An Arc-Consistency Algorithm for the Minimum Weight All Different Constraint

Meinolf Sellmann

University of Paderborn
Department of Mathematics and Computer Science
Fürstenallee 11, D-33102 Paderborn
sello@uni-paderborn.de

Abstract. Historically, discrete minimization problems in constrained logical programming were modeled with the help of an isolated bounding constraint on the objective that is to be decreased. To overcome this frequently inefficient way of searching for improving solutions, the notion of *optimization constraints* was introduced. Optimization constraints can be viewed as global constraints that link the objective with other constraints of the problem at hand. We present an arc-consistency (actually: hyper-arc-consistency) algorithm for the *minimum weight all different constraint* which is an optimization constraint that consists in the combination of a linear objective with an all different constraint.

Keywords: optimization constraint, cost based filtering, all different constraint, minimum weight all different constraint, MinWeightAllDiff, IlcAllDiffCost

1 Introduction

In recent years, a joint effort of the constraint programming (CP) and operations research (OR) community has yield new concepts that can improve approaches for discrete optimization problems which are hard both in terms of feasibility and optimality. An important contribution is the idea of *optimization constraints* [10]. Though never explicitly stated as constraints, in the OR world optimization constraints are frequently used for bound computations and variable fixing. From a CP perspective, they can be viewed as global constraints that link the objective with some other constraints of the problem.

The constraint structure of many discrete optimization problems can be modeled efficiently using all different constraints. As a matter of fact, the all different constraint was one of the first global constraints that were considered [11]. Regarding the combination of the all different constraint and a linear objective, in [3], Y. Caseau and F. Laburthe introduced the *MinWeightAllDiff* constraint. In first applications [4], it was used for pruning purposes only. In [8, 9], F. Focacci et al. showed how the constraint (they refer to it as the *IlcAllDiffCost constraint*) can also be used for domain filtering by exploiting reduced cost information.

In this paper, we present an arc-consistency algorithm for the minimum weight all different constraint. It is based on standard operations research algorithms for the computation of minimum weighted bipartite matchings and shortest paths with non-negative edge weights. We show, that arc-consistency can be achieved in time $O(n(d+$

$m \log m)$), 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.

The remaining paper is structured as follows: In Section 2, we formally define the minimum weight all different constraint. The arc-consistency algorithm for the constraint is presented in Section 3. Finally, we conclude in Section 4.

2 The Minimum Weight All Different Constraint

Given a natural number $n \in \mathbb{N}$ and variables $X_1, \ldots, X_n$, we denote with $D_1 := D(X_1), \ldots, D_n := D(X_n)$ the domains of the variables, and let $D := \{\alpha_1, \ldots, \alpha_m\} = \bigcup_i D_i$ denote the union of all domains, whereby $m = |D|$. Further, given costs $c_{ij} \geq 0$ for assigning value α_j to variable X_i (whereby c_{ij} may be undefined if $\alpha_j \notin D_i$), we add a variable for the objective $Z = Z(X, c) = \sum_{i, X_i = \alpha_j} c_{ij}$ to be minimized. Note, that the non-negativity restriction on c can always be achieved by setting $\hat{c}_{ij} := c_{ij} - \min_{i,j} c_{ij}$, which will change the objective by the constant $n \min_{i,j} c_{ij}$.

In the course of optimization, once we have found a feasible solution with associated objective value β, we are searching for improving solutions only, thus requiring $Z < \beta$. Then, we define:

Definition 1. *The* minimum weight all different constraint *is the conjunction of an all different constraint on variables $X_1, \ldots, X_n$ and a bound constraint on the objective Z, i.e.:*

$$\text{MinWeightAllDiff}(X_1, \ldots, X_n, c, \beta) := \text{AllDiff}(X_1, \ldots, X_n) \wedge (Z < \beta).$$

Because otherwise there exists no feasible assignment, in the following we will assume $m \geq n$. There is a tight correlation between the minimum weight all different constraint and the *weighted bipartite perfect matching problem* that can be formalized by setting $G := G(X, D, c) := (V_1, V_2, E, c)$ where $V_1 := \{X_1, \ldots, X_n\}$, $V_2 := \{\alpha_1, \ldots, \alpha_m\}$ and $E := \{\{X_i, \alpha_j\} \mid \alpha_j \in D_i\}$. It is easy to see that any perfect matching[1] in G defines a feasible assignment of all different values to the variables. Therefore, there is also a one-to-one correspondence of cost optimal variable assignments and minimum weighted perfect matchings in G.

For the latter problem, a series of efficient algorithms have been developed. Using the *Hungarian method* or the *successive shortest path algorithm*, it can be solved in time $O(n(d + m \log m))$, where $d := \sum_i |D_i|$ denotes the number of edges in the given bipartite graph. For a detailed presentation of approaches for the weighted bipartite matching problem, we refer to [1].

Because there are efficient algorithms available, there is no need to apply a tree search to compute an optimal variable assignment if the minimum weight all different constraint is the only constraint of a discrete optimization problem. However, the situation changes when the problem consists of more than one minimum weight all different constraint or a combination with other constraints. Then, a tree search may very well be the favorable algorithmic approach to tackle the problem [3].

In such a scenario, we can exploit the algorithms developed in the OR community to compute a bound on the best possible variable assignment that can still be reached

[1] With the term “perfect matching” we refer to a subset of pairwise non-adjacent edges of cardinality $n \leq m$.

in the subtree rooted at the current choice point. Also, it has been suggested to use reduced cost information to perform cost based filtering at essentially no additional computational cost [9].

In the following, we describe an algorithm that achieves arc-consistency in the same worst case running time as is needed to compute a minimum weighted perfect matching when using the Hungarian method or the successive shortest path algorithm.

3 An Arc-Consistency Algorithm

To achieve arc-consistency of the minimum weight all different constraint, we need to remove all values from variable domains that cannot be part of any feasible assignment of values to variables with associated costs $Z < \beta$. That is, in the graph interpretation of the problem, we need to compute and remove the set of edges that cannot be part of any perfect matching with costs less than β.

For any perfect matching M, we set $\operatorname{cost}(M) := \sum_{\{X_i,\alpha_j\}\in M} c_{ij}$. Further, we define the *corresponding network* $N^M := (V_1, V_2, A, c^M)$ whereby

$$A := \{(X_i, \alpha_j) \mid \{X_i, \alpha_j\} \in M\} \cup \{(\alpha_j, X_i) \mid \{X_i, \alpha_j\} \notin M\},$$

and $c_{ij}^M := -c_{ij}$ if $\{X_i, \alpha_j\} \in M$, and $c_{ij}^M := c_{ij}$ otherwise. That is, we transform the graph G into a directed network by directing matching edges from V_1 to V_2 and all other edges from V_2 to V_1. Furthermore, the cost of arcs going from V_1 to V_2 is multiplied by -1.

In the following, we will make some key observations that we will use later to develop an efficient arc-consistency algorithm. For a cycle C in N^M, we set $\operatorname{cost}(C) := \sum_{e\in C} c_e^M$. Let M denote a perfect matching in G.

Lemma 1. *Given an edge $e \notin M$, and assume that there exists a minimum cost cycle C_e in N^M that contains e.*[2]

a) There is a perfect matching M_e in G that contains e, and it holds that $\operatorname{cost}(M_e) = \operatorname{cost}(M) + \operatorname{cost}(C_e)$.
b) The set M is a minimum weighted perfect matching in G, iff there is no negative cycle in N^M.
c) If M is of minimum weight, then for every perfect matching M_e that contains e it holds that $\operatorname{cost}(M_e) \geq \operatorname{cost}(M) + \operatorname{cost}(C_e)$.

Proof. a) Let C_e^+ and C_e^- denote the edges in E that correspond to arcs in C_e that go from V_2 to V_1, or from V_1 to V_2, respectively. We define $M_e := (M \setminus C_e^-) \cup C_e^+$. Obviously, $e \in M_e$, and because of $|C_e^+| = |C_e^-|$, M_e is a perfect matching in G. It holds: $\operatorname{cost}(M_e) = \operatorname{cost}(M) - \operatorname{cost}(C_e^-) + \operatorname{cost}(C_e^+) = \operatorname{cost}(M) + \operatorname{cost}(C_e)$.
b) Follows directly from (a).
c) It is easy to see that the symmetric difference $M \oplus M_e = M \setminus M_e \cup M_e \setminus M$ forms a set of cycles $C_1, \ldots, C_r$ in G that also correspond to cycles in N^M. Moreover, it holds that $\operatorname{cost}(M_e) = \operatorname{cost}(M) - \operatorname{cost}(M \setminus M_e) + \operatorname{cost}(M_e \setminus M)$, and thus $\operatorname{cost}(M_e) = \operatorname{cost}(M) + \sum_i \operatorname{cost}(C_i)$. Without loss of generality, we may assume that $e \in C_1$. Then, because of (b) and $\operatorname{cost}(C_e) \leq \operatorname{cost}(C_1)$, we have that $\operatorname{cost}(M_e) \geq \operatorname{cost}(M) + \operatorname{cost}(C_1) \geq \operatorname{cost}(M) + \operatorname{cost}(C_e)$. □

[2] Here and in the following we identify an edge $e \in G$ and its corresponding arc in the directed network N^M.

Theorem 1. *Let M denote a minimum weight perfect matching in G, and $e \in E \setminus M$. There exists a perfect matching M_e with $e \in M_e$ and $\mathrm{cost}(M_e) < \beta$, iff there exists a cycle C_e in N^M that contains e with $\mathrm{cost}(C_e) < \beta - \mathrm{cost}(M)$.*

Proof. Let C_e denote the cycle in N^M with $e \in C_e$ and minimal costs.

- $\Rightarrow$ Assume that there is no such cycle. Then, either there is no cycle in N^M that contains e, or $\mathrm{cost}(C_e) \geq \beta - \mathrm{cost}(M)$. In the first case, there exists no matching M_e that contains e.[3] And in the latter case, with Lemma 1(c), we have that $\mathrm{cost}(M_e) \geq \mathrm{cost}(M) + \mathrm{cost}(C_e) \geq \beta$, which is a contradiction.
- $\Leftarrow$ We have that $\mathrm{cost}(C_e) < \beta - \mathrm{cost}(M)$. With Lemma 1(a) this implies that there exists a perfect matching M_e that contains e, and for which it holds that $\mathrm{cost}(M_e) = \mathrm{cost}(M) + \mathrm{cost}(C_e) < \beta$. □

With Theorem 1, now we can characterize values that have to be removed from variable domains in order to achieve arc-consistency. Given a minimum weight perfect matching M in G, infeasible assignments simply correspond to arcs e in N^M that are not contained in any cycle C_e with $\mathrm{cost}(C_e) < \beta - \mathrm{cost}(M)$.

Of course, if $\mathrm{cost}(M) \geq \beta$ we know from Lemma 1(b) that the current choice point is inconsistent, and we can backtrack right away. So let us assume that $\mathrm{cost}(M) < \beta$. Then, using empty cycles C_e with $\mathrm{cost}(C_e) = 0 < \beta - \mathrm{cost}(M)$ we can show that all edges $e \in M$ are valid assignments. Thus, we only need to consider $e \notin M$. By construction, we know that the corresponding edge in N^M is directed from V_2 to V_1, i.e. $e = (\alpha_j, X_i)$. Denote with $\mathrm{dist}(X_i, \alpha_j, c^M)$ the shortest path distance from X_i to α_j in N^M. Then, for the minimum weight cycle C_e with $e \in C_e$ it holds: $\mathrm{cost}(C_e) = c_{ij} + \mathrm{dist}(X_i, \alpha_j, c^M)$. Thus, it is sufficient to compute the shortest path distances from V_1 to V_2 in N^M.

We can ease this work by eliminating negative edge weights in N^M. Consider potential functions $\pi^1 : V_1 \to \mathbb{R}$ and $\pi^2 : V_2 \to \mathbb{R}$. It is a well known fact, that the shortest path structure of the network remains intact if we change the cost function by setting $\overline{c}_{ij}^M := c_{ij}^M + \pi_i^1 - \pi_j^2$ for all $(i,j) \in M$, and $\overline{c}_{ij}^M := c_{ij}^M - \pi_i^1 + \pi_j^2$ for all $(i,j) \notin M$. Then, $\mathrm{dist}(X_i, \alpha_j, c^M) = \mathrm{dist}(X_i, \alpha_j, \overline{c}_{ij}^M) - \pi_i^1 + \pi_j^2$. If the network does not contain negative weight cycles (which is true because M is a perfect matching of minimum weight, see Lemma 1(b)), we can choose potential functions such that $\overline{c}^M \geq 0$. This idea has been used before in the all-pairs shortest path algorithm by Johnson [6].

In our context, after having computed a minimum weight perfect matching, we get the node potential functions π^1 and π^2 for free by using the dual and negative dual values corresponding to the nodes in V_1 and V_2, respectively. As a matter of fact, the resulting cost vector $\overline{c}^M$ is exactly the vector of reduced costs $\overline{c}$.

We summarize: To achieve arc-consistency, first we compute a minimum weight perfect matching in a bipartite graph in time $O(n(d + m \log m))$. We obtain an optimal matching M, dual values π^1, π^2, and reduced costs $\overline{c}$. If $\mathrm{cost}(M) \geq \beta$, we can backtrack. Otherwise, we set up a network $N = (V_1, V_2, A, \overline{c})$ and compute n single source shortest paths with non-negative edge weights, each of them requiring time $O(d + m \log m)$ when using Dijkstra's algorithm in combination with Fibonacci

[3] Note, that this observation is commonly used in domain filtering algorithms for the all different constraint [11].

heaps [6]. We obtain distances $\mathrm{dist}(X_i, \alpha_j, \overline{c})$ for all variables and values. Finally, we remove value α_j from the domain of X_i, iff

$$\begin{aligned} \overline{c}_{ij} + \mathrm{dist}(X_i, \alpha_j, \overline{c}) &= c_{ij} + \mathrm{dist}(X_i, \alpha_j, \overline{c}) - \pi_i^1 + \pi_j^2 \\ &= c_{ij} + \mathrm{dist}(X_i, \alpha_j, c^M) = \mathrm{cost}(C_{\{i,j\}}) \geq \beta - \mathrm{cost}(M), \end{aligned}$$

where $C_{\{i,j\}}$ is the shortest cycle in N^M that contains $\{i, j\}$. Obviously, this entire procedure runs in time $O(n(d + m \log m))$.

Interestingly, the idea of using reduced cost shortest path distances has been considered before to strengthen reduced cost propagation [9]. For an experimental evaluation of this idea, we refer to that paper. Now we have shown that this enhanced reduced cost propagation is powerful enough to guarantee arc-consistency for the minimum weight all different constraint.

4 Conclusions

We have introduced an arc-consistency algorithm for the minimum weight all different constraint that runs in time $O(n(d + m \log m))$. At first sight this sounds optimal, because it is the same time that is needed by algorithms for the weighted bipartite perfect matching problem such as the Hungarian method or the successive shortest path algorithm. However, two questions remain open: 1. Can we base a cost based filtering algorithm on the cost scaling algorithm that gives the best known time bound for assignment problems that satisfy the similarity assumption? And 2. Can the above filtering method be implemented to run incrementally faster?

References

1. R.K. Ahuja, T.L. Magnati, J.B. Orlin. *Network Flows.* Prentice Hall, 1993.
2. N. Barnier and P. Brisset. Graph Coloring for Air Traffic Flow Management, *Proc. of CPAIOR'02*, Le Croisic, France, pp. 133–147, March 2002.
3. Y. Caseau and F. Laburthe. Solving Various Weighted Matching Problems with Constraints, *Proc. of CP'97*, LNCS 1330, pp. 17–31, 1997.
4. Y. Caseau and F. Laburthe. Solving Small TSPs with Constraints, *Proc. of ICLP'97*, The MIT Press, 1997.
5. Y. Caseau and F. Laburthe. Heuristics for large constrained routing problems, *Journal of Heuristics*, 5:281–303, 1999.
6. T.H. Cormen, C.E. Leiserson, R.L. Rivest. *Introduction to Algorithms.* The MIT Press, 1990.
7. T. Fahle. Cost based Filtering vs. Upper Bounds for maximum Clique, *Proc. of CPAIOR'02*, Le Croisic, France, pp. 93–107, March 2002.
8. F. Focacci, A. Lodi, M. Milano. Solving TSP through the Integration of OR and CP Techniques, *Workshop on Large Scale Combinatorial Optimization and Constraints, CP'98*, Electronic Notes in Discrete Mathematics, 1998.
9. F. Focacci, A. Lodi, M. Milano. Integration of CP and OR methods for Matching Problems, *Proc. of CPAIOR'99*, Ferrara, Italy, 1999.
10. F. Focacci, A. Lodi, M. Milano. Cost-Based Domain Filtering. *Proceedings of the CP'99* Springer LNCS 1713:189–203, 1999.
11. J.C. Regin. A Filtering Algorithm for Constraints of Difference in CSPs, *Proc. of AAAI*, pp. 362–367, 1994.