (1)\ \} \qquad (2)$$

Definition of $\Delta(x^*, \widetilde{x})$ (cont.d)

- When all integer-constrained variables are binary (again $Ax \geq b$ include $0 \leq x_j \leq 1, \ \forall j \in \mathcal{I}$) no additional variables x_j^+ and x_j^- (1) are required in the definition of $\Delta(x^*, \widetilde{x})$, which attains the simpler form:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{I}: \widetilde{x}_j = 1} (1 - x_j) \tag{3}$$

Definition of $\Delta(x^*, \widetilde{x})$ (cont.d)

- When all integer-constrained variables are binary (again $Ax \geq b$ include $0 \leq x_j \leq 1, \ \forall j \in \mathcal{I}$) no additional variables x_j^+ and x_j^- (1) are required in the definition of $\Delta(x^*, \widetilde{x})$, which attains the simpler form:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{I}: \widetilde{x}_j = 1} (1 - x_j) \tag{3}$$

- An important feature of the method is related to the infeasibility measure used to guide $\widetilde{x}$ towards feasibility: instead of taking a weighted combination of the degree of violation of the single linear constraints, as is customary in MIP heuristics, we use the distance $\Delta(x^*, \widetilde{x})$ of $\widetilde{x}$ from polyhedron P.

Definition of $\Delta(x^*, \widetilde{x})$ (cont.d)

- When all integer-constrained variables are binary (again $Ax \geq b$ include $0 \leq x_j \leq 1, \ \forall j \in \mathcal{I}$) no additional variables x_j^+ and x_j^- (1) are required in the definition of $\Delta(x^*, \widetilde{x})$, which attains the simpler form:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I}: \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{I}: \widetilde{x}_j = 1} (1 - x_j) \qquad (3)$$

- An important feature of the method is related to the infeasibility measure used to guide $\widetilde{x}$ towards feasibility: instead of taking a weighted combination of the degree of violation of the single linear constraints, as is customary in MIP heuristics, we use the distance $\Delta(x^*, \widetilde{x})$ of $\widetilde{x}$ from polyhedron P.

- This distance can be interpreted as a sort of 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^*.

Definition of $\Delta(x^*, \widetilde{x})$ (cont.d)

- When all integer-constrained variables are binary (again $Ax \geq b$ include $0 \leq x_j \leq 1, \ \forall j \in \mathcal{I}$) no additional variables x_j^+ and x_j^- (1) are required in the definition of $\Delta(x^*, \widetilde{x})$, which attains the simpler form:

$$\Delta(x, \widetilde{x}) := \sum_{j \in \mathcal{I} : \widetilde{x}_j = 0} x_j + \sum_{j \in \mathcal{I} : \widetilde{x}_j = 1} (1 - x_j) \qquad (3)$$

- An important feature of the method is related to the infeasibility measure used to guide $\widetilde{x}$ towards feasibility: instead of taking a weighted combination of the degree of violation of the single linear constraints, as is customary in MIP heuristics, we use the distance $\Delta(x^*, \widetilde{x})$ of $\widetilde{x}$ from polyhedron P.

- This distance can be interpreted as a sort of 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^*.

- Hence the name of the heuristic: feasibility pump (FP).

A first FP 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 $x^*_j,\ j \in \mathcal{I}$ chosen so as to minimize the increase in the current value of $\Delta(x^*, \widetilde{x})$.

A first FP implementation

- MAIN PROBLEM, stalling issues:

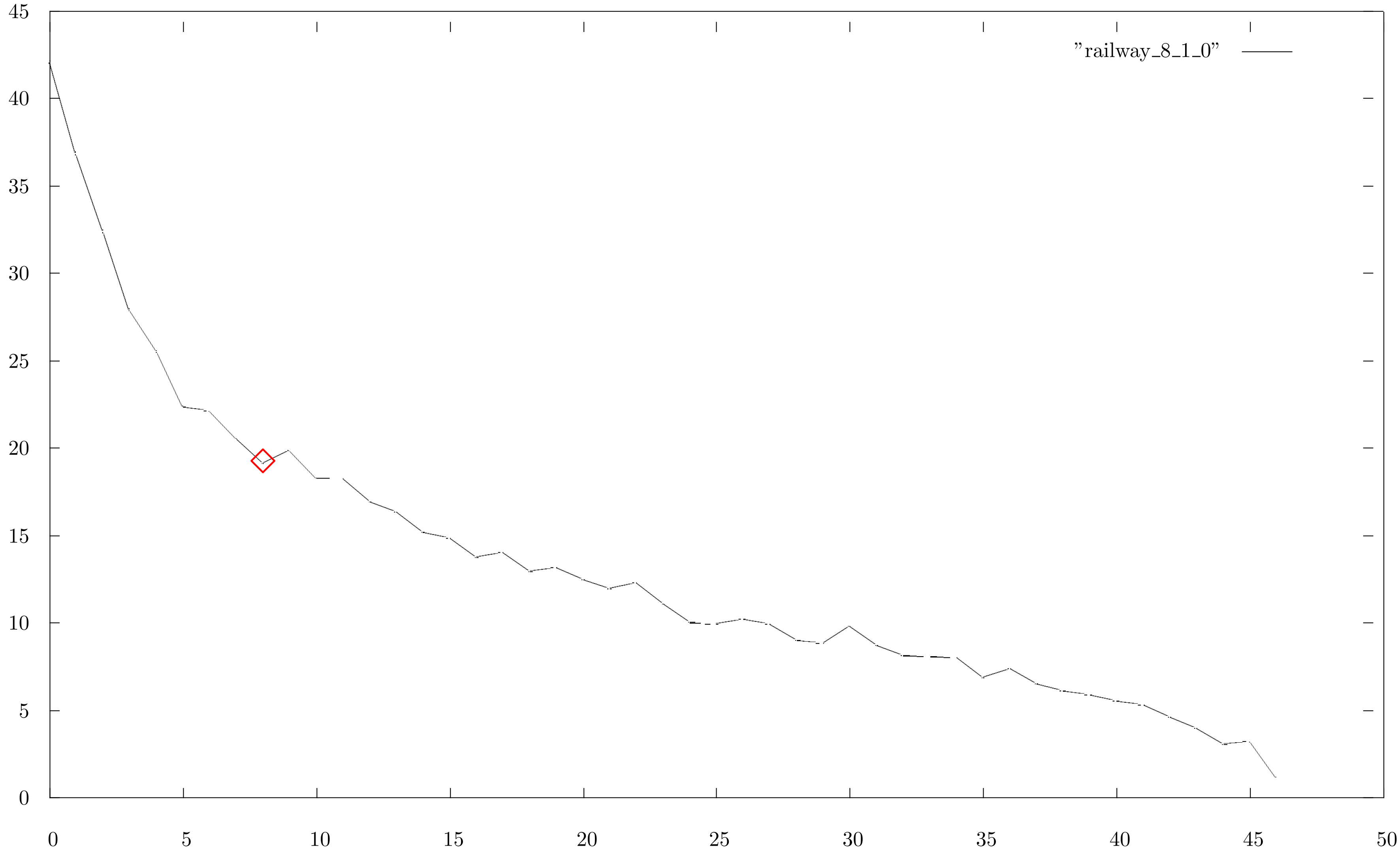
  as soon as $\Delta(x^*, \widetilde{x})$ is not reduced when replacing $\widetilde{x}$ by x^*.

  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 $x^*_j,\ j \in \mathcal{I}$ chosen so as to minimize the increase in the current value of $\Delta(x^*, \widetilde{x})$.

initialize $x^* :=$ argmin$\{c^T x : Ax \geq b\}$ and $\widetilde{x} :=$ rounding of x^*;
nIter := 0;
while ($\Delta(x^*, \widetilde{x}) > 0$ and nIter < maxIter) do
  nIter := nIter$+1$;
  $x^* :=$ argmin$\{\Delta(x, \widetilde{x}) : Ax \geq b\}$;
  if $\Delta(x^*, \widetilde{x}) > 0$ then
    for each $j \in \mathcal{I}$ define the flip score $\sigma_j := |x^*_j - \widetilde{x}_j|$;
    flip all entries $\widetilde{x}_j$ with $j \in \mathcal{I} : \sigma_j > 0.5$, for a total of (say) k variables;
    if k < T, then flip the T-k new entries of $\widetilde{x}$ with highest score
  endif
enddo

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



FP and local branching

- The FP can also be viewed as modified *local branching* (LB) strategy [Fischetti & Lodi, 2002]

FP and local branching

- The FP can also be viewed as modified *local branching* (LB) strategy [Fischetti & Lodi, 2002]

- Indeed, at each pumping cycle we have an incumbent (infeasible) solution $\widetilde{x}$ satisfying the integer requirement, and we face the problem of finding a feasible solution (if any) within a small-distance neighborhood, i.e., changing only a small subset of its variables.

FP and local branching

- The FP can also be viewed as modified *local branching* (LB) strategy [Fischetti & Lodi, 2002]

- Indeed, at each pumping cycle we have an incumbent (infeasible) solution $\widetilde{x}$ satisfying the integer requirement, and we face the problem of finding a feasible solution (if any) within a small-distance neighborhood, i.e., changing only a small subset of its variables.

- In the LB context, this subproblem would have been modeled by the MIP:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}, (1),\ \Delta(x, \widetilde{x}) \leq k\}$$

for a suitable value of parameter k, and solved through an enumerative MIP method.

FP and local branching

- The FP can also be viewed as modified *local branching* (LB) strategy [Fischetti & Lodi, 2002]

- Indeed, at each pumping cycle we have an incumbent (infeasible) solution $\widetilde{x}$ satisfying the integer requirement, and we face the problem of finding a feasible solution (if any) within a small-distance neighborhood, i.e., changing only a small subset of its variables.

- In the LB context, this subproblem would have been modeled by the MIP:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}, (1), \ \Delta(x, \widetilde{x}) \leq k\}$$

  for a suitable value of parameter k, and solved through an enumerative MIP method.

- In the FP context, instead, the same subproblem is modeled in a relaxed way through the LP:

$$\min\{\Delta(x, \widetilde{x}) : Ax \geq b, (1)\}$$

  where the "small distance" requirement is translated in terms of objective function.

FP and local branching

- The FP can also be viewed as modified *local branching* (LB) strategy [Fischetti & Lodi, 2002]

- Indeed, at each pumping cycle we have an incumbent (infeasible) solution $\widetilde{x}$ satisfying the integer requirement, and we face the problem of finding a feasible solution (if any) within a small-distance neighborhood, i.e., changing only a small subset of its variables.

- In the LB context, this subproblem would have been modeled by the MIP:

$$\min\{c^T x : Ax \geq b, x_j \text{ integer } \forall j \in \mathcal{I}, (1), \ \Delta(x, \widetilde{x}) \leq k\}$$

  for a suitable value of parameter k, and solved through an enumerative MIP method.

- In the FP context, instead, the same subproblem is modeled in a relaxed way through the LP:

$$\min\{\Delta(x, \widetilde{x}) : Ax \geq b, (1)\}$$

  where the "small distance" requirement is translated in terms of objective function.

- Hypothesis: the objective function $\Delta(x, \widetilde{x})$ will discourage x^* for be too far from $\widetilde{x}$.

  Hence, we expect a large number of the integer-constrained (integer-valued) variables in $\widetilde{x}$ will maintain their value also in the optimal x^*.

Computational results

49 hard 0-1 MIPs - Pentium M 1.6 GHz notebook - ILOG-Cplex halted at the root node

	T = 10			ILOG-Cplex 8.1, emp=1		ILOG-Cplex 8.1 default	
Name	value	nIT	time	value	time	value	time
danoint	**N/A**	3	0.00	**N/A**	1.60	66.50	1.57
markshare1	70.00	0	0.00	710.00	0.01	710.00	0.00
markshare2	648.00	2	0.00	1,735.00	0.00	1,735.00	0.00
seymour	443.00	6	3.91	463.00	3.85	463.00	4.11
nsrand_ipx	336,000.00	2	0.68	62,560.00	0.76	62,560.00	0.76
van	7.68	3	986.93	5.09	3594.95	5.09	3594.95
biella1	3,400,802.15	3	11.99	**N/A**	10.40	**N/A**	37.00
dc1c	5,163,390.90	3	20.53	**N/A**	25.60	**N/A**	82.10
dc1l	17,055,833.44	3	155.57	751,003,858.46	75.20	751,003,858.46	73.71
dolom1	199,787,276.17	4	121.74	**N/A**	31.90	**N/A**	121.30
siena1	129,121,289.71	5	721.28	**N/A**	87.60	**N/A**	271.80
trento1	27,186,350.03	1	86.61	**N/A**	25.60	45,717,270.00	45.92
rail507	181.00	2	34.79	211.00	36.15	211.00	36.89
rail2536c	709.00	0	166.67	763.00	16.48	763.00	16.49
rail2586c	994.00	2	132.27	1,078.00	57.05	1,078.00	57.49
rail4284c	1,130.00	2	516.19	1,226.00	180.30	1,226.00	181.46
rail4872c	1,611.00	4	617.19	1,736.00	239.43	1,736.00	241.22

	T = 10			ILOG-Cplex 8.1, emp=1		ILOG-Cplex 8.1 default	
Name	value	nIT	time	value	time	value	time
A1C1S1	15,463.18	7	2.87	**N/A**	1.30	**N/A**	15.10
A2C1S1	17,503.02	5	2.26	20,865.33	0.09	20,865.33	0.09
B1C1S1	37,986.94	10	4.12	69,933.52	0.10	69,933.52	0.10
B2C1S1	43,716.58	9	4.77	70,575.52	0.13	70,575.52	0.13
tr12-30	261,826.00	11	0.11	**N/A**	0.30	140,084.00	2.11
sp97ar	1,187,905,237.44	3	4.66	729,774,537.92	3.93	729,774,537.92	3.98
sp97ic	834,114,625.76	1	2.17	495,919,360.00	2.19	495,919,360.00	2.26
sp98ar	873,197,861.44	2	4.34	604,367,012.64	4.05	604,367,012.64	4.10
sp98ic	795,108,323.36	1	1.84	542,322,911.84	1.77	542,322,911.84	1.79
blp-ic98	13,211.71	3	0.97	**N/A**	3.00	**N/A**	7.30
blp-ir98	5,659.48	1	0.27	**N/A**	1.30	**N/A**	3.20
berlin_5_8_0	76.00	14	0.22	**N/A**	0.30	**N/A**	0.80
railway_8_1_0	434.00	46	0.73	**N/A**	0.20	474.00	0.33
bg512142	120,738,665.00	0	0.18	120,670,203.50	0.29	120,670,203.50	0.29
dg012142	153,406,921.50	0	0.96	153,397,300.00	1.01	153,397,300.00	1.00
ljb2	7.24	0	0.05	**N/A**	0.20	1.69	0.43
ljb7	8.61	0	0.53	**N/A**	1.70	0.96	4.74
ljb9	9.48	0	0.72	**N/A**	2.10	9.48	5.57
ljb10	7.31	0	0.89	**N/A**	2.70	2.36	4.72
ljb12	6.20	0	0.70	**N/A**	2.10	6.20	6.03

Summary of the results (1)

- Over 37 hard 0-1 MIP instances:

  FP failed in finding a feasible solution only in 1 case, while

  ILOG-Cplex 8.1 (emp=1) failed 18 times, and

  ILOG-Cplex 8.1 (default) failed 8 times.

Summary of the results (1)

- Over 37 hard 0-1 MIP instances:

  FP failed in finding a feasible solution only in 1 case, while

  ILOG-Cplex 8.1 (emp=1) failed 18 times, and

  ILOG-Cplex 8.1 (default) failed 8 times.

- The quality of the solutions obtained is generally comparable, as well as the computing times.

Summary of the results (1)

- Over 37 hard 0-1 MIP instances:

  FP failed in finding a feasible solution only in 1 case, while

  ILOG-Cplex 8.1 (emp=1) failed 18 times, and

  ILOG-Cplex 8.1 (default) failed 8 times.

- The quality of the solutions obtained is generally comparable, as well as the computing times.
- There are still 12 0-1 MIPs on the testbed which cannot be solved by the three algorithms.

Summary of the results (1)

- Over 37 hard 0-1 MIP instances:

  FP failed in finding a feasible solution only in 1 case, while

  ILOG-Cplex 8.1 (emp=1) failed 18 times, and

  ILOG-Cplex 8.1 (default) failed 8 times.

- The quality of the solutions obtained is generally comparable, as well as the computing times.
- There are still 12 0-1 MIPs on the testbed which cannot be solved by the three algorithms.
- When ILOG-Cplex is not able to find a feasible solution obviously it resorts to branching, and it is then able to find a feasible solution:

  to all MIPs from a min of 30 to a max of 207,918 nodes for ILOG-Cplex (emp=1),

  to all but 3 MIPs from a min of 10 to a max of 37,320 nodes for ILOG-Cplex (default) within a time limit of 1,200 CPU seconds.

Summary of the results (1)

- Over 37 hard 0-1 MIP instances:

  FP failed in finding a feasible solution only in 1 case, while

  ILOG-Cplex 8.1 (emp=1) failed 18 times, and

  ILOG-Cplex 8.1 (default) failed 8 times.

- The quality of the solutions obtained is generally comparable, as well as the computing times.
- There are still 12 0-1 MIPs on the testbed which cannot be solved by the three algorithms.
- When ILOG-Cplex is not able to find a feasible solution obviously it resorts to branching, and it is then able to find a feasible solution:

  to all MIPs from a min of 30 to a max of 207,918 nodes for ILOG-Cplex (emp=1),

  to all but 3 MIPs from a min of 10 to a max of 37,320 nodes for ILOG-Cplex (default) within a time limit of 1,200 CPU seconds.

- Better results have been obtained by Ed Rothberg by avoiding preprocessing!!

Summary of the results (2)

- The problem with FP is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.

Summary of the results (2)

- The problem with FP is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.
- Simple anti-stalling and anti-cycling rules borrowed from metaheuristics can be implemented within the basic FP framework.

Summary of the results (2)

- The problem with `FP` is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.
- Simple anti-stalling and anti-cycling rules borrowed from metaheuristics can be implemented within the basic `FP` framework.
- Instead, we found extremely effective the idea of introducing some enumeration.

Summary of the results (2)

- The problem with FP is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.
- Simple anti-stalling and anti-cycling rules borrowed from metaheuristics can be implemented within the basic FP framework.
- Instead, we found extremely effective the idea of introducing some enumeration.
- Let x^F (F for fractional) be the LP point x^* (as computed at step 5) which is as close as possible to its rounding $[x^F]$, chosen among those generated by the FP procedure before cycling: typically, the infeasibility degree $\Delta(x^F, [x^F])$ is small.

Summary of the results (2)

- The problem with FP is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.
- Simple anti-stalling and anti-cycling rules borrowed from metaheuristics can be implemented within the basic FP framework.
- Instead, we found extremely effective the idea of introducing some enumeration.
- Let x^F (F for fractional) be the LP point x^* (as computed at step 5) which is as close as possible to its rounding $[x^F]$, chosen among those generated by the FP procedure before cycling: typically, the infeasibility degree $\Delta(x^F, [x^F])$ is small.
- Therefore, before doing anything else, it seems reasonable to fix x^F and use a truncated enumerative MIP method in the attempt of finding a feasible integer point close to x^F.

Summary of the results (2)

- The problem with FP is that, due to the flipping at step 9 of the algorithm, some cycling is possible: a same sequence of points x^* and $\widetilde{x}$ is visited again and again.
- Simple anti-stalling and anti-cycling rules borrowed from metaheuristics can be implemented within the basic FP framework.
- Instead, we found extremely effective the idea of introducing some enumeration.
- Let x^F (F for fractional) be the LP point x^* (as computed at step 5) which is as close as possible to its rounding $[x^F]$, chosen among those generated by the FP procedure before cycling: typically, the infeasibility degree $\Delta(x^F, [x^F])$ is small.
- Therefore, before doing anything else, it seems reasonable to fix x^F and use a truncated enumerative MIP method in the attempt of finding a feasible integer point close to x^F.
- For 0-1 MIPs, this amounts to optimize $\min\{\Delta(x^F, x) : Ax \geq b,\ x_j \text{ integer } \forall j \in \mathcal{I}\}$, where:

$$\Delta(x^F, x) = \sum_{j \in \mathcal{I}} [(1 - x_j^F)x_j + x_j^F(1 - x_j)] = \sum_{j \in \mathcal{I}} x_j^F + \sum_{j \in \mathcal{I}} (1 - 2x_j^F)x_j$$

is a suitable redefinition of the distance function of a generic integer point x with respect to the given fractional point x^F.

Improving the basic FP scheme

Name	value	nIT	nR	nH	initial $\Delta(x^*, \widetilde{x})$	final $\Delta(x^*, \widetilde{x})$	B&B nodes	B&B time	total time
danoint	82.00	3	0	1	3.0	3.0	33	0.69	0.87
glass4	4.10e9	100	0	1	0.3	0.1	0	0.01	0.38
net12	296.00	7	0	1	84.1	4.0	0	1.20	6.31
blp-ar98	14,269.65	23	0	1	13.7	3.4	340	8.21	12.41
blp-ic97	6,573.63	16	0	1	5.1	0.4	0	0.78	2.35
CMS750_4	517.00	44	0	1	234.4	131.7	550	12.23	18.94
usAbbrv.8.25_70	164.00	58	0	1	110.3	1.0	0	0.16	1.60
manpower1	6.00	4	0	1	80.3	60.5	0	1.46	3.15
manpower2	6.00	8	0	1	80.7	47.3	10	2.80	7.59
manpower3	6.00	7	0	1	114.7	56.5	13	7.34	11.32
manpower3a	7.00	10	0	1	88.0	42.5	19	5.18	11.03
manpower4	6.00	9	0	1	88.9	24.5	30	5.83	10.68
manpower4a	7.00	10	0	1	80.7	15.2	8	2.24	8.77

As a measure of the effectiveness of FP + redefinition of the objective function + branching, the overall number of B&B nodes of the improved version of FP, ILOG-Cplex 8.1 (emp=1), and ILOG-Cplex 8.1 (default) is 1003, 224576 and 13016, respectively.