Preference Elicitation for DCOPs

Atena M. Tabakhi[1]([✉]), Tiep Le[2], Ferdinando Fioretto[3], and William Yeoh[1]

[1] Department of Computer Science and Engineering,
Washington University in St. Louis, St. Louis, USA
{amtabakhi,wyeoh}@wustl.edu

[2] Department of Computer Science, New Mexico State University,
Las Cruces, USA
tile@cs.nmsu.edu

[3] Department of Industrial and Operations Engineering,
University of Michigan, Ann Arbor, USA
fioretto@umich.edu

Abstract. *Distributed Constraint Optimization Problems* (DCOPs) offer a powerful approach for the description and resolution of cooperative multi-agent problems. In this model, a group of agents coordinate their actions to optimize a global objective function, taking into account their preferences or constraints. A core limitation of this model is the assumption that the preferences of all agents or the costs of all constraints are specified a priori. Unfortunately, this assumption does not hold in a number of application domains where preferences or constraints must be elicited from the users. One of such domains is the *Smart Home Device Scheduling* (SHDS) problem. Motivated by this limitation, we make the following contributions in this paper: **(1)** We propose a general model for preference elicitation in DCOPs; **(2)** We propose several heuristics to elicit preferences in DCOPs; and **(3)** We empirically evaluate the effect of these heuristics on random binary DCOPs as well as SHDS problems.

Keywords: Distributed Constraint Optimization · Smart homes · Preference elicitation

1 Introduction

The importance of constraint optimization is outlined by the impact of its application in a range of *Weighted Constraint Satisfaction Problems* (WCSPs), also known as *Constraint Optimization Problems* (COPs), such as supply chain management [34] and roster scheduling [1]. When resources are distributed among a set of autonomous agents and communication among the agents are

---

restricted, COPs take the form of *Distributed Constraint Optimization Problems* (DCOPs) [10,27,33,45]. In this context, agents coordinate their value assignments to minimize the overall sum of resulting constraint costs. DCOPs are suitable to model problems that are distributed in nature and where a collection of agents attempts to optimize a global objective within the confines of localized communication. They have been employed to model various distributed optimization problems, such as meeting scheduling [44,46], sensor networks [9], coalition formation [40], and smart grids [14,24].

The field of DCOP has matured significantly over the past decade since its inception [27]. DCOP researchers have proposed a wide variety of solution approaches, from complete approaches that use distributed search-based techniques [27,28,44] to distributed inference-based techniques [33,41]. There is also a significant body of work on incomplete methods that can be similarly categorized into local search-based methods [9,23], inference-based techniques [41], and sampling-based methods [11,29,31]. Researchers have also proposed the use of other off-the-shelf solvers such as logic programming solvers [21,22] and mixed-integer programming solvers [17].

One of the core limitations of all these approaches is that they assume that the constraint costs in a DCOP are known a priori. Unfortunately, in some application domains, these costs are only known after they are queried or elicited from experts or users in the domain. One such application is the *Smart Home Device Scheduling* (SHDS) problem [13]. In this problem, agents have to coordinate with each other to schedule smart devices (e.g., smart thermostats, smart lightbulbs, smart washers, etc.) distributed across a network of smart homes, where the goal is to schedule them in such a way that optimizes the preferences of occupants in those homes subject to a larger constraint that the peak energy demand in the network does not exceed an energy utility defined limit. Through the introduction of a number of smart devices in the commercial market, they are starting to become ubiquitous in today's very interconnected environment, consistent with the Internet-of-Things paradigm [25]. Therefore, we suspect that this SHDS problem will become more important in the future.

DCOPs are a natural framework to represent this problem as each home can be represented as an agent and the preferences of occupants can be represented as constraints. Furthermore, due to privacy reasons, it is preferred that the preferences of each occupant are not revealed to other occupants. The DCOP formulation allows the preservation of such privacy since agents are only aware of constraints that they are involved in. We further describe this motivating application and its mapping to DCOPs in more detail in Sect. 3.

A priori knowledge on the constraint costs is infeasible in our motivating SHDS application. A key challenge is thus in the elicitation of user preferences to populate the constraint cost tables. Due to the infeasibility of eliciting preferences to populate *all* preferences, in this paper, we introduce the *preference elicitation problem* for DCOPs, which studies *how to select a subset of k cost tables to elicit from each agent* with the goal of choosing those having a large impact on the overall solution quality. We propose several methods to select this

subset of cost tables to elicit, based on the notion of *partial orderings*. Additionally, we extend the SHDS problem to allow for the encoding and elicitation of soft preferences, and evaluate our methods on this extended SHDS problem as well as on random graphs to show generality. Our results illustrate the effectiveness of our approach in contrast to a baseline evaluator that randomly selects cost tables to elicit. While the description of our solution focuses on DCOPs, our approach is also suitable to solve WCSPs.

2   Background

WCSP: A *Weighted Constraint Satisfaction Problem* (WCSP) [20,36] is a tuple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F} \rangle$, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a finite set of variables, $\mathcal{D} = \{D_1, \ldots, D_n\}$ is a set of finite domains for the variables in $\mathcal{X}$, with D_i being the set of possible values for the variable x_i, and $\mathcal{F}$ is a set of *weighted constraints* (or *cost tables*). A weighted constraint $f_i \in \mathcal{F}$ is a function, $f_i : \times_{x_j \in \mathbf{x}^{f_i}} D_j \to \mathbb{R}_0^+ \cup \{\bot\}$, where $\mathbf{x}^{f_i} \subseteq \mathcal{X}$ is the set of variables relevant to f_i, referred to as the *scope* of f_i, and $\bot$ is a special element used to denote that a given combination of value assignments is not allowed. 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 $\mathbf{x}$. A solution $\mathbf{x}$ is said *complete* if $X_\mathbf{x} = \mathcal{X}$. The goal is to find an optimal complete solution $\mathbf{x}^* = \mathrm{argmin}_\mathbf{x} \mathbf{F}_\mathcal{P}(\mathbf{x})$.

DCOP: When the elements of a WCSP are distributed among a set of autonomous agents, we refer to it as a *Distributed Constraint Optimization Problem* (DCOP) [27,33,45]. Formally, a DCOP is described by a tuple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$, where $\mathcal{X}, \mathcal{D}$, and $\mathcal{F}$ are the set of variables, their domains, and the set of cost functions, defined as in a classical WCSP, $\mathcal{A} = \{a_1, \ldots, a_p\}$ $(p \leq n)$ is a set of autonomous agents, and $\alpha : \mathcal{X} \to \mathcal{A}$ is a surjective function, from variables to agents, mapping the control of each variable $x \in \mathcal{X}$ to an agent $\alpha(x)$. The goal in a DCOP is to find a complete solution that minimizes its cost: $\mathbf{x}^* = \mathrm{argmin}_\mathbf{x} \mathbf{F}_\mathcal{P}(\mathbf{x})$. A DCOP can be described by a *constraint graph*, where the nodes correspond to the variables in the DCOP, and the edges connect pairs of variables in the scope of the same cost functions. Following [12], we introduce the following definitions:

Definition 1. *For each agent* $a_i \in \mathcal{A}$, $\mathbf{L}_i = \{x_j \in \mathcal{X} \mid \alpha(x_j) = a_i\}$ *is the set of its* local *variables.* $\mathbf{I}_i = \{x_j \in \mathbf{L}_i \mid \exists x_k \in \mathcal{X} \wedge \exists f_s \in \mathcal{F} : \alpha(x_k) \neq a_i \wedge \{x_j, x_k\} \subseteq \mathbf{x}^{f_s}\}$ *is the set of its* interface *variables.*

Definition 2. *For each agent* $a_i \in \mathcal{A}$, *its local constraint graph* $G_i = (\mathbf{L}_i, \mathcal{E}_{\mathcal{F}_i})$ *is a subgraph of the constraint graph, where* $\mathcal{F}_i = \{f_j \in \mathcal{F} \mid \mathbf{x}^{f_j} \subseteq \mathbf{L}_i\}$.

Figure 1(a) shows the constraint graph of a sample DCOP with 3 agents a_1, a_2, and a_3, where $\mathbf{L}_1 = \{x_1, x_2\}$, $\mathbf{L}_2 = \{x_3, x_4\}$, $\mathbf{L}_3 = \{x_5, x_6\}$, $\mathbf{I}_1 = \{x_2\}$, $\mathbf{I}_2 = \{x_4\}$, and $\mathbf{I}_3 = \{x_6\}$. The domains are $D_1 = \cdots = D_6 = \{0, 1\}$. Figure 1(b) shows the cost table of all constraints; all constraints have the same cost table for simplicity.
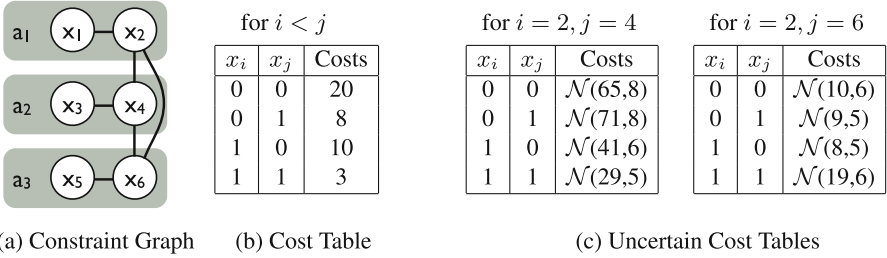
for $i < j$		
x_i	x_j	Costs
0	0	20
0	1	8
1	0	10
1	1	3

for $i = 2, j = 4$		
x_i	x_j	Costs
0	0	$\mathcal{N}(65,8)$
0	1	$\mathcal{N}(71,8)$
1	0	$\mathcal{N}(41,6)$
1	1	$\mathcal{N}(29,5)$

for $i = 2, j = 6$		
x_i	x_j	Costs
0	0	$\mathcal{N}(10,6)$
0	1	$\mathcal{N}(9,5)$
1	0	$\mathcal{N}(8,5)$
1	1	$\mathcal{N}(19,6)$

(a) Constraint Graph   (b) Cost Table   (c) Uncertain Cost Tables

Fig. 1. Example DCOP and uncertain DCOP

3 Motivating Domain: Smart Home Device Scheduling Problem

We now provide a description of (a variant of) the *Smart Home Device Scheduling* (SHDS) problem [13]. An SHDS problem is composed of a neighborhood $\mathcal{H}$ of smart homes $h_i \in \mathcal{H}$ that are able to communicate with one another and whose energy demands are served by an energy provider. The energy prices are set according to a real-time pricing schema specified at regular intervals t within a finite time horizon H. We use $\mathbf{T} = \{1, \ldots, H\}$ to denote the set of time intervals and $\theta : \mathbf{T} \to \mathbb{R}^+$ to represent the price function associated with the pricing schema adopted, which expresses the cost per kWh of energy consumed by a consumer.

Within each smart home h_i there is a set of (smart) electric devices $\mathcal{Z}_i$ networked together and controlled by a home automation system. We assume all the devices are uninterruptible (i.e., they cannot be stopped once they are started) and use s_{z_j} and δ_{z_j} to denote, respectively, the start time and duration (expressed in multiples of time intervals) of device $z_j \in \mathcal{Z}_i$. The energy consumption of each device z_j is ρ_{z_j} kWh for each hour that it is *on*. It will not consume any energy if it is *off*. We use the indicator function $\phi_{z_j}^t$ to indicate the state of the device z_j at time step t:

$$\phi_{z_j}^t = \begin{cases} 1 \text{ if } & s_{z_j} \leq t \wedge s_{z_j} + \delta_{z_j} \geq t \\ 0 \text{ otherwise} \end{cases}$$

Additionally, the usage of a device z_j is characterized by a *cost*, representing the monetary expense to schedule z_j at a given time. The aggregated cost of the home h_i at time step t is denoted with C_i^t and expressed as:

$$C_i^t = E_i^t \cdot \theta(t) \tag{1}$$

where $E_i^t = \sum_{z_j \in \mathcal{Z}_i} \phi_{z_j}^t \cdot \rho_{z_j}$ is the aggregated energy consumed by home h_i at time step t.

The SHDS problem seeks a schedule for the devices of each home in the neighborhood in a coordinated fashion so as to minimize the monetary costs and, at the same time, ensure that user-defined scheduling constraints (called

active scheduling rules in [13]) are satisfied. The SHDS problem is also subject to the following constraints:

$$1 \leq s_{z_j} \leq T - \delta_{z_j} \qquad \forall h_i \in \mathcal{H}, z_j \in \mathcal{Z}_i \qquad (2)$$

$$\sum_{t \in \mathbf{T}} \phi_{z_j}^t = \delta_{z_j} \qquad \forall h_i \in \mathcal{H}, z_j \in \mathcal{Z}_i \qquad (3)$$

$$\sum_{h_i \in \mathcal{H}} E_i^t \leq \ell^t \qquad \forall t \in \mathbf{T} \qquad (4)$$

where $\ell^t \in \mathbb{R}^+$ is the maximum allowed total energy consumed by all the homes in the neighborhood at time step t. This constraint is typically imposed by the energy provider and is adopted to guarantee reliable electricity delivery. Constraint (2) expresses the lower and upper bounds for the start time associated to the schedule of each device. Constraint (3) ensures the devices are scheduled and executed for exactly their duration time. Constraint (4) ensures the total amount of energy consumed by the homes in the neighborhood does not exceed the maximum allowed threshold.

3.1   DCOP Representation

Fioretto *et al.* introduced a mapping of the SHDS problem to a DCOP [13]. At a high level, each home $h_i \in \mathcal{H}$ is mapped to an autonomous agent in the DCOP. For each home, the start times s_{z_j}, indicator variables $\phi_{z_j}^t$, and aggregated energy in the home are mapped to DCOP variables, which are controlled by the agent for that home. Constraints (2) to (4) are enforced by the DCOP constraints. Finally, the objective function of the SHDS is expressed through agents' preferences.

4   Encoding and Eliciting Preferences in SHDS

The above SHDS problem thus far includes exclusively hard constraints and has no soft constraints (i.e., preferences for when devices are scheduled). Thus, we will describe in this section how to integrate such preferences as soft constraints into SHDS.

We consider the scenario in which a single home h_i may host multiple users $u \in \mathbf{U}_{h_i}$, with $\mathbf{U}_{h_i}$ denoting the set of users in h_i. In modeling agents' preferences, we introduce *discomfort values* $d_{z_j,u}^t \in \mathbb{R}_0^+$ describing the degree of dissatisfaction for a user u to schedule the device z_j at a given time step t. Note that the monetary cost is the same for all users while the degree of dissatisfaction is user dependent. Thus, to avoid conflicting users' decision over the control of the device, we assume that there is one user who has exclusive access to a device $z \in \mathcal{Z}_i$ at any point in time. In this paper, for each device $z_j \in \mathcal{Z}_i$ in home h_i and each time step t, we assume the likelihood for a user to gain exclusive access on a device z_j is expressed through a probability $Pr_{z_j}^t$ (i.e., $\forall u \in \mathbf{U}_{h_i}, Pr_{z_j}^t(u) \in [0,1]$ and $\sum_{u \in \mathbf{U}_{h_i}} Pr_{z_j}^t(u) = 1$). Additionally, we use $\mathbf{d}_i^t = \sum_{z_j \in \mathcal{Z}_i} \phi_{z_j}^t \cdot d_{z_j}^t$ to denote

the aggregated discomfort in home h_i at time step t, where $d_{z_j}^t$ is the discomfort value of the user who has exclusive access to the device z_j at time step t.

We can update the SHDS objective to take into account the users' preferences in addition to minimizing the monetary costs. While this is a multi-objective problem, we combine the two objectives into a single one through the use of a weighted sum:

$$\textbf{minimize} \quad \sum_{t \in \textbf{T}} \sum_{h_i \in \mathcal{H}} \alpha_c \cdot C_i^t + \alpha_u \cdot \textbf{d}_i^t \tag{5}$$

where α_c and α_u are weights in the open interval $(0, 1) \subseteq \mathbb{R}$ such that $\alpha_c + \alpha_u = 1$.

While, in general, the real-time pricing schema θ that defines the cost per kWh of energy consumed and the energy consumption ρ_{z_j} of each device z_j are well-defined concepts and can be easily acquired or modeled, the preferences on the users' discomfort values $d_{z_j,u}^t$ on scheduling a device z_j at time step t are subjective and, thus, more difficult to model explicitly.

We foresee two approaches to acquire these preferences: *(1)* eliciting them directly from the users and *(2)* estimating them based on historical preferences or from preferences of similar users. While the former method will be more accurate and reliable, it is cumbersome for the user to enter their preference for every device z_j and every time step t of the problem. Therefore, in this paper, we assume that a combination of the two approaches will be used, where a subset of preferences will be elicited and the remaining preferences will be estimated from historical sources or similar users. We believe that this strategy is especially important in application domains such as the SHDS problem, where users' preferences may be learned over time, thus, ensuring a continuous elicitation process of the unknown users preferences.

5   Preference Elicitation in DCOPs

A key drawback of existing DCOP approaches is the underlying assumption of a *total* knowledge of the model, which is not the case for a number of applications involving users' preferences, including the SHDS problem. Due to the infeasibility of eliciting *all* users' preferences—and, thus, their associated complete cost tables—in this paper, we study how to choose a subset of k cost tables to elicit. We first cast this problem as an optimization problem, before describing our proposed techniques.

Let $\hat{\mathcal{P}} = \langle \mathcal{X}, \mathcal{D}, \hat{\mathcal{F}}, \mathcal{A}, \alpha \rangle$ denote a DCOP with partial knowledge on the cost tables in $\hat{\mathcal{F}}$. The constraints $\hat{\mathcal{F}} = \mathcal{F}_r \cup \mathcal{F}_u$ are composed of *revealed constraints* $\mathcal{F}_r$, whose cost tables are accurately revealed, and *uncertain constraints* $\mathcal{F}_u$, whose cost tables are unrevealed and must be either estimated from historical sources or elicited. We refer to this problem as the *uncertain DCOP*.

In this paper, we assume that the costs of the uncertain constraints are sampled from Normal distributions that can be estimated from historical sources.[1]

---

[1] Other forms of distributions can also be used, but our minimax regret heuristics require that the form of the distributions have the following property: The sum of two distributions has the same form as their individual distributions.
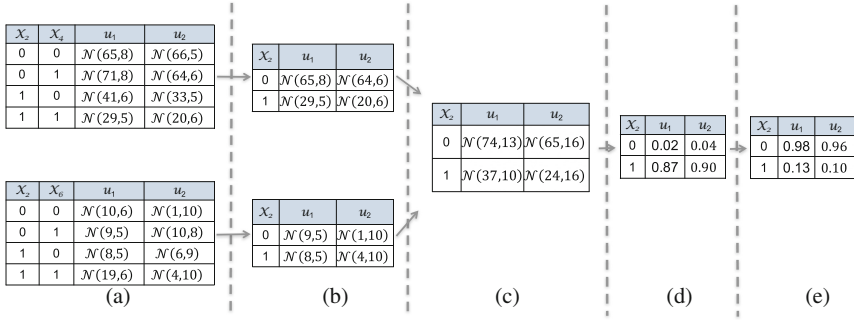
X_2	X_4	u_1	u_2
0	0	$\mathcal{N}(65,8)$	$\mathcal{N}(66,5)$
0	1	$\mathcal{N}(71,8)$	$\mathcal{N}(64,6)$
1	0	$\mathcal{N}(41,6)$	$\mathcal{N}(33,5)$
1	1	$\mathcal{N}(29,5)$	$\mathcal{N}(20,6)$

X_2	u_1	u_2
0	$\mathcal{N}(65,8)$	$\mathcal{N}(64,6)$
1	$\mathcal{N}(29,5)$	$\mathcal{N}(20,6)$

X_2	u_1	u_2
0	$\mathcal{N}(74,13)$	$\mathcal{N}(65,16)$
1	$\mathcal{N}(37,10)$	$\mathcal{N}(24,16)$

X_2	u_1	u_2
0	0.02	0.04
1	0.87	0.90

X_2	u_1	u_2
0	0.98	0.96
1	0.13	0.10

X_2	X_6	u_1	u_2
0	0	$\mathcal{N}(10,6)$	$\mathcal{N}(1,10)$
0	1	$\mathcal{N}(9,5)$	$\mathcal{N}(10,8)$
1	0	$\mathcal{N}(8,5)$	$\mathcal{N}(6,9)$
1	1	$\mathcal{N}(19,6)$	$\mathcal{N}(4,10)$

X_2	u_1	u_2
0	$\mathcal{N}(9,5)$	$\mathcal{N}(1,10)$
1	$\mathcal{N}(8,5)$	$\mathcal{N}(4,10)$

(a)   (b)   (c)   (d)   (e)

Fig. 2. Minimax regret example

Further, we assume that the distribution for each cost value is independent from the distribution of all other cost values. Figure 1(c) illustrates an uncertain cost table whose costs are modeled via random variables obeying Normal distributions, and u_1 and u_2 denote two distinct users that can control the associated device.

5.1   The Preference Elicitation Problem

The preference elicitation problem in DCOPs is formalized as follows: Given an oracle DCOP $\mathcal{P}$ and a value $k \in \mathbb{N}$, construct an uncertain DCOP $\hat{\mathcal{P}}$ that reveals only k constraints per agent (i.e., $|\mathcal{F}_r| = k \cdot |\mathcal{X}|$) and minimizes the error:

$$\epsilon_{\hat{\mathcal{P}}} = \mathbb{E}\big[\mathbf{F}_{\mathcal{P}}(\hat{\mathbf{x}}^*) - \mathbf{F}_{\mathcal{P}}(\mathbf{x}^*)\big] \tag{6}$$

where $\hat{\mathbf{x}}^*$ is the optimal solution for a *realization* of the uncertain DCOP $\hat{\mathcal{P}}$, and $\mathbf{x}^*$ is the optimal complete solution for the oracle DCOP $\mathcal{P}$. A realization of an uncertain DCOP $\hat{\mathcal{P}}$ is a DCOP (with no uncertainty), whose values for the cost tables are sampled from their corresponding Normal distributions. Note that the possible numbers of uncertain DCOPs that can be generated is $\binom{|\mathcal{F}|}{k \cdot |\mathcal{X}|}$. Since solving each DCOP is NP-hard [26], the preference elicitation problem is a particularly challenging one. Thus, we propose a number of heuristic methods to determine the subset of constraints to reveal, and to construct an uncertain problem $\hat{\mathcal{P}}$.

5.2   Preference Elicitation Heuristics

Let us first introduce a general concept of dominance between cost tables of uncertain constraints. Given two cost tables of uncertain constraints $f_{z_i}, f_{z_j} \in \mathcal{F}_u \subset \hat{\mathcal{F}}$, let $\succeq_\circ$ denote the dominance between the two cost tables according to partial ordering criteria $\circ$. In other words, $f_{z_i} \succeq_\circ f_{z_j}$ means that f_{z_i} *dominates* f_{z_j} according to criteria $\circ$. We now introduce the heuristic methods for different possible ordering criteria $\circ$.

Minimax Regret: *Minimax regret* is a well-known strategy that minimizes the maximum regret, and it is particularly suitable in a risk-neutral environment. At a high level, the minimax regret approach seeks to approximate and minimize the impact of the worst-case scenario. The idea of using minimax regret in our domain of interest is derived by the desire of taking into account the possible different outcomes occurring when eliciting the preferences of different users for a single device. Further, we assume that constraints that can be elicited are either unary or binary constraints. We leave to future work the extension to higher arity constraints. We now describe how to compute the regret for a single user u, and later how to combine the regrets across multiple users.

We use $Pr_{x_i}(d)$ to estimate the likelihood of an assignment $d \in D_i$ to a variable x_i:

$$Pr_{x_i,u}(d) = \Pi_{d' \in D_i \setminus \{d\}} Pr(\psi_{x_i,u}^d \le \psi_{x_i,u}^{d'}) \tag{7}$$

where $\psi_{x_i,u}^d$ is the random variable representing the total cost incurred by x_i if it is assigned value d from its domain under user u. Then, the value

$$d_{x_i,u}^* = \operatorname*{argmax}_{d} \; Pr_{x_i,u}(d) \tag{8}$$

with the largest probability is the one that is most likely to be assigned to x_i.

The probability $Pr(\psi_{x_i,u}^d \le \psi_{x_i,u}^{d'})$ can be computed using:

$$Pr(\psi_{x_i,u}^d \le \psi_{x_i,u}^{d'}) = \int_{c'=0}^{\infty} \int_{c=0}^{c'} Pr_{x_i,u}^d(c) Pr_{x_i,u}^{d'}(c') \, dc \, dc' \tag{9}$$

where $Pr_{x_i,u}^d$ is the *probability distribution function* (PDF) for random variable $\psi_{x_i,u}^d$. Unfortunately, the PDF $Pr_{x_i,u}^d$ is not explicitly defined in the uncertain DCOP. There are two challenges that one needs to address to obtain or estimate this PDF:

i. First, the total cost incurred by an agent is the summation of the costs over all constraints of that agent. Thus, the PDF for the total cost needs to be obtained by summing over the PDFs of all the individual constraint costs. Since we assume that these PDFs are all Normally distributed, one can efficiently construct the summed PDF, which is also a Normal distribution. Specifically, if $\mathcal{N}(\mu_i, \sigma_i^2)$ is the PDF for random variables c_i ($i = 1, 2$), then $\mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$ is the PDF for $c_1 + c_2$.

ii. Second, the cost associated to a variable for each constraint is not only dependent on its value but also on the value of the other variables constrained with it. In turn, the value of those variables depend on the variables that they are constrained with, and so on. As a result, estimating the true PDF requires the estimation of all the constraint costs in the entire DCOP. To simplify the computation process and introduce an independence between the costs of all variables, we propose the three following variants, each of which *estimates* the true PDF $Pr_{x_i,u}^{d,f}$ of a random variable $\psi_{x_i,u}^{d,f}$, representing the cost incurred by x_i from constraint f if assigned value d when its control is under user u:

- OPTIMISTIC: In this variant, the agent will optimistically choose the PDF with smallest mean among all the PDFs for all possible values of variables $x_j \in \mathbf{x}^f \setminus \{x_i\}$ in the scope of constraint f:

$$Pr_{x_i,u}^{d,f} = \mathcal{N}(\mu^*, \sigma_{\hat{d}}^2) \tag{10}$$

$$\mu^* = \min_{\hat{d} \in D_j} \mu_{\hat{d}} \tag{11}$$

where $\mathcal{N}(\mu_{\hat{d}}, \sigma_{\hat{d}}^2)$ is the PDF of the constraint cost if $x_i = d$ and $x_j = \hat{d}$ under user u. For example, in the uncertain cost tables in Fig. 1(c), the estimated PDF of the cost incurred for the choice $x_2 = 0$ from constraint f_{24} is $Pr_{x_2,u}^{0,f_{24}} = \mathcal{N}(65, 8^2)$, which *optimistically* assumes that x_4 will be assigned value 0 to minimize the incurred cost.

- PESSIMISTIC: In this variant, the agent chooses the PDF with largest mean among all the PDFs for all possible values of $x_j \in \mathbf{x}^f \setminus \{x_i\}$:

$$Pr_{x_i,u}^{d,f} = \mathcal{N}(\mu^*, \sigma_{\hat{d}}^2) \tag{12}$$

$$\mu^* = \max_{\hat{d} \in D_j} \mu_{\hat{d}} \tag{13}$$

In Fig. 1(c), the estimated PDF of the cost incurred by $x_2 = 0$ from constraint f_{24} is $Pr_{x_2,u}^{0,f_{24}} = \mathcal{N}(71, 8^2)$, which *pessimistically* assumes that x_4 will be assigned value 1 to maximize the incurred cost.

- EXPECTED: In this variant, the agent chooses the PDF with the "average" value of all the PDFs for all possible values of $x_j \in \mathbf{x}^f \setminus \{x_i\}$:

$$Pr_{x_i,u}^{d,f} = \mathcal{N}\left(\frac{1}{|D_j|} \sum_{\hat{d} \in D_j} \mu_{\hat{d}}, \frac{1}{|D_j|^2} \sum_{\hat{d} \in D_j} \sigma_{\hat{d}}^2 \right) \tag{14}$$

In Fig. 1(c), the estimated PDF of the cost incurred by $x_2 = 0$ from constraint f_{24} is $Pr_{x_2,u}^{0,f_{24}} = \mathcal{N}(68, 8^2)$, assuming that $x_4 = 0$ or $x_4 = 1$ with equal probability.

The *regret* $R_{x_i,u}^d$ of variable x_i being assigned value d is defined as:

$$R_{x_i,u}^d = 1 - Pr_{x_i,u}(d) \tag{15}$$

Each variable x_i will most likely be assigned the value $d_{x_i}^*$ with the smallest regret by definition (see Eqs. 7 and 8). We thus define the regret $R_{x_i,u}$ for each variable x_i to be the regret for this value:

$$R_{x_i,u} = R_{x_i,u}^{d_{x_i}^*} = \min_{d \in D_i} R_{x_i,u}^d \tag{16}$$

To generalize our approach to also handle multiple users in each house, where the PDFs differ across users, we take the maximum regret over all users u for

each variable x_i and its value d before taking the minimum over all values. More precisely,

$$R_{x_i} = \min_{d \in D_i} \max_s R^d_{x_i,u} \qquad (17)$$

Therefore, the minimax regret approach seeks to approximate the impact of the worst-case scenario. Finally, we define the regret R_{f_i} for a constraint f_i to be the absolute difference between the regrets of the variables in the scope of the function:

$$R_{f_i} = |R_{x_{i_1}} - R_{x_{i_2}}| \qquad (18)$$

where $\mathbf{x}^{f_i} = \{x_{i_1}, x_{i_2}\}$.

While defining the regret to be the sum of the two variables' regrets may be more intuitive, our experimental results show that the above definition provides better results. Intuitively, if the regret of a variable x_i is large, then there is little confidence that it will take on value $d^*_{x_i}$ with the smallest regret because the PDFs for all its values are very similar and have significant overlaps. Thus, eliciting a constraint between two variables with large regrets will likely not help in improving the overall solution quality since the PDFs for all value combinations for that constraint are likely to be similar.

Similarly, if the regret of a variable is small, then there is a high confidence that it will be assigned value with the smallest regret because the PDFs for its values are sufficiently distinct that regardless of the actual realizations of the random variables (costs in the cost table), the value with the smallest regret will be the one with the smallest cost. Therefore, eliciting a constraint between two agents with small regrets will also not help. Therefore, we define the regret of a constraint to be the difference in the regrets of the variables in its scope (see Eq. 18).

If we order the constraints using the ordering criteria $\circ = MR[\cdot]$, that is, according to the minimax regret criterion, then, given two uncertain constraints $f_i, f_j \in \mathcal{F}_u$, we say that $f_i \succeq_{MR} f_j$ iff $MR[f_i] \geq MR[f_j]$, where $MR[f_j] = R_{f_j}$ is the regret as defined in Eq. 18.

Figure 2 illustrates a partial trace of this approach on the example DCOP of Fig. 1 with two users u_1 and u_2. Figure 2(a) shows the uncertain cost tables for constraint f_{24} between variables x_2 and x_4 and constraint f_{26} between variables x_2 and x_6. Figure 2(b) shows the estimated PDFs $Pr^{d}_{x_2,u}$ of the constraint costs incurred by variables x_2 from constraint f under user u if it takes on value d. In this trace, we use the "optimistic" variant of the algorithm, and the PDFs are estimated using Eqs. 10 and 11. Figure 2(c) shows estimated summed PDFs $Pr^{d,f}_{x_2,u}$ of the *total* constraint costs incurred by the agent, summed over all of its constraints. Here, we only sum the PDFs for the two constraints f_{24} and f_{26}. Figure 2(d) shows the probabilities $Pr_{x_2,u}(d)$ of $x_2 = d$ under user u, computed using Eq. 7, and Fig. 2(e) shows the regrets $R^d_{x_2,u}$, computed using Eq. 15. Thus, the regret R_{x_2} for x_2 is 0.13, computed using Eq. 17. Assume that the regret R_{x_1} of x_1 is 0.50. Then, the regret $R_{f_{12}}$ of constraint $f_{12} = |R_{x_1} - R_{x_2}| = |0.50 - 0.13| = 0.37$.

Maximum Standard Deviation: We now propose a different heuristic that makes use of the degree of uncertainty in the constraint costs $\chi_{f,u}^v$ for constraint f, value combination $v = \langle x_{i_1} = d_{i_1}, \ldots, x_{i_k} = d_{i_k} \rangle$, and user u, where $\mathbf{x}^f = \{x_{i_1}, \ldots, x_{i_k}\}$, $d_{i_1} \in D_{i_1}$, ..., and $d_{i_k} \in D_{i_k}$.

Assume that there is only a single user u. Then, using the same motivation described for the minimax regret heuristic, and assuming that *variables be assigned a different value for different constraints*, the value combination chosen for a constraint f will be the $v^* = \operatorname{argmin}_v \chi_{f,u}^v$ that has the smallest cost. Unfortunately, the actual constraint costs are not known and only their PDFs $\mathcal{N}(\mu_{f,u}^v, (\sigma_{f,u}^v)^2)$ are known.

Since the constraint costs' distribution means are known, we assume that the value combination chosen for a constraint f will be the value $v^* = \operatorname{argmin}_v \mu_{f,u}^v$ that has the smallest mean. The degree of uncertainty in the constraint cost for that constraint f is thus the standard deviation associated $\sigma_{f,u}^{v^*}$ with that value combination v^*.

To generalize this approach to multiple users, we take the maximum standard deviations over all users u. More precisely, the degree of uncertainty in the constraint costs for a constraint f is:

$$\sigma_f = \max_s \sigma_{f,u}^{v^*} \tag{19}$$

One can then use this maximum standard deviation criterion to order the constraints. In other words, if the ordering criteria $\circ = MS[\cdot]$ is done according to the maximum standard deviation criterion, then, given two unknown functions $f_i, f_j \in \mathcal{F}_u$, we say that $f_i \succeq_{MS} f_j$ iff $MS[f_i] \geq MS[f_j]$, where $MS[f_j] = \sigma_{f_j}$ is the maximum standard deviation as defined in Eq. 19.

6   Related Work

There is an extensive body of work on the topic of modeling preferences [16]. In particular, Rossi *et al.* discussed *conditional-preference networks* (CP-nets) for handling preferences [35], which provide a qualitative graphical representation of preferences reflecting the conditional dependence of the problem variables. Differently from CP-nets, our proposal focuses on the notion of *conditional additive independence* [3], which requires the utility of an outcome to be the sum of the "utilities" of the different variable values of the outcome.

In terms of preference elicitation, two major approaches are studied in the literature [6]: A Bayesian approach [4,7] and a minimax regret approach [5,15,42]. The former is typically adopted when the uncertainty can be quantified probabilistically, and preference elicitation is often formalized as a *partially-observable Markov decision process* (POMDP) [18] that assumes each query to a user is associated with a finite set of possible responses. In contrast, our proposal follows the minimax regret approach [5,15,42]. The proposed framework differs from other proposals in the literature in the following ways: We assume the unknown costs are sampled from a Normal distribution and compute the regret based on

such distributions. In contrast, other minimax regret based methods have different assumptions. For example, Boutilier *et al.* assumes that a set of (hard) constraints together with a graphical utility model captures user preferences [5]. While the structure of the utility model is known, the parameters of this utility model are imprecise, given by upper and lower bounds. The notion of regret is computed based on those upper and lower bounds. Differently, Wang and Boutilier computes regrets under the assumption that constraints over unknown utility values are linear [42]. Finally, Gelain *et al.* computes regrets by taking the minimum among the known utilities associated to the projections of an assignment, that is, of the appropriated sub-tuples in the constraints [15].

Finally, preference elicitation has never been applied directly on DCOPs before. The closest DCOP-related problem is a class of DCOPs where agents have partial knowledge on the costs of their constraints and, therefore, they may discover the unknown costs via exploration [39, 47]. In this context, agents must balance the coordinated *exploration* of the unknown environment and the *exploitation* of the known portion of the rewards, in order to optimize the global objective [37]. Another orthogonal related DCOP model is the problem where costs are sampled from probability distribution functions [30]. In such a problem, agents seek to minimize either the worst-case regret [43] or the expected regret [21].

7   Empirical Evaluation

We evaluate our preference elicitation framework on distributed random binary graphs and smart home device scheduling (SHDS) problems [13], where we compared our four heuristics–minimax regret with the three variants: optimistic (MR-O), pessimistic (MR-P), and expected (MR-E) and maximum standard deviation (MS)–against a random baseline (RD) that chooses the constraints to elicit randomly. All the problems are modeled and solved optimally on multiple computers with Intel Core i7-3770 CPU $3.40\,\text{GHz}$ and $16\,\text{GB}$ of RAM. We use MiniZinc [38], an off-the-shelf centralized CP solver, to solve all the DCOPs.

In our experiments the preference elicitation heuristics are evaluated in terms of the *normalized error* $\frac{\epsilon_{\hat{\mathcal{P}}}}{\mathbf{F}_{\mathcal{P}}(\mathbf{x}^*)}$, where $\epsilon_{\hat{\mathcal{P}}}$ is the error as defined by Eq. 6. An accurate computation of this error requires us to generate all possible realizations for the uncertain DCOPs. Due to the complexity of such task, we create $m = 50$ realizations of the uncertain DCOPs and compute the error $\epsilon_{\hat{\mathcal{P}}}$ in this reduced sampled space.

7.1   Random Graphs

We create 100 random graphs whose topologies are based on the Erdős and Rényi model [8] with the following parameters: $|\mathcal{X}| = 50$, $|\mathcal{A}| = 5$, and $|D_i| = 2$ for all variables $x_i \in \mathcal{X}$. Each agent a_i has $|\mathbf{L}_i| = 10$ local variables with density $p_1 = 0.8$ that produces $|\mathcal{F}_i| = 36$ local constraints per agent. These constraints are unknown (uncertain constraints) and we set two scenarios (called
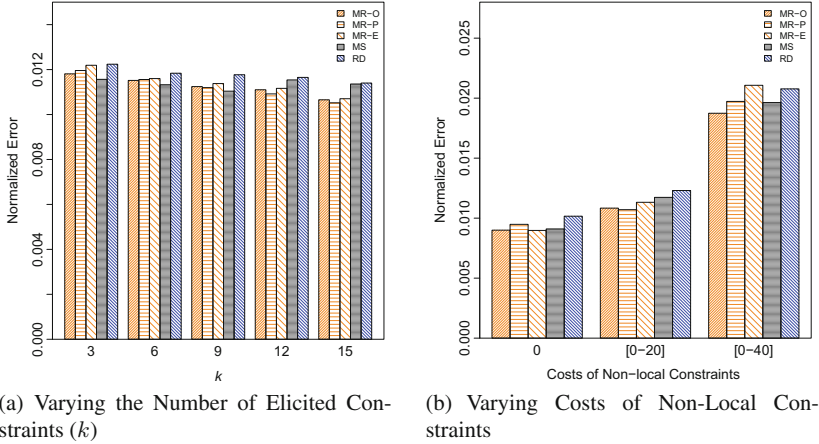
(a) Varying the Number of Elicited Constraints (k)

(b) Varying Costs of Non-Local Constraints

Fig. 3. Random graphs preference elicitation

users in Sect. 5) for all uncertain cost tables. All constraint costs are modeled as random variables following a Normal distribution $\mathcal{N}(\mu, \sigma^2)$, where σ is uniformly sampled from the range $[5, 10]$ and the means μ are uniformly sampled from one of the following six ranges $[5, 70]$, $[5, 80]$, $[5, 90]$, $[5, 100]$, $[5, 110]$, and $[5, 120]$. The different ranges are to introduce some heterogeneity into the constraints. We set the non-local constraints (i.e., inter-agent constraints) to be uncertain constraints as well, where we vary the mean μ of their Normal distributions to be from different distributions: $\mu = 0$; $\mu \in [0, 20]$; and $\mu \in [0, 40]$. Finally, we allow only local constraints (or preferences) to be elicited.

Figure 3(a) illustrates the normalized errors of our heuristics and that of the random baseline heuristic, where the mean μ of the non-local constraints are uniformly sampled from the range $[0, 20]$. We control k so that the number of the constraints per agent elicited from the oracle DCOP varies from 3 to 15 with increment of 3. We make the following observations:

- As the number of constraints (k) to elicit increases, the errors of the MR-P and MR-O heuristic decrease for all values of k as opposed to the random heuristic which is approximately the same for all values of k. The reason is, as we increase k, the random heuristic randomly selects k constraint to elicit with high likelihood of choosing the wrong constraints. However, since the regret-based heuristic (e.g., MR-P) takes into account the uncertain cost of the constraints it chooses those minimizing the regret.
- The MS heuristic performs slightly better than random heuristic. The reason is that MS orders the uncertain constraints by their degrees of uncertainty (i.e., σ) corresponding to the most likely value combinations to be assigned (i.e., the ones with the smallest μ). In contrast, the random heuristic chooses constraint randomly without taking into account the degree of uncertainty.

Table 1. Smart devices and their energy consumption (in kWh)

Dish-washer	Washer	Dryer	Hob	Oven	Microwave	Laptop	Desktop	Vacuum cleaner	Fridge	Electrical vehicle
0.75	1.20	2.50	3.00	5.00	1.70	0.10	0.30	1.20	0.30	3.50

- All regret-based heuristics outperform the baseline heuristic, especially for larger values of k, indicating that they are able to effectively take the regrets of the constraints into account.

Figure 3(b) illustrates the normalized errors for the random problems, where we vary the mean values μ of the non-local constraints, sampled from different distributions; we set $k = 15$ for all cases. The same trends observed above apply here. However, the normalized error increases as the range of the mean increases for all heuristics. The reason is because the magnitude in the error (when variables are assigned wrong value due to wrong guesses in the cost of the constraints) increases when the range increases. However, generally, the optimistic and pessimistic variants of the minimax regret heuristics still perform better in all three cases.

7.2   Smart Home Device Scheduling (SHDS) Problems

SHDS Problem Construction: We now describe how we construct SHDS problems. As the only uncertain element in the uncertain constraints are the discomfort values $d_{z_j,u}^t$ (defined in Sect. 3) for devices z_j, time steps t, and users u, we model these values as random variables following a Normal distribution (e.g., one could fit a Normal distribution to the historical data). As the distribution for one user may be different from the distribution for a different user in a home, for each user u, we generate a discomfort table composed of a Normal distribution $\mathcal{N}(\mu_{z_j,u}^t, (\sigma_{z_j,u}^t)^2)$ for each device z_j and time step t. Each user u can gain the exclusive access to a device z_j with the probability $Pr_{z_j}^t$, and the Normal distribution of the discomfort of device z_j at time step t is the Normal distribution of the user that has exclusive access for that device and time step.

Next, let $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$ denote the DCOP whose constraints $\mathcal{F}$ have *accurate* cost tables that depend only on external parameters and are easily obtained (e.g., price function θ and energy consumption of devices ρ_{z_j}) or they depend on user preferences that are accurately obtained through an oracle. Using the same process described above, we combine the discomfort tables for multiple users into a single aggregated discomfort table $\mathcal{U}$. Note that this aggregated discomfort table may be different than the one $\hat{\mathcal{U}}$ for the uncertain DCOP if there are multiple users. Then, the actual discomfort value $d_{z_j}^t$ for each device z_j and time step t is sampled from the Normal distribution $\mathcal{N}(\mu_{z_j}^t, (\sigma_{z_j}^t)^2)$ for that device and time step in the aggregated discomfort table $\mathcal{U}$. We refer to this problem as the *oracle DCOP*. In summary, when a constraint is elicited, the actual discomfort values are retrieved from the oracle DCOP.

Experimental Setup: In our experiments, we consider $|\mathcal{H}| = 10$ homes, each controlling $|\mathcal{Z}_i| = 10$ smart devices, listed in Table 1 along with their energy consumption. We populate the set of smart devices $\mathcal{Z}_i$ of each home by randomly sampling 10 elements from $\mathcal{Z}$. Thus, a home might control multiple devices of the same type. We set a time horizon $H = 6$ with increments of 4 h. We use the same real-time pricing schema as proposed by Fioretto *et al.* [13], which is the one used by the Pacific Gas and Electric Company for their Californian consumers during peak summer months.[2]

To generate the discomfort table for each user, we assume that there is a weak correlation between the price of energy and the level of discomfort of the user. Specifically, we assume that users will prefer (i.e., they are more comfortable) using their devices when prices are low to save money. Therefore, the higher the price, the more uncomfortable the user will be at using the device at that time. Based on this assumption, for each home, user, and device, the mean μ^t at each time step t is an integer that is uniformly sampled from the range $[\max\{1, \theta(t) - 50\}, \theta(t) + 50]$, where $\theta(t)$ is the real-time pricing at time step t used by the Pacific Gas and Electric Company. Therefore, the range of the means differ across time steps but are the same for all devices as the discomfort level is primarily motivated by the pricing schema.

The weights α_c and α_u of the objective function defined in Eq. 5 are both set to 0.5. These settings are employed to create both an oracle DCOP and the corresponding uncertain DCOP, except that the values of the constraints of the uncertain DCOPs are not realized (i.e., they are distributions).

Finally, since all uncertain constraints in an SHDS problem are unary constraints, all three variants of the minimax regret heuristics are identical, and we use "MR" to label this heuristic.

Single User Experiments: In the first set of experiments, we set each home to have only one user. Figure 4(a) plots the error for our heuristics compared against the random baseline heuristic. The results are averaged over 100 randomly generated SHDS problem instances. We make the following observations:

- As expected, for all elicitation heuristics, the error decreases as the number of cost tables to elicit increases.
- Both the MR and MS heuristics consistently outperform the random heuristic for all values of k. Like the results in random graphs, the random heuristic has a higher likelihood of choosing the wrong constraint to elicit, while MR and MS choose better constraints.
- Interestingly, we observe that MS selects the constraints slightly better than MR, indicating that despite the fact that MS is a simpler heuristic, it is well-suited in problems with single users. The reason is that the key feature of MR—maximizing the regret over all users—is ignored when there is only one user.

---

[2] https://www.pge.com/en_US/business/rate-plans/rate-plans/peak-day-pricing/peak-day-pricing.page. Retrieved in November 2016.
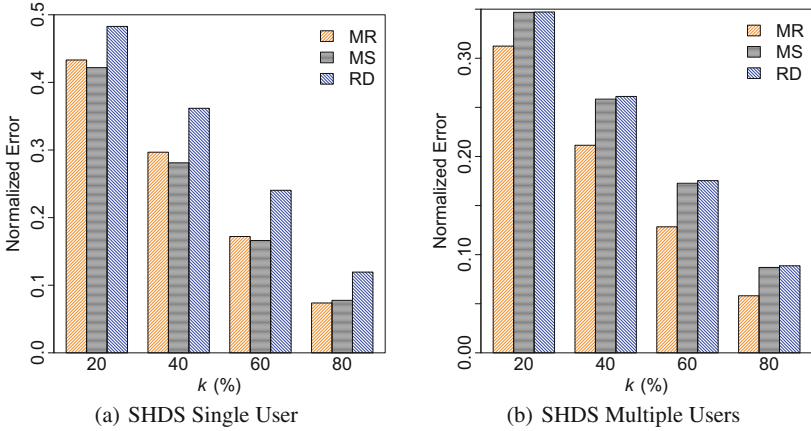
Fig. 4. Smart homes device scheduling preference elicitation

Multiple User Experiments: In the second set of experiments, we set each home to have two users, where both users have equal likelihood of controlling the devices (i.e., $P_{u_1} = P_{u_2} = 0.5$). Figure 4(b) shows the results. The trends for this experiment is similar to that shown for Fig. 4(a), where our MR heuristic outperforms the random heuristic. However, the MS heuristic performs poorly in this experiment, with similar performance as the random baseline. In general, our results show that the regret-based method outperforms other heuristics in multiple users scenarios, as it takes into account the discomfort values of all users, orders the constraints to elicit based on their minimum regrets. Similar to random graph results, MR performs better in the scenarios that multiple users take control of the devices in a building.

Finally, the SHDS and random graph experiment results demonstrate that the regret-based elicitation heuristics achieve approximately 30% and 11% improvement over the baseline random heuristic in minimizing the error, respectively. The improvements in SHDS problems are larger than those in random graphs because variables are highly connected ($p_1 = 0.8$) in random graph problems. In contrast, variables in SHDS problems are mostly independent as they mostly have unary constraints. The higher dependency between variables in random graphs reduces the improvements of our heuristics over the baseline random heuristic.

8   Conclusions and Future Work

DCOPs have been used to model a number of multi-agent coordination problems including the smart home device scheduling (SHDS) problem. However, one of the key assumptions in DCOPs—that constraint costs are known a priori—do not apply to many applications including SHDS. Thus, in this paper, we propose the problem of preference (i.e., constraint cost) elicitation for DCOPs; introduce

minimax regret based heuristics to elicit the preferences; and evaluate them on random binary DCOPs as well as SHDS problems. Our results show that our methods are better than a baseline method that elicits preferences randomly. This paper thus makes the foundational contributions that are necessary in the deployment of DCOP algorithms on practical applications, where preferences or constraint costs must be elicited or estimated. Future work includes incorporating real-world datasets [2,19,32] to generate the uncertain constraint costs as well as conducting comprehensive experiments in the many other applications that DCOPs have been used (e.g., meeting scheduling problems), where preferences are typically unknown and must be elicited too.

References

1. Abdennadher, S., Schlenker, H.: Nurse scheduling using constraint logic programming. In: Proceedings of the Conference on Innovative Applications of Artificial Intelligence (IAAI), pp. 838–843 (1999)
2. Anderson, B., Lin, S., Newing, A., Bahaj, A., James, P.: Electricity consumption and household characteristics: implications for census-taking in a smart metered future. Comput. Environ. Urban Syst. **63**, 58–67 (2017)
3. Bacchus, F., Grove, A.J.: Utility independence in a qualitative decision theory. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 542–552 (1996)
4. Boutilier, C.: A POMDP formulation of preference elicitation problems. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 239–246 (2002)
5. Boutilier, C., Patrascu, R., Poupart, P., Schuurmans, D.: Regret-based utility elicitation in constraint-based decision problems. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 929–934 (2005)
6. Braziunas, D.: Computational approaches to preference elicitation. Technical report (2006)
7. Braziunas, D., Boutilier, C.: Local utility elicitation in GAI models. In: Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI), pp. 42–49 (2005)
8. Erdös, P., Rényi, A.: On random graphs, I. Publ. Math. (Debr.) **6**, 290–297 (1959)
9. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.: Decentralised coordination of low-power embedded devices using the Max-Sum algorithm. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 639–646 (2008)
10. Fioretto, F., Pontelli, E., Yeoh, W.: Distributed constraint optimization problems and applications: a survey. CoRR, abs/1602.06347 (2016)
11. Fioretto, F. Yeoh, W., Pontelli, E.: A dynamic programming-based MCMC framework for solving DCOPs with GPUs. In: Proceedings of Principles and Practice of Constraint Programming (CP), pp. 813–831 (2016)
12. Fioretto, F., Yeoh, W., Pontelli, E.: Multi-variable agent decomposition for DCOPs. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2016)
13. Fioretto, F., Yeoh, W., Pontelli, E.: A multiagent system approach to scheduling devices in smart homes. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 981–989 (2017)

14. Fioretto, F., Yeoh, W., Pontelli, E., Ma, Y., Ranade, S.: A DCOP approach to the economic dispatch with demand response. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 981–989 (2017)
15. Gelain, M., Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Elicitation strategies for fuzzy constraint problems with missing preferences: algorithms and experimental studies. In: Stuckey, P.J. (ed.) CP 2008. LNCS, vol. 5202, pp. 402–417. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85958-1_27
16. Goldsmith, J., Junker, U.: Preference handling for artificial intelligence. AI Mag. **29**(4), 9–12 (2008)
17. Hatano, D., Hirayama, K.: DeQED: an efficient divide-and-coordinate algorithm for DCOP. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 566–572 (2013)
18. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artif. Intell. **101**(1–2), 99–134 (1998)
19. Kolter, J.Z., Johnson, M.J.: REDD: a public data set for energy disaggregation research. In: Proceedings of the Workshop on Data Mining Applications in Sustainability, pp. 59–62 (2011)
20. Larrosa, J.: Node and arc consistency in weighted CSP. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 48–53 (2002)
21. Le, T., Fioretto, F., Yeoh, W., Son, T.C., Pontelli, E.: ER-DCOPs: a framework for distributed constraint optimization with uncertainty in constraint utilities. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2016)
22. Le, T., Son, T.C., Pontelli, E., Yeoh, W.: Solving distributed constraint optimization problems with logic programming. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2015)
23. Maheswaran, R., Pearce, J., Tambe, M.: Distributed algorithms for DCOP: a graphical game-based approach. In: Proceedings of the International Conference on Parallel and Distributed Computing Systems (PDCS), pp. 432–439 (2004)
24. Miller, S., Ramchurn, S., Rogers, A.: Optimal decentralised dispatch of embedded generation in the smart grid. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 281–288 (2012)
25. Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I.: Internet of things: vision, applications and research challenges. Ad Hoc Netw. **10**(7), 1497–1516 (2012)
26. Modi, P.: Distributed constraint optimization for multiagent systems. Ph.D. thesis, University of Southern California, Los Angeles (United States) (2003)
27. Modi, P., Shen, W.-M., Tambe, M., Yokoo, M.: ADOPT: asynchronous distributed constraint optimization with quality guarantees. Artif. Intell. **161**(1–2), 149–180 (2005)
28. Netzer, A., Grubshtein, A., Meisels, A.: Concurrent forward bounding for distributed constraint optimization problems. Artif. Intell. **193**, 186–216 (2012)
29. Nguyen, D.T., Yeoh, W., Lau, H.C., Gibbs, D.: A memory-bounded sampling-based DCOP algorithm. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 167–174 (2013)
30. Nguyen, D.T., Yeoh, W., Lau, H.C., Zilberstein, S., Zhang, C.: Decentralized multi-agent reinforcement learning in average-reward dynamic DCOPs. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 1447–1455 (2014)
31. Ottens, B., Dimitrakakis, C., Faltings, B.: DUCT: an upper confidence bound approach to distributed constraint optimization problems. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 528–534 (2012)

32. Paatero, J.V., Lund, P.D.: A model for generating household electricity load profiles. Int. J. Energy Res. **30**(5), 273–290 (2006)
33. Petcu, A., Faltings, B.: A scalable method for multiagent constraint optimization. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1413–1420 (2005)
34. Rodrigues, L., Magatao, L.: Enhancing supply chain decisions using constraint programming: a case study. In: Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI), pp. 1110–1121 (2007)
35. Rossi, F., Venable, K.B., Walsh, T.: Preferences in constraint satisfaction and optimization. AI Mag. **29**(4), 58–68 (2008)
36. Shapiro, L.G., Haralick, R.M.: Structural descriptions and inexact matching. IEEE Trans. Pattern Anal. Mach. Intell. **5**, 504–519 (1981)
37. Stranders, R., Delle Fave, F., Rogers, A., Jennings, N.: DCOPs and bandits: exploration and exploitation in decentralised coordination. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 289–297 (2012)
38. Stuckey, P.J., Becket, R., Brand, S., Brown, M., Feydy, T., Fischer, J., de la Banda, M.G., Marriott, K., Wallace, M.: The evolving world of MiniZinc. In: Constraint Modelling and Reformulation, pp. 156–170 (2007)
39. Taylor, M., Jain, M., Tandon, P., Yokoo, M., Tambe, M.: Distributed on-line multiagent optimization under uncertainty: balancing exploration and exploitation. Adv. Complex Syst. **14**(03), 471–528 (2011)
40. Ueda, S., Iwasaki, A., Yokoo, M.: Coalition structure generation based on distributed constraint optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 197–203 (2010)
41. Vinyals, M., Rodríguez-Aguilar, J., Cerquides, J.: Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law. J. Auton. Agents Multi-Agent Syst. **22**(3), 439–464 (2011)
42. Wang, T., Boutilier, C.: Incremental utility elicitation with the minimax regret decision criterion. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 309–318 (2003)
43. Wu, F., Jennings, N.: Regret-based multi-agent coordination with uncertain task rewards. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 1492–1499 (2014)
44. Yeoh, W., Felner, A., Koenig, S.: BnB-ADOPT: an asynchronous branch-and-bound DCOP algorithm. J. Artif. Intell. Res. **38**, 85–133 (2010)
45. Yeoh, W., Yokoo, M.: Distributed problem solving. AI Mag. **33**(3), 53–65 (2012)
46. Zivan, R., Okamoto, S., Peled, H.: Explorative anytime local search for distributed constraint optimization. Artif. Intell. **212**, 1–26 (2014)
47. Zivan, R., Yedidsion, H., Okamoto, S., Glinton, R., Sycara, K.: Distributed constraint optimization for teams of mobile sensing agents. J. Auton. Agents Multi-Agent Syst. **29**(3), 495–536 (2015)