ECAI 2020
G.D. Giacomo et al. (Eds.)


doi:10.3233/FAIA200109

# Analysis of Reduced Costs Filtering for Alldifferent and Minimum Weight Alldifferent Global Constraints

**Guillaume Claus, Hadrien Cambazard** and **Vincent Jost** [1]

**Abstract.** An incomplete filtering technique known as *variable fixing* has been used in integer programming for a long time. It relies on the reduced costs of the variables given by an optimal dual solution of the linear relaxation. Reduced-costs are used to detect some of the 0/1 variables that must be fixed to either 0 or 1 in any solution improving the best known. Reduced cost based filtering was introduced in CP for a global constraint referred to as MINIMUM WEIGHT ALLDIFFERENT and to the best of our knowledge, no analysis of this filtering technique has ever been performed. We therefore propose an analysis of reduced costs filtering for this constraint, showing that arc-consistency can be achieved with reduced-costs of $n$ dual solutions and that this bound is sharp. For ALLDIFFERENT, a single dual solution is enough. From a practical side, our end goal is the design of incomplete but anytime primal-dual filtering approaches. We illustrate this idea on the MINIMUM WEIGHT ALLDIFFERENT where a near-complete filtering can be done in shorter times.

## 1 Introduction

Mixed integer programming (MIP) and Constraint Programming (CP) have often been combined in the past to take advantage of the complementary strengths of the two frameworks. Many approaches have been proposed to benefit from their modeling and solving capabilities [6, 21, 19, 3, 2]. A typical integration of the two approaches is to use the linear relaxation of the entire problem in addition to the local consistencies enforced by the CP solver. The relaxation can detect infeasibility and is often added to provide a bound on the objective.

A number of previous works have also proposed to use the linear relaxation for filtering the domains in a constraint programming framework [18, 19, 3, 2, 10]. Based on the relaxation, filtering can be performed using a technique referred to as reduced cost based filtering [10, 13]. It is a specific case of cost-based filtering [9] that aims at filtering out values leading to non-improving solutions. It originates from *variable fixing* [16] which is performed in MIP to detect some 0/1 variables that must be fixed to either 0 or 1 in any solution improving the best known. *Variable fixing* relies on the reduced costs of the variables given by an optimal dual solution of the linear relaxation. It is known to be incomplete because it strongly depends on the specific dual solution used. Alternatively, it was recently shown in [11] that a complete filtering, namely arc-consistency, can be achieved with a linear relaxation when the problem considered is a satisfaction problem with an ideal integer programming formulation. Such formulations can be found for a number of common global constraints such as ELEMENT, ALLDIFFERENT, GLOBALCARDINALITY or GEN-SEQUENCE [19, 11]. The approach does not apply to global constraints involving a cost variable such as MINIMUMWEIGHTALLDIFFERENT [7, 10] even though it has an ideal LP formulation. A natural extension to the work [11] is to handle an objective function i.e a cost variable from the constraint point of view. We are therefore interested in the design of filtering algorithms based on linear programming for polynomial global constraints with a cost variable. Note that when an ideal LP formulation is available for the constraint, a naive approach, typically used in practice when checking or designing propagators is to solve one LP for each variable-value pair.

Since the approach of [11] does not easily extend, we go back to reduced cost based filtering and investigate the specific case of the MINIMUMWEIGHTALLDIFFERENT global constraint (referred to as MINWALLDIFF for short in the rest of the paper). This constraint enforces $n$ distinct values to be assigned to $n$ variables so that the cost of the assignment is no more than a given upper-bound. Assigning to a variable $X_i$ a value $j$ of its domain incurs a cost $c_{ij} \in \mathbb{N}$ and the overall cost is the sum of all individual assignment costs. This constraint is related to the assignment problem for which a well-known LP ideal formulation is available. Interestingly, cost-based filtering was introduced in CP with the MINWALLDIFF [10] and reduced costs of the linear relaxation were used to perform filtering. An arc-consistency algorithm is first given in [20] for the more general case of the GLOBALCARDINALITY constraint with costs. Later on, [22] focuses on MINWALLDIFF and achieves arc-consistency in $\mathcal{O}(n(d + mlog(m)))$, where $n$ denotes the number of variables, $m$ is the cardinality of the union of all variable domains, and $d$ denotes the sum of the cardinalities of the variable domains. Let's give some details about reduced costs filtering to properly state the results of the present paper. In general, the consistency of a given value $j$ of a variable $X_i$ is established by computing the minimum increase of the optimal objective due to the assignment $X_i = j$. The optimal value of the problem restricted with $X_i = j$ is referred to as the $(i, j)$-optimal value and denoted $z^*_{|ij}$. Value $j$ of $X_i$ is inconsistent if $z^*_{|ij}$ is greater than the maximum cost allowed denoted $\overline{z}$. A typical lower bound of $z^*_{|ij}$ is the LP reduced cost, $r_{u^*,ij}$ available from an optimal dual solution $u^*$ of the linear relaxation of the assignment problem (namely $z^* + r_{u^*,ij} \leq z^*_{|ij}$). It was used to perform an incomplete filtering in [10]. However, the value of $r_{u,ij}$ depends on the dual solution $u$ found and greatly varies in practice from one solution to another. We are now ready to state the results presented in this paper.

We prove that there always exists an optimal dual solution $u^*$ such that the reduced cost $r_{u^*,ij}$ provides the $(i, j)$-optimal value (*i.e.* such that $z^* + r_{u^*,ij} = z^*_{|ij}$). Moreover, we show that $n$ dual solutions are sufficient to compute all $(i, j)$-optimal values and this bound is tight. These results also show that arc-consistency for

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP G-SCOP, 38000 Grenoble, France

ALLDIFFERENT can be achieved with the reduced costs of a single dual solution, which is consistent with [11] but gives a different view-point.

Finally, we propose a preliminary primal-dual algorithm to enforce filtering. It remains preliminary but highlights the motivation for this line of research. By enumerating dual solutions, filtering is performed in an opportunistic manner and can be interrupted even if it is incomplete. We believe such an anytime algorithm is key for very costly global constraints where arc-consistency is rarely worth a high runtime complexity such as $O(n^3)$. See for instance the discussion in [8] where the arc-consistency algorithm for MINWALLDIFF is found too costly and the filtering of [10] used as a baseline is too weak. Primal-dual filtering techniques could be a very good framework to design anytime and adaptive consistency algorithms [5]. Our experiments on random cost matrices show that a near-complete filtering can be done with reduced costs in much shorter times than the proof of arc-consistency.

## 2 Notations and problem's definition

A constraint satisfaction problem is made of a set of variables, each with a given domain, *i.e.* a finite set of possible values, and a set of constraints specifying the allowed combinations of values for a subset of variables. In the following, the variables, *e.g.* $X_i$, are written with upper case letters for the constraint programming models as opposed to the variables of linear programming models that are in lower case. $D(X_i) \subseteq \mathbb{Z}$ denotes the domain of $X_i$. The minimum and maximum values in $D(X_i)$ are respectively denoted $\underline{X_i}$ and $\overline{X_i}$. A constraint $C$ over a set of variables $\langle X_1, \ldots, X_n \rangle$ is defined by the allowed combinations of values (tuples) of its variables. Such tuples of values are also referred to as solutions of the constraint $C$. Given a constraint $C$ with a scope $\langle X_1, \ldots, X_n \rangle$, a **support** for $C$ is a tuple of values $\langle a_1, \ldots, a_n \rangle$ that is a solution of $C$ and such that $a_i \in D(X_i)$ for all variables $X_i$ in the scope of $C$. Consider a variable $X_i$ in the scope of $C$, the domain $D(X_i)$ is said **arc-consistent** for $C$ if and only if all the values of $D(X_i)$ belong to a support for $C$. A constraint $C$ is said arc-consistent if and only if all its variable's domains are arc-consistent.

The MINWALLDIFF constraint considers a cost $c_{ij} \in \mathbb{N}$ for assigning value $j$ to variable $X_i$. It enforces $n$ variables $(X_1, \ldots, X_n)$ to take distinct values so that the cost of the assignment is no more than a given cost variable $Z$. The cost is defined as the sum of the individual assignment costs so that if $X_i$ is assigned to value $a_i$, it is computed as $\sum_{i=1}^{n} c_{i,a_i}$.

**Definition 1.** MINWALLDIFF$(X_1, \ldots, X_n, Z, c)$ *has a solution if and only if the following constraint network has a solution:*

$$\text{ALLDIFFERENT}(X_1, \ldots, X_n)$$
$$\sum_{i=1}^{n} c_{i,X_i} \leq Z$$

For sake of simplicity, we consider the specific case of a permutation where $\bigcup_{i=1}^{n} D(X_i) = \{1, \ldots, n\}$. The results presented below, namely properties 1, 2, 3 and 4 hold if there are more values than variables (the proofs remain identical).

The minimum weight alldifferent is strongly related to the assignment problem or weighted bipartite perfect matching problem stated in the graph $G = (U, V, E, c)$ referred to as the **weighted variable-value bipartite graph**. The set $U = \{X_1, \ldots, X_n\}$ relates to the variables, $V = \{a_1, \ldots, a_m\}$ to the set of values and edge $(X_i, a_j)$ of cost $c_{ij}$ (also denoted as a triplet $(X_i, a_j, c_{ij})$) is in $E$ if and only if $a_j \in D(X_i)$. A perfect matching $\mathcal{M}$ in $G$ is a set of $n$ vertex-disjoint edges that define a feasible assignment of distinct values to the variables. A minimum weighted perfect matching in $G$ is denoted $\mathcal{M}^*$ and is a minimum cost assignment of the $X$ variables.

A useful graph representation associated to a matching $\mathcal{M}$ of $G$ is the **residual graph** $G_{\mathcal{M}}$ as introduced in [20]. $G_{\mathcal{M}} = (U, V, A, c')$ is a directed bipartite graph with the same node sets as G and with arcs defined as follows:

$$A = \{(X_i, a_j, -c_{ij}) \mid (X_i, a_j) \in \mathcal{M}\}$$
$$\cup \{(a_j, X_i, c_{ij}) \mid (X_i, a_j) \in E \setminus \mathcal{M}\}$$

In other words, the edges from $\mathcal{M}$ are directed from $U$ to $V$ with a cost multiplied by $-1$ and the remaining edges are directed from $V$ to $U$ with their original cost. The total cost of a set of weighted edges or arcs $S$ is denoted $c_S$ and defined as $c_S = \sum_{(i,j) \in S} c_{ij}$. Figure 1 illustrates these notions with an example made of three variables that will serve later on. A non-optimal matching $\mathcal{M}$ of cost $c_{\mathcal{M}} = 1$ is shown on $G$ in bold with its residual graph. Assuming $\overline{Z} = 2$, values 2 and 1 from $X_1$ and $X_2$ respectively are not consistent and the arc-consistent domains are shown on the right of the figure.

## 3 Filtering algorithms for Minimum Weight AllDifferent

We briefly review the filtering algorithm achieving arc-consistency that was initially given in [20] and detailed in [22]. A support for $\underline{Z}$, the lower bound of $Z$ is a matching $\mathcal{M}^*$ of minimal cost in the weighted variable-value bipartite graph $G$. Such an optimal matching can be computed with the famous Hungarian algorithm [15]. For all edges $e \in E \setminus \mathcal{M}^*$, there exists a perfect matching of cost less than $\overline{Z}$ that contains $e$ if and only there exists a cycle $C_e$ in the residual graph $G_{\mathcal{M}^*}$ containing $e$ and such that $c_{C_e} + c_{\mathcal{M}^*} \leq \overline{Z}$. Inconsistent values can be characterized as arcs that are not contained in any such cycles. This can be checked by computing the shortest path distances from $U$ to $V$ in $G_{\mathcal{M}}$ with an all-pairs shortest path algorithm such as Johnson's algorithm [14]. All $(i, j)$-optimal values are known at this stage and all inconsistent values can be removed. The complete procedure runs in time $O(n(d + m\log(m)))$.

A more practical and cheaper incomplete filtering technique is based on the use of linear reduced costs. It is based on the assignment problem and its Integer Programming (IP) formulation. Recall that $G = (U, V, E, c)$ denotes the weighted variable-value bipartite graph. Variables $x_{ij} \in \{0, 1\}$ encode the assignment so that $x_{ij} = 1$ means that $X_i$ is assigned to $j$. The IP formulation is as follows:

$$(\mathcal{P}_{IP}) \begin{cases} z^* = & \min \sum_{(i,j) \in E} x_{ij} c_{ij} & & & \\ \text{s.t.} & \sum_{j:(i,j) \in E} x_{ij} & = & 1 & \forall i \in U \quad (1) \\ & \sum_{i:(i,j) \in E} x_{ij} & = & 1 & \forall j \in V \quad (2) \\ & x_{ij} & \in & \{0, 1\} & \forall (i, j) \in E \quad (3) \end{cases}$$

The objective is to minimize the cost of the assignment. Constraint (1) states that each vertex $i$ of $U$ is assigned to exactly one vertex of $V$. Conversely, constraint (2) enforces each vertex $j$ of $V$ to be assigned to a single vertex of $U$. We denote by $(\mathcal{P})$ the linear relaxation of $(\mathcal{P}_{IP})$ *i.e.* the formulation where constraints (3) stating the domains $x_{ij} \in \{0, 1\}$ have been replaced by $x_{ij} \geqslant 0$. It is well-known that $(\mathcal{P})$ is an ideal formulation so that an integer solution is returned by the simplex algorithm. Finally, the dual $(\mathcal{D})$ of $(\mathcal{P})$ is the following:

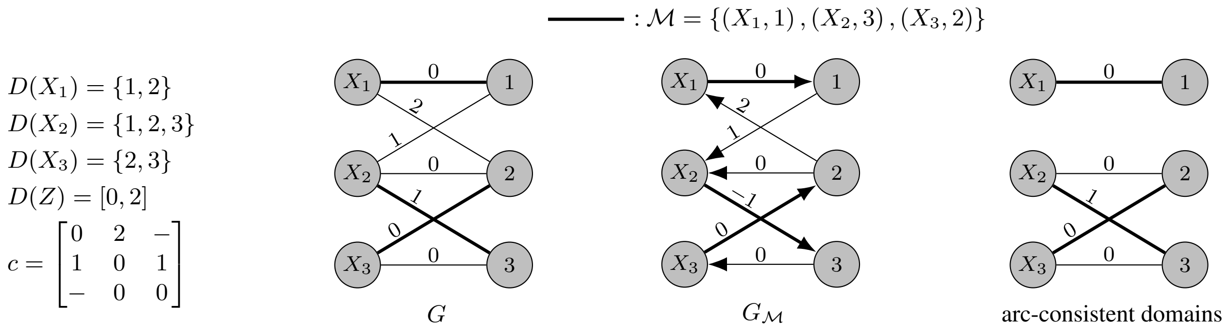


**Figure 1.** Example of MINIMUMWEIGHTALLDIFFERENT$(X_1, X_2, X_3, Z, c)$ with the weighted variable-value graph, the residual graph and the arc-consistent domains.

$$(\mathcal{D}) \begin{cases} w^* = & \max \sum_{i \in U} u_i + \sum_{j \in V} v_j & & \\ \text{s.t.} & u_i + v_j \leqslant c_{ij} & \forall (i,j) \in E & (4) \\ & u_i \in \mathbb{R} & \forall i \in U & (5) \\ & v_j \in \mathbb{R} & \forall j \in V & (6) \end{cases}$$

Variables $u_i$ and $v_j$ are respectively the dual variables related to constraints (1) and (2) of the primal. $u_i$ and $v_j$ are often referred to as the potentials of each $i \in U$ and $j \in V$. The **reduced cost** of an edge $(i,j) \in E$ for a dual solution $(u, v)$ is denoted $r_{u,ij}$ and is the slack of constraint (4):

$$r_{u,ij} = c_{ij} - u_i - v_j$$

Recall that $\overline{Z}$ is the maximum cost allowed, reduced cost based filtering (or variable fixing) is performed from an optimal dual solution $u^*$ of value $z^*$ and states that if $z^* + r_{u^*,ij} > \overline{Z}$ then $x_{ij}$ is set to 0 in any solution of cost less than or equal to $\overline{Z}$. In other words, from the view point of our global constraint, value $j$ can be removed from the domain of $X_i$. Figure 2 shows the dual $(D)$ of the example used previously and two dual optimal solutions $(u, v) = (2, 0, 0, -2, 0, 0)$ and $(u, v) = (0, 1, 1, 0, -1, -1)$ with the corresponding reduced costs. It can be easily checked that the objective value is $z^* = w^* = 0$ for both solutions and that both are feasible. Since $\overline{Z} = 2$, the first solution is able to filter value 1 from $X_2$ whereas the second solution filters value 2 from $X_1$. Thus, each of these dual solution performs an incomplete filtering.

If we consider $\overline{Z} = 1$, it is possible to filter both values with a single dual solution such as $(0.5, 0, 0, -0.5, 0, 0)$. Dual values can also be used to detect variables that must be set to 1. This is meaningful when the filtering is incomplete (It is otherwise implied by the fact that all remaining values of the domain are forbidden). We do not discuss this aspect in the present paper and refer the reader to [11] for the general statement of the rules used for reduced cost filtering.

## 4 Analysis of reduced costs based filtering

Recall that $z^*$ is the optimal value of the assignment problem and $z^*_{|ij}$ denotes the optimal value of formulation $(\mathcal{P})$ with the additional constraint $x_{ij} = 1$. This restricted formulation is denoted $(\mathcal{P}_{|ij})$ and $z^*_{|ij}$ is referred to as the $(i,j)$-optimal value. Both $(\mathcal{P})$ and $(\mathcal{P}_{|ij})$ are known to have the integrality property and an integer solution can be found by solving the linear relaxation with the simplex algorithm.

**Definition 2.** *The **exact reduced cost** $R_{ij}$ of an edge $(i,j) \in E$ is defined as*

$$R_{ij} = z^*_{|ij} - z^*$$

$R_{ij} = +\infty$ *if $(i,j)$ does not belong to a perfect matching of $G = (U, V, E, c)$.*

The exact reduced cost is defined from a primal point view even though its name refers to the dual problem. This will be justified with Property 1. $R_{ij}$ can be seen as the minimum increase of the optimal value $z^*$ when forcing the edge $(i,j)$ in a solution. It provides the $(i,j)$-optimal value from an optimal solution. On our running example, it is easy to see that $R_{12} = R_{21} = 3$ and $R_{23} = R_{32} = 1$. Note that these exact reduced costs can be obtained as the reduced costs of two dual optimal solutions. For instance, $(2, 0, 0, -2, 0, 0)$ provides $R_{21}$ and $R_{23}$ whereas $(0, 1, 0, 0, -1, 0)$ provides $R_{12}$ and $R_{32}$.

An $(i,j)$-optimal value can be computed from a non-optimal solution. The notion of **complete** set of dual solutions is introduced to avoid referring to optimal solutions.

**Definition 3.** *A set $\{u^t\}_{t=1}^q$ of dual solutions is said to be **complete** if and only if $\max_{t=1,\dots,q} (w^t + r_{u^t,kl}) = z^*_{|kl}$ for each edge $(k,l)$ of $E$ where $w^t$ denotes the objective value of solution $u^t$.*

A complete set of dual solutions provide all $(i,j)$-optimal values and ensures arc-consistency. Reversely, a set of dual solutions is said **incomplete** when there exists at least one edge $(i,j)$ for which the $(i,j)$-optimal value is not reached in any of the solutions of the set.

Finally, we assume from now on that **all edges of $E$ belong to a perfect matching of** $G$ to avoid handling the case of unbounded exact reduced costs ($R_{ij} = +\infty$). This assumption can be enforced by the filtering algorithm of the ALLDIFFERENT global constraint as a pre-preprocessing.

### 4.1 Analysis

**Property 1.** *For each edge $(k,l)$ of $E$, there exists an optimal dual solution $u^*$ such that $r_{u^*,kl} = R_{kl}$.*

*Proof.* Let's build explicitly the dual solution $u^*$. Let $\left(\tilde{\mathcal{P}}\right)$ be the primal problem identical to $(\mathcal{P})$ except for the cost of edge $(k,l)$ so that:

$$\begin{cases} \tilde{c}_{kl} = c_{kl} - (z^*_{|kl} - z^*) & \\ \tilde{c}_{ij} = c_{ij} & \forall (i,j) \in E \backslash (k,l) \end{cases}$$

Recall that $(k,l)$ belongs to at least one perfect matching of $G$ so that $\tilde{c}_{kl}$ is finite. Let $\tilde{z}^*$ be the optimal value of $\left(\tilde{\mathcal{P}}\right)$ and $\tilde{u}^*$ be any optimal dual solution for $\left(\tilde{\mathcal{D}}\right)$, the dual of $\left(\tilde{\mathcal{P}}\right)$. We show below that

$$
\begin{aligned}
&\max \textstyle\sum_{i=1}^{3} u_i + \sum_{j=1}^{3} v_j \\
&\quad u_1 + v_1 \leq 0 \\
&\quad u_1 + v_2 \leq 2 \\
&\quad u_2 + v_1 \leq 1 \\
&\quad u_2 + v_2 \leq 0 \\
&\quad u_2 + v_3 \leq 1 \\
&\quad u_3 + v_2 \leq 0 \\
&\quad u_3 + v_3 \leq 0 \\
&\quad u_1, u_2, u_3, v_1, v_2, v_3 \in \mathbb{R}
\end{aligned}
$$

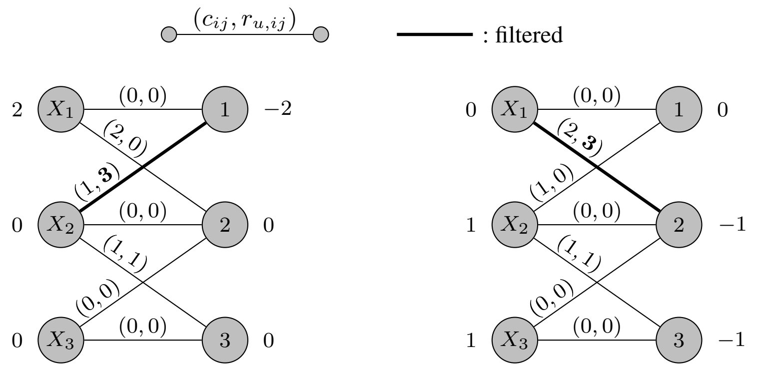


**Figure 2.** Example MINIMUMWEIGHTALLDIFFERENT$(X_1, X_2, X_3, Z, c)$ continued with two dual solutions and the corresponding reduced costs.

$\tilde{u}^*$ is not only feasible and optimal for $(\mathcal{D})$ but also gives the exact reduced cost for edge $(k, l)$:

- Since $z^*_{|kl} \geqslant z^*$ the modified cost $\tilde{c}_{kl}$ is always lower than the original cost so that $\tilde{c}_{kl} \leqslant c_{kl}$. As a result, $\forall (i,j) \in E$, $\tilde{u}^*_i + \tilde{v}^*_j \leqslant \tilde{c}_{ij} \leqslant c_{ij}$ and **$\tilde{u}^*$ is feasible for $(\mathcal{D})$**.
- Let's show that the optimal value is unchanged *i.e.* $\tilde{z}^* = z^*$. Since $\left(\tilde{\mathcal{P}}\right)$ is an ideal formulation, it has (at least) one optimal integer solution. Suppose such an optimal matching of $\left(\tilde{\mathcal{P}}\right)$ has a cost lower than $z^*$ ($\tilde{z}^* < z^*$). On the one hand, if it didn't contain $(k, l)$, it would have the same cost for $(\mathcal{P})$ contradicting $z^*$ as the optimum of $(\mathcal{P})$. On the other hand, if it contained $(k, l)$, its cost would be smaller than $z^*_{|kl}$ in $(\mathcal{P})$ since $(k, l)$ is the only modified cost by the quantity $(z^*_{|kl} - z^*)$. In either cases, it is not possible. Moreover, since any optimal matching in $(\mathcal{P})$ has also a cost of $z^*$ in $\left(\tilde{\mathcal{P}}\right)$, we have $\tilde{z}^* = z^*$. Thus **$\tilde{u}^*$ is an optimal solution for $(\mathcal{D})$** since it is feasible with value $z^*$.
- Finally, an optimal solution of $\left(P_{|kl}\right)$ is an optimal solution of $\left(\tilde{\mathcal{P}}\right)$ of value $z^*$ with the primal variable $x_{kl}$ set to 1. From the complementary slackness theorem, the constraint associated with $x_{kl}$ in $\tilde{u}^*$ must be tight.
Therefore
$$
\begin{aligned}
& & \tilde{u}^*_k + \tilde{v}^*_l &= \tilde{c}_{kl} \\
&\iff & \tilde{u}^*_k + \tilde{v}^*_l &= c_{kl} - (z^*_{|kl} - z^*) \\
&\iff & c_{kl} - \tilde{u}^*_k - \tilde{v}^*_l &= z^*_{|kl} - z^* \\
&\iff & \boldsymbol{r_{\tilde{u}^*,kl}} &= \boldsymbol{R_{kl}}
\end{aligned}
$$
$\square$

The previous property shows that it is possible to establish the consistency of a given value using an appropriate optimal dual solution. But a given solution always provides $|E|$ reduced costs and, as discussed later on, a small number of dual solutions can often detect most of the inconsistent values [4]. We show below that $n$ dual solutions are always enough to express all exact-reduced costs.

**Property 2.** *There exists a complete set of $n$ dual optimal solutions* i.e. *a set $\{u^{t*}\}_{t=1}^{n}$ such that $\max\limits_{t=1,\dots,n} r_{u^{t*},kl} = R_{kl}$ for each edge $(k, l)$ of $E$*

*Proof.* We explicitly build the corresponding $n$ optimal dual solutions. Each one of them is related to a vertex $k \in U$ and is built from the modified primal problem $\left(\tilde{\mathcal{P}}^k\right)$ which is identical to $(\mathcal{P})$ to the exception of the costs related to edges connected to $k$. More precisely:

$$
\begin{cases}
\tilde{c}^k_{kl} = c_{kl} - (z^*_{|kl} - z^*) & \forall l \in V \\
\tilde{c}^k_{ij} = c_{ij} & \forall (i,j) \in U \backslash k \times V
\end{cases}
$$

Let $\tilde{u}^{k*}$ be an optimal solution for $\left(\tilde{\mathcal{D}}^k\right)$, the dual of $\left(\tilde{\mathcal{P}}^k\right)$, and $\tilde{z}^{k*}$ its value. We must show that $\tilde{u}^{k*}$ is feasible and optimal for $(\mathcal{D})$ while providing the exact-reduced costs of all edges connected to vertex $k$. The proof is nearly identical to the one of property 1 and not detailed here. The key additional idea to notice is that any perfect matching $(\mathcal{P})$ contains exactly one edge connected to $k$. Since no more than one such edge with a modified cost can be used, the optimal value $z^*$ remains the same $z^* = \tilde{z}^{k*}$ and the reasonings of Property 1 hold. Overall, $\{\tilde{u}^{k*}\}_{k \in U}$ is a complete set of $|U| = n$ optimal dual solutions. $\square$

Note that the property hold even if $|V| > |U|$ i.e even if there are more values than variables. We now show that $n$ is sharp, *i.e.* that there exists cases where it is not possible to find all $(i, j)$-optimal values with less than $n$ dual solutions.

**Property 3.** *For any $n$, there exists an instance with $n$ variables for which any complete set of dual solutions contains at least $n$ solutions.*

*Proof.* Consider a complete bipartite graph $G = (U, V, E, c)$ with $|U| = n$ and where the costs $c$ are defined as follows:

$$
c_{ij} = \begin{cases} 0 & \text{if } i \geq j \\ 1 & \text{if } i < j \end{cases}
$$

Note that the optimal value is null ($z^* = 0$), the only optimal primal solution is the matching $\mathcal{M}^* = \{(1,1), (2,2), \dots, (n,n)\}$ and all $(i, j)$-optimal values are equal to 1 (to the exception of the edges of $\mathcal{M}^*$). Consider two edges $(X_\alpha, \alpha - 1)$ and $(X_\beta, \beta - 1)$ with $1 < \alpha < \beta$ as shown on Figure 3 with $\alpha = 2$ and $\beta = 4$.

Let's show that edges $(X_\alpha, \alpha - 1)$ and $(X_\beta, \beta - 1)$ are incompatible *i.e.* that a single dual solution can not provide the $(i, j)$-optimal values of both edges.

Suppose that such a dual solution $u$ exists with value $\omega$. Then $u$ would be such that $\omega + r_{u,(\alpha,\alpha-1)} = z^*_{|\alpha,\alpha-1}$ which means $\omega - u_\alpha - v_{\alpha-1} = 1$. Similarly, $\omega - u_\beta - v_{\beta-1} = 1$. By summing the two equalities, we have:

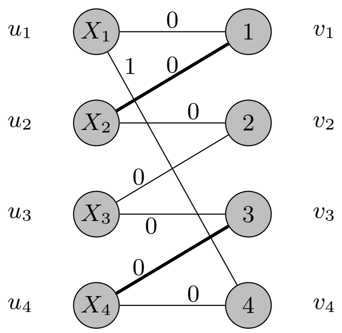


**Figure 3.** Edges $(X_2, 1)$ and $(X_4, 3)$ are incompatible.

$$2\omega - u_\alpha - v_{\alpha-1} - u_\beta - v_{\beta-1} = 2$$
$$\Leftrightarrow \omega + \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j - u_\alpha - v_{\alpha-1} - u_\beta - v_{\beta-1} = 2$$
$$\Leftrightarrow \omega + \sum_{\substack{i=1 \\ i\neq\alpha,\, i\neq\beta}}^{n} u_i + \sum_{\substack{j=1 \\ j\neq\alpha-1,\, j\neq\beta-1}}^{n} v_j = 2$$

Since $u$ is a feasible dual solution, we also have:

$$\Leftrightarrow \omega = 2 - \left( \sum_{\substack{i=1 \\ i\notin\{\alpha,\alpha-1,\beta,\beta-1\}}}^{n} \underbrace{(u_i + v_i)}_{\leqslant 0} \right) - \underbrace{(u_{\alpha-1} + v_\beta)}_{\leqslant 1} - \underbrace{(u_{\beta-1} + v_\alpha)}_{\leqslant 0}$$

$\Leftrightarrow \omega \geqslant 1$ which contradicts $z^* = 0$.

Moreover, the same contradiction occurs for the pair $(X_\alpha, \alpha - 1)$ and $(X_1, n)$. Overall, this highlights a set $\{(X_\alpha, \alpha - 1)\}_{\alpha=2}^{n} \cup \{(X_1, n)\}$ of $n$ edges that are pairwise incompatible. Therefore, at least $n$ dual solutions are required to express all $(i, j)$-optimal values. Any set of less than $n$ dual solutions is incomplete. □

Firstly, note that arc-consistency would also require $n$ dual solutions when setting $\overline{z} = 1$ on the example of Property 3 since exact-reduced costs are equal to 1, their exact values must be computed to assert the consistency of each value.

Secondly, note that no improvement can be expected with costs $c$ restricted to $\{0, 1\}$ (compared to integer costs) since the proof of property 3 is using an instance in $\{0, 1\}$.

Finally, the AllDifferent problem can be seen as a specific case: it can be encoded as an assignment problem in a complete bipartite graph with costs in $\{0, 1\}$ and an upper bound of $\bar{z} = 0$. The edges of the variable-value graph are given a cost of 0 and edges with a cost of 1 are added to make the bipartite graph complete. An edge belongs to a solution of AllDifferent if and only if it belongs to an assignment of cost 0. In this set-up, any positive ($> 0$) reduced cost exhibits an inconsistent edge and a single dual solution can rule them all out. Let $\mathcal{I}$ be the set of all the inconsistent edges.

**Property 4.** *There exists an optimal dual solution $u^*$ s.t.*

$$r_{u^*,ij} > 0 \quad \forall (i, j) \in \mathcal{I}$$

*Proof.* Since the $\{0, 1\}$ encoding presented above implies $R_{ij} \geqslant 1 \;\; \forall (i, j) \in \mathcal{I}$, we can consider a set of optimal dual solutions $\{\tilde{u}^{ij*} : (i, j) \in \mathcal{I}\}$ with $r_{\tilde{u}^{ij*},ij} \geqslant 1$.

Let $\tilde{u}^*$ be the average solution of the previous set: $\tilde{u}^* = \sum_{(i,j)\in\mathcal{I}} \frac{\tilde{u}^{ij*}}{|\mathcal{I}|}$.

This solution is feasible, optimal, and has a positive reduced cost for each $(i, j) \in \mathcal{I}$. □

Therefore, **a single dual solution** is enough to achieve arc-consistency of AllDifferent. Such a solution can be seen as an interior point of the dual problem since a positive reduced cost is a positive slack of a dual constraint. It can be found with the method given in [11] and sheds a different light on this result from a reduced cost point of view.

## 5 Towards a primal dual algorithm for filtering

We suggest a very simple enhancement, based on reduced costs, of the known algorithm to achieve arc-consistency initially proposed by [20] and refined in [22]. Note that we only intend to motivate our analysis and illustrate the design of anytime filtering algorithms based on primal/dual iterations. Consider the algorithm of [22]:

1. Remove from $E$, all edges $(i, j)$ that do not belong to a perfect matching of $G = (U, V, E, c)$ (see AllDifferent).
2. Solve the assignment problem with the Hungarian algorithm [15]. Let $u^*$ and $\mathcal{M}^*$ respectively denote the optimal dual and primal solution found at the end.
3. For each $k \in U$
   - Replace all $c_{ij}$ by the reduced costs $r_{u^*,ij}$ so that all costs are positive (see Johnson's algorithm [14] and [22]).
   - Compute the shortest path distances $d(l)$ from $k$ to all vertices of $l \in U \cup V$ in $G_{\mathcal{M}^*}$ with Dijkstra algorithm to get the exact reduced-cost $R_{kl}$ for all $l \in V$ s.t $(k, l) \in E$.

We propose to add two filtering steps over all edges of $E$ so that the algorithm can be interrupted at any time while still proving filtering over all domains. Firstly, at the end of step 2, $u^*$ can be used to perform reduced cost based filtering on all edges as done by [10]. Note that this can be done even if the Hungarian algorithm is interrupted by using the dual feasible solution it provides when interrupted. Secondly, after each iteration $k$ of step 3, reduced cost filtering for all remaining edges can be performed with an optimal dual solution built as follows: $\tilde{u}_i = u_i^* + d(i)$ and $\tilde{v}_j = v_j^* - d(j)$. It can be easily checked that it is dual feasible:

- For each $(i, j) \in E \backslash \mathcal{M}^*$, the shortest path distances satisfy the inequality $d(i) \leqslant d(j) + r_{u^*,ij}$ implying that:

$$\tilde{u}_i + \tilde{v}_j \leqslant u_i^* - d(j) + v_j^* + d(j) + r_{u^*,ij} = c_{ij}$$

- For each $(i, j) \in \mathcal{M}^*$, $r_{u^*,ij=0}$. Therefore $d(i) = d(j)$ and $\tilde{u}_i + \tilde{v}_j = c_{ij}$.

The time complexity remains in $O(n^3)$ in the worst case. The filtering algorithm can be seen as producing a sequence of dual feasible solutions $u^1, \ldots, u^{n+1}$ whose reduced-costs are used for filtering the entire domains. The only interest of this approach is to derive an anytime algorithm and to stop the filtering after $Q$ dual solutions: $u^1, \ldots, u^Q$. The $Q - 1$ vertices (variables) from which the dual solutions are built can be chosen heuristically. In the experiments and as a mean of illustration we chose them randomly and fixed $Q = 1 + 0.1n'$ where $n'$ is the number of ungrounded variables (initially $n$). Steps 2 and 3 enhanced with reduced cost filtering are illustrated in Figure 4. The variable-value graph is given in (a), an optimal dual solution (provided by the Hungarian) is shown in (b) and value 4 is filtered from $D(X_3)$ (step 2). The dual solution obtained for $k = 1$ after computing the shortest path distances from $X_1$ (step 3) is shown in (c). The exact reduced costs of $D(X_1)$ lead to filtering value 2 from $D(X_1)$. Additionally, value 3 is removed from $D(X_2)$ by the same dual solution.

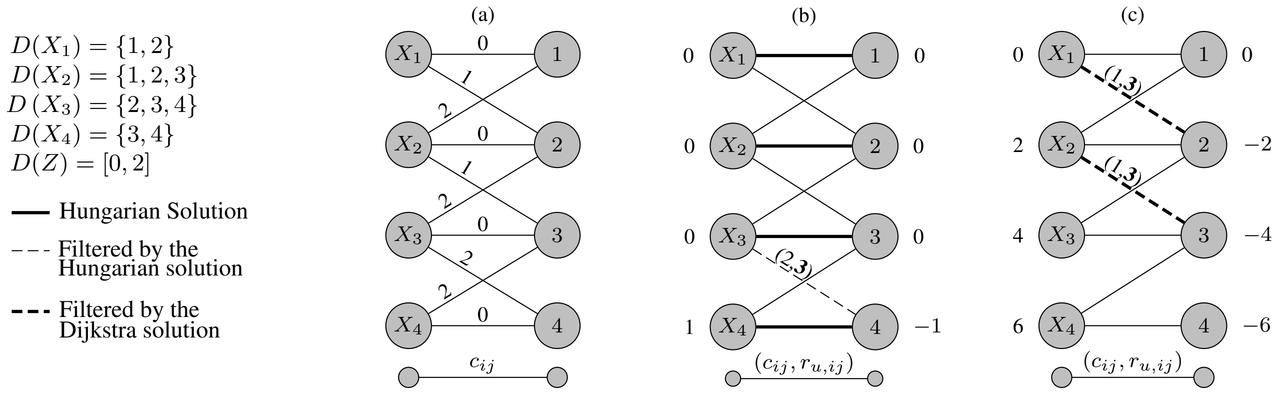


**Figure 4.** Example. (a): Variable-value graph ; (b): Hungarian solution ; (c): Dual solution given by the Dijkstra algorithm from $X_1$.

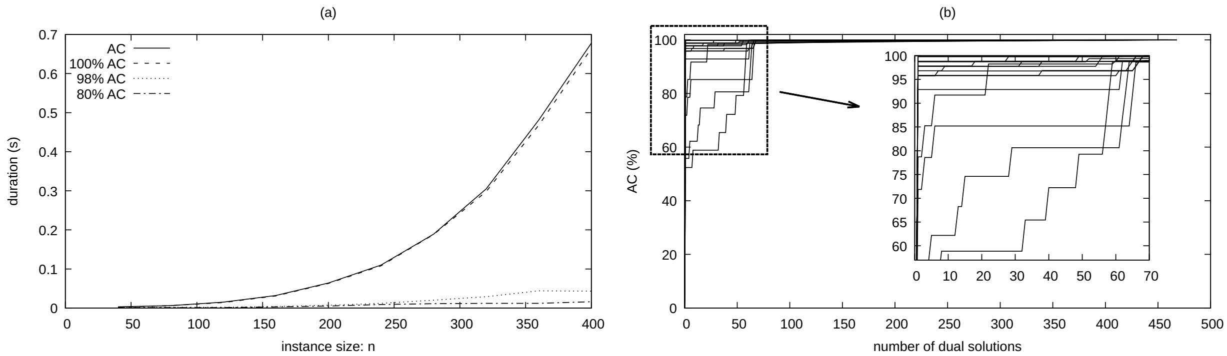


**Figure 5.** (a): cpu time (in s) needed to filter 80%, 98% and 100% of the inconsistent values for different problem sizes $n$ ranging from 20 to 400. The line AC shows the time needed to prove arc-consistency. (b): Level of consistency depending on the cumulative number of dual feasible solutions enumerated ($n = 400$, $c_{ij} \in \{0, \ldots, 100\}$ and $\overline{z} = 1.2z^*$).

## 6 Experimental results

We analyse experimentally the filtering obtained with reduced costs and the behavior of the primal-dual approach proposed in Section 5. Levels of consistency are reported as a percentage of the total number of inconsistent values (for a given upper bound $\overline{z}$). All experiments ran on a laptop Dell Precision 5530 (i7-8850H 2.60GHz 16Go Ram, Linux Gentoo 64bits) using a single thread.

**A single call to the filtering.** The results reported on Figure 5 are obtained with random costs matrices of sizes ranging from $n = 20$ to $n = 400$ and where each $c_{ij}$ is drawn in $\{0, \ldots, 100\}$ with a uniform distribution. The value of $\overline{z} = 1.2z^*$ is used to act as the upper bound of $z^*$. Figure 5.a shows the cpu time needed to achieve three levels of consistencies (80 %, 98 % and 100 % of AC) as well as the time to perform and to prove that arc-consistency is enforced (denoted AC). Figure 5.b shows the level of consistency reached after each dual feasible solution for 20 random instances of size $n = 400$. Each line is one of the instances and a point $(x, y)$ means that $y$ % of the complete filtering has been obtained with the $x$ first dual solutions $u^1, \ldots, u^x$ of the sequence produced by the primal dual approach. We include (in the sequence) the dual solutions provided after each primal/dual iteration of the Hungarian method itself to visualize better what would happen when interrupting the Hungarian thus the number of iterations can be larger than $n+1$ in the results. Table 1 presents the same results with more precise numerical values by explicitly giving the value of $x$ required to achieve a given percentage $p$ of filtering (namely $p \in \{0.80, 0.98, 1\}$) as well as the corresponding time in seconds (column T). Column $Q$ gives the total number of dual solutions produced when proving arc-consistency. Moreover, we report the percentage of filtering that would be performed by the optimal dual solution alone (column F), *i.e.* the solution found at the end of the Hungarian algorithm which is the traditional approach for filtering with reduced-costs [10]. There are in average around 25000

| | $\overline{z}$ | $p = 80\%$ | | $p = 98\%$ | | $p = 100\%$ | | AC | | F |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $x$ | T | $x$ | T | $x$ | T | Q | T | |
| mean | $1.20z^*$ | 12.7 | 0.02 | 39.5 | 0.05 | 452.8 | 0.69 | 460.7 | 0.70 | 99 % |
| median | $1.20z^*$ | 1.0 | 0.00 | 56.0 | 0.06 | 455.5 | 0.66 | 461.5 | 0.67 | 99 % |
| max | $1.20z^*$ | 58 | 0.06 | 66 | 0.14 | 465 | 1.11 | 467 | 1.11 | 99 % |
| mean | $1.235z^*$ | 80.0 | 0.51 | 316.3 | 1.84 | 450.3 | 2.67 | 460.7 | 2.75 | 80 % |
| median | $1.235z^*$ | 58.0 | 0.18 | 308.5 | 1.55 | 451.5 | 2.37 | 462.0 | 2.42 | 85 % |
| max | $1.235z^*$ | 274 | 2.93 | 453 | 4.80 | 464 | 5.34 | 469 | 5.52 | 94 % |

**Table 1.** Numerical details of the results presented in Figure 5. (20 random instances ; n=400)

values removed in the first case ($\overline{z} = 1.20z^*$) and around 2500 in the second case ($\overline{z} = 1.235z^*$).

- For $\overline{z} = 1.20z^*$, most of all inconsistent values are identified in a fraction of the total time needed to enforce arc-consistency (See Figure 5.a and Table 1 for $p = 98\%$). Typically, 98% of the values are removed in less than 10% of the total time.

- Filtering is gathered with intermediate dual solutions.
- The optimal dual solution alone provides nearly all the filtering for $\overline{z} = 1.20z^*$ but the algorithm can be interrupted earlier with very little loss.
- As the gap increases to $\overline{z} = 1.235z^*$, the transition is abrupt and achieving more than 80% of the filtering requires to go beyond the dual solution of the Hungarian algorithm.

**Preliminary results during resolution.** We investigate the Resource Constrained Assignment Problem (RCAP) [1] to illustrate the same ideas during resolution. The problem is to find a minimum weight assignment that satisfies one or several resource constraints. It is NP-Hard and can be formulated with the traditional model ($\mathcal{P}_{IP}$) given in Section 3 (using a complete set $E = U \times V$ of edges) and a set $K$ of additional resource constraints:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^k x_{ij} \leq b^k \quad \forall k \in K$$

$d_{ij}^k$ denotes the consumption of resource $k$ by assigning value $j$ to variable $i$ and $b^k$ is the capacity of the same resource. Moreover we denote by $b'$ (resp. $d'$) the sum of all capacities (resp. consumption) i.e $d' = \sum_{k\in R} b^k$ and $d'_{ij} = \sum_{k\in R} d_{ij}^k$. Each resource can be seen as an assignment problem and the CP model can be written as follow:

$$\begin{array}{lll} & \min Z & \\ \text{s.t.} & Z = \sum_{i=1}^{n} c_{i,X_i} & (7) \\ & \textsc{MinWAllDiff}(X_1,\ldots,X_n,Z,c) & (8) \\ & \textsc{MinWAllDiff}(X_1,\ldots,X_n,d^k,b^k) \quad \forall k \in R & (9) \\ & \textsc{MinWAllDiff}(X_1,\ldots,X_n,d',b') & (10) \end{array}$$

Constraint (7) is implemented using ELEMENT constraints. Constraint (8) enforces the $X$ variables to be all different and gives the strong filtering dedicated to the objective function. Constraints (9) model the resources. Finally (10) is a redundant constraint that was found useful when resources are tight. Overall, this model involves $K+2$ MINWALLDIFF constraints. We compare the following filtering algorithms:

- The arc-consistency algorithm of [22] denoted *ac*.
- The initial approach of [10] denoted *hung*.
- A version of the primal-dual filtering with $Q = 1{+}0.1n'$ (denoted *pd10*) where $n'$ is the number of ungrounded variables (initially $n$). The $0.1n'$ variables to build the dual solutions are chosen randomly. The intention is that only 10% of the variables are filtered precisely following the original algorithm of [22] but dual solutions provide filtering over potentially all remaining variables.

We use a 600s time limit and the Choco 3 solver [17]. Instances are generated with a uniform distribution for costs/resources in $\{0, ..., 100\}$ and capacities between 0.1 and 0.6 of $100n$. We consider 6 classes of instances with $n \in \{100, 200, 500\}$ and $k \in \{2, 6\}$ with 5 instances per class so 30 in total. The search is performed by ordering the variables lexicographically in non-increasing resource consumption. The consumption is computed (for $X_i$) as $\sum_{j=1}^{n}\sum_{k\in K} \tilde{d}_{ij}^k$ where $\tilde{d}$ are the consumptions normalized in $[0, 1]$ to be able to compare resources with various capacities. The value ordering heuristic picks the value $j$ with the minimum resource consumption ($j = arg\, min_{v\in D(X_i)} \sum_{k\in K} \tilde{d}_{iv}^k$). Finally, to make sure the tree is traversed at various depths and the search does not remain stuck in a subtree at a very high or very low depth, we use a Limited Discrepancy Search [12] provided by the solver. Table 2 reports the number of nodes opened per second (**nodes/s**), the gap (**Δ Sol**) between the best solution found compared to the one found by *ac* (a negative gap is an improvement), the number of times the algorithm obtained the best solution among the three approaches (♯ **best Sol**) and the total percentage of the filtering that was performed compared to arc-consistency (**% AC**). This last metric was obtained by instrumenting the code and re-running the solving process for the same number of nodes, to count (at each call to the filtering) the number of values arc-consistency would have removed and the number of values actually removed by the algorithm under evaluation.

| n | k | Algo | nodes/s | % AC | Δ Sol | ♯ best Sol |
|---|---|---|---|---|---|---|
| 100 | 2 | ac | 105.7 | 100 % | - | 1 |
| | | hung | 619.2 | 87.1 % | −8.6 % | 0 |
| | | pd10 | 406.1 | 92 % | −16.2 % | 4 |
| 100 | 6 | ac | 73 | 100 % | - | 0 |
| | | hung | 366.5 | 73.2 % | −1.6 % | 0 |
| | | pd10 | 351 | 83.4 % | −3.2 % | 5 |
| 200 | 2 | ac | 27.6 | 100 % | - | 0 |
| | | hung | 249.4 | 84.6 % | −18.5 % | 0 |
| | | pd10 | 179.6 | 92.4 % | −25.9 % | 5 |
| 200 | 6 | ac | 16.9 | 100 % | - | 0 |
| | | hung | 108.5 | 75.6 % | −9.8 % | 0 |
| | | pd10 | 90.2 | 87.5 % | −13.6 % | 4 |
| 500 | 2 | ac | 0.9 | 100 % | - | 0 |
| | | hung | 36.1 | 68.9 % | −2.4 % | 0 |
| | | pd10 | 30.6 | 79.8 % | −2.8 % | 5 |
| 500 | 6 | ac | 0.1 | 100 % | - | 0 |
| | | hung | 15.7 | 58.9 % | $-\infty$ | 0 |
| | | pd10 | 12 | 73.6 % | $-\infty$ | 5 |

**Table 2.** Results on the RCAP problem.

*pd10* performs (roughly) between 4 to 120 times more nodes per seconds than *ac* while still achieving between 73 % and 92 % of the complete filtering in average. It significantly improves the quality of the solutions found. Although *hung* can be even faster in nodes per seconds, it misses too much of the filtering and is not as competitive with *pd10*. Note that for $n = 500$ and $k = 6$, *ac* does not find any feasible solution within the time limit. Note that this experiment is only an example of a specific solving context where *pd10* provides a better "inference versus search" trade-off than *hung* and *ac*.

## 7 Conclusion

We conducted an analysis of reduced cost based filtering for a very fundamental global constraint related to the assignment problem: MINIMUMWEIGHTALLDIFFERENT. Reduced-cost filtering was proposed in 1999 by Foccaci and al [10] on the very same global constraint and has been used since without detailed analysis. The present work shows that arc-consistency can be achieved with the use of reduced costs and that a minimum number of $n$ dual feasible solutions are always required in the worst-case. It also shows that arc-consistency of ALLDIFFERENT can be established with the reduced costs of a single dual solution giving a different view-point on the result of [11]. The analysis is based on the LP formulation of the constraint and its integrality property (ideal formulation). In particular, it does not rely on the shortest path sub-problems of the dedicated arc-consistency algorithm or the flow structure of [20]. The proofs presented are new and we aim at generalizing these results to global constraints with an ideal LP formulation and a cost variable which encompasses a large class of constraints. The next step is to turn this analysis into more efficient filtering algorithms based on primal/dual iterations.

# References


[1] Ronny Aboudi and Kurt Jørnsten. Resource constrained assignment problems. *Discrete Applied Mathematics*, 26(2-3):175–191, 1990.

[2] Tobias Achterberg, Timo Berthold, Thorsten Koch, and Kati Wolter. *Constraint Integer Programming: A New Approach to Integrate CP and MIP*, pages 6–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[3] Ionuţ Aron, John N. Hooker, and Tallys H. Yunes. *SIMPL: A System for Integrating Optimization Techniques*, pages 21–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[4] Omid Sanei Bajgiran, André A. Ciré, and Louis-Martin Rousseau. A first look at picking dual variables for maximizing reduced cost fixing. In Domenico Salvagnin and Michele Lombardi, editors, *Integration of AI and OR Techniques in Constraint Programming - 14th International Conference, CPAIOR 2017, Padua, Italy, June 5-8, 2017, Proceedings*, volume 10335 of *Lecture Notes in Computer Science*, pages 221–228. Springer, 2017.

[5] Amine Balafrej, Christian Bessiere, Anastasia Paparrizou, and Gilles Trombettoni. *Adapting Consistency in Constraint Solving*, pages 226–253. Springer International Publishing, Cham, 2016.

[6] Alexander Bockmayr and Thomas Kasper. Branch and infer: A unifying framework for integer and finite domain constraint programming. *INFORMS Journal on Computing*, 10(3):287–300, 1998.

[7] Yves Caseau and François Laburthe. Solving various weighted matching problems with constraints. *Constraints*, 5(1/2):141–160, 2000.

[8] Sascha Van Cauwelaert and Pierre Schaus. Efficient filtering for the resource-cost alldifferent constraint. *Constraints*, 22(4):493–511, 2017.

[9] Filippo Focacci, Andrea Lodi, and Michela Milano. Cost-based domain filtering. In Joxan Jaffar, editor, *Principles and Practice of Constraint Programming - CP'99, 5th International Conference, Alexandria, Virginia, USA, October 11-14, 1999, Proceedings*, volume 1713 of *Lecture Notes in Computer Science*, pages 189–203. Springer, 1999.

[10] Filippo Focacci, Andrea Lodi, and Michela Milano. Embedding relaxations in global constraints for solving tsp and tsptw. *Ann. Math. Artif. Intell.*, 34(4):291–311, 2002.

[11] Grigori German, Olivier Briant, Hadrien Cambazard, and Vincent Jost. Arc consistency via linear programming. In *Principles and Practice of Constraint Programming - 23rd International Conference, CP 2017, Melbourne, VIC, Australia, August 28 - September 1, 2017, Proceedings*, pages 114–128, 2017.

[12] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. pages 607–613. Morgan Kaufmann, 1995.

[13] John. N. Hooker. Operations research methods in constraint programming. In Francesca Rossi, Peter van Beek, and Toby Walsh, editors, *Handbook of Constraint Programming*, chapter 15. Elsevier, 2006.

[14] Donald B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1):1–13, 1977.

[15] Harold W Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008*, pages 29–47. Springer, 2010.

[16] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.

[17] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca. *Choco Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., 2016.

[18] Philippe Refalo. *Tight Cooperation and Its Application in Piecewise Linear Optimization*, pages 375–389. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[19] Philippe Refalo. Linear formulation of constraint programming models and hybrid solvers. In Rina Dechter, editor, *Principles and Practice of Constraint Programming - CP 2000, 6th International Conference, Singapore, September 18-21, 2000, Proceedings*, volume 1894 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2000.

[20] Jean-Charles Régin. Arc consistency for global cardinality constraints with costs. In Joxan Jaffar, editor, *Principles and Practice of Constraint Programming – CP'99*, pages 390–404, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[21] Robert Rodosek, Mark G. Wallace, and Mozafar T. Hajian. A new approach to integrating mixed integer programming and constraint logicprogramming. *Annals of Operations Research*, 86(0):63–87, 1999.

[22] Meinolf Sellmann. An arc-consistency algorithm for the minimum weight all different constraint. In Pascal Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002, 8th International Conference, CP 2002, Ithaca, NY, USA, September 9-13, 2002, Proceedings*, volume 2470 of *Lecture Notes in Computer Science*, pages 744–749. Springer, 2002.