Investigation on Stochastic Local Search for Decentralized Asymmetric Multi-objective Constraint Optimization Considering Worst Case

Toshihiro Matsui
Nagoya Institute of Technology, Gokiso-cho Showa-ku Nagoya Aichi 466-8555, Japan

Keywords: Asymmetric Distributed Constraint Optimization, Multi-Objective, Stochastic Local Search, Multiagent.

Abstract: The Distributed Constraint Optimization Problem (DCOP) has been studied as a fundamental problem in multiagent cooperation. With the DCOP approach, various cooperation problems including resource allocation and collaboration among agents are represented and solved in a decentralized manner. Asymmetric Multi-Objective DCOP (AMODCOP) is an extended class of DCOPs that formalizes the situations where agents have individual objectives to be simultaneously optimized. In particular, the optimization of the worst case objective value among agents is important in practical problems. Existing works address complete solution methods including extensions with approximation. However, for large-scale and dense problems, such solution methods are insufficient. Although the existing studies also address a few simple deterministic local search methods, there are opportunities to introduce stochastic local search methods. As the basis for applying stochastic local search methods to AMODCOPs for the preferences of agents, we introduce stochastic local search methods with several optimization criteria. We experimentally analyze the influence of the optimization criteria on perturbation in the exploration process of search methods and investigate additional information propagation that extends the knowledge of the agents who are performing the local search.

1 INTRODUCTION

The Distributed Constraint Optimization Problem (DCOP) has been studied as a fundamental problem in multiagent cooperation (Fioretto et al., 2018). With the DCOP approach, various cooperation problems including resource allocation and collaboration among agents are represented and solved in a decentralized manner. Asymmetric Multi-Objective DCOP (AMODCOP) is an extended class of DCOPs that formalizes the situations where agents have individual objectives to be simultaneously optimized (Matsui et al., 2018a). In particular, the optimization of the worst case objective value among agents is important in practical problems. Existing works address complete solution methods including extensions with approximation. However, for large-scale and dense problems, such solution methods are insufficient. Although the current studies also address a few simple deterministic local search methods (Matsui et al., 2018b), there are opportunities to introduce stochastic local search methods. As the basis for applying stochastic local search methods to AMODCOPs for the preferences of agents, we introduce stochastic local search methods with several optimization criteria. We experimentally analyze the influence of the optimization criteria on perturbation in the exploration process of the search methods. We also investigate additional information propagation, which extends the knowledge of the agents that are performing the local search.

The contribution of this study is as follows. 1) We apply several variants of a fundamental stochastic local search method to a class of Asymmetric Multi-Objective DCOPs where the worst case cost value among agents is improved. 2) The effect of the stochastic local search with different optimization criteria is experimentally investigated, and we show the cases where a leximin based criterion is effective with the stochastic local search.

The rest of our paper is organized as follows. In the next section, we present our preliminary study that includes standard DCOPs, solution methods for DCOPs, Asymmetric Multi-Objective DCOPs that consider the worst case, social welfare, and the scalability issues of solution methods for Asymmetric Multi-Objective DCOPs. In Section 3, we propose decentralized stochastic local search methods for AMODCOPs considering the worst case cost value among agents. We apply fundamental stochastic lo-

Matsui, T.
Investigation on Stochastic Local Search for Decentralized Asymmetric Multi-objective Constraint Optimization Considering Worst Case.
DOI: 10.5220/0010395504620469
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 1*, pages 462-469
ISBN: 978-989-758-484-8

cal search methods with several optimization criteria to this class of problems. We experimentally investigate the proposed approach in Section 4. We briefly discuss our work in Section 5 and conclude in Section 6.

2 PRELIMINARY

We address a class of DCOPs where agents asymmetrically evaluate their local objectives, and multiple objectives of the agents are simultaneously optimized. Our problem is based on the definition in (Matsui et al., 2018a), and we mainly concentrate on the problems where the worst case cost among agents is minimized.

2.1 Distributed Constraint Optimization Problems

A Distributed Constraint Optimization Problem (DCOP) is defined by $\langle A, X, D, F\rangle$ where A is a set of agents, X is a set of variables, D is a set of the domains of the variables, and F is a set of objective functions. Variable $x_i \in X$ represents a state of agent $i \in A$. Domain $D_i \in D$ is a discrete finite set of values for x_i. Objective function $f_{i,j}(x_i, x_j) \in F$ defines a utility extracted for each pair of assignments to x_i and x_j. The objective value of assignment $\{(x_i, d_i), (x_j, d_j)\}$ is defined by binary function $f_{i,j} : D_i \times D_j \to \mathbb{N}_0$. For assignment $\mathcal{A}$ of the variables, global objective function $F(\mathcal{A})$ is defined as $F(\mathcal{A}) = \sum_{f_{i,j} \in F} f_{i,j}(\mathcal{A}_{\downarrow x_i}, \mathcal{A}_{\downarrow x_j})$, where $\mathcal{A}_{\downarrow x_i}$ is the projection of assignment $\mathcal{A}$ on x_i.

The value of x_i is controlled by agent i, which locally knows the objective functions that are related to x_i in the initial state. The goal is to find global optimal assignment $\mathcal{A}^*$ that minimizes the global objective value in a decentralized manner. For simplicity, we focus on the fundamental case where the scope of constraints/functions is limited to two variables, and each agent controls a single variable.

2.2 Solution Methods for DCOPs

The solution methods for DCOPs are categorized into exact and inexact solution methods (Fioretto et al., 2018). The former are based on tree search and dynamic programming (Fioretto et al., 2018). However, the time/space complexity of such exact methods is generally exponential for one of size parameters of problems. Therefore, applying the exact methods to large-scale and densely constrained problems is difficult. On the other hand, inexact solution methods consist of a number of approaches (Fioretto et al., 2018) including hill-climbing local search, stochastic random sampling and belief propagation.

In this study, we focus on Distributed Stochastic search Algorithm (DSA) (Zhang et al., 2005; Zivan, 2008), which is a baseline algorithm of inexact methods. DSA is the simplest stochastic local search method for DCOPs where agents locally determine the assignment to their own variables by exchanging their assignments with neighborhood agents that are related by constraints/functions. For each agent i, an iteration of the algorithm is synchronously performed as follows.

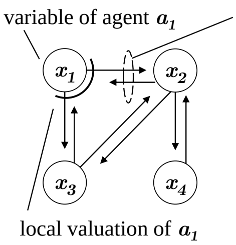
1. Set the initial assignment to variable x_i of agent i.
2. Collect the current assignment of the variables of the neighborhood agents in Nbr_i. With the collected assignment and functions related to agent i, locally evaluate all the assignments of variable x_i. Then select a new assignment to x_i with a stochastic strategy and the evaluation for x_i.
3. Update the assignment to x_i. Repeat from step 2.

Each agent i has a view of assignments to the variables of the neighborhood agents Nbr_i that is denoted by $\mathcal{A}_i$. Agent i's local cost is defined as $f_i(\mathcal{A}_i) = \sum_{j \in Nbr_i} f_{i,j}(\mathcal{A}_{i\downarrow x_i}, \mathcal{A}_{i\downarrow x_j})$. Here we assume that $\mathcal{A}_i$ also contains i's own assignment. The agent only interests to improve its local cost value and does not receive local cost values of other agents.

Although there are a few versions of stochastic search strategies, most of them randomly select 1) hill-climb, 2) stay as is, or 3) switch to one of other assignments to its own variable whose local objective value equals the current local objective value. With preliminary experimental investigation, we prefer the following settings in this work. 1) If there are other assignments to the agent's own variable that improves the local objective, the agent performs a hill-climb with probability P_a and randomly selects one of the assignments related to the best improvement with uniform distribution. 2) Otherwise, the agent randomly selects one of other assignments with probability P_b. With a relatively large P_a and a relatively smaller P_b, the hill-climb and escape from local optimal solutions are stochastically performed.

Although there are a number of sophisticated solution methods for DCOPs, DSA is considered a baseline method. Therefore, there are opportunities to investigate the performance of the stochastic local search for the problems based on min-max problems.

Several random-walk solution methods require/employ snapshot algorithms to capture global objective values and related global solutions. A fun-



x_1	x_2	$f_{1,2}$	$f_{2,1}$
0	0	a	b
0	1	c	d
1	0	e	f
1	1	g	h

Figure 1: AMODCOP.

damental approach is based on a spanning tree on a constraint graph and the implicit/explicit timestamps of locally evaluated objective values to aggregate them in a synchronized manner (Mahmud et al., 2020). With such methods, global objective values can be aggregated with the number of communication cycles that equals the depth of the spanning tree. A similar additional snapshot algorithm has been applied to DSA (Zivan, 2008). Note that it is possible to employ similar mechanism so that the agents symmetrically and simultaneously collect snapshots using the same single spanning tree, where each agent performs as a root node of the spanning tree. Here we concentrate on a solution method and employ snapshots on a simulator for simplicity.

2.3 Asymmetric Multi-objective Distributed Constraint Optimization Problem

An Asymmetric Multiple Objective DCOP on the preferences of agents (AMODCOP) is defined by $\langle A, X, D, F \rangle$, where A, X and D are similarly defined for the DCOP in Section 2.1. Agent $i \in A$ has its local problem defined for $X_i \subseteq X$. For neighboring agents i and j, $X_i \cap X_j \neq \emptyset$. F is a set of objective functions $f_i(X_i)$. Function $f_i(X_i) : D_{i_1} \times \cdots \times D_{i_k} \rightarrow \mathbb{N}_0$ represents the objective value for agent i based on the variables in $X_i = \{x_{i_1}, \cdots, x_{i_k}\}$. For simplicity, we concentrate on the case where each agent has a single variable and relates to its neighborhood agents with binary functions. The functions are asymmetrically defined and locally aggregated. Variable x_i of agent i is related to other variables by objective functions. When x_i is related to x_j, agent i evaluates objective function $f_{i,j}(x_i, x_j)$. On the other hand, j evaluates another function $f_{j,i}(x_j, x_i)$. Each agent i has function $f_i(X_i)$ that represents the local problem of i that aggregates $f_{i,j}(x_i, x_j)$. We define the local evaluation of agent i as summation $f_i(X_i) = \sum_{j \in Nbr_i} f_{i,j}(x_i, x_j)$ for neighborhood agents $j \in Nbr_i$ related to i by objective functions.

Global objective function $\mathbf{F}(\mathcal{A})$ is defined as $[f_1(\mathcal{A}_1), \cdots, f_{|A|}(\mathcal{A}_{|A|})]$ for assignment $\mathcal{A}$ to all the variables. Here $\mathcal{A}_i$ denotes the projection of assignment $\mathcal{A}$ on X_i . The goal is to find assignment $\mathcal{A}^*$ that minimizes the global objective based on a set of aggregation and evaluation structures. Figure 1 shows an example of AMODCOP.

2.4 Multiple Objectives and Social Welfare

Since multiple objective problems among individual agents cannot be simultaneously optimized in general cases, several criteria such as Pareto optimality are considered. However, there are generally a huge number of candidates of optimal solutions based on such criteria. Therefore, several social welfare and scalarization functions are employed. With aggregation and comparison operators $\oplus$ and $\prec$, the minimization of the objectives is represented as follows: $\mathcal{A}^* = \mathrm{argmin}^{\prec}_{\mathcal{A}} \oplus_{i \in A} f_i(\mathcal{A})$.

Several types of social welfare (Sen, 1997) and scalarization methods (Marler and Arora, 2004) are employed to handle objectives. In addition to the summation and comparison of scalar objective values, we consider several criteria based on the worst case objective values (Matsui et al., 2018a). Although some operators and criteria are designed for the maximization problems of utilities, we employ similar criteria for minimization problems.

Summation $\sum_{a_i \in A} f_i(X_i)$ only considers the total utilities. Min-max criterion $\min \max_{a_i \in A} f_i(X_i)$ improves the worst case cost value. This criterion is called the Tchebycheff function. Although it does not consider global cost values and is not Pareto optimal, the criterion to improve the worst case is practically important in situations without trades of cost (utility) values. To improve the global cost values, ties of min-max are broken by comparing the summation values, and the criterion is Pareto optimal. We employ the lexicographic augmented Tchebycheff function that independently compares maximum and summation values (Marler and Arora, 2004).

Lexmin for maximization problems is an extension of the max-min that is the maximization version of min-max. With this criterion, utility values are represented as a vector whose values are sorted in ascending order, and the comparison of two vectors is based on the dictionary order of the values in the vectors. Maximization with leximin is Pareto optimal and relatively improves the fairness among objectives. We address 'leximax', which is an inverted lexmin for minimization problems, where objective values are sorted in descending order. See the literature for details (Sen, 1997; Marler and Arora, 2004; Matsui et al., 2018a).

In this study, we mainly concentrate on improving the worst case value, since it is a challenge for decentralized local search methods.

To evaluate experimental results, we also consider the Theil index T, which is a measurement of unfairness: $T = \frac{1}{|A|}\sum_{i=1}^{|A|}\left(\frac{f_i(X_i)}{\overline{f}} \ln \frac{f_i(X_i)}{\overline{f}}\right)$, where $\overline{f}$ denotes the average value for all $f_i(X_i)$. T takes zero if all $f_i(X_i)$ are identical.

2.5 Scalability of Exact Solution Methods and Stochastic Local Search

Several exact solution methods based on tree search and dynamic programming for AMODCOPs with preferences for individual agents have been proposed (Matsui et al., 2018a). However, such methods cannot be applied to large-scale and complex problems with dense constraints/functions where the tree-width of constraint graphs, which is related to the number of combinations of assignments in partial problems, is intractable. Although several approximations have been proposed for those types of solution methods (Matsui et al., 2018b), the accuracy of the solutions decreases when a number of constraints are eliminated from densely constrained problems.

Therefore, opportunities can be found for employing local search methods. Although several local search methods were addressed in earlier studies for such problems (Matsui et al., 2018b), stochastic local search methods have not been adequately addressed. Since such solution methods are an important class of baseline methods, we focus on the stochastic local search for AMODCOPs with variants of a min-max criterion for the worst case cost value among agents.

3 DECENTRALIZED STOCHASTIC LOCAL SEARCH FOR AMODCOPS WITH WORST CASE CRITERIA

3.1 Applying Stochastic Local Search to AMODCOPs

To handle multiple objectives, the objective values of other agents must be evaluated in addition to the objective value of each agent even in a local search with a narrow view. For Asymmetric Multi-Objective problems, we adjusted the original DSA so that each agent collects the local objective values of the neighborhood agents. Each agent i repeats the following steps in each iteration.

1. Set the initial assignment to variable x_i of agent i.
2. Collect the current assignment of the variables of the neighborhood agents in Nbr_i. With the assignment and functions related to each agent, evaluate local objective value $f_i(\mathcal{A}_i)$ to the current assignment.
3. Collect the current objective values f_j of the neighborhood agents $j \in Nbr_i$. With the objective values of the neighborhood agents, the current assignment and functions related to each agent, locally evaluate all the assignments to x_i. Then select a new assignment with a stochastic strategy and the evaluation of assignments of x_i.
4. Update the assignment to variable x_i of agent i. Repeat from step 2.

Here we insert an additional step 3. Each agent updates its current objective value based on a received partial assignment to the variables of neighborhood agents and then receives the current objective values of the neighborhood agents. This step is necessary to improve the consistency of the current assignment and the related objective values.

The local evaluation is also modified to consider the current cost values f_j for neighborhood agents $j \in Nbr_i$. Although each agent can employ several social welfares and scalarization functions, their scope is limited to the objective values of neighborhood agents. In this work, it is assumed that all the agents employ the same criterion in the optimization process for a common goal.

3.2 Adjusting for the Criteria

With the modification of the collected information, each agent has a view of the current objective values of the neighborhood agents. In a local search, each agent only evaluates the multiple objectives within the view and the evaluation partially overlaps thoughout the whole system.

An issue is that local evaluation with a criterion of social welfare is inexact, and the choice of assignments to variables might unexpectedly influence the perturbation in the search process. For example, minimization of the maximum objective value might improve perturbation more than the minimization of the summation for the best solution based on a different criterion. Therefore, we simultaneously employ several different criteria to monitor and capture snapshots of the best solution. We capture the best solutions based on different multiple criteria, although the search process is performed based on a single criterion. As mentioned in Section 2.2, we capture snapshots of the best solutions by a simulator.

3.3 Considering Opposite Constraints

In the original problem setting, each agent locally aggregates its own asymmetric constraints related to the agent. However, when each agent locally searches for the assignment to its variable, each agent cannot evaluate the influence of the new assignment on the neighborhood agents. By revealing the opposite part of an asymmetric constraint to neighborhood agents, the change of the objective values of the neighborhood agents can also be evaluated. This might be too optimistic, since each agent independently estimates the improvement of the cost values.

3.4 Local Agreement

Since each agent locally determines the assignment to its own variable, the selection usually causes a mismatch among neighborhood agents. To reduce such cases, we employ an agreement mechanism based on a local leader election that resembles MGM (Fioretto et al., 2018), which is a local search method for DCOPs. For a leader election, an additional communication step is introduced as follows.

1. Set the initial assignment to variable x_i of agent i.
2. Collect the current assignment of the variables of the neighborhood agents in Nbr_i. With the assignment and functions related to each agent, evaluate local objective value $f_i(\mathcal{A}_i)$ to the current assignment.
3. Collect the current objective values f_j of the neighborhood agents $j \in Nbr_i$. With the objective values of the neighborhood agents, the current assignment and functions related to each agent, locally evaluate all the assignments to x_i. Then select a new assignment with a stochastic strategy and the evaluation of the assignment to x_i. Compute the improvement g_i of the objective value by the new assignment.
4. Collect the improvement g_j of objective values of the neighborhood agents $j \in Nbr_i$. Evaluate the best improvement value g^*_k among the neighborhood agents with a tie-breaker based on the identifiers of the agents.
5. Update the assignment to a variable of agent i if its improvement is the best value g^*_i. Repeat from step 2.

In Step 4, agents compute the improvement g_i of the local evaluations and exchange information about the improvement with the neighborhood agents. The information of the improvement is the same type as the information for the criterion of optimization. For example, a sorted objective vector is exchanged for leximax criterion.

3.5 Employing Global Information

Although each agent locally searches for the assignment to its local value, there are opportunities to access some global information. We employ global objective values by assuming the snapshot methods mentioned in Section 2.2. Here we simply employ snapshots in a simulator by inserting a sufficient delay of iterations that equals the number of agents so that the agents do not access the global objective values too early.

While the global summation of objective values cannot be compared with the local evaluation of each agent, the worst case objective value among the agents can directly be a bound for each agent. Therefore, we limit the locally worst case objective value in each agent's view by the globally worst case value. The locally maximum cost value f^{max}_k in a view of agent i is limited by globally maximum cost value F^{max} as $f^{max}_k \leftarrow \min(f^{max}_k, F^{max})$. In the case of leximax, the globally maximum value is employed instead of the first value in the sorted objective vector of the 'maximum' cost values. We do not employ this information when the optimization criterion is the summation.

4 EVALUATION

4.1 Settings

We experimentally evaluated our proposed methods. The benchmark problems consist of n agents/variables and c asymmetric constraints, which are a pair of binary objective fictions for two agents. Each variable takes a value in its domain that contains d discrete values. We employed the following types of cost functions for minimization problems. 1) rnd: The cost values are randomly set to integer values in $[1, 100]$ based on uniform distribution. 2) gmm: The cost values are random integer values in $[1, 100]$ that are rounded down from random values based on gamma distribution with $\alpha = 9$ and $\beta = 2$.

In a stochastic local search, one of the following criteria is employed. 1) sum: The summation of the objective values in the view of each agent. 2) max:

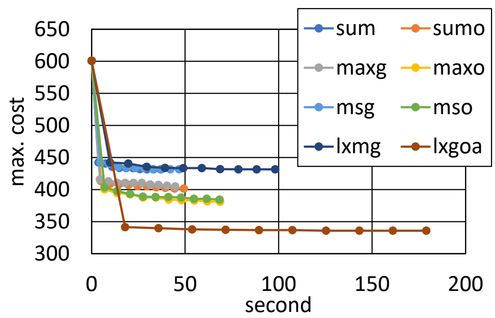


Figure 2: Example of anytime curve (rnd, $n = 100$, $d = 3$, $c = 250$, $(P_a, P_b) = (0.9, 0.1)$, max. cost).

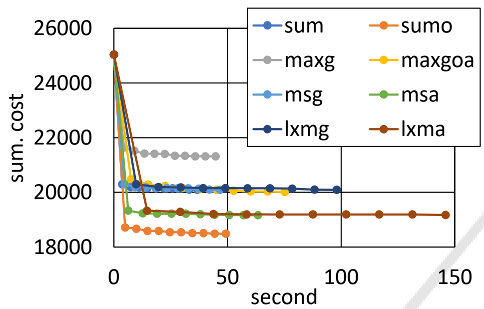


Figure 3: Example of anytime curve (rnd, $n = 100$, $d = 3$, $c = 250$, $(P_a, P_b) = (0.9, 0.1)$, sum. cost).

The maximum of the objective values in the view of each agent. 3) ms: An extension of 'max' where ties are broken by 'sum'. 4) lxm: The leximax criteria for cost values that is similar to leximin for utility values. These criteria are also employed to capture snapshots of the best solutions in each iteration. We captured four different best solutions based on different criteria, including the one for optimization. We also evaluated the Theil index values for the solutions.

For stochastic searches, we set probabilities $(P_a, P_b) = (1, 0)$ and $(0.9, 0.1)$ by preliminary experiments. The former setting is a deterministic local search method.

We denote the additional mechanisms in Section 3 as follows. 1) (none): base method. 2) o: evaluation of part of the opposite objective functions. 3) a: local agreement with neighborhood agents. 4) g: modification of local evaluation by globally worst cost value. These are added to the names of optimization criteria (e.g. 'maxgoa').

We compared the best solutions that are captured with several criteria at the cutoff iteration 200000. The results are averaged over ten trials with different initial solutions on ten problem instances for each setting.

4.2 Results

Table 1 shows the solution quality for 'rnd', $n = 50$, $d = 3$, $c = 150$ and $(P_a, P_b) = (1, 0)$). For each optimization criterion, we selected the variants of solution that result the best maximum and summation cost values. In this result, the solution quality of the variants of 'max' was relatively worse than the others. A possible reason is that the search can easily be captured by locally optimal solutions, since the local search process is deterministic in these settings. The results of 'ms' and 'lxm' that are assisted by additional characteristics of the criteria are relatively better. In particular, 'lxm' with 'oa' significantly outperformed the other methods for the maximum cost value. It can be considered that the effect of leximax to improve fairness by comparing more than secondly worst cases might also improved the local search. In addition, several variants of 'lxm' relatively improved the Theil index. It reveals the possibility that the property of leximax to improve fairness is effective in several settings of local search.

Tables 2 and 3 show the solution quality in the case of 'rnd' and $(P_a, P_b) = (0.9, 0.1)$. In these cases, the solution quality of variants of 'max' was relatively better in comparison to the deterministic search. It is considered the effect of the stochastic escape form locally optimal solutions. With additional information, the solution quality was improved in several cases. Although 'o' that referred opposite cost functions was too optimistic without different additional information in several cases, it was effective with some additional information. We found that 'g' that employed the global min-max values looks improved the variants of 'max' in a few cases. However, the effect decreased by employing other types of additional information. As mentioned above, several variants of 'lxm' were relatively better, while its computational overhead to maintain sorted objective vectors is significantly higher than 'max'.

Table 4 shows the solution quality in the case of gmm, $n = 50$, $d = 3$, $c = 150$ and $(P_a, P_b) = (0.9, 0.1)$. Here, the effectiveness of the solution methods was different from the previous cases. It can be considered that the non-uniform distribution of cost values prevented some cooperative evaluation among neighborhood agents.

Figures 2 and 3 show typical anytime curves of maximum and summation cost values in a case of 'rnd'. The snapshots of cost values are independently captured with the globally maximum and summation values. Since the evaluated optimization criteria need different computational overheads, we plot the results considering computational time. This experi-

Table 1: Solution quality (rnd, $n = 50$, $d = 3$, $c = 150$, $(P_a, P_b) = (1, 0)$).

alg.	sum			max			ms			lxm		
	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil
sum	537.1	12224.4	0.1070	**449.1**	12926.0	0.0915	**449.1**	**12902.6**	0.0922	**449.1**	12912.2	0.0917
suma	536.5	**12160.2**	0.1107	466.2	12572.0	0.0995	466.2	12565.0	0.0996	466.2	12593.1	0.0991
maxo	536.4	**13443.6**	0.0834	**480.2**	14053.7	0.0730	**480.2**	**14014.6**	0.0725	**480.2**	14047.3	0.0725
ms	537.1	12224.4	0.1070	**449.1**	12926.0	0.0915	**449.1**	**12902.6**	0.0922	**449.1**	12912.2	0.0917
msa	536.5	**12146.2**	0.1148	462.6	12711.9	0.0972	462.6	12678.5	0.0974	462.6	12696.4	0.0977
msg	537.1	12224.4	0.1070	**449.1**	12926.0	0.0915	**449.1**	**12902.6**	0.0922	**449.1**	12912.2	0.0917
lxma	528.9	**12160.3**	0.1117	458.8	12766.4	0.0963	458.8	12745.6	0.0971	458.8	12760.3	0.0964
lxmgoa	410.7	13061.3	0.0631	**394.1**	13308.0	0.0570	**394.1**	**13196.0**	0.0581	**394.1**	13266.9	0.0569
lxmoa	410.7	13061.3	0.0631	**394.1**	13308.0	0.0570	**394.1**	**13196.0**	0.0581	**394.1**	13266.9	0.0569

Table 2: Solution quality (rnd, $n = 50$, $d = 3$, $c = 150$, $(P_a, P_b) = (0.9, 0.1)$).

alg.	sum			max			ms			lxm		
	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil
sumo	486.1	**11371.2**	0.1075	**391.8**	12150.1	0.0822	**391.8**	**12115.7**	0.0823	**391.8**	12156.2	0.0820
maxgoa	532.6	**12214.8**	0.1047	431.9	13033.2	0.0780	431.9	12859.9	0.0822	431.9	13004.5	0.0764
maxo	476.9	12523.4	0.0841	**391.1**	13277.3	0.0593	**391.1**	**13270.7**	0.0591	**391.1**	13277.4	0.0586
msa	517.7	**11907.9**	0.1135	426.6	12541.6	0.0929	426.6	12519.7	0.0932	426.6	12534.3	0.0930
mso	477.4	12512.4	0.0844	**391.7**	13321.3	0.0585	**391.7**	**13316.8**	0.0587	**391.7**	13320.5	0.0587
lxma	508.4	**11872.4**	0.1109	420.9	12579.0	0.0898	420.9	12548.6	0.0904	420.9	12581.1	0.0896
lxmgoa	391.2	12459.9	0.0641	**369.4**	12973.6	0.0554	**369.4**	**12748.6**	0.0576	**369.4**	12986.5	0.0524
lxmoa	391.2	12459.9	0.0641	**369.4**	12973.6	0.0554	**369.4**	**12748.6**	0.0576	**369.4**	12986.5	0.0524

Table 3: Solution quality (rnd, $n = 100$, $d = 3$, $c = 250$, $(P_a, P_b) = (0.9, 0.1)$).

alg.	sum			max			ms			lxm		
	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil
sumo	512.5	**18690.5**	0.1402	**389.7**	20282.9	0.1140	**389.7**	**20235.7**	0.1141	**389.7**	20269.8	0.1140
maxgoa	531.9	**19829.8**	0.1169	383.5	20971.4	0.0937	383.5	20811.9	0.0948	383.5	20948.5	0.0923
maxo	477.5	21242.9	0.1010	**377.9**	22416.4	0.0798	**377.9**	**22393.9**	0.0796	**377.9**	22406.0	0.0795
msa	527.7	**19173.4**	0.1435	413.3	20228.6	0.1276	413.3	20182.9	0.1283	413.3	20228.4	0.1273
mso	468.9	21209.3	0.0997	**379.4**	22350.2	0.0807	**379.4**	**22342.2**	0.0808	**379.4**	22354.8	0.0806
lxma	518.5	**19199.3**	0.1409	412.5	20387.0	0.1254	412.5	20314.6	0.1256	412.5	20376.5	0.1250
lxmgoa	352.0	20376.1	0.0793	**334.4**	21292.6	0.0706	**334.4**	**20765.7**	0.0757	**334.4**	21161.3	0.0703
lxmoa	352.0	20376.1	0.0793	**334.4**	21292.6	0.0706	**334.4**	**20765.7**	0.0757	**334.4**	21161.3	0.0703

Table 4: Solution quality (gmm, $n = 50$, $d = 3$, $c = 150$, $(P_a, P_b) = (0.9, 0.1)$).

alg.	sum			max			ms			lxm		
	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil	max.	sum.	theil
sumo	177.3	**4483.9**	0.0794	**153.5**	4719.7	0.0697	**153.5**	**4697.8**	0.0699	**153.5**	4723.8	0.0691
maxo	166.5	**4711.4**	0.0677	**145.5**	4938.3	0.0569	**145.5**	**4899.7**	0.0577	**145.5**	4930.6	0.0568
msa	178.7	**4555.6**	0.0811	160.8	4706.6	0.0741	160.8	4696.0	0.0743	160.8	4711.3	0.0737
mso	165.9	4699.4	0.0673	**145.9**	4936.2	0.0572	**145.9**	**4908.9**	0.0581	**145.9**	4937.9	0.0570
lxma	181.2	**4557.4**	0.0815	160.1	4726.0	0.0726	160.1	4700.4	0.0732	160.1	4720.9	0.0725
lxmo	164.6	4708.6	0.0664	**145.8**	4948.7	0.0563	**145.8**	**4919.7**	0.0564	**145.8**	4940.4	0.0564

ment is performed on a computer with g++ (GCC) 4.4.7, Linux version 2.6, Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz and 32GB memory. Note that there are opportunities to improve our experimental implementation.

For each optimization criterion, we selected the fastest variant and the best-solution-quality variant. While the optimization process with complex criteria and more additional information results relatively better solutions, the process with 'light' criteria has chance to perform more iterations. However, for min-max cost values of relatively large-scale problems, the best variant of 'lxm' optimization was relatively better. We note that for 'max', 'ms' and 'lxm' variants, employing snapshots of globally maximum cost values were slightly faster than the simplest one. This was a side-effect where part of evaluation was conditionally skipped due to the threshold based on the globally maximum cost value.

5 DISCUSSION

There are several decentralized min-max problem settings that require several assumptions including the convexity and continuity of objective functions. However, the aim of DCOPs with similar criteria is general discrete problems with non-convexity. To solve large-scale and complex problems of these classes, inexact methods are necessary. As a baseline investigation, we addressed a class of stochastic hill-climb methods that is an extension of a fundamental approach for traditional DCOPs.

An approximation of complete solution methods for AMODCOPs that are based on tree search and dynamic programming has been presented. However, since the approximation partially eliminates or ignores several constraints, its accuracy decreases when the induced width (Fioretto et al., 2018) of constraint graphs is relatively larger. In such situations, there are opportunities to employ stochastic local search methods. While deterministic local search methods for this class of problems have been addressed in early studies, we investigated stochastic local search methods that also employ some additional information.

In the original DSA and several variants, the probability of the algorithms is generally employed to determine whether the agents select the hill-climb or stay as is. Therefore, the methods are basically local search that cannot escape from locally optimal solutions. We mainly investigated the practical case where the agents are basically greedy but stochastically escape from locally optimal solutions.

6 CONCLUSION

We investigated the decentralized stochastic search methods for Asymmetric Multi-Objective Distributed Constraint Optimization Problems considering the worst cases among the agents. The experimental results show the effect and influence of different optimization criteria and additional information in stochastic local search processes. In particular, the results show that the minimization of the maximum cost value is not straightforward with min-max based criteria that are also designed to improve the global objective value or fairness. Min-max optimization with several additional bits of information performs better perturbation. We believe such an investigation of fundamental search methods for different optimization criteria can provide a foundation for more sophisticated sampling based methods for extended classes of DCOPs. Our future work will investigate other stochastic and sampling-based algorithms with fairness and Pareto optimality.

ACKNOWLEDGEMENTS

This work was supported in part by JSPS KAKENHI Grant Number JP19K12117.

REFERENCES

Fioretto, F., Pontelli, E., and Yeoh, W. (2018). Distributed constraint optimization problems and applications: A survey. *Journal of Artificial Intelligence Research*, 61:623–698.

Mahmud, S., Choudhury, M., Khan, M. M., Tran-Thanh, L., and Jennings, N. R. (2020). AED: An Anytime Evolutionary DCOP Algorithm. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 825–833.

Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395.

Matsui, T., Matsuo, H., Silaghi, M., Hirayama, K., and Yokoo, M. (2018a). Leximin asymmetric multiple objective distributed constraint optimization problem. *Computational Intelligence*, 34(1):49–84.

Matsui, T., Silaghi, M., Okimoto, T., Hirayama, K., Yokoo, M., and Matsuo, H. (2018b). Leximin multiple objective dcops on factor graphs for preferences of agents. *Fundam. Inform.*, 158(1-3):63–91.

Sen, A. K. (1997). *Choice, Welfare and Measurement*. Harvard University Press.

Zhang, W., Wang, G., Xing, Z., and Wittenburg, L. (2005). Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87.

Zivan, R. (2008). Anytime local search for distributed constraint optimization. In *Twenty-Third AAAI Conference on Artificial Intelligence*, pages 393–398.