A First Look at Picking Dual Variables for Maximizing Reduced Cost Fixing

Omid Sanei Bajgiran[1,3], Andre A. Cire[1], and Louis-Martin Rousseau[2,3]

[1]Department of Management, University of Toronto Scarborough
[2]Department of Mathematics and Industrial Engineering, Polytechnique Montréal
[3]Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract. In this paper, we investigate a new variable-fixing methodology for arbitrary mixed-integer linear programming models. Our technique is based on expanding the classical reduced cost-based filtering by searching for the optimal dual values that maximize propagation. The resulting method can be naturally incorporated into existing solvers. Preliminary results on a large set of benchmark instances suggest that the method can effectively reduce solution times on hard instances with respect to a state-of-the-art commercial solver.

Keywords: mixed-integer programming; Variable Fixing Methodology; Reduced-cost based filtering

1 Introduction

A key feature of modern mathematical programming solvers refers to the wide range of techniques that are applied to *simplify* an instance. Typically considered during a preprocessing stage, these techniques aim at fixing variables, eliminating redundant constraints, and identifying structure that can either lead to speed-ups in solution times or provide useful information about the model at hand. Examples of valuable information include, e.g., potential numerical issues or, if the model is infeasible, which subset of inequalities and variables may be responsible for the infeasibility [10]. These simplification methods alone reduce solution times by half in state-of-the-art solvers such as CPLEX, Gurobi, or SCIP [3], thereby constituting an important tool in the use of mixed-integer linear programming (MILP) in practical real-world problems [9].

In this paper we investigate a new simplification technique that expands upon the well-known *reduced cost fixing method*, first proposed by Balas and Martin [1] and largely used both in the mathematical programming and the constraint programming (CP) communities. The underlying idea of the method is straightforward: Given a linear programming (LP) model and any optimal solution to such a model, the *reduced cost* of a variable indicates the marginal change in the objective function when the value of the variable in that solution is increased [4]. In cases where the LP encodes a relaxation of an arbitrary optimization problem, we can therefore filter all values from a variable domain that, based on the reduced cost, incur a new objective function value that is worse than a known solution to the original problem. The result is a tighter variable bound which can then trigger further variable fixing and other simplifications.

This simple but effective technique is widely applied in MILP presolving [3,9,10] and plays a key role in a variety of propagation methods for global constraints in CP [5,7,6]. It can be easily incorporated into solvers since the reduced costs are directly derived from any optimal set of duals, which in turn can be efficiently obtained by solving an LP once. The technique is also a natural way of exploiting the strenghts of MILP within a CP framework, since the dual values incorporate a global bound information that is potentially lost when processing constraints one at a time (a concept that is explored, e.g., in [13,15,2]).

However, in all cases typically only *one* reduced cost per variable is considered for this methodology (e.g., the one obtained after solving the LP relaxation of a MILP). In theory, any optimal set of duals yields valid reduced costs that may lead, in turn, to quite different variable bound tightenings. Our goal in this work is to investigate a notion of *consistency* for reduced cost fixing; that is, we wish to obtain the strongest variable bounds that can be derived from this technique considering all possible set of valid duals, and verify whether this can be effective in practice when simplifying a model. In particular, we view the proposed techniques as a first direction towards some of the main questions in the field of *CP-based Lagrangian relaxation* [12,2], in that we wish to find the set of duals that maximize propagation and understand its structural properties.

In this paper we restrict ourselves to MILPs and formulate the problem of finding the strongest bound reductions as an optimization problem defined over the space of valid duals. We show that the resulting model can be solved in polynomial time on the number of variables and constraints of the original problem, but which is prohibitive in practice as it requires solving a large number of LPs. To circumvent this issue, we propose an approximate technique that aims at maximizing the number of fixed variables by solving an alternative MILP problem. The resulting technique can be seamlessly incorporated into existing solvers, and preliminary results over the MIPLIB indicate that it can provide substantial solution time improvements, motivating further research on the quality of the duals within both MILP and CP technology.

The paper is organized as follows. Section 2 introduces the necessary notation and the basic concepts of reduced cost fixing. We then introduce the notion of reduced cost consistency in Section 3 and investigate its computational complexity. Next, we discuss one alternative to obtain an approximate consistency in Section 4. Finally, we present a preliminary numerical study in Section 5 and conclude in Section **??**.

2 Preliminaries

For the purposes of this paper, consider the problem

$$z_P := \min\{c^T x : Ax \geq b, x \geq 0\} \tag{P}$$

with $A \in \mathbb{R}^{n \times m}$ and $b, c \in \mathbb{R}^n$ for some $n, m \geq 1$. We assume that (P) represents the LP relaxation of an MILP problem P_S with an optimal solution value of $z^* \geq z_P$ and where variables $\{x_i : i \in S\}$ are subject to integrality constraints. The dual of the problem (P) can be written as

$$z_D := \max\{u^T b : u^T A \leq c, u \geq 0\} \tag{D}$$

where $u \in \mathbb{R}^m$ is the vector of *dual variables*. We assume for exposition that P_S, (P), and (D) are both feasible and bounded (the results presented here can be easily generalized when that is not the case).

We have $z_P = z_D$ (strong duality) and for every optimal solution x^* of (P), there exists an optimal solution u^* to (D) such that $u^{*T}(b - Ax^*) = 0$ (complementary slackness). Moreover, for some j such that $x_j^* = 0$, the quantity

$$\bar{c}_j = c_j - u^{*T} A_j$$

is the *reduced cost* of variable x_j and yields the marginal increase in the objective function if x_j^* moves away from to its lower bound. Thus, if a given known feasible solution with value $z^{UB} \geq z^*$ is available to the original MILP, the *reduced cost fixing* technique consists of fixing $x_j^* = 0$ if

$$z_P + \bar{c}_j \geq z^{UB}, \tag{RC}$$

since any solution with $x_j^* > 0$ can never improve upon the existing upper bound z^{UB}. We refer to Wolsey [14] and Nemhauser & Wolsey [11] for the formal proofs of correctness.

We remark in passing that the condition (RC) can be generalized to established more general bounds on a variable. That is, we can use the reduced cost $\bar{c}_j$ to deduce values l_j and u_j such that either $x_j^* \geq l_j$ or $x_j^* \leq u_j$ in any optimal solution (see, e.g., [8]). In this paper we restrict our attention to the classical case described above.

3 Reduced Cost Fixing Consistency

The duals variables u^* for the computation of (RC) can be obtained with very little computational effort after finding an optimal solution x^* to (P) (e.g., they are computed simultaneously to x^* when using the Simplex method). In practical known implementations concerning MILP presolving and CP propagation methods, the reduced cost fixing is typically carried out using the single u^* computed after solving every LP relaxation [5,10].

Note, however, that (D) may contain multiple optimal solutions, each potentially yielding a different reduced cost $\bar{c}_j$ that may or may not satisfy condition (RC). We therefore do not need to restrict our attention to a unique x^*, and we can potentially improve the number of variables that are fixed if we refocus our attention to the dual spaced instead. To cast this formally, we now define the concept of *reduced cost fixing consistency* as follows.

Definition 1. *Let P_S be an MILP model with linear programming relaxation (P) and its corresponding dual (D). The model P_S is reduced cost fixing consistent (or RCF-consistent) with respect to an upper bound z^{UB} to P_S if*

(a) No variables of P_S are fixed to 0; and
(b) For any optimal solution u^* to (D) and its associated reduced cost $\overline{c}$ vector, condition (RC) is never satisfied, i.e., $z_P + \overline{c}_j < z^{UB}$ for all $j = 1, \ldots, n$.

If a model is RCF-consistent according to Definition 1, then it is not possible to fix any variable x_j via reduced costs. Specifically, condition (a) of the definition implies that the model cannot be simplified any further if some variable is known to be fixed to zero (which could potentially strengthen the LP relaxation and yield new reduced costs). For condition (b), recall that for any optimal dual u^* and optimal primal x^*, the feasibility constraints and strongly duality implies

$$u^T Ax \leq c^T x = u^T b \leq u^T Ax$$

and therefore the inequality holds as equality throughout. We then have $u^T b = u^T Ax$ or $u^T(b - Ax) = 0$, i.e., complementary slackness. Any optimal dual therefore suffices to verify condition (RC), which does not depend on the actual point x^* but only on the optimal solution value z_P.

We now show that any MILP formulation can be efficiently converted into a RCF-consistent formulation.

Theorem 1. *An RCF-consistent model with respect to an upper bound z^{UB} can be derived from a MILP P_S in weakly polynomial time.*

Proof. Given an MILP model P_S and the primal (P) and dual (D) of its associated linear programming relaxation, the set of optimal dual solution coincides with the polyhedral set $\mathcal{D} = \{u \in \mathbb{R}^m : u^T A \geq c,\, u^T b = z_P\}$. Thus, a variable x_j can be fixed to zero if the optimal solution $\overline{c}_j^*$ of the problem

$$\overline{c}_j^* = \max\{c_j - u^T A_j : u \in \mathcal{D}\} \qquad (D_j)$$

is such that $z_P + \overline{c}_j^* \geq z^{UB}$. If that is the case, we can then generate a new MILP model by fixing x_j to zero and adapting all constraints appropriately. We can repeat this procedure for variables $x_1, \ldots, x_n$ in order. If a variable x_j is fixed to zero, we need to restart the procedure from the first unfixed variable as the reduced costs may change.

For the complexity, note that every restart require solving the primal LP to obtain z_P and solving at most n models (D_j). Rewriting the model to remove a variable for which $x_j = 0$ can be done in linear time in n and m. This means that the complexity of the procedure is dominated by the cost of solving $O(n^2)$ LP models, each of which can be done in weakly polynomial time [14]. □

Notice that Theorem 1 only guarantees that the resulting model is *minimal* with respect to variables that can be removed according to the reduced cost fixing. For instance, different orderings considered in the proof of Theorem 1 may lead to additional fixing. The question whether there exists an unique *minimum* RCF-consistency model is still open.

4 Approximate Consistency

Based on Theorem 1, one could derive an RCF-consistent formulation by solving $O(n^2)$ LP models, which is impractical when the model has any reasonably large number of variables. In this section we propose a simple alternative model that exploits the underlying concept of searching in the space of dual variables for maximizing propagation.

Namely, as opposed to solving an LP for each variable, we will search for the dual variables that maximize the number of fixed variables. This can be written as the following MILP model:

$$\max \quad \sum_{i=1}^{n} y_i \tag{A-RCF}$$

$$\text{s.t.} \quad u^T A \leq c \tag{1}$$

$$u^T b = z_P \tag{2}$$

$$z_P + (c - u^T A) \geq z^{UB} y - (\mathbf{1} - y)M \tag{3}$$

$$u \geq 0 \tag{4}$$

$$y \in \{0,1\}^n \tag{5}$$

In the model (A-RCF) above, we are searching for the dual variables u that maximize the number of variables fixed. Specifically, we will define a binary variable y_i in such a way that $y_i = 1$ if and only if we fix x_j to 0.

To enforce this, let M be a sufficiently large number, and $\mathbf{1}$ an n-dimensional vector containing all ones. Constraints (1), (2), and (4) ensure that u^* is dual optimal. If $y_i = 1$, then inequality (3) reduces to condition (RC) and the variable should be fixed. Otherwise, the right-hand side of (3) is arbitrary small (in particular to account for arbitrarily small negative reduced costs). Finally, constraint (5) defines the domain of the y variable and the objective maximizes the number of variables fixed.

The model (A-RCF) does not necessarily achieve RCF-consistency as it yields a single reduced cost vector. However, our experimental results indicate that the model can be solved quite efficiently and yields interesting bounds. Notice also that any feasible solution to (A-RCF) corresponds to a valid set of variables to fix, and hence any solutions found during search can be used to our purposes.

5 Preliminary Numerical Study

We present a preliminary numerical study of our technique on the MIPLIB 2010 benchmark[1]. We implemented our technique using ILOG CPLEX Optimization Studion 12.6.3. All experiments ran on a single thread of an Intel Core i7 CPU 3.40GHz with 8.00 GB RAM. We performed tests on all instances of the MIPLIB, but kept only the ones who could be solved within 30,000 seconds by all

[1] http://miplib.zib.de/miplib2010-benchmark.php

methods or that would not exceed our machine memory limit, which resulted in 64 instances in total.

In our experiments, we solved model (A-RCF) at the root node and kept the resulting dual solution u^R. Once the optimality gap reached 1% (hopefully due to a better z^{UB}), we rechecked condition (RC) using u^R. The procedure was then repeated for geometrically smaller gaps, i.e., 0.50%, 0.25%, and so on.

The implementation required the use of *callbacks* in CPLEX, which fundamentally changes its dynamic behaviour. To try to make a better comparison, we have included a "dummy" callback for CPLEX that computes the variables x_j to be fixed, but adds an innocuous inequality $x_j \geq 0$ instead. The additional time to compute such constraint was subtracted in the corresponding total times in the case that our procedure should not be considered. We have also included the times for CPLEX default (with one thread) for comparison purposes.

The solution times and number of nodes obtained for our runs are summarized in Table 1 and detailed in Table 2. The columns with `CPLEX`, A-RCF, and `Default CPLEX` represent the results for the modified CPLEX with innocuous callbacks, CPLEX modified with our procedure, and the default CPLEX, respectively. We also separate the problems into general mixed-integer linear programs ("MILP instances") and binary problems ("BP").

Table 2 shows that, on average, a more careful selection of the dual variables can lead to performance improvement on a variety of instances (such as `aflow40b`, `rol3000`, `biella` and `rococo`), but the converse may be true as well (such as `neos-476283`). However, for the majority of the cases, the method either improves upon the modified CPLEX version or has a little detrimental effect.

We also show some statistics on the number of variables that were fixed with respect to the classical reduced cost fixing method... **todo**.

Table 1: Summary results

	`CPLEX`		A-RCF		`Default`	
	Time	Nodes	Time	Nodes	Time	Nodes
Arithmetic mean	2,698	580,138	1,838	471,324	528	189,928
Geometric mean	165	14,299	189	12,226	79	8,168

6 Conclusion

In this paper, we expanded upon the classical reduced cost fixing method by searching the dual variables that maximize propagation. We investigated theoretical properties of finding the optimal set of duals, and performed experiments on an model that maximizes the number of variables that can be fixed. Preliminary results on a set of benchmark instances from MIPLIB 2010 showed that the approach could be promising in terms of improving solution times.

Table 2: General Results

MILP instances						
	CPLEX		A-RCF		Default	
Instance	Time	Nodes	Time	Nodes	Time	Nodes
30n20b8	6	7	6	7	6	322
aflow40b	668	232354	352	137891	108	27,066
beasleyC3	1993	526842	997	215350	5	536
bienst2	392	57193	391	57191	53	72,854
binkar10_1	6	3516	23	2871	3	2,376
core2536-691	28	289	154	288	37	1,080
biella1	489	13000	258	6206	195	2,051
csched010	1913	431359	1900	437579	3506	229,856
danoint	2589	835753	2589	835753	1029	512,097
dfn-gwin-UUM	46	25453	40	21421	49	19,790
enlight13	0.05	1	0.05	1	1	1
gmu-35-40	164	305396	98	240535	105	746,177
lectsched-4-obj	3	580	3	580	5	595
mcsched	208	63846	204	63830	47	12,167
mik-250-1-100-1	823	9465262	575	7133239	4	29,115
mzzv11	5	4	134	1	21	665
n4-3	23	772	17	499	29	1,760
neos-1396125	61	4037	60	3997	22	1,011
neos13	42	6392	42	6392	55	5,850
neos-1601936	238	5261	351	10042	79	543
neos-476283	60	332	165	346	69	1,037
neos-916792	3	225	3	222	212	69,649
neos-934278	68	20	68	20	153	30
net12	1078	3339	1054	3326	170	690
newdano	8674	965768	8598	965738	1061	1,590,405
noswot	35	263427	35	263427	36	204,651
ns1208400	2512	60531	2512	60531	633	39,053
ns1830653	105	10614	103	10612	61	4,926
pg5_34	175	93417	188	103731	282	171,451
pigeon-10	796	6,375,886	796	6,375,886	684	5,747,874
pw-myciel4	115	20187	109	19446	23	2,942
qiu	9	3255	9	3255	5	1,877
rail507	26	3979	26	3979	69	2,917
ran16x16	61	62673	57	61564	39	27,345
rmatr100-p10	23	914	369	966	23	891
rmatr100-p5	42	677	36	742	29	684
rocII-4-11	113	90351	34	21502	168	23,925
rococoC10-001000	123	13621	66	10053	115	15,795
roll3000	41643	5878758	25195	5820934	19	3,042
satellites1-25	2441	26749	2648	35259	218	1,989
sp98ir	34	9677	31	5418	24	4,477
timtab1	955	1759829	892	1738307	403	389,468
zib54-UUE	4442	59197	4384	59173	2025	18,823
BP instances						
acc-tight5	22	22	487	487	98	2,339
air04	5	165	131	103	5	509
bab5	13,026	8,260	813,597	402,386	2472	60,138
bnatt350	9,959	9,959	41,678	41,678	5809	63,016
cov1075	7	7	3,875	3,875	5	2,338
eil33-2	50	50	16,697	17,203	38	5,222
eilB101	80	71	19,289	17,528	120	8,414
iis-100-0-cov	6,299	6,299	1,017,192	1,017,192	1175	177,293
iis-bupa-cov	6,042	6,042	525,287	525,287	3569	343,293
iis-pima-cov	335	335	22,371	22,371	323	15,427
n3div36	51,793	17,910	5,589,084	1,913,476	6,274	646,994
neos-1109824	2	1	417	74	2	512
neos-1337307	18	229	1,241	1241	8	500
neos18	14	14	7,574	7,574	14	5,933
neos-849702	4,889	4,889	708,570	708,570	297	32,610
ns1688347	8	8	383	383	28	1,393
opm2-z7-s2	137	308	2,005	1,984	83	1,164
reblock67	1,590	1,590	212,800	212,800	265	245,356
rmine6	2,309	2,190	190,188	174,590	995	574,430
sp98ic	1,306	1,310	146,788	120,583	585	135,988
tanglegram2	5	5	3	3	2	3

References

1. Egon Balas and Clarence H. Martin. Pivot and complement–a heuristic for 0-1 programming. *Management Science*, 26(1):86–96, 1980.
2. David Bergman, Andre A. Cire, and Willem-Jan van Hoeve. *Improved Constraint Propagation via Lagrangian Decomposition*, pages 30–38. Springer International Publishing, Cham, 2015.
3. E Robert Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. Mip: Theory and practice—closing the gap. In *System modelling and optimization*, pages 19–49. Springer, 2000.
4. Vasek Chvátal. *Linear programming.* Freeman, New York (N. Y.), 1983. Reprints : 1999, 2000, 2002.
5. Filippo Focacci, Andrea Lodi, and Michela Milano. Cost-based domain filtering. In *International Conference on Principles and Practice of Constraint Programming*, pages 189–203. Springer, 1999.
6. Filippo Focacci, Andrea Lodi, Michela Milano, and Daniele Vigo. Solving tsp through the integration of or and cp techniques. *Electronic notes in discrete mathematics*, 1:13–25, 1999.
7. Filippo Focacci, Michela Milano, and Andrea Lodi. Solving tsp with time windows with constraints. In *Proceedings of the 1999 international conference on Logic programming*, pages 515–529. Massachusetts Institute of Technology, 1999.
8. Diego Klabjan. A new subadditive approach to integer programming. In *Integer Programming and Combinatorial Optimization*, pages 384–400. Springer, 2002.
9. Andrea Lodi. Mixed integer programming computation. In *50 Years of Integer Programming 1958-2008*, pages 619–645. Springer, 2010.
10. Ashutosh Mahajan. Presolving mixed–integer linear programs. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.
11. George L Nemhauser and Laurence A Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.
12. Meinolf Sellmann. *Theoretical Foundations of CP-Based Lagrangian Relaxation*, pages 634–647. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
13. Erlendur S. Thorsteinsson and Greger Ottosson. Linear relaxations and reduced-cost based propagation of continuous variable subscripts. *Annals of Operations Research*, 115(1):15–29, 2002.
14. Laurence A Wolsey. Integer programming john wiley & sons. *New York, NY*, 4, 1998.
15. Tallys Yunes, Ionuţ D. Aron, and J. N. Hooker. An integrated solver for optimization problems. *Operations Research*, 58(2):342–356, 2010.