

Parameterized Heuristics for Incomplete Weighted CSPs

Atena M. Tabakhi

Washington University in St. Louis
 One Brookings Dr. Jolley Hall, Suite 304
 St. Louis, MO, 63130
 amtabakhi@wustl.edu

Abstract

The key assumption in *Weighted Constraint Satisfaction Problems* (WCSPs) is that all constraints are specified a priori. This assumption does not hold in some applications that involve users preferences. *Incomplete WCSPs* (IWCSPs) extend WCSPs by allowing some constraints to be partially specified. Unfortunately, existing IWCSP approaches either guarantee to return optimal solutions or not provide any quality guarantees on solutions found. To bridge the two extremes, we propose a number of parameterized heuristics that allow users to find *boundedly-suboptimal solutions*, where the error bound depends on user-defined parameters. These heuristics thus allow users to trade off solution quality for fewer elicited preferences and faster computation times.

Introduction

In *Weighted Constraint Satisfaction Problems* (WCSPs), the goal is to find an optimal solution, given a set of preferences expressed by means of cost functions (Shapiro and Haralick 1981; Schiex, Fargier, and Verfaillie 1995). A key assumption in all these constraint-based models is that *all* the constraints are specified or known a priori. In some applications, such as roster and meeting scheduling problems, some constraints encode the preferences of human users. As such, they may not be fully specified simply because it is unrealistic to accurately know the preferences of users for all possible scenarios in an application. Motivated by such applications, (Tabakhi et al. 2017a; 2017b; 2018) assumed all cost functions are not specified a priori and ask users a number of preset questions to elicit cost functions before the search. While (Gelain et al. 2010) proposed the *Incomplete WCSP* (IWCSP) problem formulation, which extends WCSPs by allowing some constraints to be partially specified (i.e., the costs for some constraints are unknown). To solve IWCSPs, they introduced a series of algorithms that interleave the search process, which seeks to find a good solution, and the preference elicitation process, which seeks to obtain some subset of cost functions from the user.

Unfortunately, existing approaches fall into two extremes – they are either guaranteed to find optimal solutions or they do not provide any quality guarantees on the quality of solutions found. In this paper, we seek to bridge the two ex-

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

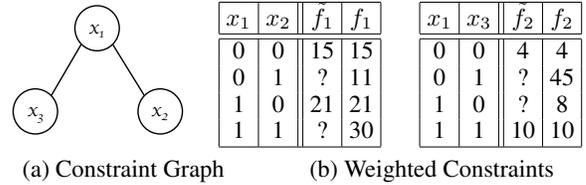


Figure 1: IWCSP Example

trems by introducing parameterized heuristics that allow one to find *boundedly-suboptimal solutions*. In other words, these heuristics will allow users to trade off solution quality for fewer elicited preferences and faster computation times.

An *Incomplete Weighted Constraint Satisfaction Problem* (IWCSP) extends WCSPs and is defined by a tuple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \tilde{\mathcal{F}} \rangle$, where \mathcal{X} , \mathcal{D} , and \mathcal{F} are a set of finite variables, domains, and weighted constraints, respectively, which are the same as WCSPs. The key difference is that the set of *fully-specified* constraints \mathcal{F} are not known to an IWCSP algorithm. Instead, only the set of partially-specified constraints $\tilde{\mathcal{F}}$ are known. More formally, $\tilde{\mathcal{F}} = \{ \tilde{f}_1, \dots, \tilde{f}_m \}$ is a set of *partially-specified* weighted constraints (or cost tables). Each partially-specified constraint is a function $\tilde{f}_i : \times_{x_j \in \mathbf{x}^{f_i}} D_j \rightarrow \mathbb{R}_0^+ \cup \{ \infty, ? \}$, where $?$ is a special element denoting that the cost for a given combination of value assignment is not specified. Further, the costs $\mathbb{R}_0^+ \cup \{ \infty \}$ that are specified are exactly the costs of the corresponding specified constraints $f_i \in \mathcal{F}$. A *solution* \mathbf{x} is a value assignment to a set of variables $X_{\mathbf{x}} \subseteq \mathcal{X}$ that is consistent with the variables' domains. The cost $\mathbf{F}_{\mathcal{P}}(\mathbf{x}) = \sum_{f \in \mathcal{F}, \mathbf{x}^f \subseteq X_{\mathbf{x}}} f(\mathbf{x})$ is the sum of the costs of all the applicable cost functions in \mathbf{x} . The goal is to find an optimal complete solution $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \mathbf{F}_{\mathcal{P}}(\mathbf{x})$ while eliciting as few unspecified costs as possible.

Figure 1(a) shows the constraint graph of an example IWCSP with three variables x_1 , x_2 , and x_3 where the domains are $D_1 = D_2 = D_3 = \{0, 1\}$. Figure 1(b) shows both the partially-specified and fully-specified cost tables for all constraints. In this example, the optimal complete solution is $\mathbf{x}^* = \langle x_1 = 0, x_2 = 1, x_3 = 0 \rangle$, which has a cost of 15. This can theoretically be found, but not proven to be optimal, by only eliciting the unknown cost $\tilde{f}_1(\langle x_1 = 0, x_2 = 1 \rangle)$ since $f_2(\langle x_1 = 0, x_3 = 0 \rangle)$ is known.

$ \mathcal{X} $	# unknown costs	# of elicited costs		runtime (sec)	
		MBE	SCE	MBE	SCE
5	12.51	8.26	10.73	3.14	0.10
6	18.39	12.31	14.89	31.89	0.51
7	24.18	16.5	18.27	260.48	3.49
8	33.06	22.28	25.46	179.88	36.67
9	42	31.44	33.76	3832.33	381.28
10	54	43.26	44.07	23630.17	4497.11

Table 1: Varying Number of Variables $|\mathcal{X}|$

Parameterized Heuristics

A simple and straightforward extension of the *depth-first branch-and-bound* (DFBnB) search algorithm to solve IWCSF is as follows: Before expanding a node n with some unknown costs, it elicits all the *unknown costs* associated with that node and adds those costs to the *known costs* associated with that node before proceeding. We refer to these costs as $g(n)$. Further, one can use heuristics, referred to as $h(n)$, to estimate the cost to complete the partial solution at node n and if those heuristics are underestimates on the true cost, then they can be used to better prune the search space, that is, when $f(n) = g(n) + h(n) \geq \mathbf{F}_{\mathcal{P}}(\mathbf{x})$, where \mathbf{x} is the best complete solution found so far.

Smallest Cost Elicitation (SCE) Heuristic

The *Smallest Cost Elicitation* (SCE) heuristic counts the minimal number of yet-to-be-elicited unknown costs that must be elicited to complete the partial solution corresponding to that node. Let $\phi(n)$ denote this number. Then, an underestimate on the minimal cost to complete the partial solution is $h(n) = \phi(n) \cdot \mathcal{L}$, where \mathcal{L} is the smallest cost across all functions $f \in \mathcal{F}$. Thus, when using this heuristic, DFBnB prunes a node n if $g(n) + w \cdot \phi(n) \cdot \mathcal{L} + \epsilon \geq \mathbf{F}_{\mathcal{P}}(\mathbf{x})$, where \mathbf{x} is the best complete solution found so far, $w \geq 1$ is the user-defined relative error bound, and $\epsilon \geq 0$ is the user-defined additive error bound. The solution found using this pruning condition is then guaranteed to have a cost that is bounded from above by $w \cdot \mathbf{F}_{\mathcal{P}}(\mathbf{x}^*) + \epsilon$, where \mathbf{x}^* is an optimal complete solution.

Matrix-Based Elicitation (MBE) Heuristic

The *Matrix-Based Elicitation* (MBE) heuristic builds a frequency matrix, where each row i corresponds to a range of known and elicited costs $[\ell_i, u_i]$ and each column j corresponds to a count j of yet-to-be-elicited unknown costs. Each element $e_{i,j}$ of the matrix then corresponds to the number of complete solutions under the subtree rooted at n whose sum of known and elicited costs is in the range $[\ell_i, u_i]$ and whose number of yet-to-be-elicited unknown costs is j . To use this heuristic, when evaluating a node n , DFBnB first constructs a submatrix of the first i rows of the frequency matrix, where $\ell_i \leq \mathbf{F}_{\mathcal{P}}(\mathbf{x}) < u_i$ and \mathbf{x} is the best complete solution found so far. Then, DFBnB prunes the node n if all elements of this submatrix are zeros (i.e., all complete solutions in the subtree rooted at n have known costs that are no smaller than $\mathbf{F}_{\mathcal{P}}(\mathbf{x})$).

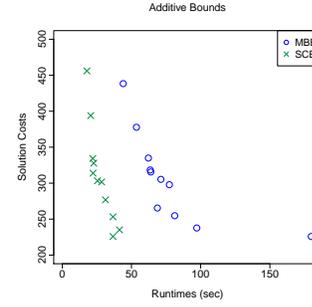


Figure 2: Solution Quality Trade off for Faster Runtime

Experimental Results

We generate 100 random (binary) graphs, where we varied the number of variables $|\mathcal{X}|$ from 5 to 10, with the domain size $|D_i| = 4$ for all variables, the constraint density $p_1 = 0.4$ and the fraction of unknown costs in each constraint $i = 0.2$. All known and elicited costs are randomly sampled from $[2, 100]$. Table 1 tabulates the results for the number of elicited constraint costs and the runtimes of DFBnB algorithm using our heuristics for solving IWCSFs. We empirically show that our solver does not need to elicit all unknown costs to find the optimal solution. In addition, Figure 2 shows the trade off between the solution quality and the runtime. The runtime decreases by increasing the user-defined additive error bound from 0 to 500 with increment of 50.

Conclusions and Future Work

In this paper, we introduce parameterized heuristics that allow one to find optimal solutions as well as boundedly-suboptimal solutions that also allow to trade off solution quality for fewer elicited preferences and smaller runtimes.

In the future, we plan to investigate different forms of eliciting preferences (e.g., eliciting actual constraint costs versus eliciting ordering of constraint costs).

References

- Gelain, M.; Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2010. Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies. *Artificial Intelligence*.
- Schiex, T.; Fargier, H.; and Verfaillie, G. 1995. Valued constraint satisfaction problems: Hard and easy problems. In *proceedings of IJCAI*.
- Shapiro, L. G., and Haralick, R. M. 1981. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Tabakhi, A. M.; Le, T.; Fioretto, F.; and Yeoh, W. 2017a. Preference elicitation for DCOPs. In *proceedings of CP*.
- Tabakhi, A. M.; Tourani, R.; Natividad, F.; Yeoh, W.; and Misra, S. 2017b. Pseudo-tree construction heuristics for dcops and evaluations on the ns-2 network simulator. In *proceedings of ICTAI*.
- Tabakhi, A. M.; Yeoh, W.; Tourani, R.; Natividad, F.; and Misra, S. 2018. Communication-sensitive pseudo-tree heuristics for dcop algorithms. *International Journal on Artificial Intelligence Tools*.