Layered Distributed Constraint Optimization Problem for Resource Allocation Problem in Distributed Sensor Networks

Kazuhiro Ota, Toshihiro Matsui, and Hiroshi Matsuo

Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Aichi, 466-8555 Japan
ohta@matlab.nitech.ac.jp, {matsui.t,matsuo}@nitech.ac.jp

Abstract. Distributed sensor network is an important research area of multi-agent systems. We focus on a type of distributed sensor network systems that cooperatively observe multiple targets with multiple autonomous sensors that can control their own view. The problem of allocating observation resource of the distributed sensor network can be formalized as distributed constraint optimization problems. However, in the previous works, the computation cost to solve the resource allocation problem highly increases with its scale/density. In this work, we divide the problem into two layers of problems, and two layered cooperative solvers are applied to those problems. The result of the experiment shows that our proposed method reduces the number of message cycles.

Keywords: Distributed Constraint Optimization Problem, Multi-agent, Distributed sensor network.

1 Introduction

Distributed Constraint Optimization Problem (DCOP) is an important research area of multi-agent systems[1][2][3][4]. In DCOPs, agent's state is represented by variables; and, relations between agents are represented by constraints and cost functions. Each agent decides/determines the values of its own variables by exchanging information with other agents. The goal is to assign global optimal values to the variables. DCOPs are an important model that represents cooperated resource scheduling problems in distributed systems. On the other hand, distributed sensor networks are studied as practical problems of multi-agent systems. In previous studies, resource allocation problems of the distributed sensor network are formalized as DCOPs[5][6]. There are various purposes of distributed sensor networks. An important purpose is to give information in a large observation area. Other purposes include cooperative navigation of robots using distributed sensor networks. In this paper, we focus on a type of observation system that cooperatively observes multiple targets with multiple autonomous sensors that can control own view. An important problem in the observation system can be formalized as an allocation problem of observation resources. In this paper, we considered the problem at a snapshot for an initial examination of the proposed method. In the actual system, the environment changes dynamically in situations such as observation targets are moving. Therefore, it is necessary to allocate observation resources responding to the changing environment. That problem can be represented by repeatedly solving a consecutive set of snapshot problems. On the other hand, the time for a snapshot is limited. Therefore it is reasonable to apply stochastic methods that can find solution in comparatively short time. However, the problem including cooperation by agents and the resource allocations is complex. Then, in this paper, we propose a method that divides the problem into two layers: layer of leader election and the layer of observation resource allocation. It is expected that our proposed method reduces the complexity of the problem and efficiently solves the problem.

In section 2, DCOP and a search algorithm for the DCOP are shown. In section 3, we explain how to represent the resource allocation problem in the distributed sensor network. Then, some conventional formalizations of the problem are shown. A model based agency is also shown. In section 4, we propose the method that divides the problem into two layers. In section 5, our proposed method is evaluated. And we conclude in section 6.

2 Distributed Constraint Optimization Problem

DCOP consists of a set of agents. Each agent a_i has some variables $X_i = \{x_i^1, \cdots, x_i^k\}$. x_i^k takes a value from discrete finite domain D_i^k. a_i is the only agent that can decide the values of X_i. That is, the variable shows agent's state and decision. The relation between a set of variables is defined as a constraint c. A cost function f_c defines the cost for a set of variables. f_c is the cost function of c. A cost value represents the degree of violation on constraint c. There are constraints which cannot be relaxed and constraints which can be relaxed. The constraints which cannot be relaxed are defined as hard constraint. The constraints which can be relaxed are defined as soft constraint. A goal of the problem is to find optimal assignments of variables that minimize global cost value.

ADOPT[1] and DPOP[7] have been proposed as exact methods for DCOP. ADOPT performs as distributed version of branch and bound/A$*$ search based on depth first search tree for constraint network. DPOP is based on dynamic programming. In these methods, search iterations or memory uses exponentially increase according to induced-width[7] of the depth first search tree. On the other hand, DSA[2] and DSTS[3] have been proposed as stochastic algorithms. The solution found by these methods may not be optimal. However, these stochastic methods find suboptimal solutions with less number of cycles than ones of exact algorithms. In this work, we apply DSTS to the resource allocation problem in the distributed sensor network.

DSTS is a distributed stochastic search algorithm based on DSA. DSTS employs a tabu search to get out from local optimal solution. In search processing, each agent exchanges the values of its variables. Then each agent a_i calculates costs for assignments. The costs are evaluated for assignments of a_i's variables

```
initialize own variables;
empty tabu_list;
send variables' values to agents related by constraints;
while not terminated do
    receive variables' values from other agents rlated by constraints;
    call maintainance;
end while
procedure maintainance
if all variables' values are in the tabu_list then
    noting to do;
else if NA_VALUEs have been received from all agents related by constraints
    then
    assign new values to variables;
else if $\Delta \geq 0$ then
    assign new values to variables with $p_1$;
else if current cost $> 0$ then
    assign new values to variables with $p_2$;
end if
if all values are in the tabu_list then
    send NA_VALUE to agents related by constraints;
else if new valiables's value is assigned then
    send own variable's value to agents related by constraints;
    add new variable's value to the tabu_list;
end if
end procedure
```

Fig. 1. Pseudo code of DSTS

with values of other agent's variables that are related by constraints. According to the costs, the agent a_i stochastically changes its variable's value to value which obtains best cost with probability p_1. Moreover, each agent uses the tabu search to get out from local optimal solution. Each agent adds variable's value to tabu_list. It prevents the variable from changing its value for a certain term(i.e. tabu period). Improvement of the cost value Δ takes negative value because of tabu search. In that case, each agent changes the values of variables with probability p_2. A pseudo code of DSTS is shown in Fig. 1. In Fig. 1, NA_VALUE represents that all values of a_i's variables are in the tabu_list.

3 Resource Allocation Problems in Distributed Sensor Network

In this section, a model of resource allocation problem for the distributed sensor network is shown. Then we show some formalization for the resource allocation problem based on DCOPs. Another framework based on the concept of agency is also shown.

3.1 Grid Model

Grid model is a basic representation of allocation problem that allocates observation resource of a sensor to a target. In the grid model, sensors are arranged

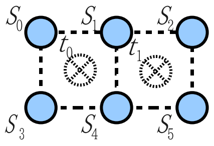


Fig. 2. Example of grid model

on the nodes of a uniform grid. Targets are located within the area enclosed by the grid. In related studies, models like this were used[6][5]. In this work, we focus on a type of observation system that consists of autonomous sensor nodes. That is, a sensor is an agent. In Fig. 2, s_i represents the sensor and t_j represent the target. We assume that only one target can exist in one area.

There is the limitation about the view of the sensor. This restriction cannot be compromised. The limitation about the view is modeled as a constraint such that the sensors can observe adjoining areas. Moreover, the sensors cannot observe multiple targets at a time. This limitation is modeled as a constraint such that the sensors can observe only one target at a time. On the other hand, it is preferable to observe one target with multiple sensors because more information about the target is obtained. This purpose is modeled as a constraint such that each target has to be observed by three sensors. However, this purpose can be relaxed because the targets can be observed by a smaller number of sensors.

3.2 Formalization Based on DCOP

In the following, two DCOP based formalizations for sensor resource allocation problem are shown. These are STAV[1] and TAV[5].

STAV (Sensor-Target As Variable): STAV is a model of formalization which defines a variable for a pair of a sensor and a target. An example of a sensor network shown in Fig. 2 is formalized as a STAV problem shown in Fig. 3. In Fig. 3, $x_{t_j}^{s_i}$ represents a variable of s_i for target t_j. For each sensor s_i, variables are defined for targets that can be observed by s_i. In this example, s_0, s_2, s_3, and s_5 have one variable because they can observe only one target. s_1 and s_4 have two variables because they can observe two targets. A value of $x_{t_j}^{s_i}$ represents which sensors are allocated to t_j. If a set of sensors that

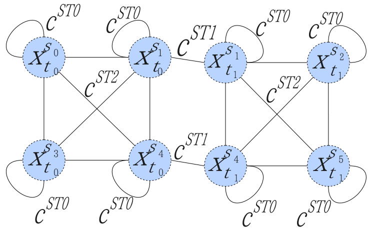


Fig. 3. Constraint network with STAV

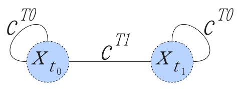


Fig. 4. Constraint network with TAV

can observe t_j is $\{s_0, \cdots, s_n\}$, $x^{s_i}_{t_j}$ takes a combination of sensors as a value. $\{\phi, \{s_0\}, \cdots, \{s_n\}, \{s_0, s_1\}, \cdots, \{s_0, \cdots, s_n\}\}$ is the domain of $x^{s_i}_{t_j}$. With this formalization, three types of constraints are defined, and they are shown in Fig. 3 as c^{ST0}, c^{ST1} and c^{ST2}. Details of the constraints are as follows.

- $c^{ST0}(x^{s_i}_{t_j})$: Allocating sensors to observation target
  This constraint represents a requirement that three sensors are allocated to a target. If the number of sensors allocated to t_j is fewer than three, the constraint is not fully satisfied. In such case, this constraint can be relaxed. The cost function $f_{c^{ST0}}$ for c^{ST0} is defined as follows. $w^{c^{ST0}}$ is the weight parameter which represents degree of violation. In expression (1), a value of n_{t_j} represents the number of sensors allocated to t_j.

$$f_{c^{ST0}(x^{s_i}_{t_j})} = \begin{cases} w^{c^{ST0}}_0 & n_{t_j} = 0 \\ w^{c^{ST0}}_1 & n_{t_j} = 1 \\ w^{c^{ST0}}_2 & n_{t_j} = 2 \\ 0 & otherwise \end{cases} \tag{1}$$

- $c^{ST1}(x^{s_i}_{t_j}, x^{s_i}_{t_{j'}})$: Restriction of observation resource
  This constraint represents a restriction about the number of targets where a sensor can be allocated. If a sensor is allocated to multiple targets, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{ST1}}$ for c^{ST1} is defined as follows. $w^{c^{ST1}}$ is the weight parameter.

$$f_{c^{ST1}(x^{s_i}_{t_j}, x^{s_i}_{t_{j'}})} = \begin{cases} w^{c^{ST1}} & x^{s_i}_{t_j} \cap x^{s_i}_{t_{j'}} \neq \phi \\ 0 & otherwise \end{cases} \tag{2}$$

- $c^{ST2}(x^{s_i}_{t_j}, x^{s_{i'}}_{t_j})$: Consistency of allocation of observation resource
  This constraint represents cooperation that sensors are allocated to targets without contradiction. If the variables' values for the same target are different between sensors, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{ST2}}$ for c^{ST2} is defined as follows. $w^{c^{ST2}}$ is the weight parameter.

$$f_{c^{ST2}(x^{s_i}_{t_j}, x^{s_{i'}}_{t_j})} = \begin{cases} w^{c^{ST2}} & x^{s_i}_{t_j} \neq x^{s_{i'}}_{t_j} \\ 0 & otherwise \end{cases} \tag{3}$$

The variable and the constraint increase because this model represents the explicit agreement between agents.

TAV (Target As Variable): TAV is a model of formalization which defines a variable for a target. An example of a sensor network shown in Fig. 2 is formalized as a TAV problem shown in Fig. 4. In Fig. 4, x_{t_j} represents a variable for target t_j. A value of x_{t_j} represents which sensors are allocated to t_j. If a set of sensors that can observe t_j is $\{s_0, \cdots, s_n\}$, x_{t_j} takes a combination of sensors as a value. $\{\phi, \{s_0\}, \cdots, \{s_n\}, \{s_0, s_1\}, \cdots, \{s_0, \cdots, s_n\}\}$ is the domain of x_{t_j}. With this formalization, two types of constraints are defined. They are shown in Fig. 4 as c^{T0} and c^{T1}. Details of the constraints are as follows.

- $c^{T0}(x_{t_j})$: Allocating sensors to observation target
  This constraint represents a requirement that three sensors are allocated to a target. If the number of sensors allocated to t_j is fewer than three, the constraint is not fully satisfied. In such case, this constraint can be relaxed. The cost function $f_{c^{T0}}$ for c^{T0} is defined as follows. $w^{c^{T0}}$ is the weight parameter which represents the degree of violation. In expression (4), a value of n_{t_j} represents the number of sensors allocated to t_j.

$$f_{c^{T0}(x_{t_j})} = \begin{cases} w_0^{c^{T0}} & n_{t_j} = 0 \\ w_1^{c^{T0}} & n_{t_j} = 1 \\ w_2^{c^{T0}} & n_{t_j} = 2 \\ 0 & otherwise \end{cases} \tag{4}$$

- $c^{T1}(x_{t_j}, x_{t_{j'}})$: Restriction of observation resource
  This constraint represents a restriction about the number of targets where a sensor is allocated. If a sensor is allocated to multiple targets, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{T1}}$ for c^{T1} is defined as follows. $w^{c^{T1}}$ is the weight parameter.

$$f_{c^{T1}(x_{t_j}, x_{t_{j'}})} = \begin{cases} w^{c^{T1}} & x_{t_j} \cap x_{t_{j'}} \neq \phi \\ 0 & otherwise \end{cases} \tag{5}$$

In TAV, variables are defined for targets. It seems that target has a variable. However, in the system assumed in this paper, targets are not agents. It is not clear what variables agents have. Therefore, TAV cannot be applied.

3.3 Cooperation Model with Agency

The observation system by distributed cooperative processing with the agency[8] has been proposed besides the frame of DCOP. An agency is a group of agents. Fig. 5 shows the concept of the agency based cooperative observation system. This system consists of camera agents which can control a view (AVA: Active Vision Agent) and there are some observation targets. Each AVA operates autonomously. The basic characteristic of autonomous camera agents is similar to

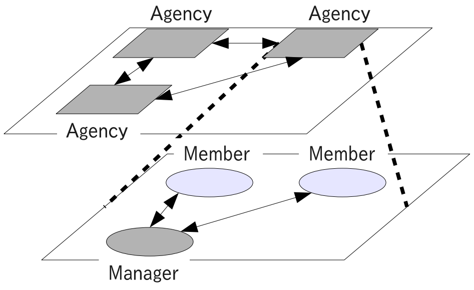


Fig. 5. Cooperation model based on agency[8]

that of sensors as it is assumed in this paper. The outline of this system is as follows.

- Each AVA is an observation resource which can be allocated to a target.
- When AVA detects a target, each AVA makes an agency.
- One of AVAs in a agency performs as a manager(AM:Agency Manager). Other AVAs follows the AM's decision.
- Each AM exchanges information, and decides the distribution of the observation resource.
- Observed information is gathered and managed by each AM.

The efficiency of this system was demonstrated by a small-scale experimental environment with real machines. Therefore, it is thought that use of layered structures for cooperation is more effective. However, in this system, the problem is not formalized as an optimization problem like DCOP, and optimization method in DCOP frameworks is not used.

4 Applying Layered Structure into Formalization by DCOP

In our proposed method, we apply a layered structure into formalization by DCOP. The original resource allocation problem in the distributed sensor network is divided into two problems that represent the leader election problem and the observation resource allocation problem. These two problems are rather easy compared to the original problem. It is thought that this formalization can integrate efficient cooperated operation by agency with flexible problem description with constraint network.

4.1 Layer1: Leader Election Problem

In this layer, some agents are elected as leader of a target. In the following, each sensor is identified with an agent. The leader can be considered as the manager of the agency based cooperation model[8]. In this work, we define a rule that each leader must be allocated to its relative target. An example of a sensor network shown in Fig. 2 is formalized as a leader election problem shown in Fig. 6. In

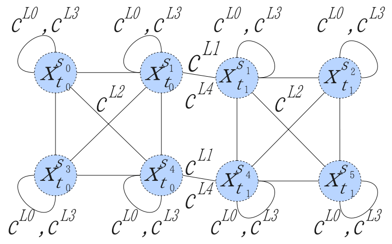


Fig. 6. Leader election problem layer

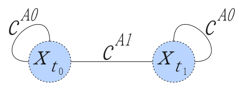


Fig. 7. Resource allocation problem layer

Fig. 6, $x_{t_j}^{s_i}$ represents a variable of s_i for t_j. For each sensor s_i, variables are defined for targets that can be observed by s_i. A value of $x_{t_j}^{s_i}$ represents which sensor is a leader of t_j. If a set of agents that can observe t_j is $\{s_0, \cdots, s_n\}$, the value of $x_{t_j}^{s_i}$ is selected from $\{\phi, s_0, \cdots, s_n\}$. With this formalization, five types of constraints are defined, and they are shown in Fig. 6 as c^{L0}, c^{L1}, c^{L2}, c^{L3} and c^{L4}. Details of the constraints are as follows.

– $c^{L0}(x_{t_j}^{s_i})$: Requirement of a leader for a target
This constraint represents a requirement that a leader is selected for a target. If there is no leader of the target, the constraint is not fully satisfied. This constraint can be relaxed. The cost function $f_{c^{L0}}$ for c^{L0} is defined as follows. $w^{c^{L0}}$ is the weight parameter which represents the degree of violation.

$$f_{c^{L0}(x_{t_j}^{s_i})} = \begin{cases} w_0^{c^{L0}} & x_{t_j}^{s_i} = \phi \\ 0 & otherwise \end{cases} \tag{6}$$

– $c^{L1}(x_{t_j}^{s_i}, x_{t_{j'}}^{s_i})$: Restriction of a leader
This constraint represents that each sensor is allocated to one target as its leader. If a sensor is the leader of multiple targets, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{L1}}$ for c^{L1} is defined as follows. $w^{c^{L1}}$ is the weight parameter.

$$f_{c^{L1}(x_{t_j}^{s_i}, x_{t_{j'}}^{s_i})} = \begin{cases} w^{c^{L1}} & x_{t_j}^{s_i} = x_{t_{j'}}^{s_i} \\ 0 & otherwise \end{cases} \tag{7}$$

– $c^{L2}(x_{t_j}^{s_i}, x_{t_j}^{s_{i'}})$: Consistency of allocation of leader
This constraint represents cooperation that all leaders are elected without contradiction. If the variables' values for the same target are different between sensors, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{L2}}$ for c^{L2} is defined as follows. $w^{c^{L2}}$ is the weight parameter.

$$f_{c^{L2}(x_{t_j}^{s_i}, x_{t_j}^{s_{i'}})} = \begin{cases} w^{c^{L2}} & x_{t_j}^{s_i} \neq x_{t_j}^{s_{i'}} \\ 0 & otherwise \end{cases} \tag{8}$$

– $c^{L3}(x_{t_j}^{s_i})$: Ensuring allocation candidate 1
This constraint represents a requirement which ensure allocation candidates. Because the leader has to observe the target where the leader has been allocated as mentioned above, the leader is excluded from the allocation candidate of other targets. That may prevent fair sensor resource allocation. For that reason, there are situations where it is impossible to allocate a sufficient number of sensors to a target. To avoid this situation, the number of targets which the leader can observe should be small. If the number of targets that the leader can observe is not minimum value in candidate sensors, the constraint is not fully satisfied. This constraint can be relaxed. The cost function $f_{c^{L3}}$ for c^{L3} is defined as follows. $w^{c^{L3}}$ is the weight parameter. In

expression (9), S_{t_j} represents that set of sensors which can observe t_j and v_i represents the number of targets which can be observed by s_i.

$$f_{c^{L3}(x^{s_i}_{t_j})} = \begin{cases} w^{c^{L3}} & \exists s \in S_{t_j}, v_s < v_{x^{s_i}_{t_j}} \\ 0 & otherwise \end{cases} \tag{9}$$

– $c^{L4}(x^{s_i}_{t_j}, x^{s_i}_{t_{j'}})$: Ensuring allocation candidate 2
This constraint represents a requirement which ensure allocation candidates. This is a constrain for avoiding situations where areas that can be observed by leaders are overlapped. Especially, in the grid model, each sensor can observe its four adjoining areas. If two or more adjoining areas of a leader overlap with another leader's adjoining areas, the constraint is not fully satisfied. This constraint can be relaxed. The cost function $f_{c^{L4}}$ for c^{L4} is defined as follows. $w^{c^{L4}}$ is the weight parameter. In expression (10), R_{s_i} represents set of areas that can be observed by s_i.

$$f_{c^{L4}(x^{s_i}_{t_j}, x^{s_i}_{t_{j'}})} = \begin{cases} w^{c^{L4}} & |R_{x^{s_i}_{t_j}} \cap R_{x^{s_i}_{t_{j'}}}| > 1 \\ 0 & otherwise \end{cases} \tag{10}$$

The number of variables in the leader election problem is the same as STAV. However, in this problem, each variable's domain contains five values because there are four agents which can observe each target. In STAV, each variable's domain contains 16 values. Therefore, the leader election problem is easier than STAV. On the other hand, this model can be considered that agents decide which agents own variables in TAV.

4.2 Layer2: Observation Resource Allocation Problem

In this layer, the observation resource allocation problem is solved by leaders. The leaders exchange information and solve the problem. Agents which are not leader follow leaders' decisions. Because one leader is elected each target in the layer of leader election problem, an example of a sensor network shown Fig. 2 is formalized as an observation resource allocation problem shown in Fig. 7. In Fig. 7, x_{t_j} is a variable of leader of t_j. A value of x_{t_j} represents which sensors are allocated to t_j. If a set of sensors that can observe t_j is $\{s_0, \cdots, s_n\}$, x_{t_j} takes a combination of sensors as a value. $\{\phi, \{s_0\}, \cdots, \{s_n\}, \{s_0, s_1\}, \cdots, \{s_0, \cdots, s_n\}\}$ is the domain of x_{t_j}. With this formalization, two types of constraints are defined, and they are shown in Fig. 7 as c^{A0} and c^{A1}. Details of the constraints are as follows.

– $c^{A0}(x_{t_j})$: Allocating sensors to targets
This constraint represents a requirement that three sensors are allocated to a target. If the number of sensors allocated to t_j is fewer than three, the constraint is not fully satisfied. In such case, this constraint can be relaxed. The cost function $f_{c^{A0}}$ for c^{A0} is defined as follows. $w^{c^{A0}}$ is the weight parameter which represents degree of violation. In expression (11), a value of n_{t_j} represents the number of sensors allocated to t_j.

$$f_{c^{A0}(x_{t_j})} = \begin{cases} w_0^{c^{A0}} & n_{t_j} = 0 \\ w_1^{c^{A0}} & n_{t_j} = 1 \\ w_2^{c^{A0}} & n_{t_j} = 2 \\ 0 & otherwise \end{cases} \tag{11}$$

- $c^{A1}(x_{t_j}, x_{t_{j'}})$: Restriction of observation resource
  This constraint represents a restriction about the number of targets to which a sensor is allocated. If a sensor is allocated to multiple targets, the constraint is violated. This constraint cannot be relaxed. The cost function $f_{c^{A1}}$ for c^{A1} is defined as follows. $w^{c^{A1}}$ is the weight parameter.

$$f_{c^{A1}(x_{t_j}, x_{t_{j'}})} = \begin{cases} w^{c^{A1}} & x_{t_j} \cap x_{t_{j'}} \neq \phi \\ 0 & otherwise \end{cases} \tag{12}$$

Because the representation of the constraint network of this layer is similar to using TAV, the number of variables in this layer is fewer than the number of variables in using STAV. So the constraint network is sparser. Moreover, because the leader must observe at a target, the number of values that can be taken by variables is reduced. Therefore, it is thought that the problem in this layer is solved easily.

4.3 Synchronization between Two Layers

It is necessary to synchronize between two layers in the proposed method. On the other hand, each agent needs only information of agents which are related by constraints. In other words, each agent can solve own problem by receiving information from agents which relate by constraints. In the proposed method, each agent elected as a leader sends messages to agents which have the possibility of relating by constraint on a resource allocation problem. In this way, each agent can realize which agent is a leader and shift to solving the resource allocation problem. Conditions to judge that s_i has been selected as a leader are defined as follows.

- The agent is a leader for one of targets: if J is a set of targets which can be observed by s_i, this condition is defined as follows.

$$l_1 = \begin{cases} true & \exists t_j \in J, x_{t_j}^{s_i} = s_i \\ false & otherwise \end{cases} \tag{13}$$

- All of hard constraints of s_i in leader election problem are satisfied: if C is a set of hard constraints of s_i in leader election problem, this condition is defined as follows.

$$l_2 = \begin{cases} true & \forall c \in C, f_c = 0 \\ false & otherwise \end{cases} \tag{14}$$

If $l_1 \wedge l_2 = true$, the agent is a leader and sends messages to other agents. When the problem has not been globally solved, there is a possibility that $l_1 \wedge l_2$ become

```
 1 initialize variables;
 2 is_leader ← false;
 3 empty leader_list;
 4 **while** not terminated **do**
 5 previous_status ← is_leader;
 6 receive messageses from other agents related by constraints;
 7 clear current_status;
 8 if new leader has been elected then
 9     add new leader to leader_list;
10     store the information of new leader into current_status;
11 end if
12 if elected leader came off then
13     remove old leader from leader_list;
14     store the information of removed leader into current_status;
15 end if
16 call maintenance of DSTS for solving leader election;
17 if previous_status = $false$ then
18     if $l_1 \wedge l_2 = true$ then
19         is_leader ← $true$;
20         store is_leader into current_status;
21 end if
22 if previous_status = $true$ then
23     if $l_1 \wedge l_2 = false$ then
24         is_leader ← $false$;
25         store is_leader into current_status;
26     end if
27 end if
28 if is_leader = $true$ then
29     call maintenance of DSTS for resource allocation problem
30 end if
31 send current_status and variables' values to agents related by the constraints.
32 end while
```

Fig. 8. Pseudo code of the proposed method

$false$ again. At that time, the agent send message about $l_1 \wedge l_2 = false$ to the others. In proposed method, the agent send messages to other agents which are related by constraints, instead of sending to all agents. That reduces message passing cost. However, non-neighborhood agents on the grid can be related by constraints. Considering that case, each agent propagates the message. In the grid model, each agent has to propagate message only one hop. A pseudo code of the proposed method is shown in Fig. 8. In Fig. 8, is_leader represents whether the agent is a leader. leader_list represents list of leaders. Information about the election of leaders is stored in current_status. "maintenance" procedures in Fig. 1 are performed for each layer. As shown in the line 31 of Fig. 8, a couple of current_status and variables' values of two layers of problems are sent at a same time.

5 Experiments

We compared the proposed methods with previous method using STAV and evaluated the efficiency of dividing the problem into two problems which can be solved comparatively easily. The previous method is shown as STAV, and

Table 1. Parameters of DSTS

	p_1	p_2	tabu period
STAV	0.8	0.4	1 cycle
LYR			
Layer1	0.9	0.3	2 cycles
Layer2	0.7	0.2	1 cycle

Table 2. Weight parameters

STAV		LYR			
		Layer1		Layer2	
c^{ST0}	$w_0^{c^{ST0}} = 15$	c^{L0}	$w^{c^{L0}} = 10$	c^{A0}	$w_0^{c^{A0}} = 15$
	$w_1^{c^{ST0}} = 5$	c^{L1}	$w^{cL1} = 200$		$w_1^{c^{A0}} = 5$
	$w_2^{c^{ST0}} = 1$	c^{L2}	$w^{c^{L2}} = 100$		$w_2^{c^{A0}} = 1$
c^{ST1}	$w^{c^{ST1}} = 200$	c^{L3}	$w_0^{c^{L3}} = 1$	c^{A1}	$w^{c^{A1}} = 200$
c^{ST2}	$w^{c^{ST2}} = 100$	c^{L4}	$w^{c^{L4}} = 1$		

the proposed method is shown as LYR. While several algorithms are applied to similar problem[5]. We applied DSTS that is a extended version of DSA in both methods. In LYR, we applied DSTSs to two layers of problems shown in Fig. 8. Original DSTS does not have termination detection mechanism. Therefore we apply a simple rule to DSTS for termination. We define the suboptimal solution as a solution which satisfies all hard constraints. We also evaluated the cases that any soft constraints are not optimized. In other words, the weight parameters of soft constraints are 0. Moreover, in LYR, we evaluated the cases that the soft constraints are not optimized in resource allocation problem. The experiment aims to evaluate impact of optimality of leader election problem to global optimality. Those methods are shown as follows.

– STAV: Previous method that all constraints are considered in optimization.
– STAV-NoOpt: Previous method that any soft constraints are not optimized.
– LYR: Proposed method that all constraints are considered in optimization.
– LYR-NoOpt: Proposed method that any soft constraints are not optimized.
– LYR-NoOptInAllo: Proposed method that any soft constraints relaxed in resource allocation problem are not optimized.

The experiments are performed using simulation programs. The simulation iterates cycles of globally synchronized processing shown as follows.

(1) Each agent receives messages and processes the local part.
(2) Each agent sends messages to the other agents if necessary.

We evaluated the number of cycles when a solution is found, distance of cost of suboptimal solution and cost of optimal solution, the number of sensors allocated to a target, and the number of messages. The maximum number of cycles for a trial is limited to 1000. If a solution could not be found within the 1000 cycles, the number of cycles is considered as the upper limit value. In such case, the result of the trial is not included in other evaluations. We prepare four problem classes. There are ten instances in each class. The result of a class is the average of result of all instances in the class. For each instance 1000 trials are performed.

5.1 Parameters of DSTS and Weight Parameters of Constraints

Parameters of DSTS are shown in Table 1. The parameters are decided according to the best results of preliminary experiments. Weight parameters of DSTS are shown in Table 2. The total degrees of violations of soft constraints must be smaller than a weight parameter of a hard constraint. Moreover, the weight parameters of hard constraints are set according to the total number of constraints. If the type of hard constraints is mostly contained in problem instances, relatively small weight value is set for the constraint. That aims to keep a balance on total weight parameters. The balance is expected to reduce number of search iterations because it reduces cases of local optima. In leader election problem, in order to elect a leader in all targets as much as possible, w^{L0} is bigger than w^{L3} and w^{L4}.

5.2 Change Conditions of Variable's Value and Termination Rule

In addition to the above parameter settings, we modified change conditions of variable's value of DSTS. The modifications aim to obtain the best result for each method. And as shown above, we have applied termination rule to DSTS.

– Condition to take other values of variables
  In leader election problem of LYR, the variable's value is changed with probability p_1 if $\Delta > 0$. On the other hand, in STAV and resource allocation problem of LYR, the variable's value is changed with probability p_1 if $\Delta \geq 0$. This condition increases frequency to get out from local optimal because it increases neighborhood solutions that can be selected as the locally best solution.
– Condition to find the solution early
  Each agent changes the variable's value with p_2 if $\Delta < 0$ when the agent has a violation for hard constraints. This condition is applied to STAV and LYR. By this condition, each agent does not change the values if $\Delta < 0$ when the agent satisfied the all of the hard constraints. It aims to find suboptimal solution in fewer number of cycles.
– Termination rule to decide a suboptimal solution
  If each agent changes the variable's value when $\Delta \geq 0$, the agent cannot decide a solution because there is no rule to choose one solution from the set of solutions which have same degree of violation . Therefore, when the agent satisfied all of the hard constraints, the agent changes the variable's value with probability p_1 if $\Delta > 0$. This rule is applied to DSTS of STAV and LYR. By this rule, each agent does not change variable's value even if there are some solutions which have same degree of violation.

5.3 Problem Settings

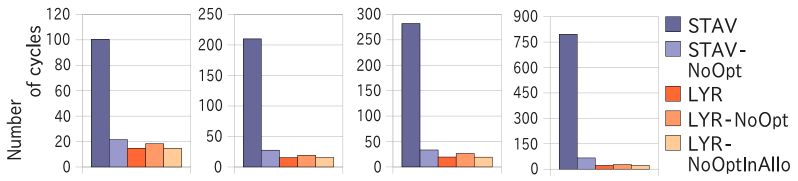
We prepared four classes of problem. Each problem is generated according to t and n as parameters. t decides the number of targets. n decides the limitation number of targets which exist in each target's adjoining area. We used the parameters $t = 5, 10$ and $n = 2, 3$. Problems are generated as follows.

(1) The first target is placed on an area at random. (2) The next candidate areas to place a new target are the empty adjoining areas of target. (3) The area where

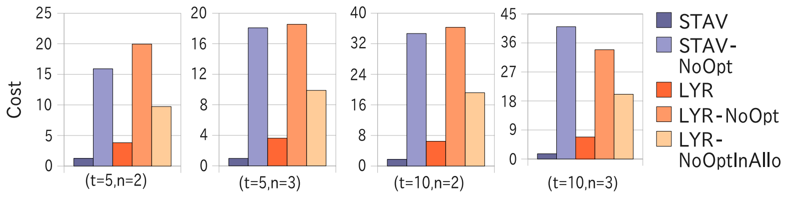
the number of adjacent targets is more than n is excluded from the candidates. (4) The new target is placed into an area that is randomly selected from the candidate areas. (5) Repeat 2 to 4 until the number of target becomes t.

5.4 Results

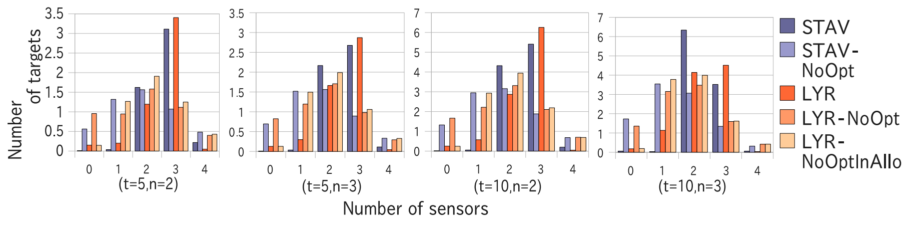
The number of cycles until finding suboptimal solution are shown as Fig. 9(a). The results show that LYR needs less number of cycles. Main reason of the result is that complexity of the problem is reduced by dividing the problem into two layers. In the results of STAV-NoOpt, less number of cycles is required, when compared to STAV. It is a reasonable result because optimization problem is more difficult than satisfaction problem.



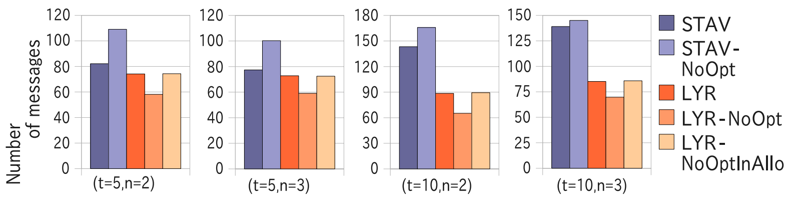
(a) Number of cycles



(b) Distance from optimal solution



(c) Number of sensors allocated to a target



(d) Number of messages

Fig. 9. Results

The difference of cost between suboptimal solution and optimal solution are shown as Fig. 9(b). The result shows that LYR's cost is larger than STAV's cost. A reason of this drawback is local optimal solution in leader election problem. Such local optimal solution causes a situation that some targets are ignored in sensor resource allocation by relaxing soft constraints. In addition, number of sensors which can be allocated for each target is often disproportional in sensor resource allocation layer. That is caused by a bias of greedy decision in leader election. There might be targets which can not be allocated enough number of sensors according to the leader's arrangement. Costs of STAV-NoOpt, LYR-NoOpt and LYR-NoOptInAllo are relatively large, because these methods do not optimize soft constraints as mentioned above.

The number of sensors allocated to a target are shown as Fig. 9(c). The number of targets with no sensors in LYR is more than in STAV. However, this difference is relatively small. In LYR, the number of sensors allocated to a target is disproportional due to the bias in leader election. In the case of STAV-NoOpt, LYR-NoOpt and LYR-NoOptInAllo, the number of sensors allocated to targets is relatively few.

The number of messages per cycle are shown as Fig. 9(d). Each agent sends messages to other agents which are related by constraints. However, each agent does not have to send messages if its variables' value are not changed. Therefore, in the leader election of LYR, each agent that has been found a local suboptimal solution does not send the message. Leaders have to send message of resource allocation problem. In resource allocation problem, the number of messages sent by leader is fewer than the number of messages in leader election. The reason of less number of messages is that only leader agents are related with constraints in the resource allocation problem. In addition, the messages for synchronization are not sent if it is not necessary. Therefore, the number of messages per cycle is reduced by LYR. On the other hand, in the later period of search, the number of agents that have been found local suboptimal solution is increased. That relatively decreases average number of messages in the methods that need much number of cycles. Therefore, in STAV and LYR-NoOpthe, the number of messages per cycle is comparatively few.

In LYR, although the number of cycles is significantly reduced, the number of sensors allocated to a target is disproportional. However, we think this type of trade off can be acceptable in some actual systems that need fast reasons. On the other hand, the number of sensors allocated to target by LYR is more than that by STAV-NoOPt, LYR-NoOPt and LYR-NoOptInAllo. Therefore LYR is more effective than those methods.

6 Conclusions

In this paper, we proposed a DCOP based cooperative resource allocation method for distributed sensor network systems. In the proposed method, the problem is divided into two layers of sub-problems: leader election and sensor resource allocation. Then a stochastic DCOP solvers is applied to each layer of problems. The solvers cooperate with partial synchronization. Experimental result shows that proposed method significantly reduces number of cycles in distributed search processing. It is efficient to divide the complex problem into two or more comparatively easy problems. On the other hand, the number of sensors allocated to a target is disproportional due to bias of leader election problems. More detailed analysis, applying other previous algorithms including DSA[2], improving design of constraints and search strategy, and applying to practical observation systems are included in future works.

References

1. Modi, P.J., Shen, W.M., Tambe, M., Yokoo, M.: ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees. Artifi. Intell. 161, 149–180 (2005)
2. Zhang, W., Wang, O., Wittenburg, L.: Distributed stochastic search for constraint satisfaction and optimization: Parallelism, phase transitions and performance. In: AAAI Workshop on Probabilistic Approaches in Search, pp. 53–59 (2002)
3. Iizuka, Y., Suzuki, H., Takeuchi, I.: Multi-agent tabu search method for distributed constraint satisfaction problems (in japan). The IEICE Transactions on Information and Systems J90-D(9), 2302–2313 (2007)
4. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.R.: Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In: AAMAS, pp. 639–646 (2008)
5. Bejar, R., Domshilak, C., Fernandez, C., Gomes, C., Krishnamachari, B., Selman, B., Valls, M.: Sensor networks and distributed csp. Artif. Intell. 161, 117–147 (2005)
6. Ali, S., Koenig, S., Tambe, M.: Preprocessing techniques for accelerating the DCOP algorithm ADOPT. In: AAMAS, July 2005, pp. 1041–1048 (2005)
7. Petcu, A., Faltings, B.: A scalable method for multiagent constraint optimization. In: IJCAI, August 2005, pp. 266–271 (2005)
8. Ukita, N.: Real-time Dense Communication among Agents for Active Tracking. In: AAMAS, July 2005, pp. 1335–1336 (2005)