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, Toronto, Canada
omid.saneibajgiran@rotman.utoronto.ca
[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montreal, Canada
[3] Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, Canada

Abstract. Reduced-cost-based filtering in constraint programming and variable fixing in integer programming are techniques which allow to cut out part of the solution space which cannot lead to an optimal solution. These techniques are, however, dependent on the dual values available at the moment of pruning. In this paper, we investigate the value of picking a set of dual values which maximizes the amount of filtering (or fixing) that is possible. We test this new variable-fixing methodology for arbitrary mixed-integer linear programming models. 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 which subset of inequalities and variables may be responsible for the infeasibility [12], if that is the case. These simplification methods alone reduce solution times by half in state-of-the-art solvers such as CPLEX, SCIP, or Gurobi [4], thereby constituting an important tool in the use of mixed-integer linear programming (MILP) in practical real-world problems [11].

In this paper we investigate a new simplification technique that expands upon the well-known *reduced cost fixing method*, first mentioned by Balas and

D. Salvagnin and M. Lombardi (Eds.): CPAIOR 2017, LNCS 10335, pp. 221–228, 2017.
DOI: 10.1007/978-3-319-59776-8_18

Martin [2] 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 linear change in the objective function when the value of the variable in that solution is increased [5]. 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 [4,11,12] and plays a key role in a variety of propagation methods for global constraints in CP [7–9]. 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 strengths 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 [3,17,19]).

However, in all cases typically only *one* reduced cost per variable is considered, that is, the one obtained after solving the LP relaxation of a MILP. In theory, any set of feasible duals provides valid reduced costs that may lead, in turn, to quite different variable bound tightenings. This question was originally raised by Sellmann [16], who demonstrated that not only distinct dual vectors would result in significantly different filtering behaviors, but that potentially sub-optimal dual vectors could yield much more pruning than the optimal ones.

Our goal in this work is to investigate the potential effect of *picking* the dual vector that maximizes reduced-cost-based filtering. By doing so, we revisit the notion of *relaxed consistency* for reduced costs fixing; that is, we wish to influence the choice of the dual values given by a relaxation so as to maximize the amount of pruning that can be performed. We view the proposed techniques as a first direction towards answering some of the interesting questions raised in the field of *CP-based Lagrangian relaxation* [3,16], in particular related to how to select the dual variables (or, equivalently, the Lagrangian multipliers) to maximize propagation.

The contribution of this paper is to formulate the problem of finding the dual vectors that maximize the number of reductions as an optimization problem defined over the space of optimal (or just feasible) dual values. We compare this approach to achieving full *relaxed consistency*, which can be obtained by solving a large (but polynomial) number of LP problems. The resulting technique can be seamlessly incorporated into existing solvers, and preliminary results over the MIPLIB indicate that it can significantly reduce solution time as well as the size of the branching tree when proving the optimality of a primal bound. We hope to motivate further research on the quality of the duals used within both ILP and CP technology.

For the sake of clarity and without loss of generality, the proposed approaches will be detailed in the context of integer linear programs (ILPs), i.e., where all variables are integers, as opposed to mixed-integer linear programming models. This technique is also applicable in the context of CP, if one can derive a (partial) linear relaxation of the model [15].

The paper is organized as follows. Section 2 introduces the necessary notation and the basic concepts of reduced cost fixing and the related consistency notions. Next, we discuss one alternative to obtain an approximate consistency in Sect. 3. Finally, we present a preliminary numerical study in Sect. 4 and conclude in Sect. 5.

2 Reduced Cost Fixing and Relaxed-Consistency

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 ILP 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

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

where $u \in \mathbb{R}^m$ is the vector of *dual variables*. We assume for exposition that P_S, (P), and (D) are 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

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

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

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

since any solution with $x_j^* > 0$ can never improve upon the existing upper bound z^{UB}. We remark in passing that the condition (RCF) can be generalized to establish 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., [10]). In this paper we restrict our attention to the classical case described above. We refer to Wolsey [18] and Nemhauser and Wolsey [13] for the formal proofs of correctness.

The dual 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 the most of practical known implementations concerning ILP presolving and CP propagation methods, the reduced cost fixing is typically carried out using the single u^* computed after solving every LP relaxation [7,12]. Note, however, that (D) may contain multiple optimal solutions, each potentially yielding a different reduced cost $\overline{c}_j$ that may or may not satisfy condition (RCF).

One therefore does not need to restrict its attention to a unique u^*, and can potentially improve the number of variables that are fixed if the whole dual space is considered instead. This would mean that if there exists a reduced cost $\overline{c}_j$ such that variable x_j can be fixed to 0 with respect to z^{UB}, then there exist a dual vector u such that $u^T b + (c_j - u^T A_j) \geq z^{UB}$. This was first demonstrated by [16] in the context of CP-based Lagrangian Relaxation, which also pointed out that often more filtering occurs when u is not an optimal dual vector and therefore we might have that $u^T b < z_P$.

Achieving the notion of *Relaxed Consistency*, as defined in [6], is thus quite consuming in such a condition. This form of consistency can be casted, in the context of ILP, as follows.

Definition 1. *Let P_S be an ILP model with linear programming relaxation (P) and its corresponding dual (D). The model P_S is relaxed consistent (or relaxed-P-consistent) with respect to an upper bound z^{UB} to P_S if for any dual feasible u^* to (D) and its associated reduced cost $\overline{c}$ vector, condition (RCF) is never satisfied, i.e., $z_P + \overline{c}_j < z^{UB}$ for all $j = 1, \ldots, n$.*

If a model is *relaxed consistent* according to Definition 1, then it is not possible to fix any variable x_j via reduced costs.

Consistent with the theory presented in [6,16], any ILP formulation can be efficiently converted into a relaxed consistent formulation in polynomial time. Given an ILP 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 $\mathcal{D} = \{u \in \mathbb{R}^m : u^T A \geq c, u \geq 0\}$. Thus, a variable x_j can be fixed to zero if the optimal solution $\overline{c}_j^*$ of the problem $\overline{c}_j^* = \max\{c_j - u^T A_j : u \in \mathcal{D}\}$ is such that $z_P + \overline{c}_j^* \geq z^{UB}$. This means that the complexity of the procedure is dominated by the cost of solving $O(n)$ LP models, each of which can be done in weakly polynomial time [18].

3 Dual Picking for Maximum Reduced Cost Fixing

Establishing *relaxed consistency* by solving $O(n)$ LP models is impractical when a model has any reasonably large number of variables. We thus propose a simple alternative model that exploits the underlying concept of searching in the dual space for maximizing filtering. Namely, as opposed to solving an LP for each

variable, we will search for the dual variables that together maximize the number of variables that can be fixed. This can be written as the following MILP model:

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

$$\begin{aligned}
\text{s.t.} \quad & u^T A \leq c && (1)\\
& u^T b = z_P && (2)\\
& u^T b + (c_j - u^T A_j) \geq z^{UB} - (\mathbf{1} - y_j)M \quad \forall j && (3)\\
& u \geq 0 && (4)\\
& y \in \{0,1\}^n && (5)
\end{aligned}$$

In the model (DP-RCF) above, we are searching for the dual variables u, on the optimal dual face, 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 it allows to deduce that x_j can be fixed to 0.

To enforce this, let M be a sufficiently large number, and $\mathbf{1}$ an n-dimensional vector containing all ones. Constraints (1), and (4) ensure that u^* is dual feasible. If $y_i = 1$, then inequality (3) reduces to condition (RCF) and the associated x_i should be fixed to 0. Otherwise, the right-hand side of (3) is arbitrary small (in particular to account for arbitrarily small negative reduced costs). Constraint (2) enforces strong duality and the investigation of optimal dual vectors only, it can be omitted in order to explore the whole dual feasible space (as sub-optimal dual vectors can perhaps filter more [16]). Finally, constraint (5) defines the domain of the y variable and the objective maximizes the number of variables fixed.

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

4 Preliminary Numerical Study

We present a preliminary numerical study of our technique on a subset of the MIPLIB 2010 benchmark [1]. All experiments were performed using IBM ILOG CPLEX 12.6.3 on a single thread of an Intel Core i7 CPU 3.40 GHz with 8.00 GB RAM.

Our goal for these experiments is twofold: We wish first to evaluate the filtering achieved by the dual picking models (DP-RCF) in comparison to the full relaxed-consistent model, and next verify what is the impact of fixing these variables in CPLEX when proving optimality. For the first criteria, we considered the number of fixed variables according to three different approaches:

1. The model (DP-RCF) with a time limit of 10 min, denoted by `DP`.
2. A modified version of the model (DP-RCF) without constraint (2), i.e., we increased our search space by considering any *feasible* dual solution, also fixing a time limit of 10 min. We denote this approach by `DP-M`.
3. Solving the $O(n)$ LP models to achieve relaxed consistency in view of Definition 1, with no time limit. This method provides the maximum number of variables that can be fixed thought RCF and thus an upperbound for both `DP` and `DP-M`, it is denoted by `RCC` (*relaxed reduced-cost consistency*).

In all the cases above, we considered the optimal solution value of the instance as the upper bound z^{UB} for the (RCF) condition, which results in the strongest possible filtering for a fixed set of duals.

Next, to assess the impact of fixing variables in the ILP solution process, we ran the default CPLEX for each approach above, specifically setting the fixed variables to zero and providing the optimal solution value of each instance to the solver. We have also ran default CPLEX without any variable fixing and with the optimal value as an input, in order to evaluate the impact of variable fixing in proving the optimality of a particular bound.

As a benchmark we considered all "easy" instances of the MIPLIB that could be solved within our memory or limit of 8 GB and a time limit of 60,000 s. We have also eliminated all instances where *relaxed consistency* could not fix any variable. This resulted in 36 instances.

The results are depicted in Table 1, where `Aux` stands for the CPU time required to solve the auxiliary model (DP-RCF) and `Vars` is the number of variables which could be fixed. Moreover, `Cons` and `Vars` in the `Dimension` category indicate the number of constraints and decision variables of each instance, respectively, and `DC` shows the number of variables that default CPLEX can fix as a result of the final dual solution calculated by ourselves. Omitted rows indicates the auxiliary problem reached its time limit. Due to space restrictions, instances *neos-16...*, *neos-47...*, *neos-93...*, *neos-13...*, *neos-84...*, *rmatr-p5*, *rmatr-p10*, *core253...*, *neos-93...*, and *sat...* represent instances *neos-1601936*, *neos-476283*, *neos-934278*, *neos-1337307*, *neos-849702*, *rmatr100-p5*, *rmatr100-p10*, *core2536-691*, *neos-934278*, and *satellites1-25*, respectively.

We first notice that achieving full *relaxed consistency* is quite time consuming and not practical with respect to the default solution time of CPLEX. When looking at both `DP` models, it is obvious that restricting the search to the optimal dual face, rather than the whole dual feasible space, yields practically the same amount of filtering while being orders of magnitude faster. In fact, in many cases the (DP-RCF) model can be solved in less than a second.

To determine whether such filtering is worth the extra effort, we compare the solution time of `DP` against the default solution time of CPLEX. For each instance we compute the speedup as the solution time of CPLEX divided by the sum of both solution and dual picking (i.e., solving (DP-RCF)) time of DP. We then compute the geometric mean of all speedups, which yields an average speed up of 20% when using our dual picking methodology over the default CPLEX.

Table 1. General results

Instance	Dimension		DC	CPLEX Default		DP				DP-M				RCC			
	Cons	Vars	Vars	Time	Nodes	Time	Nodes	Aux	Vars	Time	Nodes	Aux	Vars	Time	Nodes	Aux	Vars
MILP instances																	
30n20b8	578	18380	0	3	260	3	260	0.5	7282					3	90	494	13603
aflow40b	1442	2728	33	187	17461	1961	200477	0.03	499					281	21635	42	532
binkar10_1	1026	2298	165	7	1567	8	2135	0.1	200					5	902	51	281
core253...	2539	15293	2494	38304	239195	2414	8816	600	3046					4464	72010	79173	4818
biella1	1203	7328	219	942	1477	257	1107	52	429					204	1089	18220	1540
gmu-35-40	424	1205	0	68	351765	68	351765	0.02	0	124	≈6e5	9	485	26	94209	19	493
mik-...	151	251	50	1	2205	0.7	1115	0.01	50	0.7	1115	0.5	50	0.7	1115	0.8	50
mzzv11	9499	10240	29	19	57	17	46	26	213					26	329	18679	678
neos13	20852	1827	8	6	485	5	399	1	8	5	399	68	8	6	7	588	179
neos-16...	3131	4446	867	327	4264	137	2259	2	1766					377	7549	7747	1782
neos-47	10015	11915	36	1054	1565	195	1129	2	69					195	1129	38293	69
neos-93...	11495	23123	15843	430	97	74	12	3	15981					132	39	86254	16088
net12	14021	14115	0	162	440	16	50	1	32					150	880	3057	334
ns1208400	4289	2883	2	21	860	21	860	0.08	286	21	860	107	286	32	1465	1286	592
ns1830653	2932	1629	0	208	24606	74	9185	0.2	850	217	31211	952	853	110	5203	65	915
pw-myciel4	8164	1059	0	0.4	17	0.4	117	0.05	0	0.4	117	6	0	1	580	54	160
rmatr-p10	7260	7359	100	20	455	27	345	0.2	100					27	345	3260	100
rmatr-p5	8685	8784	100	3	2	5	6	2	100					5	6	9601	100
rococo...	1293	3117	0	523	13193	4475	54077	0.3	511					531	15979	35	553
roll3000	2295	1166	0	2645	177058	2645	177058	0.08	120					1015	61271	50	149
sat...	5996	9013	99	220	442	25	497	2	2499					11	758	13845	4989
sp98ir	1531	1680	110	244	8607	241	13057	1.5	285					204	18659	86	524
timtab1 171	397	0	2393	≈1e6	2393	≈1e6	0.02	13	2393	≈1e6	4	13	2393	≈1e6	1	13	
Binary-Only Problems																	
acc-tight5	3052	1339	1339	97	2262	97	2262	0	1339	97	2262	0	1339	97	2262	0	1339
air04	823	8904	0	7	493	7	46	6	64					5	54	1500	126
bab5	4964	21600	0	8313	49311	2903	12717	369	373					883	7904	18563	2309
eil33-2	32	4516	1004	112	15041	18	6825	6	1035					0.3	305	33	4013
eilB101	100	2818	45	595	26753	318	21666	1	125					84	10309	115	1588
n3div36	4484	22120	4755	57917	≈2.7e6	57017	≈2.7e6	1	4755					3518	83900	6568	6372
neos-13...	5687	2840	14	9	300	6	110	5	77					5	30	1098	150
neos18	11402	3312	0	0.4	1	0.4	1	0.08	84					0.4	1	142	84
neos-84...	1041	1737	1737	105	8223	105	8223	0	1737	105	8223	0	1737	105	8223	0	1737
ns1688347	4197	2685	0	8	260	5	1310	0.25	267					1	1	56	716
opm2-z7-s2	31798	2023	0	230	1000	230	1000	1	0	230	1000	35	0	25	260	2484	459
rmine6	7078	1096	0	69	15233	31	5920	0.25	1	31	5920	10	1	52	10497	116	66
sp98ic	825	10894	6902	29041	149629	40306	466555	62	7098					345	35647	4724	8488

5 Conclusion

In this paper, we revisited the notion of reduced-cost based filtering and variable fixing, which are known to be dependent on the available dual information. We defined the problem of identifying the set of dual values that maximize the number of variables which can be fixed as an optimization problem. We demonstrated that looking for a good set of such dual on the optimal dual face is considerably faster and filter almost as many variables as when considering the full feasible dual space. In many cases fixing more variable lead to a reduced search tree that can be explored faster. However, in a good number of cases, solution time increases when more variables are fixed, which is probably due to the fact that early in the tree the search takes a different path.

Future research will consider dual picking during search, so as to try to fix variables when the relative gap becomes small enough in a subtree, as well as applying the techniques in the context of constraint programming.

References

1. MIPLIB2010. http://miplib.zib.de/miplib2010-benchmark.php
2. Balas, E., Martin, C.H.: Pivot and complement-a heuristic for 0–1 programming. Manage. Sci. **26**(1), 86–96 (1980)
3. Bergman, D., Cire, A.A., Hoeve, W.-J.: Improved constraint propagation via lagrangian decomposition. In: Pesant, G. (ed.) CP 2015. LNCS, vol. 9255, pp. 30–38. Springer, Cham (2015). doi:10.1007/978-3-319-23219-5_3
4. Bixby, E.R., Fenelon, M., Gu, Z., Rothberg, E., Wunderling, R.: MIP: theory and practice — closing the gap. In: Powell, M.J.D., Scholtes, S. (eds.) CSMO 1999. ITIFIP, vol. 46, pp. 19–49. Springer, Boston, MA (2000). doi:10.1007/978-0-387-35514-6_2
5. Chvátal, V.: Linear Programming. Freeman, New York (1983). Reprints: (1999), (2000), (2002)
6. Fahle, T., Sellmann, M.: Cost-based filtering for the constrained knapsack problem. Ann. Oper. Res. **115**, 73–93 (2002)
7. Focacci, F., Lodi, A., Milano, M.: Cost-based domain filtering. In: Jaffar, J. (ed.) CP 1999. LNCS, vol. 1713, pp. 189–203. Springer, Heidelberg (1999). doi:10.1007/978-3-540-48085-3_14
8. Focacci, F., Lodi, A., Milano, M., Vigo, D.: Solving TSP through the integration of OR and CP techniques. Electron. Notes Discrete Math. **1**, 13–25 (1999)
9. Focacci, F., Milano, M., Lodi, A.: Solving TSP with time windows with constraints. In: Proceedings of the 1999 International Conference on Logic programming, Massachusetts Institute of Technology, pp. 515–529 (1999)
10. Klabjan, D.: A new subadditive approach to integer programming. In: Cook, W.J., Schulz, A.S. (eds.) IPCO 2002. LNCS, vol. 2337, pp. 384–400. Springer, Heidelberg (2002). doi:10.1007/3-540-47867-1_27
11. Lodi, A.: Mixed integer programming computation. In: Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A. (eds.) 50 Years of Integer Programming 1958–2008, pp. 619–645. Springer, Heidelberg (2010)
12. Mahajan, A.: Presolving mixed-integer linear programs. In: Wiley Encyclopedia of Operations Research and Management Science (2010)
13. Nemhauser, G.L., Wolsey, L.A.: Integer Programming and Combinatorial Optimization (1988)
14. Chichester, W., Nemhauser, G.L., Savelsbergh, M.W.P., Sigismondi, G.S.: Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin, vol. 20, pp. 8–12 (1992)
15. Refalo, P.: Linear formulation of constraint programming models and hybrid solvers. In: Dechter, R. (ed.) CP 2000. LNCS, vol. 1894, pp. 369–383. Springer, Heidelberg (2000). doi:10.1007/3-540-45349-0_27
16. Sellmann, M.: Theoretical foundations of CP-based lagrangian relaxation. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 634–647. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30201-8_46
17. Thorsteinsson, E.S., Ottosson, G.: Linear relaxations and reduced-cost based propagation of continuous variable subscripts. Ann. Oper. Res. **115**(1), 15–29 (2002)
18. Wolsey, L.A.: Integer Programming, vol. 4. Wiley, New York (1998)
19. Yunes, T., Aron, I.D., Hooker, J.N.: An integrated solver for optimization problems. Oper. Res. **58**(2), 342–356 (2010)