

Multi-Context System for Optimization Problems

Tiep Le, Tran Cao Son, Enrico Pontelli

Department of Computer Science
New Mexico State University
{tle, tson, epontell}@cs.nmsu.edu

Abstract

This paper proposes *Multi-context System for Optimization Problems (MCS-OP)* by introducing conditional cost-assignment bridge rules to Multi-context Systems (MCS). This novel feature facilitates the definition of a preorder among equilibria, based on the total incurred cost of applied bridge rules. As an application of MCS-OP, the paper describes how MCS-OP can be used in modeling Distributed Constraint Optimization Problems (DCOP), a prominent class of distributed optimization problems that is frequently employed in multi-agent system (MAS) research. The paper shows, by means of an example, that MCS-OP is more expressive than DCOP, and hence, could potentially be useful in modeling distributed optimization problems which cannot be easily dealt with using DCOPs. It also contains a complexity analysis of MCS-OP.

Introduction

The paradigm of *Multi-context Systems (MCS)* has been introduced in (McCarthy 1987; Roelofsen and Serafini 2005; Brewka and Eiter 2007) as a framework for integration of knowledge from different sources. Intuitively, an MCS (Brewka and Eiter 2007) consists of several theories, referred to as *contexts*, e.g., representing different reasoning components in distributed settings or different *agents* in *multi-agent systems (MAS)*. The contexts may be *heterogeneous*, in the sense that each context could use a different logical language for its knowledge base (e.g., deductive databases, triple-stores, description logic, logic programming (Gelfond and Lifschitz 1991)) and thus may use a different inference system. The flow of information between contexts are modeled via *bridge rules* in a declarative way, where bridge rules describe how the beliefs of one context depend on the beliefs of other contexts. The semantics of MCS is defined in terms of *equilibria* (Brewka and Eiter 2007).

An equilibrium of an MCS can be seen as a state where each context holds a set of its beliefs, and those sets of beliefs of contexts must *together satisfy* the conditions specified in the bridge rules. Therefore, MCSs are very suitable for modeling *distributed satisfaction problems*. However, it

does not provide a means for modeling *distributed optimization problems* where distributed components (or agents) coordinate with each other to achieve a most preferred (best) solution. This is because an MCS may have many equilibria but there is no means to express the preference among them as well as obtain a most preferred equilibrium.

In a different line of research, within the MAS community, *Distributed Constraint Optimization Problems (DCOPs)* (Modi et al. 2005; Petcu and Faltings 2005; Mailler and Lesser 2004; Gershman, Meisels, and Zivan 2009; Yeoh and Yokoo 2012) have emerged as a prominent agent model to govern the agents' autonomous behavior in solving distributed optimization problems. A DCOP is typically specified by a finite set of variables and a finite set of constraints among these variables. Each constraint indicates a utility for each value assignment of the variables involved in it. Agents in DCOPs need to coordinate value assignments of their variables, in a decentralized and distributed manner, to optimize their *objective functions*. Researchers have used DCOPs to model various multi-agent coordination and resource allocation problems (Maheswaran et al. 2004; Zivan, Grinton, and Sycara 2009; Zivan, Okamoto, and Peled 2014; Lass et al. 2008; Kumar, Faltings, and Petcu 2009; Ueda, Iwasaki, and Yokoo 2010; Léauté and Faltings 2011).

Although both the MCS and DCOP research communities have the similar goal of developing a general framework for modeling multi-agent and distributed systems, it is interesting to observe that there is little connection between the two communities. Obviously, the two frameworks cannot be more different: DCOP is homogeneous and MCS is heterogeneous. DCOP emphasizes optimization and MCS satisfaction. There have been several systems developed for computing solutions of DCOP whilst only a few experimental systems for computing equilibria of MCSs are available. *Can we develop a framework that exploits the advantages of both MCS and DCOP?*

In this paper, we make the first steps to bridge the two different research directions, MCS and DCOP, exploiting their advantages in modeling distributed optimization problems. We introduce a novel, more general form of MCS, called *MCS for Optimization Problems (MCS-OP)*, where each bridge rule can be associated to a *conditional cost* via the so-called *cost-assignment bridge rules*. This feature al-

lows MCS-OP to derive preferences among its equilibria using their total incurred cost.

This paper contributes to both areas of MCS and DCOP by formally showing how a DCOP can be modeled using an MCS-OP. For the MCS community, this demonstrates that MCS-OP can be used to represent a large and interesting class of optimization problems that DCOP can, something beyond the immediate capabilities of the original MCS model, and hence, a useful and much-needed extension of MCS to model distributed optimization problems. For the DCOP community, MCS-OP can be viewed as an extension of DCOP allowing heterogeneous agents to work with each other.

The paper is organized as follows: after recalling the background on MCS, we introduce the MCS-OP framework, study its computational properties. Next we illustrate how to use MCS-OP to model DCOP and show that MCS-OP is more expressive than DCOP. This is followed by a discussion on the related work. The paper ends with its conclusion and future works¹.

Background: Multi-context System

A logic L is a tuple (KB_L, BS_L, ACC_L) where KB_L is the set of well-formed knowledge bases of L —each being a set of formulae. BS_L is the set of possible belief sets; each element of BS_L is a set of syntactic elements representing the beliefs L may adopt. $ACC_L : KB_L \rightarrow 2^{BS_L}$ describes the “semantics” of L by assigning to each element of KB_L a set of acceptable sets of beliefs.

An MCS $M = (C_1, \dots, C_n)$ consists of contexts $C_i = (L_i, kb_i, br_i)$, ($1 \leq i \leq n$), where $L_i = (KB_i, BS_i, ACC_i)$ is a logic, $kb_i \in KB_i$ is a specific knowledge base of L_i , and br_i is a set of L_i -bridge rules of the form:

$$(r_k^i) s \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \text{not } (c_{j+1} : p_{j+1}), \dots, \text{not } (c_m : p_m) \quad (1)$$

where, r_k^i is the name of the bridge rule (e.g., the k -th bridge rule in the context C_i) and for each $1 \leq t \leq m$, we have that: $1 \leq c_t \leq n$, p_t is an element of some belief set of L_{c_t} , and $kb_i \cup \{s\} \in KB_i$. Intuitively, a bridge rule r allows us to add s to a context, depending on the beliefs in the other contexts. We denote with $head(r)$ the part s of r , and denote with $body(r)$ the right-hand part of the arrow in r . The semantics of MCS is described by the notion of belief states. Let $M = (C_1, \dots, C_n)$ be an MCS. A belief state is a tuple $S = (S_1, \dots, S_n)$ where each S_i is an element of BS_i . Let Br_i be the set of the names of all bridge rules in br_i .

Given a belief state $S = (S_1, \dots, S_n)$ and a bridge rule r of the form (1), we say that r is *applicable* in S if $p_v \in S_{c_v}$ for each $1 \leq v \leq j$ and $p_k \notin S_{c_k}$ for each $j+1 \leq k \leq m$. By $app(B, S)$ we denote the set of the bridge rules $r \in B$ that are applicable in S .

¹Due to space constraint, most proofs of lemmata and theorems are presented in the supplemental file at the URL <https://www.cs.nmsu.edu/~tile/AAAI19/supplementary.pdf>

A belief state $S = (S_1, \dots, S_n)$ of M is an *equilibrium* if, for all $1 \leq i \leq n$, we have that $S_i \in ACC_i(kb_i \cup \{head(r) \mid r \in app(br_i, S)\})$.

Example 1. Consider a scenario in which two people want to travel together. The first person C_1 needs to decide whether they purchase the business (b) or economic (e) class for their flight whilst the second person C_2 chooses whether they will stay in Hilton (h) or Inn (i) hotel. Note that the knowledge bases of C_1 and C_2 are given under Answer Set Programming (ASP) (Gelfond and Lifschitz 1991) and Propositional Logic under the closed world assumption (PL), respectively. There are constraints among such selections in order to have an affordable travel together. If C_2 chooses to stay in a fancy hotel like h (resp. affordable hotel like i), C_1 would choose a cheaper flight like e (resp. more comfortable flight like b). Similarly, if C_1 picks b (resp. e) class for their flights, C_2 would select i (resp. h) to stay.

This scenario can be modeled as an MCS $M_{fh} = (C_1, C_2)$ where L_1 and L_2 are ASP and PL, respectively². The knowledge bases and the bridge rules are given as follows.

$$kb_1 = \{1\{b; e\}1 \leftarrow\} \quad kb_2 = \{(h \wedge \neg i) \vee (\neg h \wedge i)\}$$

$$br_1 = \begin{cases} (r_1^1) e \leftarrow (2 : h) \\ (r_2^1) b \leftarrow (2 : i) \end{cases} \quad br_2 = \begin{cases} (r_1^2) h \leftarrow (1 : e) \\ (r_2^2) i \leftarrow (1 : b) \end{cases}$$

It is possible to see that M_{fh} has two equilibria:

$$S = (\{e\}, \{h\}). \quad S' = (\{b\}, \{i\}).$$

S (resp. S') corresponds to the group decision in which C_1 selects economic class (e) and C_2 selects Hilton (h) (resp. C_1 selects business class (b) and C_2 selects Inn (i)) for their flight and hotel respectively.

MCS-OP: Multi-context System for Optimization Problems

Motivation

Roughly speaking, there is no qualitative distinction among the equilibria of an MCS. As such, MCSs are ideal for modeling (distributed) satisfaction problems; e.g., the scenario in Example 1 where a decision for C_1 and C_2 to travel together need to satisfy the constraints among them. In many scenarios, some selection among the equilibria might be preferred. Let us consider the following example.

Example 2. Let us continue with Example 1. Assume that business class and economic class tickets cost \$100 and \$50 respectively. Further, staying in Hilton or Inn hotels cost \$120 or \$60 respectively. The most preferred group decision is to select the flight and the hotel in such a way that minimizes the total cost.

Intuitively, the preference of C_1 and C_2 indicates that equilibrium S' is more preferred than S since the total cost associated with S' (\$160) is smaller than the total cost associated with S (\$170).

²The logics of ASP and PL were given in (Brewka and Eiter 2007).

As we have mentioned earlier, MCSs were not focused on providing a mechanism for preferring an equilibrium over another. For this reason, to utilize MCS in solving the problem in Example 2, certain modifications to M_{fh} needs to be done. For example, one might have to add to M_{fh} (i) rules to encode the total cost of a selection to some contexts; and (ii) rules to eliminate the less desirable equilibrium. We next introduce *MCS for Optimization Problems (MCS-OP)*, an extension of MCS, that allows for a principled and efficient way to model distributed optimization problems.

Syntax

There are many possible ways to take into consideration preferences among equilibria. Previous work in this direction such as the framework on Multi-Context System with Preferences (MCSP) in (Le, Son, and Pontelli 2018) focuses on creating an order among equilibria from individual preferences which are expressed in each context. In this paper, we focus on establishing a notion of the aggregate preference of an equilibrium and using this notion in defining an order among equilibria. Inspired by the formulation of Distributed Constraint Optimization Problem (see below), where, intuitively, constraints among agents that could be considered as bridges between contexts are associated with costs/utilities, we consider that the aggregate preference of an equilibrium depends on the bridge rules. For this reason, we introduce an extra component to MCS, the cost-assignment bridge rules. Formally, a *cost-assignment bridge rule* is defined as follows.

Definition 1. Let $M = (C_1, \dots, C_n)$ be a multi-context system where $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$. A C_k -cost-assignment bridge rule over M , $1 \leq k \leq n$, is of the form

$$r @ x \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \\ \text{not } (c_{j+1} : p_{j+1}), \dots, \text{not } (c_m : p_m) \quad (2)$$

where:

- $r \in Br_k$ is the name of a bridge rule in br_k ;
- $x \in \mathbb{R} \cup \{+\infty\}$ that specifies the cost of the bridge rule r ;
- for each $1 \leq k \leq m$, we have that: $1 \leq c_k \leq n$, p_k is an element of some belief set of L_{c_k} .

Let cr be a cost-assignment bridge rule of the form (2). We denote with the *head*(cr) (resp. *body*(cr)) of cr , the left-hand (resp. right-hand) part of the arrow in cr . Intuitively, (2) states that if its right hand side is satisfied then the context C_k will incur a cost x .

Definition 2. A Multi-Context System for Optimization Problems (*MCS-OP*) $M = (C_1, \dots, C_n)$ consists of a collection of cost-assignment contexts $C_i = (L_i, kb_i, br_i, cbr_i)$, where $M' = (C'_1, \dots, C'_n)$ with $C'_i = (L_i, kb_i, br_i)$ is a MCS and cbr_i is a finite set of C_i -cost-assignment bridge rules over M' .

For simplicity, given an MCS-OP $M = (C_1, \dots, C_n)$ where $C_i = (L_i, kb_i, br_i, cbr_i)$, we denote with $MCS(M)$ the multi-context system $MCS(M) = (C'_1, \dots, C'_n)$ in which $C'_i = (L_i, kb_i, br_i)$.

Example 3. To model the problem in Example 2, we extend the MCS M_{fh} described

in Example 1 into an MCS-OP M where: $MCS(M) = M_{fh}$; and $cbr_1 = \{r_1^1 @ 50 \leftarrow; r_2^1 @ 100 \leftarrow\}$, $cbr_2 = \{r_1^2 @ 120 \leftarrow; r_2^2 @ 60 \leftarrow\}$.

The proposal to use cost-assignment bridge rules deserves some discussion. One can observe that it is possible to have bridge rules deriving costs straightforwardly, we decide to use cost-assignment bridge rules mainly because of the following reason. We want to extend original MCS in a modular way, where bridge rules are dedicated to model the flow of information between contexts while cost-assignment bridge rules are dedicated exclusively to assign costs to bridge rules. One benefit of such a modular extension is that one can apply *preference elicitation* techniques to elicit the costs of bridge rules in future work.

Most Preferred Equilibria

For an MCS-OP M , a belief state of M is defined as a belief state of $MCS(M)$. Given a belief state S of M , a bridge rule r is applicable in S with respect to M if r is applicable in S with respect to $MCS(M)$. A belief state S is an equilibrium of an MCS-OP M iff S is an equilibrium of $MCS(M)$.

As our goal in proposing cost-assignment bridge rules is to select “best” equilibria, we establish a *preference order* among belief states of an MCS-OP. Clearly, this order has to be based on the set of cost-assignment bridge rules. We first define the *applicability* of a cost-assignment bridge rule.

Definition 3. A cost-assignment bridge rule of the form (2) is fireable in a belief state $S = (S_1, \dots, S_n)$ if

- r is applicable in S ; and
- its body is satisfied in S , i.e., $p_v \in S_{c_v}$ for each $1 \leq v \leq j$ and $p_k \notin S_{c_k}$ for each $j+1 \leq k \leq m$.

Given a belief state $S = (S_1, \dots, S_n)$ and a cost-assignment bridge rule cr of the form (2), we define the *cost impact* of cr in S , denoted with $c_S(cr)$, as follows:

- $c_S(cr) = 0$ if cr is not fireable in S , that is, either r is not applicable or $p_k \notin S_{c_k}$ for some $k \in \{1, \dots, j\}$, or some $p_k \in S_{c_k}$ for $k \in \{j+1, \dots, m\}$; otherwise,
- $c_S(cr) = x$ if cr is fireable in S .

Intuitively, cr has some impact on the belief state S if it is fireable in S . As such, for case (i), since cr is not fireable, the cost impact of cr is 0. For case (ii), cr is fireable, its body is satisfied and the bridge rule r is actually applicable in S . As such, the indicated cost of x is incurred, for the application of r . Then, the cost impact of cr is x .

Definition 4. Let $C_i = (L_i, kb_i, br_i, cbr_i)$ be a context of an MCS-OP M . The *cost impact of the context C_i in a belief state S of M* , denoted by c_S^i , is defined as

$$c_S^i = \sum_{cr \in cbr_i} c_S(cr)$$

The cost impact of an MCS-OP in a belief state S is the summation of the cost impacts of all contexts in S .

Definition 5. Let $M = (C_1, \dots, C_n)$ be an MCS-OP and S be a belief state of M . The *cost impact of M in S* , denoted by c_S^M , is defined as $c_S^M = \sum_{C_i \in M} c_S^i$.

When no confusion is possible, we omit the superscript M in c_S^M . It is worth to notice that given an MCS-OP M , a belief state S , and a cost-assignment bridge rule cr , by the definition of the cost impact of cr in S , $c_S(cr)$ is always determined. As such, the following proposition is derived immediately given Definitions 4 and 5.

Proposition 1. *Given an MCS-OP M , c_S^M is well-defined, i.e., it is determined for every belief state S of M .*

The cost impact can be used to define a preference order among belief states as follows.

Definition 6. *Let S_1 and S_2 be two belief states of an MCS-OP M . We write $S_1 \succeq S_2$ if $c_{S_1} \leq c_{S_2}$, $S_1 \succ S_2$ if $c_{S_1} < c_{S_2}$, and $S_1 \approx S_2$ if $c_{S_1} = c_{S_2}$.*

\succeq is referred to as the preference order among belief states.

Proposition 2. *The ordering \succeq is a total preorder, i.e., \succeq is total, reflexive, and transitive.*

Since \succeq is a preorder, we can use this ordering to define an order among the equilibria of an MCS-OP as follows.

Definition 7. *A belief state S of an MCS-OP M is a most preferred equilibrium of M if*

- S is an equilibrium of M , and
- There does not exist another equilibrium S' of M where $S' \neq S$ such that $S' \succ S$.

Let E_M be the set of all equilibria of an MCS-OP M . In other words, a most preferred equilibrium S^* of M can be defined mathematically as $S^* = \arg \min_{S \in E_M} c_S$. Note that, in this paper, we consider *cost* instead of *utility*, i.e., whenever a bridge rule is applied, a *cost* is incurred instead of a *utility* is produced. As such, a most preferred equilibrium minimizes the total incurred cost. The framework can thus easily be adapted to consider utility instead of cost, i.e., defining $S_1 \succeq S_2$ if $c_{S_1} \geq c_{S_2}$ and maximizing utility by using $S^* = \arg \max_{S \in E_M} c_S$.

Example 4. *The MCS-OP M in Example 3 has two equilibria S and S' that are the two equilibria of the MCS M_{fh} in Example 1 because $MCS(M) = M_{fh}$. In addition, $c_S = c_S^1 + c_S^2 = 50 + 120 = 170$. Similarly, $c_{S'} = c_{S'}^1 + c_{S'}^2 = 100 + 60 = 160$. $S' \succ S$ since $c_{S'} < c_S$, and thus, S' is the most preferred equilibrium of M .*

Complexity Analysis

We present some results on complexity analysis of MCS-OP M in this section. We consider here two problems:

- $CONS(M)$: Decide whether M has a most preferred equilibrium;
- $MOST(M, S)$: Decide whether a belief state S is a most preferred equilibrium of M .

For this, we utilize the approach of *output-projected* belief states similar as in (Eiter et al. 2014; Brewka et al. 2011). Given an MCS-OP M , br_M (resp. cb_{r_M}) denotes the set of all bridge rules (resp. cost-assignment bridge rules) in M .

Definition 8. *Given an MCS-OP M and a context C_i of M , $OUT_i = \{p \mid p \text{ is a belief of } C_i, (i : p) \text{ or not } (i : p) \text{ occurs in the body of some bridge rule or cost-assignment}$*

| $CC(M)$ in | \mathbf{P} | $\Sigma_i^{\mathbf{P}}$ | \mathbf{PSPACE} | $\mathbf{EXPTIME}$ |
|--------------|-----------------------------|-----------------------------|-------------------|--------------------|
| $CONS(M)$ | \mathbf{NP} | $\Sigma_i^{\mathbf{P}}$ | \mathbf{PSPACE} | $\mathbf{EXPTIME}$ |
| $MOST(M, S)$ | $\mathbf{D}_0^{\mathbf{P}}$ | $\mathbf{D}_i^{\mathbf{P}}$ | \mathbf{PSPACE} | $\mathbf{EXPTIME}$ |

Table 1: Complexity of problems $CONS(M)$ and $MOST(M, S)$ with $i \geq 1$.

bridge rule of some other context C_j in $M\}$ is called the set of output beliefs of C_i .

Furthermore, given a belief state $S = (S_1, \dots, S_n)$ of M , the output-projected belief state $S' = (S'_1, \dots, S'_n)$ is the projection of S to output beliefs of M : $S'_i = S_i \cap OUT_i$.

An output-projected belief state provides sufficient information to evaluate the applicability of bridge rules, and thus one can define witnesses for equilibria using this projection.

Definition 9. *An output-projected belief state $S' = (S'_1, \dots, S'_n)$ of an MCS-OP M is an output-projected equilibrium iff the following holds: for all $1 \leq i \leq n$, $S'_i \in ACC_i(kb_i \cup \{\text{head}(r) \mid r \in \text{app}(br_i, S')\}) \cap OUT_i$.*

An output-projected belief state S' contains information about all output beliefs, which is sufficient to determine whether a bridge rule is applicable and its cost impact as well. Thus, $\text{app}(br_i, S) = \text{app}(br_i, S')$, and $c_S = c_{S'}$. In addition, equilibria of an MCS-OP M are defined as equilibria of $MCS(M)$ which is identical to M excluding cost-assignment bridge rules. As a result, Lemma 1 below is obtained immediately from Lemma 1 in (Eiter et al. 2010).

Lemma 1. *For each equilibrium S of an MCS-OP M , its output-projected belief state S' is an output-projected equilibrium. Conversely, for each output-projected equilibrium S' of M , there exists at least one equilibrium T of M and its output-projected belief state T' such that $T' = S'$.*

Following (Eiter et al. 2010), we also let the *context complexity* of a context C_i be the complexity of the problem: given an output-projected belief state $S' = (S'_1, \dots, S'_n)$ of an MCS-OP M , decide whether there exists an $S_i \in ACC_i(kb_i \cup \{\text{head}(r) \mid r \in \text{app}(br_i, S')\})$ such that $S'_i = S_i \cap OUT_i$.

The context complexity of an MCS-OP M , denoted with $CC(M)$, is a (smallest) upper bound for the context complexity classes of all contexts C_i in M .

For complexity analysis, as in (Brewka et al. 2018), we only consider *finite* MCS-OP, i.e., we do not consider rule schemas (which stand potentially for an infinite amount of bridge rules or cost-assignment bridge rules), and assume that all knowledge bases in the given MCS-OP are finite. The complexity of MCS-OP are specified in Theorem 1.

Theorem 1. *Table 1 summarizes the complexities of membership of problems $CONS(M)$ and $MOST(M, S)$ depending on the context complexity. Here, the class $\mathbf{D}_0^{\mathbf{P}}$ contains decision problems which are the “conjunction” of a \mathbf{P} and an independent **co-NP** decision problem. Further, the class of $\mathbf{D}_i^{\mathbf{P}}$ with $i \geq 1$ contains decision problems which are the “conjunction” of a $\Sigma_i^{\mathbf{P}}$ and an independent $\Pi_i^{\mathbf{P}}$ decision problem (e.g., for $i = 1$, a SAT and an independent UNSAT instance).*

Proof. For $CONS(M)$: By Proposition 2, the problem $CONS(M)$ is equivalent with consistency checking (equilibrium existence) of M . As such, by the definition of equilibria of M , the complexity of $CONS(M)$ is obtained based on the complexity of consistency checking (equilibrium existence) of $MCS(M)$ whose result was investigated in (Brewka and Eiter 2007) and is given in Table 1.

For $MOST(M, S)$: Checking whether a belief state $S = (S_1, \dots, S_n)$ is a most preferred equilibrium includes: (T1) checking whether S is an equilibrium of M , and (T2) checking there is no other equilibrium T that is preferred to S (i.e., there is no T such that $c_T < c_S$).

- For (T1): this can be decided by (T1.1) evaluating the bridge rules on S , yielding for each context C_i a set H_i of applied bridge rule heads with respect to S —which is linear with the *finite* number of bridge rules in C_i ; and (T1.2) decides whether $S_i \in ACC_i(kb_i \cup H_i)$ —which is the context complexity of C_i . As such, (T1) can be decided in $CC(M)$.
- For (T2): this can be decided by:
 - (T2.1) guessing all output-projected belief states T' ;
 - (T2.2) evaluating the bridge rules on each $T' \in \mathcal{T}'$, yielding for each context C_i a set H_i of applied bridge rule heads with respect to T' —which is linear with the *finite* number of bridge rules in C_i ;
 - (T2.3) deciding, whether there exists an $T = (T_1, \dots, T_n)$ such that $T_i \in ACC_i(kb_i \cup H_i)$ and $T'_i = T_i \cap OUT_i$ —which is the context complexity of C_i .
 - (T2.4) checking whether $c_T = c_{T'} < c_S$. Note that if there are many T that satisfy (T2.3), then they will all incur identical costs, which equal to $c_{T'}$.
 - (T2.4) the instance is yes instance iff all such checks fail, leading to the class **co-NP** (resp. Π_1^P) if $CC(M) = \mathbf{P}$ (resp. $CC(M) = \Sigma_1^P$).

If $CC(M) = \mathbf{P}$, the complexity of $MOST(M, S)$ is the class \mathbf{D}_0^P that contains decision problems which are the “conjunction” of a \mathbf{P} (i.e., for (T1)) and an independent **co-NP** decision problem (i.e., for (T2)). Further, if $CC(M) = \Sigma_i^P$ with $i \geq 1$, the complexity of $MOST(M, S)$ is the class of \mathbf{D}_i^P that contains decision problems which are the “conjunction” of a Σ_i^P (i.e., for (T1)) and an independent Π_1^P decision problem (i.e., for (T2)). Similarly, if $CC(M) = \mathbf{PSPACE}$ (resp. **EXPTIME**), then $MOST(M, S)$ is the class of **PSPACE** (resp. **EXPTIME**). \square

Modeling Distributed Constraint Optimization

Distributed Constraint Optimization Problems (DCOPs) have emerged as a prominent agent model to govern the agents’ autonomous behavior, where agents coordinate their value assignments, in a decentralized and distributed manner, to optimize their objective functions. DCOPs can effectively model a wide range of problems in MAS setting (Fioretto, Pontelli, and Yeoh 2018). Next, we will review DCOP and show how it can be modeled by MCS-OP.

A *Distributed Constraint Optimization Problem* (DCOP) (Modi et al. 2005; Petcu and Faltings 2005;

| x_1 | x_2 | f_1 | x_2 | x_3 | f_2 | x_3 | x_1 | f_3 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 10 | 0 | 0 | 7 | 0 | 0 | 9 |
| 0 | 1 | 20 | 0 | 1 | 9 | 0 | 1 | 2 |
| 1 | 0 | 35 | 1 | 0 | 6 | 1 | 0 | 27 |
| 1 | 1 | 40 | 1 | 1 | 15 | 1 | 1 | 35 |

Figure 1: Three Constraints of DCOP in Example 5

Gershman, Meisels, and Zivan 2009; Yeoh and Yokoo 2012) is a tuple $\mathcal{D} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$ where

- $\mathcal{X} = \{x_1, \dots, x_m\}$ is a finite set of (decision) *variables*;
- $\mathcal{D} = \{D_1, \dots, D_m\}$ is a set of finite *domains*, where each D_i is the domain of variable $x_i \in \mathcal{X}$;
- $\mathcal{F} = \{f_1, \dots, f_p\}$ is a finite set of *constraints*, where each k_j -ary constraint $f_j : D_{j_1} \times D_{j_2} \times \dots \times D_{j_{k_j}} \mapsto \mathbb{R} \cup \{+\infty\}$ specifies the *cost* of each combination of values of the variables in its *scope*; the scope of f_j is denoted by $scp(f_j) = \{x_{j_1}, \dots, x_{j_{k_j}}\}$,³
- $\mathcal{A} = \{a_1, \dots, a_n\}$ is a finite set of *agents*;
- $\alpha : \mathcal{X} \mapsto \mathcal{A}$ maps each variable to an agent.

In the DCOP literature, the constraints f_j are also called *objective functions* or *cost functions*. We say that an agent a_i owns the variable x_j if $\alpha(x_j) = a_i$. Each constraint in \mathcal{F} can be either *hard* (i.e., some value combinations result in a utility of $+\infty$ and must be avoided), or *soft*, (i.e., all value combinations result in a finite cost and need not be avoided). Given a constraint f_j and a *complete value assignment* x (i.e., a value assignment for all variables), we denote with x_{f_j} a *partial value assignment* from x for all variables in $scp(f_j)$. For simplicity, given two value assignments x and x' , we write $x = x'$ whenever (1) if $(x_i = d_i) \in x$, then $(x_i = d_i) \in x'$, and (2) conversely, if $(x_i = d_i) \in x'$, then $(x_i = d_i) \in x$. A *solution* of a DCOP is a complete value assignment x , and its corresponding *total cost* is the evaluation of all cost functions on x . The goal of a DCOP is to find a cost-minimal solution $\mathbf{x}^* = \arg \min_x \sum_{j=1}^p f_j(x_{f_j})$.

In this paper, we adopt a *simplifying assumption* that each agent owns exactly one variable. This assumption is a common practice in the DCOP literature (Petcu and Faltings 2005; Gershman, Meisels, and Zivan 2009; Ottens, Dimitrakakis, and Faltings 2012). Thus, without loss of generality, we assume $\alpha(x_i) = a_i$. It is relatively straightforward to generalize the ideas presented below to model a general DCOP, which allows agents to own multiple variables.

Example 5. Considering a DCOP $\mathcal{D} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$ where $\mathcal{X} = \{x_1, x_2, x_3\}$, $\mathcal{D} = \{D_1, D_2, D_3\}$ in which $D_i = \{0, 1\}$ with $i \in \{1, 2, 3\}$, $\mathcal{A} = \{a_1, a_2, a_3\}$, and α is defined as $\alpha(x_i) = a_i$. Figure 1 exhibits the three constraints f_1, f_2 , and f_3 between x_1 and x_2 , between x_2 and x_3 , and between x_3 and x_1 , respectively.

It is possible to see that the solution of \mathcal{D} is the value assignment in which $x_1 = x_2 = x_3 = 1$ which yields the minimal total cost of $40 + 15 + 35 = 90$.

We will now show how a DCOP can be encoded using an MCS-OP. We first specify the space of all solutions of a

³For simplicity, we assume a given ordering of variables

DCOP $\mathcal{D} = \langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$ with n agents (i.e., $|\mathcal{A}| = n$) as equilibria of an MCS $M'_\mathcal{D} = (C_1, \dots, C_n)$ of n contexts. Each context C_i corresponds to the agent $a_i \in \mathcal{A}$. For simplicity, we specify $C_i = (L_i, kb_i, br_i)$, for $1 \leq i \leq n$, where:

- L_i is the logic of the *Answer Set Programming (ASP)* framework (Gelfond and Lifschitz 1991). We also use the extended syntax of logic programs such as choice atoms, aggregates, etc. that have been adapted and implemented in most ASP solvers.
- kb_i consists of a rule (assuming that $D_i = \{d_i^1, \dots, d_i^s\}$)

$$1\{value(x_i, d_i^1), \dots, value(x_i, d_i^s)\}1 \leftarrow \quad (3)$$

Intuitively, the rule of the form (3) enforces the selection of exactly one of the listed values (i.e., $value(x_i, d_i^\ell)$ with $1 \leq \ell \leq s$) to an answer set of kb_i . Furthermore, $value(x_i, d_i^\ell)$ in an answer set of kb_i means that the variable x_i is assigned the value $d_i^\ell \in D_i$.

- br_i consists of bridge rules of the form

$$\begin{aligned} (B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})}) \text{ cost}_{f_j}(V) \leftarrow & (j_1 : value(x_{j_1}, d_{j_1})), \\ & \dots, \\ & (j_{k_j} : value(x_{j_{k_j}}, d_{j_{k_j}})) \end{aligned} \quad (4)$$

for each constraint f_j (e.g., $scp(f_j) = \{x_{j_1}, \dots, x_{j_{k_j}}\}$) such that $\alpha(x_{j_1}) = a_i$, where for $1 \leq \ell \leq k_j$, $d_{j_\ell} \in D_{j_\ell}$ and $V = f_j(d_{j_1}, \dots, d_{j_{k_j}})$. Intuitively, V is the cost specified by the constraint f_j with respect to the respective value assignment of variables in $scp(f_j)$ that are given in the body of the bridge rule (i.e., $value(x_{j_\ell}, d_{j_\ell})$). Thus, $V \in \mathbb{R} \cup \{+\infty\}$. Note that br_i will contain $|D_{j_1}| \times |D_{j_2}| \times \dots \times |D_{j_{k_j}}|$ bridge rule(s) for every f_j such that $\alpha(x_{j_1}) = a_i$. In (4), $B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})}$ is the *unique* name of the respective bridge rule.

Given a solution x of a DCOP \mathcal{D} in which x_i is assigned value $d_i \in D_i$ (i.e., $x_i = d_i$) for $1 \leq i \leq n$, let $E(x) = (S_1, \dots, S_n)$ be a belief state where:

$$S_i = \{value(x_i, d_i)\} \cup \quad (5)$$

$$\{cost_{f_j}(V) \mid scp(f_j) = \{x_{j_1}, \dots, x_{j_{k_j}}\}, \alpha(x_{j_1}) = a_i,$$

$$V = f_j(x_{j_1} = d_{j_1}, \dots, x_{j_{k_j}} = d_{j_{k_j}})\} \quad (6)$$

Furthermore, given a belief state $S = (S_1, \dots, S_n)$ of $M'_\mathcal{D}$, we let $A(S)$ be a (partial) value assignment in which $x_i = d_i$ if $value(x_i, d_i) \in S_i$. The next theorem relates \mathcal{D} and $M'_\mathcal{D}$.

Theorem 2. *Let \mathcal{D} be a DCOP and $M'_\mathcal{D}$ be its corresponding MCS. We have:*

- *If S is an equilibrium of $M'_\mathcal{D}$, then $x = A(S)$ is a solution of \mathcal{D} ; and*
- *If x is a solution of \mathcal{D} , then $S = E(x)$ is an equilibrium of $M'_\mathcal{D}$.*

To be able to model the cost-minimal solution of DCOP, we extend the MCS $M'_\mathcal{D}$ into an MCS-OP $M_\mathcal{D}$ by adding a set of cost-assignment bridge rules (i.e., cbr_i) to contexts of $M'_\mathcal{D}$. Formally, given a DCOP \mathcal{D} let $M_\mathcal{D} = (C_1, \dots, C_n)$ be an MCS-OP with $C_i = (L_i, kb_i, br_i, cbr_i)$

where $MCS(M_\mathcal{D}) = M'_\mathcal{D}$ and cbr_i is:

$$\begin{aligned} cbr_i = \{ & B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})} @V \leftarrow \mid B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})} \in br_i, \\ & head(B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})}) \equiv cost_{f_j}(V)\} \quad (7) \end{aligned}$$

Intuitively, a cost-assignment bridge rule of the form (7) means that if the bridge rule $B_{f_j(d_{j_1}, \dots, d_{j_{k_j}})}$ is applicable and its head (e.g., $cost_{f_j}(V)$) is added to the respective knowledge base, then this bridge rule will incur a cost (e.g., a cost of V).

By Definition 5 and cost-assignment bridge rules of the form (7), it is simple to derive Lemma 2 below. Let us remind that, given a belief state S of $M_\mathcal{D}$, $c_S^{M_\mathcal{D}}$ (c_S , for short) is the cost impact of $M_\mathcal{D}$ in S .

Lemma 2. *Let S be an equilibrium of $M_\mathcal{D}$. We have*

$$c_S = \sum_{cost_{f_j}(V) \in S} V \quad (8)$$

We also can conclude the next lemma to relate c_S with the total cost on value assignment $x = A(S)$.

Lemma 3. *Let \mathcal{D} be a DCOP and $M_\mathcal{D}$ be its corresponding MCS-OP. If x is a solution of \mathcal{D} and $S = E(x)$ is an equilibrium of $M_\mathcal{D}$ (see the second item of Theorem 2), then*

$$c_S = \sum_{j=1}^p f_j(x_{f_j}) \quad (9)$$

The next theorem relates DCOP \mathcal{D} and MCS-OP $M_\mathcal{D}$.

Theorem 3. *Let \mathcal{D} be a DCOP and $M_\mathcal{D}$ be its corresponding MCS-OP. We have:*

- *If S is a most preferred equilibrium of $M_\mathcal{D}$, then $x = A(S)$ is a cost-minimal solution of \mathcal{D} ; and*
- *If x is a cost-minimal solution of \mathcal{D} , then $S = E(x)$ is a most preferred equilibrium of $M_\mathcal{D}$.*

To sum up, Theorem 3 means that one can use MCS-OP to model DCOPs, where the semantics of MCS-OP as most preferred equilibria are directly related to the semantics of DCOPs as cost-minimal solutions.

Example 6. *The DCOP \mathcal{D} in Example 5 can be modeled using a MCS-OP $M_\mathcal{D} = (C_1, C_2, C_3)$ where with $i \in \{1, 2, 3\}$ $C_i = (L_i, kb_i, br_i, cbr_i)$ in which (for short, in the following we use predicate “val” instead of “value”):*

$$kb_i = \{1\{val(x_i, 0), val(x_i, 1)\}1 \leftarrow \}$$

$$br_1 = \begin{cases} (br_1^1) \text{ cost}_{f_1}(10) \leftarrow (1 : val(x_1, 0)), (2 : val(x_2, 0)) \\ (br_2^1) \text{ cost}_{f_1}(20) \leftarrow (1 : val(x_1, 0)), (2 : val(x_2, 1)) \\ (br_3^1) \text{ cost}_{f_1}(35) \leftarrow (1 : val(x_1, 1)), (2 : val(x_2, 0)) \\ (br_4^1) \text{ cost}_{f_1}(40) \leftarrow (1 : val(x_1, 1)), (2 : val(x_2, 1)) \end{cases}$$

$$br_2 = \begin{cases} (br_1^2) \text{ cost}_{f_2}(7) \leftarrow (2 : val(x_2, 0)), (3 : val(x_3, 0)) \\ (br_2^2) \text{ cost}_{f_2}(9) \leftarrow (2 : val(x_2, 0)), (3 : val(x_3, 1)) \\ (br_3^2) \text{ cost}_{f_2}(6) \leftarrow (2 : val(x_2, 1)), (3 : val(x_3, 0)) \\ (br_4^2) \text{ cost}_{f_2}(15) \leftarrow (2 : val(x_2, 1)), (3 : val(x_3, 1)) \end{cases}$$

$$br_3 = \begin{cases} (br_1^3) \text{ cost}_{f_3}(9) \leftarrow (3 : val(x_3, 0)), (1 : val(x_1, 0)) \\ (br_2^3) \text{ cost}_{f_3}(2) \leftarrow (3 : val(x_3, 0)), (1 : val(x_1, 1)) \\ (br_3^3) \text{ cost}_{f_3}(27) \leftarrow (3 : val(x_3, 1)), (1 : val(x_1, 0)) \\ (br_4^3) \text{ cost}_{f_3}(35) \leftarrow (3 : val(x_3, 1)), (1 : val(x_1, 1)) \end{cases}$$

$$cbr_1 = \begin{cases} (br_1^1)@10 \leftarrow \\ (br_2^1)@20 \leftarrow \\ (br_3^1)@35 \leftarrow \\ (br_4^1)@40 \leftarrow \end{cases} \quad cbr_2 = \begin{cases} (br_1^2)@7 \leftarrow \\ (br_2^2)@9 \leftarrow \\ (br_3^2)@6 \leftarrow \\ (br_4^2)@15 \leftarrow \end{cases}$$

$$cbr_3 = \begin{cases} (br_1^3)@9 \leftarrow \\ (br_2^3)@2 \leftarrow \\ (br_3^3)@27 \leftarrow \\ (br_4^3)@35 \leftarrow \end{cases}$$

It is possible to see that $M_{\mathcal{D}}$ has the most preferred equilibria: $S = (\{val(x_1, 1)\}, \{val(x_2, 1)\}, \{val(x_3, 1)\})$

Expressivity: MCS-OP vs. DCOP

The previous section shows that MCS-OP can model DCOP. For a researcher in the DCOP community, it raises the question whether the use of MCS-OP would bring any advantages over the DCOP formalization or it is just yet another way to formalize DCOP. This section answers this question by presenting a problem that can be straightforwardly formalized using MCS-OP but not in DCOP.

Example 7. We extend Example 2 assuming that C_2 also decides whether they go to their hotel by shuttle (s) or taxi (t). If they stay in Hilton, using s (resp. t) costs them \$10 (resp. \$30), and if they stay in Inn, using s or t costs them \$40 or \$30 respectively. C_2 prefers the cheaper transportation because he will pay for this cost because it is not accounted to the total cost for C_1 and C_2 to travel together.

It is clear to see that since the cost of going to hotel is not accounted to the total cost, the most preferred group decision is the same as the one in Example 2 (i.e., select b and i) and thus taking a taxi to go to Inn because it is cheaper than taking a shuttle. The total cost remains the same as \$160.

Example 7 can be modeled using the MCS-OP M in Example 3, with the following modification: (i) L_2 is answer set optimization program (Brewka, Niemelä, and Truszczyński 2003), and (ii) kb_2 is extended with the two rules $s > t \leftarrow h$ and $t > s \leftarrow i$. Intuitively, the former rule states that if C_2 stays in h , he prefers s to t . The latter rule means that if he stays in i , he prefers t to s . It is possible to check that the most preferred equilibrium of this updated MCS-OP is $(\{b\}, \{i, t\})$ that corresponds to the most preferred group decision mentioned above (i.e., select b, i and t).

If we were to use DCOP to model Example 7, it is necessary to introduce three variables x_1 , x_2 , and x_3 to represent the flight, hotel, and taxi, respectively. Formally, a DCOP that may model Example 7 is $\mathcal{D} = \langle \{x_1, x_2, x_3\}, \{D_1, D_2, D_3\}, \{f_1, f_2, f_3, f_4\}, \{C_1, C_2\}, \alpha \rangle$ where $D_1 = \{b, e\}$, $D_2 = \{h, i\}$, $D_3 = \{s, t\}$, $\alpha(x_1) = C_1$, $\alpha(x_2) = C_2$, $\alpha(x_3) = C_2$, and the four constraints f_1, f_2, f_3 , and f_4 are given in Figure 2. Intuitively, f_1 and f_2 represent the cost relation for x_1 (by C_1) and x_2 (by C_2), respectively. While f_3 models the hard constraints between x_1 and x_2 (similar to br_1 and br_2), f_4 represents the cost relation between x_2 and x_3 (by C_2).

| x_1 | f_1 | x_2 | f_2 | x_1 | x_2 | f_3 | x_2 | x_3 | f_4 |
|-------|-------|-------|-------|-------|-------|----------|-------|-------|-------|
| e | 50 | h | 120 | e | h | 0 | h | s | 10 |
| b | 100 | i | 60 | e | i | ∞ | h | t | 30 |
| | | | | b | h | ∞ | i | s | 40 |
| | | | | b | i | 0 | i | t | 30 |

Figure 2: Four Constraints of DCOP Modeling Example 7

It is possible to check that this DCOP \mathcal{D} derives an unexpected cost-minimal solution (i.e., select e, h , and s) which yields the minimal total cost of \$180. It is because the preferences of C_2 on taking a taxi or a shuttle to the hotel must be encoded as a constraint (i.e., f_4), and thus its respective cost is accumulated to the total cost, deriving an unexpected minimal-cost solution. We believe that the lack of expressibility of the representation formalism of DCOP is the reason that prevents DCOP from successfully modeling the scenario in Example 7. MCS-OP is indeed more expressive than DCOP and could be useful in scenarios where some preferences among agents need to be considered locally and not be accounted for in the aggregation.

Related Work

To the best of our knowledge, *MCS with preferences* (MCSP), introduced in (Le, Son, and Pontelli 2015) and extended in (Le, Son, and Pontelli 2018) is the most similar proposal to MCS-OP in that both approaches aim at defining a preference order (or a qualitative comparison) among equilibria of MCSs. We therefore relate MCSP and MCS-OP in details before discussing others extensions of MCS.

MCSP assumes that the underlying logic of each context is a ranked logic, i.e., a logic with a preference order among its pairs of knowledge bases and belief sets. The preference order among equilibria of an MCSP is defined by pairwise combining the local preference orders. Specifically, an equilibrium $S = (S_1, \dots, S_n)$ is *strongly (weakly) preferred* to another equilibrium $S' = (S'_1, \dots, S'_n)$ if for every i , S_i is strongly (weakly) preferred to S'_i with respect to the underlying ranked logic of context i . An equilibrium is most strongly (weakly) preferred if there exists no other equilibrium that is strongly (weakly) preferred to it. Thus, a most strongly (weakly) preferred equilibrium in MCSP is similar to a Nash equilibrium of a game for “personal” preferences among the contexts. Then, it is not an easy task to use MCSP in formalizing problems such as DCOP where the preference order often spans more than a single local context.

Differently from MCSP, MCS-OP introduces an extra component, the cost-assignment bridge rules, to MCS and defines the notion of a cost impact of a belief state of the underlying MCS, which is used to define the order among belief states, and hence, equilibria. Being a non-negative value, the cost impact allows for the definition of a *preorder* among equilibria. Although it is convenient to use MCS-OP in modeling DCOP, we envision that MCS-OP might not be the best knowledge representation framework to model MCS with preferences where utilities/costs are difficult to obtain. For this reason, we see MCS-OP as a useful alternative that compensates rather than competes with MCSP. Example 7 also

shows that MCS-OP can be used in scenarios where agents do have local preferences which, desirably, should not be considered in the aggregate preferences of the whole system. In this sense, MCS-OP provides a continuum from fully cooperative agents (as in MCS with a centralized controller) to non-cooperative or self-interested agents (as in MCSP).

Beside equilibrium semantics, (Brewka and Eiter 2007) proposed *minimal/grounded equilibria* and *well-founded semantics* for MCS. For the former, one can directly utilize our approach (i.e., assigning conditional cost to bridge rules) to derive the preferences among equilibria, and thus obtain the semantics of the most preferred minimal/grounded equilibria. Moreover, it is unclear how our approach could be used to derive the preference in the well-founded semantics, as it is defined using an *antimonotone* operator $\gamma_M(\cdot)$.

Some significant extensions of the MCS framework have been recently introduced. *Managed MCS* (mMCS) (Brewka et al. 2011) allows applied bridge rules to also perform arbitrary operations on context knowledge bases, e.g., deletion or revision operators. *Reactive MCS* (rMCS) (Brewka et al. 2018) and *evolving MCS* (eMCS) (Gonçalves, Knorr, and Leite 2014) are extensions of mMCS to allow changes of *observations* over time. In order to model real-world situations of multi-agent systems (MAS), *DACMACS* (Costantini and Gasperis 2015) integrates Data-Aware Commitment-based MAS and MCS where agents not only interact among themselves, but also consult external heterogeneous data- and knowledge-bases to extract useful information. Clearly, these extensions are not designed to derive the preference among their respective equilibria, sequences of equilibria, or runs. It is our belief that an integration of cost-assignment bridge rules in these frameworks can enhance them by allowing the determination of a *best* run (or most preferred equilibrium).

Other extensions of MCS have been proposed, where the introduction of a preference order requires more in-depth study and will be the subject of future work, e.g., preference order among *supported equilibrium semantics* in *generalized MCS* (Tasharrofi and Ternovska 2014), in operational-like semantics of *asynchronous MCS* (Ellmauthaler and Pührer 2015), or in *idealized run semantics* of *streaming MCS* (Dao-Tran and Eiter 2017).

Another related line of research is *preference-based inconsistency management* in MCS. An MCS is inconsistent if it does not have an equilibrium; research has been done to analyze the inconsistency to identify the *most preferred diagnoses* (Eiter et al. 2014). A diagnosis of an MCS intuitively is a pair of the set of all bridge rules that are either removed or set to be applicable in all belief states to make the MCS consistent. There are ways to accommodate preferences among diagnoses (see (Eiter and Weinzierl 2017)), and intuitively one can see that the most preferred equilibrium of M is an equilibrium of M after being applied to the most preferred diagnosis. Different from using diagnoses, (Mu, Wang, and Wen 2016) proposed to use *Preferential MCS* (PMCS), which are stratified MCS based on predetermined partial ordering among contexts. From such ordering, to address the inconsistency, one may ask for a *maximal consistent section* as the semantics of PMCS. Compared to our

work in this paper, there is a significant difference: the works in inconsistency management are used to resolve MCS that do not admit an equilibrium, while in MCS-OP preference information is directly integrated into the semantics to select among existing equilibria.

We note that there have been attempts to extend the DCOP model. For example, Asymmetric DCOPs (Grinshpoun et al. 2013) allows different agents owning variables in the scope of a constraint can incur to different costs, given a fixed join assignment. Multi-Objective DCOPs (Marler and Arora 2004) are problems involving more than one objective function to be optimized simultaneously. However, these extensions do not address the difficulty in modeling scenarios where some preferences among agents need to be considered locally and not be accounted for in the aggregation. MCS-OP could therefore be viewed as an alternative to DCOP that avoids this difficulty.

Conclusions and Future Works

We proposed Multi-context System for Optimization Problems (MCS-OP), which associate each context with a set of cost-assignment bridge rules that assign conditional cost to its bridge rules. The preferences order among equilibria are determined by the total incurred cost of actually-applied bridge rules in the equilibria. We discussed the MCS-OP's complexity and showed how to model DCOPs using MCS-OP to illustrate the contribution of this paper to both the MCS community and the DCOP community.

It is not difficult to see that cost-assignment bridge rules can be introduced to any of the existing MCS extensions (e.g., MCSP, mMCS, asynchronous MCS (Ellmauthaler and Pührer 2015) and streaming MCS (Dao-Tran and Eiter 2017)). In other words, MCS-OP can be easily integrated with almost all of the MCS extensions and the notion of a most preferred equilibrium of such an extension can be defined as in the “*Most Preferred Equilibria*” subsection. For this reason, it would be interesting to investigate the integration of MCS-OP into other extensions of MCS and their properties (e.g., as we have mentioned in the previous subsection, the cost-assignment bridge rules could be useful in mMCS to identify the best runs). We also plan to investigate the use of a meta-reasoning transformation as in (Eiter and Weinzierl 2017), which implements a rewriting technique, to select most preferred equilibria. Furthermore, as we have mentioned in the introduction, there is only a few systems for computing MCS that have been developed. On the other hand, many efficient and scalable systems for solving DCOPs. As such, we also plan to investigate algorithms for computing MCS/MCS-OP that can take advantage of approaches used in computing DCOP solutions.

Acknowledgments

The authors acknowledge the partial support from the NSF grants HRD-1345232 and IIS-1812628.

References

- Brewka, G., and Eiter, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proc. of AAI*, 385–390.
- Brewka, G.; Eiter, T.; Fink, M.; and Weinzierl, A. 2011. Managed multi-context systems. In *Proc. of IJCAI*, 786–791.
- Brewka, G.; Ellmauthaler, S.; Gonçalves, R.; Knorr, M.; Leite, J.; and Pührer, J. 2018. Reactive multi-context systems: Heterogeneous reasoning in dynamic environments. *Artif. Intell.* 256:68–104.
- Brewka, G.; Niemelä, I.; and Truszczynski, M. 2003. Answer set optimization. In *Proc. of IJCAI*, 867–872.
- Costantini, S., and Gasperis, G. D. 2015. Exchanging data and ontological definitions in multi-agent-contexts systems. In *Proc. of RuleML*.
- Dao-Tran, M., and Eiter, T. 2017. Streaming multi-context systems. In *Proc. of IJCAI*, 1000–1007.
- Eiter, T., and Weinzierl, A. 2017. Preference-based inconsistency management in multi-context systems. *J. Artif. Intell. Res.* 60:347–424.
- Eiter, T.; Fink, M.; Schüller, P.; and Weinzierl, A. 2010. Finding explanations of inconsistency in multi-context systems. In *Proc. of KR*.
- Eiter, T.; Fink, M.; Schüller, P.; and Weinzierl, A. 2014. Finding explanations of inconsistency in multi-context systems. *Artif. Intell.* 216:233–274.
- Ellmauthaler, S., and Pührer, J. 2015. Asynchronous multi-context systems. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, 141–156.
- Fioretto, F.; Pontelli, E.; and Yeoh, W. 2018. Distributed constraint optimization problems and applications: A survey. *J. Artif. Intell. Res.* 61:623–698.
- Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput.* 9(3/4):365–386.
- Gershman, A.; Meisels, A.; and Zivan, R. 2009. Asynchronous forward bounding for distributed cops. *J. Artif. Intell. Res.* 34:61–88.
- Gonçalves, R.; Knorr, M.; and Leite, J. 2014. Evolving multi-context systems. In *Proc. of ECAI*, 375–380.
- Grinshpoun, T.; Grubshtein, A.; Zivan, R.; Netzer, A.; and Meisels, A. 2013. Asymmetric distributed constraint optimization problems. *J. Artif. Intell. Res.* 47:613–647.
- Kumar, A.; Faltings, B.; and Petcu, A. 2009. Distributed constraint optimization with structured resource constraints. In *Proc. of AAMAS*, 923–930.
- Lass, R.; Kopena, J.; Sultanik, E.; Nguyen, D.; Dugan, C.; Modi, P.; and Regli, W. 2008. Coordination of first responders under communication and resource constraints (Short Paper). In *Proc. of AAMAS*, 1409–1413.
- Le, T.; Son, T. C.; and Pontelli, E. 2015. Multi-context systems with preferences. In *Proc. of PRIMA*, 449–466.
- Le, T.; Son, T. C.; and Pontelli, E. 2018. Multi-context systems with preferences. *Fundam. Inform.* 158(1-3):171–216.
- Léauté, T., and Faltings, B. 2011. Coordinating logistics operations with privacy guarantees. In *Proc. of IJCAI*, 2482–2487.
- Maheswaran, R.; Tambe, M.; Bowring, E.; Pearce, J.; and Varakantham, P. 2004. Taking DCOP to the real world: Efficient complete solutions for distributed event scheduling. In *Proc. of AAMAS*, 310–317.
- Mailler, R., and Lesser, V. 2004. Solving distributed constraint optimization problems using cooperative mediation. In *Proc. of AAMAS*, 438–445.
- Marler, R., and Arora, J. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26(6):369–395.
- McCarthy, J. 1987. Generality in artificial intelligence. *Commun. ACM* 30(12):1029–1035.
- Modi, P. J.; Shen, W.; Tambe, M.; and Yokoo, M. 2005. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.* 161(1-2):149–180.
- Mu, K.; Wang, K.; and Wen, L. 2016. Preferential multi-context systems. *Int. J. Approx. Reasoning* 75:39–56.
- Ottens, B.; Dimitrakakis, C.; and Faltings, B. 2012. DUCT: an upper confidence bound approach to distributed constraint optimization problems. In *Proc. of AAI*.
- Petcu, A., and Faltings, B. 2005. A scalable method for multiagent constraint optimization. In *Proc. of IJCAI*, 266–271.
- Roelofsen, F., and Serafini, L. 2005. Minimal and absent information in contexts. In *Proc. of IJCAI*, 558–563.
- Tasharrofi, S., and Ternovska, E. 2014. Generalized multi-context systems. In *Proc. of KR*.
- Ueda, S.; Iwasaki, A.; and Yokoo, M. 2010. Coalition structure generation based on distributed constraint optimization. In *Proc. of AAI*, 197–203.
- Yeoh, W., and Yokoo, M. 2012. Distributed problem solving. *AI Magazine* 33(3):53–65.
- Zivan, R.; Grinton, R.; and Sycara, K. 2009. Distributed constraint optimization for large teams of mobile sensing agents. In *Proc. of IAT*, 347–354.
- Zivan, R.; Okamoto, S.; and Peled, H. 2014. Explorative anytime local search for distributed constraint optimization. *Artificial Intelligence* 212:1–26.