Computing with multi-row Gomory cuts

Daniel G. Espinoza*

Universidad de Chile, Department of Industrial Engineering,
Santiago RM, 837-0439, Chile
daespino@dii.uchile.cl

Abstract. Cutting planes for mixed integer problems (MIP) are nowadays an integral part of all general purpose software to solve MIP. The most prominent,and computationally significant, class of general cutting planes are Gomory mixed integer cuts (GMI). However finding other classes of general cuts for MIP that work well in practice has been elusive. Recent advances on the understanding of valid inequalities derived from the infinite relaxation introduced by Gomory and Johnson for mixed integer problems, has opened a new possibility of finding such an extension. In this paper, we investigate the computational impact of using a subclass of minimal valid inequalities from the infinite relaxation, using different number of tableau rows simultaneously, based on a simple separation procedure. We test these ideas on a set of MIPs, including MIPLIB 3.0 and MIPLIB 2003, and show that they can improve MIP performance even when compared against commercial software performance.

* This research was partially funded by FONDECYT grant 1070749 and by ICM grant P05-004F.

1 Introduction

The most successful approach to solve general MIP today is branch and cut, where general cutting planes are a crucial factor for the overall performance. After the great success in the 90's of using general purposes cutting planes such as GMI cuts [9, 5], a great deal of research was devoted to extend those ideas to find other families of general cuts that consistently outperform GMI cuts. However, results have been mixed, and although there are several extensions that in theory are at least as good as GMI cuts, in practice they do not seem to offer much advantage. Most of the extensions have focused on deriving inequalities from the master cyclic group problem introduced by Gomory and Johnson [11], which look at a single constrained problem.

The theoretical importance of looking at multi-row relaxations has been proved in a number of works. For instance, Cook et al. [6], show an example with infinite Chvátal-Gomory rank (i.e. obtaining the convex hull of the integer points by adding inequalities derived from one row relaxations is impossible). Andersen et al. [3], prove that by looking at inequalities generated from two row relaxations, the convex hull of the Cook-Kannan-Schrijver example, can be obtained by adding a single cut. This situation is extended to higher dimensions in the work of Yanjun Li and Jean-Philippe P. Richard.

An interesting recent development has been the work of Andersen et al. [3], Cornuéjols and Borozan [7] and Gomory [10]; who have proposed to look at the so-called infinite relaxation problem, which was also introduced by Gomory and Johnson [11], and where several constraint are considered at the same time. The novelty of this relaxation is that it works on a continuous relaxation, and looks at an arbitrary number of tableau rows at the same time. Cornuéjols and Borozan [7] show that any minimal valid inequality for the relaxation can be related to maximal, convex, lattice-free polyhedrons; thus identifying *relevant* inequalities with simple geometrical entities.

To the best of our knowledge, no computational test of the impact of using cuts derived from this relaxation have been published. The main contribution of this paper is to show that they are also very valuable in practice, not only improving the root LP integrality GAP ($\mathrm{GAP}_{\mathrm{LP}}$) closed at the root node, but also in speeding-up the overall branch and cut performance when compared with CPLEX [12] defaults.

The rest of the paper is organized as follows. Section 2 presents the definition and basic results related to the infinite relaxation. Section 3 presents the basic computational problems, tradeoffs, and main ideas that guided the implementation, and also some further ideas to speed-up cut-generation and possible alternative choices. Section 4 explain our experiments, settings, and results.

2 The infinite relaxation

Consider a general mixed integer program (MIP)

$$\begin{aligned} &\min cx \\ &s.t.\ Ax = b \\ &\quad x_i \ \in \mathbb{Z}\ \forall i \in I \\ &\quad x_i \ \geq 0\ \forall i = 1,\ldots,n, \end{aligned} \tag{1}$$

where $I \subseteq \{1,\ldots,n\}$, $A \in \mathbb{Q}^{m\times n}$ is of full row rank, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, and $x \in \mathbb{Q}^n$. Branch and cut algorithms start by solving

$$\begin{aligned} &\min cx \\ &s.t.\ Ax = b \\ &\quad x_i \ \geq 0\ \forall i = 1,\ldots,n, \end{aligned} \tag{2}$$

the LP relaxation of (1), and obtain an optimal basic solution of the form

$$x_B = f + \sum_{j\in N} r^j x_j, \tag{3}$$

where B is the set of basic variables satisfying $B \subseteq \{1,\ldots,n\}$, $|B| = m$, N is the set of non-basic variables defined as $N = \{1,\ldots,n\} \setminus B$, and $f, r^j \in \mathbb{Q}^m, \forall j \in N$, $f \geq 0$. The basic solution is $x^* = (x_B, x_N) = (f, 0)$, and is an optimal solution to (1) if and only if $x_i^* \in \mathbb{Z}, \forall i \in I' = I \cap B$. If not, then one might try to find a valid inequality cutting off x^* from the feasible region of (2).

One possibility is to consider the following relaxation of (1):

$$\begin{aligned} &z = f + \sum_{i\in N\cap I} (r^i - a^i)s_i + \sum_{i\in N\setminus I} r^i s_i, \\ &z \in \mathbb{Q}^{I'}, \\ &s \geq 0, \end{aligned} \tag{4}$$

where we drop all basic continuous variables, drop the non-negativity constraints on the basic integer variables, and where $a^i \in \mathbb{Z}^{I'}, \forall i \in I \cap N$, $z = x_{I'} - \sum_{i\in I\cap N} a^i s_i$, and then relax s_i to be continuous. This relaxation was considered in [3, 10] for the case $|I'| = 2$.

Gomory and Johnson [11] suggested relaxing (4) to an infinite-dimensional space; following the notation in [7]; it can be described as:

$$\begin{array}{l} x = f + \sum_{finite} r s_r \\ x \in \mathbb{Z}^q \\ s \geq 0 \end{array} \tag{5}$$

where s_r is defined for every $r \in \mathbb{Q}^q$, and $\sum_{finite}$ means that $|r : s_r > 0| \in \mathbb{N}$, i.e. s has finite support. This is called the *infinite relaxation* and is denoted by R_f,where the feasible solutions of R_f are vectors (x, s) with finite support satisfying (5). Note that any valid inequality for (5) yields a valid inequality for (1).

Borozan and Cornuéjols [7] studied minimal valid inequalities for (5), proving the following theorem:

Theorem 1 (Minimal Valid Inequalities for R_f [7]). *If $f \notin \mathbb{Z}^q$, then any minimal valid inequality that cuts off $(f, 0)$:*

i. Is of the form $\sum_{finite} \psi(r)s_r \geq 1$.

ii. ψ is positive, subadditive, homogeneous, convex and piecewise linear.

iii. If $B_\psi = \{x \in \mathbb{Q}^p : \psi(x - f) \leq 1\}$, then B_ψ is convex, with no integral point in its interior. Furthermore $f \in B_\psi$.

iv. If ψ is finite, then ψ is a continuous nonegative homogeneous convex piecewise linear function with at most 2^q pieces.

v. If ψ is finite, then f is in the interior of B_ψ and B_ψ is a polyhedron of at most 2^q facets, and each of its facets contains an integral point in its relative interior.

One of the consequences of Theorem 1 is that it allow us to identify minimal valid inequalities ψ with the set B_ψ, providing a simple geometrical interpretation for them. We use this interpretation to chose a sub-family of minimally valid inequalities for (5). It is worth mentioning that the results of Theorem 1 where simultaneously conjectured (and partially proved) by Gomory in [10].

3 Selecting a subclass of valid inequalities, and separating them

Thanks to the results in [7], the problem of finding minimal valid inequalities for (5), can be reduced to the problem of looking at maximal lattice-free polyhedra in $\mathbb{Q}^q$, where the lattice is just $\mathbb{Z}^q$. Although the characterization of all maximal lattice-free convex sets in the plane is known [15], such a characterization is unknown for arbitrary dimensions.

For general dimension q, we can define the following full-dimensional maximal lattice-free bounded convex sets:

1. The simplex defined by the points $\{0, \pm ke_i : i = 1, \ldots, q\}$.
2. The set $B_a = \frac{1}{2} + \{x : a^\delta x \leq a^\delta \delta, \forall \delta \in \Delta\}$ where $\Delta = \{\{-\frac{1}{2}, \frac{1}{2}\}^q\}$, $0 < a^\delta \in \mathbb{Q}^q$ and $a^\delta_i \neq 0, \forall i = 1, \ldots, q, \delta \in \Delta$.

These two classes of sets represent the two extremes in terms of number of facets; in the first family, each set has $q+1$ facets, while in the second family, each set has 2^q facets. Note also that each of their facets contains an integer point in their relative interior, thus they define minimal valid inequalities for (5).

For the case $q = 2$, Cornuéjols and Margot [8] proved that all simplex-related sets (called triangle inequalities in [3]) are facet defining for R_f, but that not all B_a sets define facets of R_f. However, is easy to see that there exist an arbitrarily small perturbation ε of a, such that $B_{a+\varepsilon}$ defines a facet of R_f. This observation, and the limited numerical precision of floating point representation, justify, from a practical point of view, overlooking the fact that some B_a do not define a facet of R_f for $q = 2$. Although a similar result for arbitrary q is unknown, it seems reasonable to conjecture that related arguments should show the importance of the sets B_a in general.

This gives us a wide range of possible sets B to choose from. However, if we restrict ourselves to sets that are symmetric with respect to each coordinate axis, then, the only possible choice for B is the family B_a, where all $a^\delta \equiv a$ for some $a \in \mathbb{Q}^q_+$ (we assume that $0 \notin \mathbb{Q}_+$). This restriction implies that the resulting cut should be invariant under multiplication of -1 to any constraint in (5).

From this point on, we focus on this kind of lattice-free sets. We assume that $f \in (0,1)$, and define $f' = f - \frac{1}{2}e$, where e is the vector of all ones. With this, ψ_a (the function related to B_a) can be defined as follows:

$$\psi_a(r) = \begin{cases} 0 & \text{if } r = 0 \\ 2 \max\limits_{\delta \in \Delta} \left\{ \frac{\phi_\delta(a,r)}{a_o - \phi_\delta(a,f')} \ : \phi_\delta(a,r) > 0 \right\} & \text{if } r \neq 0 \end{cases}, \tag{6}$$

where $\phi_\delta(a,b) = \sum(a_i \delta_i b_i : i = 1, \ldots, q)$ and $a_o = \frac{1}{2} a \cdot e$.

Note that the amount of work to compute $\psi_a(r)$ is exponential in q, however, one can speed up the process by using gray-code enumeration of Δ. In our code we use Knuth's loopless gray binary generation (LGBG) algorithm [13] to speed-up the computation of $\psi_a(r)$, moreover, we compute $\psi_a(r)$ for all required r at the same time. Additional speed gains can be achieved by noting that in LGBG, index i changes its value exactly

2^{q-i} times during the algorithm, thus sorting each row in decreasing order by number of non-zeros should decrease the amount of total work. Finally, another factor of two can be gained by maintaining a list of $r : \phi_\delta(a, r) > 0$.

Another problem is to choose appropriate vectors a. One possibility is to use branching pseudo-cost values (see [1, 14] for details on pseudo-cost branching) to define the a_i. Instead, we use $a_i = 1, \forall i = 1, \ldots, q$, but select the fractional variables to consider using branching pseudo-cost information.

For integer non-basic variables we select a^i in (4), such that $r^{i'} = r^i - a^i \in [-\frac{1}{2}, \frac{1}{2}]^q$, in the hope of obtaining small coefficients for $\psi_a(r^{i'})$. Note, that such a choice may not be the best possible.

To improve numerical stability of the cuts, we choose from fractional variables that are away from the nearest integer by at least 2^{-12}; also, the ratio between the smallest and largest absolute value in the cut should not exceed 2^{15}; if the minimum non-zero absolute coefficient in the cut ($|c|_{min}$) is above one, we divide the resulting cut by $|c|_{min}$; we discard cuts whose violation is below 2^{-10}; finally, we add cuts only at the root node of the branch and cut run. The code is available at `http://dii.uchile.cl/~daespino`.

4 Computational Results

Our computing environment is a Linux 2.6.22 machine with 1Gb. of RAM, with a 3GHz. Intel Pentium 4 CPU with 1Mb of cache; all the code is written in C, and was compiled with GCC 4.2.0 with optimization flags -O3.

Our cutting scheme was embedded as a cut-callback in CPLEX 10.2, and is called after CPLEX has added its own cuts. In every call we add at most one cut, but the procedure may be called many times during the optimization process. Our procedure adds cuts only at the root node. We compare our results against CPLEX defaults, with pre-processing turned on; this include automatic generation of clique cuts, lifted cover cuts, implied bound cuts, lifted flow cover cuts, flow path cuts and Gomory fractional cuts.

Our test set of MIP instances contains all MIPLIB 3.0 [4], MIPLIB 2003 [2], and other problems from the literature. The full test set contained 173 problems, from where we discard all problems (29) where the GAP_{LP} after solving with CPLEX 10.2 defaults was below 0.1%; then we discard all problems (34) CPLEX could solve to optimality within five seconds of CPU time; then we discarded all problems (15) where neither CPLEX nor our cutting procedure could improve the root LP bound[1]; finally, we discarded all problems (8) where our cutting procedure could not add any cut[2]. This reduced our test-set to 87 problems.

berlin_5_8_0	CMS750_4	glass4	marketshare1	marketshare2
neos19	neos818918	neos823206	net12	noswot
p2m2p1m1p0n100	railway_8_1_0	rd-rplusc-21	usAbbrv.8.25_70	van

Table 1. Problems where neither CPLEX nor our procedure could improve the root LP value.

bg512142	dano3mip	dano3mip.pre	dg012142
harp2	mod011	momentum3	neos4

Table 2. Problems where our procedure could not add any cut to the problem.

We tested six configurations, CPLEX defaults (C0), and the configurations T2N5, T5N5, T10N5, T10N1k and T15N1k[3], where TxxNyy signifies the adding of up to yy cuts generated using up to xx tableau rows. The first four configurations where run with a time limit of one hour, while the last two configurations where run with a time limit of 20 minutes.

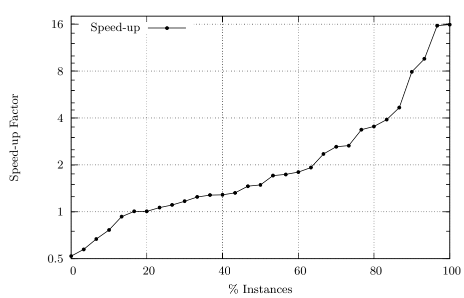

Fig. 1. Overall speed-up

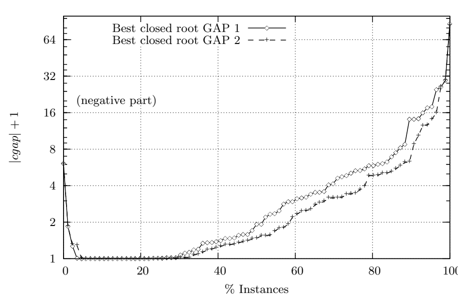

Fig. 2. Best closed $\mathrm{GAP_{LP}}$

Tables 3, 4, 5 present our computational results over the reduced test-set. The first column indicates the problem name; the following six columns, give the root LP bound and the running time for the corresponding configuration; finally, the last column, has the optimal/best known solution for instance, and then the maximum of the closed $\mathrm{GAP_{LP}}$, de-

[1] Table 1 contain the list of all such problems.
[2] Table 2 contain the list of all such problems.
[3] 1k stand for 1000, i.e. a thousand.

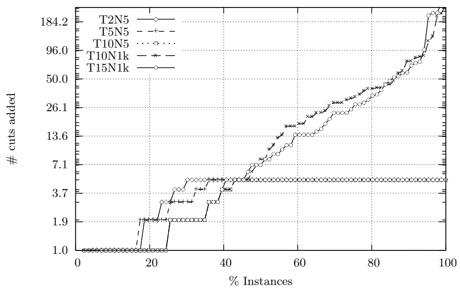

Fig. 3. Cuts added by configuration

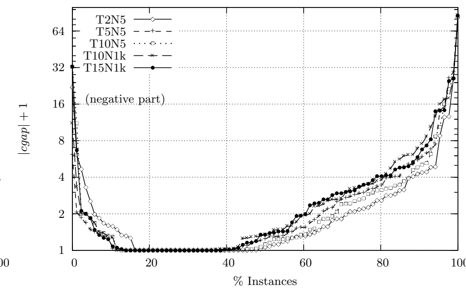

Fig. 4. Closed $\mathrm{GAP_{LP}}$ by configuration

fined as $(Z_{LP} - Z_{CPX})/(Z_{IP} - Z_{CPX})$[4] over all configurations, and then the maximum of the same quantity over the T*N5 configurations.

Figure 1 summarizes the best speed-up factor over CPLEX defaults obtained over all instances (31) that finished to optimality on all six configurations, the geometric average speed-up was 31%. Also, looking at problems where at least one configuration had finished to optimality, CPLEX was faster by at least 5% in 10 cases, while in 16, 16, 14, 11, 9 cases configurations T2N5, T5N5, T10N5, T10N1k, T15N1k, where faster by at least 5% respectivelly. Figure 2 summarizes the best closed $\mathrm{GAP_{LP}}$ over CPLEX default (1) and the best closed $\mathrm{GAP_{LP}}$ over CPLEX when we limit the code to add up-to five cuts at the root node (2). The number of cases where each configuration had the best root LP value where 18, 19, 27, 22, 43, 38, for C0, T2N5, T5N5, T10N5, T10N1k and T15N1k respectively.

Figure 3 shows the number of cuts added for each configuration. Note that for the T*N1k configurations, more than 80% of the instances required less than 40 cuts. On the other hand, it seems that the more tableau rows we use to generate cuts, the less number of total cuts our procedure needs. Figure 4 shows the closed $\mathrm{GAP_{LP}}$ for each configuration; where negative values (i.e. the procedure performs worse than CPLEX) are displayed in the left part of the graph. Again from this figure it seems clear that there are advantages to considering more than one tableau row at the same time; in fact, the results for two tableau rows are consistently poorer than configurations that use more tableau rows in the cutting procedure.

[4] where Z_{LP} is the root LP value for the configuration, Z_{IP} is the value of the optimal/best solution known for the problem, and Z_{CPX} is the value of the root LP obtained with CPLEX defaults.

5 Final Remarks and Conclusions

We have shown that even simple subclasses of inequalities derived from the infinite relaxation can have an important impact both on overall branch and cut performance, and on the $\mathrm{GAP}_{\mathrm{LP}}$ closed at the root node. These results point towards both trying to identify important classes of inequalities for R_f for higher dimensions, and to find good computational implementation choices for them.

Although the implementation is numerically conservative, still, there are some numerical issues when cuts are used within the branch and bound tree. There are also instances where the cuts generated tend to be parallel to previous ones, causing again numerical issues.

There are many possibilities to explore, like adding more than one cut in every iteration, choosing different sets of tableau to work on, choosing different ground sets B_a, and testing the impact of inequalities derived form simplex-like ground sets.

References

1. T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Op. Res. Letters*, 33:42–54, 2005.
2. T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 34:4:361–372, 2006.
3. K. Andersen, Q. Louveaux, R. Weismantel, and L. A. Wolsey. Inequalities from two rows of a simplex tableau. In M. Fischetti and D. P. Williamson, editors, *IPCO*, volume 4513 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2007.
4. R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.
5. R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice - closing the gap. In *Proceedings of the 19th IFIP TC7 Conference on System Modelling and Optimization*, pages 19–50, Deventer, The Netherlands, 2000. Kluwer, B.V.
6. W. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Math. Prog.*, 47:155–174, 1990.
7. G. Cornuéjols and V. Borozan. Minimal valid inequalities for integer constraints. George Nemhauser Symposium, Atlanta, July 2007.
8. G. Cornuéjols and F. Margot. On the facets of mixed integer programs with two integer variables and two constraints. Tepper Working Paper 2007 E-35 (2007), 2007.
9. R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, RAND Corporation, 1960.
10. R. E. Gomory. Corner polyhedra and two-equation cutting planes. George Nemhauser Symposium, Atlanta, July 2007.
11. R. E. Gomory and E. L. Johnson. Some continuous functions related to corner polyhedra, part I. *Math. Prog.*, 3:23–85, 1972.
12. ILOG CPLEX Division, Incline Village, Nevada, 89451, USA. *CPLEX 10.2 Reference Manual*, 2007.
13. D. E. Knuth. *The Art of Computer Programming*, volume 4, chapter 7.2.1.1. Addisoin-Wesley, 1st. edition, February 2005.
14. J. T. Linderoth and M. W. P. Savelsbergh. A computational study of search strategies for mixed integer programming. *INFORMS J. on Computing*, 11:173–187, 1999.
15. L. Lovász. Geometry of numbers and integer programming. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 177–210. Springer, 1989.

Configuration	C0	T2N5	T5N5	T10N5	T10N1k	T15N1k	Results
Problem	Root LP	Root LP	Root LP	Root LP	Root LP	Root LP	Best Sol.
	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	CGap1/CGap2
a1c1s1	9770.47	9797.84	9854.74	9799.46	9799.46	9823.78	11503.4
	3592.91	3593.45	3592.71	3594.60	1193.85	1194.55	4.86%/4.86%
A1C1S1	9804.58	9862.15	9869.86	9838.56	9838.56	9829.37	11503.4
	3587.77	3600.33	3594.44	3584.48	1195.51	1196.85	3.84%/3.84%
A2C1S1	9468.86	9394.16	9394.16	9320.6	9320.6	9385.61	10938.8
	3590.22	3598.84	3598.90	3591.90	1195.77	1194.16	-5.08%/-5.08%
aflow30a	1075.14	1081.59	1074.58	1074.58	1074.58	1075.27	1158
	66.62	55.56	45.79	50.67	48.33	50.86	7.79%/7.79%
aflow40b	1059.61	1058.88	1059.61	1062.55	1062.55	1062.94	1168
	3585.42	3563.11	3597.93	3596.80	1193.03	1194.28	3.07%/2.71%
air04	55645.7	55664.6	55660.4	55663.3	55667	55664.9	56137
	42.33	75.30	48.47	64.48	154.15	405.37	4.33%/3.86%
air05	25957.8	25972	25970.5	25973.7	25978.6	25973.7	26374
	31.48	83.59	79.81	66.11	500.72	295.92	5.00%/3.83%
B1C1S1	15496.3	15471.8	15463.4	15529	15529	15451.8	24881.7
	3577.55	3586.90	3586.32	3579.90	1191.87	1192.43	0.35%/0.35%
B2C1S1	16012.5	16112.2	16165.8	15969.5	15969.5	16213	26282.5
	3587.55	3590.03	3588.81	3589.90	1190.73	1192.06	1.95%/1.49%
bc1	2.47	2.47	2.47	2.47	2.48	2.47	3.34
	274.66	493.91	436.80	880.40	923.93	585.60	0.47%/0.26%
bell3a	873792	873792	873792	873792	873792	873792	878430
	7.09	7.17	7.53	7.22	7.54	7.28	0.00%/0.00%
biella1	3.0605e+06	3.0605e+06	3.0605e+06	3.0605e+06	3.0605e+06	3.0605e+06	3.0678e+06
	3590.18	3591.99	3532.93	3593.41	1191.94	1195.15	-0.27%/-0.29%
bienst1	15.11	15.49	14.98	14.99	14.99	15.25	46.75
	431.12	850.79	331.77	475.22	463.64	503.64	1.20%/1.20%
bienst2	15.28	15.35	15.35	15.28	15.28	15.27	54.6
	3586.40	3599.01	3566.95	3591.94	1194.88	1195.43	0.18%/0.18%
blp-ar98	6086.37	6086.76	6087.58	6087.57	6088.08	6088.14	6217.86
	3563.80	3559.50	3563.07	3566.11	1165.68	1179.48	1.35%/0.92%
blp-ic97	3928.3	3928.38	3928.83	3928.46	3928.64	3928.81	4057.94
	3569.97	3377.47	3586.15	3400.25	1181.92	1176.14	0.41%/0.41%
blp-ic98	4376.19	4376.8	4378.04	4378.59	4381.97	4379.05	4531.39
	3560.93	3540.84	3595.65	3541.67	1142.70	1158.75	3.72%/1.55%
blp-ir98	2283.19	2283.67	2283.51	2284.5	2286.22	2287.46	2342.32
	541.68	606.95	725.03	467.55	922.14	973.49	7.22%/2.20%
core2586-950	935.94	935.95	935.95	935.95	935.95	935.95	974
	3423.24	3502.65	3587.97	3598.87	1411.34	1205.72	0.01%/0.01%
core4284-1064	1054.08	1054.08	1054.08	1054.08	1054.08	1054.08	1086
	3598.72	3599.71	3599.47	3545.67	1196.53	1209.14	0.01%/0.00%
core4872-1529	1510.91	1510.91	1510.91	1510.91	1510.91	1510.91	1568
	3305.73	3598.09	3545.04	3599.30	1196.74	1196.19	0.00%/0.00%
danoint	62.73	62.73	62.73	62.73	62.73	62.73	65.67
	3589.38	3599.02	3598.22	3594.90	1198.09	1197.23	0.02%/0.02%
dc1c	1.7582e+06	1.7575e+06	1.7575e+06	1.7575e+06	1.7575e+06	1.7575e+06	1.8478e+06
	3591.28	3591.33	3591.63	3432.11	1192.99	1181.32	-0.84%/-0.84%
dc1l	1.7445e+06	1.7445e+06	1.7445e+06	1.7446e+06	1.7446e+06	1.7445e+06	1.8517e+06
	3588.70	3588.79	3588.84	3587.28	1194.80	1195.38	0.02%/0.02%
dolom1	6.5563e+06	6.5563e+06	6.5563e+06	6.5563e+06	6.5563e+06	6.5563e+06	1.49e+08
	3588.23	3586.81	3587.44	3585.06	1189.61	1190.51	0.00%/0.00%
ds	59.54	59.56	59.49	59.59	59.59	59.35	447.01
	3584.19	3582.82	3585.46	3584.17	1194.23	1212.81	0.01%/0.01%
fast0507	172.15	172.17	172.18	172.18	172.18	172.18	174
	2340.49	2862.75	1119.82	2467.56	1170.02	1250.19	1.95%/1.87%
gesa2-o	2.5731e+07	2.5729e+07	2.5733e+07	2.5731e+07	2.5731e+07	2.5731e+07	2.5779e+07
	8.79	6.39	5.62	6.37	6.22	15.37	4.50%/4.50%
liu	560	560	560	560	560	560	1174
	3476.32	3467.47	3463.13	3452.70	1152.16	1153.31	0.00%/0.00%

Table 3. Results over reduced test set, part I

Configuration	C0	T2N5	T5N5	T10N5	T10N1k	T15N1k	Results
Problem	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Best Sol. CGap1/CGap2
manna81	-13229.5	-13227.5	-13226.8	-13226.1	-13221.9	-13220.8	-13164
	3540.39	3548.56	3548.17	3524.98	1181.63	1180.20	13.22%/5.21%
mas74	10506.2	10555.8	10558.9	10535.5	10535.5	10538.2	11801.2
	2548.22	3203.12	2437.62	2299.57	1185.31	1180.43	4.07%/4.07%
mas76	38908	38925.2	38934.3	38911.1	38911.1	38911.8	40005.1
	241.62	189.35	175.21	325.39	319.60	210.48	2.40%/2.40%
misc07	1425	1425	1425	1425	1425	1425	2810
	23.53	38.43	28.10	44.54	61.50	21.00	0.00%/0.00%
mkc1	-611.85	-611.85	-611.85	-611.85	-611.85	-611.85	-607.21
	19.25	20.48	39.64	99.49	89.70	58.91	0.00%/0.00%
mkc	-582.39	-582.39	-582.39	-582.39	-579.15	-581.79	-563.23
	3547.53	3550.57	3562.28	3567.50	1172.77	1181.56	16.91%/0.00%
momentum1	96250.1	96250.1	96250.1	96250.1	96250.1	96250.1	109143
	3592.25	3590.55	3594.12	3593.23	1193.81	1182.12	0.00%/0.00%
momentum2	10702.1	10702.1	10705.4	10702.6	10702.6	10705.2	12314.2
	3591.55	738.17	3288.60	3593.40	1196.20	749.59	0.20%/0.20%
msc98-ip	1.9702e+07	1.9702e+07	1.9702e+07	1.9702e+07	1.9702e+07	1.9702e+07	1.98e+07
	3583.89	3549.60	3593.76	3586.20	1196.60	1195.54	0.00%/0.00%
mzzv11	-22066.1	-22067.3	-22066.2	-22063.3	-22061.5	-22067	-21718
	548.37	471.03	819.70	541.25	491.16	551.90	1.32%/0.81%
mzzv42z	-20830.7	-20826.7	-20789	-20787.2	-20787.2	-20822.6	-20540
	130.53	140.58	155.44	157.69	161.51	226.87	14.96%/14.96%
neos10	-1187.33	-1185.79	-1185.77	-1185.12	-1184.66	-1185.34	-1135
	23.15	22.85	21.30	29.77	26.81	61.57	5.10%/4.23%
neos11	6	6	6	6	6	6	9
	446.38	405.21	192.80	216.16	231.75	203.14	0.00%/0.00%
neos12	9.51	9.51	9.51	9.51	9.51	9.51	13
	644.55	971.43	1111.82	883.34	1099.59	656.93	-0.01%/-0.01%
neos13	-112.97	-112.75	-107.98	-109.42	-112.4	-112.63	-95.47
	3596.18	3596.89	3696.19	3596.15	1190.17	1197.57	28.51%/28.51%
neos14	66464.1	66387.1	66381.6	66439.7	66439.7	66536.2	74333.3
	1555.23	1181.56	1426.66	1037.64	1007.33	1193.10	0.92%/-0.31%
neos15	70411.4	70172.5	70409	70418.4	70418.4	70415.8	80835
	3591.99	3597.23	3595.22	3588.92	1197.49	1198.19	0.07%/0.07%
neos17	0.03	0.03	0.03	0.02	0.02	0.02	0.15
	3057.75	315.74	776.98	514.16	517.12	1192.86	0.13%/0.13%
neos18	13	13	13	13	13	13	16
	325.23	3575.88	2547.07	3572.41	1183.89	1051.34	0.00%/0.00%
neos20	-470.8	-470.8	-470.8	-470.8	-470.8	-470.8	-434
	186.19	61.84	84.35	134.95	124.60	124.43	0.00%/0.00%
neos21	2.72	2.74	2.73	2.73	2.74	2.75	7
	91.88	107.15	89.11	108.62	107.98	114.33	0.71%/0.45%
neos22	777536	777786	777676	777702	777702	777821	779715
	66.65	52.88	49.44	36.72	38.65	62.01	13.09%/11.51%
neos23	63.81	63.81	63.81	63.82	64.16	63.82	137
	653.86	1958.24	2709.85	813.83	1195.49	525.27	0.48%/0.02%
neos2	-4069.79	-4056.09	-4039.14	-4070.69	-4028.29	-3927.78	454.86
	14.81	15.08	13.36	16.30	19.03	52.29	3.14%/0.68%
neos3	-5664.36	-5630.32	-5655.1	-5643.04	-5643.04	-5731.26	368.84
	54.28	53.25	54.06	69.07	63.54	80.71	0.56%/0.56%
neos5	13.33	13.32	13.33	13.33	13.33	13.33	15
	3197.48	3579.64	3580.21	3367.37	1193.82	1194.55	0.00%/0.00%
neos7	692631	693168	693268	692631	692631	692532	721934
	137.84	59.85	59.72	51.88	51.01	1091.68	2.17%/2.17%
neos9	793.25	792.25	793.5	791.75	793.14	791.75	798
	3587.56	3586.40	3585.06	3586.00	1191.12	1190.05	5.26%/5.26%
nsrand-ipx	50181.8	50183.1	50184.8	50184.2	50187.8	50186.2	51200
	3588.37	3575.79	3583.57	3609.45	1185.31	1198.76	0.58%/0.30%

Table 4. Results over reduced test set, part II

Configuration	C0	T2N5	T5N5	T10N5	T10N1k	T15N1k	Results
Problem	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Root LP Time (s)	Best Sol. CGap1/CGap2
nsrand_ipx	50181.8	50183.1	50184.8	50184.2	50187.8	50186.2	51200
	3579.66	3583.67	3579.68	3571.66	1193.02	1203.12	0.58%/0.30%
nug08	204.28	204.31	204.33	204.33	204.37	204.37	214
	47.44	63.14	73.89	90.37	126.99	185.04	0.92%/0.54%
nw04	16380.3	16771.3	16779.9	16781.3	16792.6	16787.9	16862
	58.00	46.19	51.81	52.10	112.66	907.47	85.59%/83.25%
opt1217	-20	-19	-19	-19	-19	-19	-16
	3553.84	3542.62	3544.18	3530.90	1193.49	1190.35	25.00%/25.00%
pk1	0	0	0	0	0	0	11
	153.16	143.84	112.56	162.38	168.46	195.87	0.00%/0.00%
protfold	-41.09	-39.92	-41.09	-41.09	-39.42	-39.78	-31
	3599.38	3599.40	3599.40	3598.46	1199.45	1199.71	16.53%/11.63%
qap10	333.5	333.5	333.48	333.51	333.52	333.51	340
	395.29	951.54	399.66	450.92	486.25	507.83	0.34%/0.23%
qiu	-923.04	-926.83	-926.83	-923.04	-923.04	-923.04	-132.87
	77.25	92.21	95.20	77.66	74.46	68.85	0.00%/0.00%
rail507	172.15	172.17	172.17	172.17	172.18	172.18	174
	3570.21	3320.39	3564.49	3559.79	1160.16	1255.95	1.82%/1.46%
ran14x18_1	3362.27	3363.59	3363.68	3363.22	3363.22	3361.99	3735
	3588.84	3587.58	3587.68	3589.10	1195.36	1195.84	0.38%/0.38%
roll3000	12243.9	12257.5	12257.8	12259.4	12259.4	12260.1	12890
	3592.85	3596.57	3595.23	3592.53	1194.49	1197.01	2.52%/2.41%
rout	985.46	985.53	985.53	985.46	985.46	985.56	1077.56
	219.81	292.25	195.17	126.08	122.64	81.27	0.11%/0.09%
seymour1	405.86	405.9	405.92	405.93	405.97	405.93	410.76
	2186.19	2455.05	2762.13	2433.69	1200.26	1203.71	2.22%/1.33%
seymour	407.17	407.2	407.63	407.63	408.2	408.17	423
	3582.73	3590.80	3590.46	3590.24	1195.45	1191.93	6.50%/2.95%
siena1	1.0163e+07	1.0163e+07	1.0163e+07	1.0163e+07	1.0163e+07	1.0163e+07	1.58e+08
	3592.54	3593.03	3592.21	3593.56	1197.19	1199.08	0.00%/0.00%
sp97ar	6.5388e+08	6.5391e+08	6.5391e+08	6.5391e+08	6.5391e+08	6.5391e+08	6.64e+08
	3591.66	3568.81	3588.03	3568.96	1170.85	1267.79	0.36%/0.32%
sp97ic	4.2211e+08	4.2217e+08	4.2217e+08	4.2218e+08	4.2223e+08	4.2219e+08	4.3e+08
	3553.65	3554.51	3551.67	3542.83	1136.89	1190.07	1.48%/0.79%
sp98ar	5.2499e+08	5.2504e+08	5.2508e+08	5.2509e+08	5.2511e+08	5.2509e+08	5.3e+08
	3549.89	3556.06	3562.27	3558.49	1184.27	1175.80	2.57%/2.16%
sp98ic	4.4430e+08	4.4445e+08	4.4443e+08	4.4445e+08	4.4448e+08	4.4444e+08	4.5141e+08
	3550.63	3552.86	3563.69	3580.06	1190.08	1174.74	2.53%/2.16%
stein45	22	22	22	22	22	22	30
	24.73	26.15	26.43	30.23	35.56	36.11	0.00%/0.00%
swath1	338.68	339.99	339.36	340.32	340.54	340.64	379.07
	41.33	37.34	86.34	124.20	309.76	1195.45	4.85%/4.07%
swath2	343.09	343.89	344.07	344.13	344.27	344.91	385.2
	195.24	104.00	527.60	40.76	195.96	458.98	4.32%/2.47%
swath3	343.09	344.09	344.16	344.13	344.29	345.07	397.76
	783.86	535.92	789.25	186.25	393.72	866.58	3.62%/1.96%
swath	373.88	374.41	375.53	375.21	379.4	376.08	467.41
	3494.23	3533.15	3513.14	3535.65	1141.32	1199.99	5.90%/1.77%
timtab1	443780	438838	485907	446072	446072	462240	764772
	3567.05	3587.46	3589.34	3570.40	1196.11	1193.77	13.12%/13.12%
timtab2	563628	560505	566589	558124	558124	575232	1.1111e+06
	3583.77	3593.85	3592.81	3583.98	1194.40	1194.57	2.12%/0.54%
tr12-30	129675	129675	129728	129762	129762	129894	130596
	3593.70	3596.10	3586.60	3544.45	1198.44	1197.57	23.72%/9.38%
trento1	5.1830e+06	5.1830e+06	5.1830e+06	5.1830e+06	5.1830e+06	5.1830e+06	5.2874e+06
	3582.16	3590.92	3590.58	3581.38	1194.41	1195.11	0.02%/0.00%
UMTS	2.9137e+07	2.9137e+07	2.9140e+07	2.9142e+07	2.9142e+07	2.9141e+07	3.01e+07
	3554.64	3581.09	3588.94	3576.85	1194.68	1193.78	0.47%/0.47%

Table 5. Results over reduced test set, part III