A Distributed Cooperative Model for Resource Supply Networks

Toshihiro Matsui and Hiroshi Matsuo

***Abstract*—Resource allocation problems are an important domain of distributed cooperative problem solving. Such problems have a dedicated representation of resource allocation and need appropriate solvers that can be applied to them. As an approach to handling these problems, formalizations of distributed constraint optimization problems (DCOPs) can be applied. We propose a distributed cooperative model motivated by power supply networks that contain distributed power sources. The model is represented as a DCOP. The optimal solution of the problem represents the appropriate assignment of amounts of the resource that are consumed or supplied in each node of the network. A conventional algorithm is modified to apply to the problem. Behaviors of the proposed model and the solver are experimentally evaluated.**

***Index Terms*—multiagent, distributed problem solving, cooperation, smart grid, resource allocation.**

I. Introduction

Resource allocation problems are an important domain of distributed cooperative problem solving. The problems have dedicated representation of resource allocation and need appropriate solvers that can be applied to them.

As an approach to handling these problems, formalizations of distributed constraint optimization problems (DCOPs) can be applied. DCOPs [1], [2], [3], [4] have been studied as a basic framework of cooperative problem solving in multiagent systems. With DCOPs, the states of agents and the relationships between agents are formalized into a constraint optimization problem that is solved by distributed search algorithms. These studies focus on the optimization problems and the distributed search algorithms that are essentially contained in cooperative protocols of the multiagent systems.

Several cooperative problems including distributed resource scheduling and sensor networks are represented as DCOPs [4], [5]. The representation of DCOPs can be extended to meet a particular problem. In that case, a solver also has to be modified for the problem.

Similarly, several problems in the power supply network of a smart grid system can be considered as distributed resource allocation problems. In [6], a dedicated representation of DCOP and a solver for a problem of power supply restoration [7] have been proposed. On the other hand, the optimization of consumption and supplement of power on a network that contains distributed power sources is an important problem.

We propose a distributed cooperative model that is motivated by power supply networks that contain distributed power sources. The model is represented as a DCOP. The optimal solution of the problem represents the appropriate assignment of amounts of the resource that are consumed or supplied in each node of the network. A conventional algorithm is modified to be applied to the problem. Behaviors of the proposed model and the solver are experimentally evaluated.

The rest of our paper is organized as follows. In Section II, we address a resource allocation problem in a power supply network and define an example problem. The problem is formalized as a DCOP in Section III. Then, a solver is applied to the DCOP in Section IV. The model and solver are experimentally evaluated in Section V and discussed in Section VI. In Section VII, we conclude our study.

This work was supported in part by a Grant-in-Aid for Young Scientists (B), 22700144 and a grant from the Okawa Foundation for Information and Telecommunications..

T. Matsui and H. Matsuo are with the Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan. e-mail: {matsui.t, matsuo}@nitech.ac.jp.

II. Resource allocation problem in a power supply network

In this work, we consider an example problem motivated by power supply networks. As shown in Figure 1(a), a power distribution network can be modeled as a simple network that consists of power sources, sinks, and power lines.

For the network, several feeder trees that are rooted at the power sources are selected to supply power resources to sinks (Figure 1(b)). Our main focus is how to determine the amount of imported/exported resource in each node of the network. Therefore, it is assumed that a feeder tree has been built using other methods.

The network consists of the following elements.

- Nodes: consumers of the resource that have the option of supplying its resource to other nodes.
- Source node: a special power source that provides the resource to other nodes.
- Links: paths that transfer some of the resource.

There is one source node in the feeder tree. Basically, the resource is provided from the source node to other nodes. Additionally, several nodes have a certain amount of the resource that can be shared through the links. Therefore, the amounts of resource coming from each node have to be determined.

When a node exports an amount of its resource, the node obtains an amount of utility. On the other hand, when the node imports an amount of resource, the node incurs a cost. Moreover, additional incentive that affects utility and cost can be considered.

Each link transfers an amount of resource between two nodes. There are limitations on the maximum amount of resource. When an amount of the resource is transferred, a part of the resource is consumed in the link. In the following, we describe the details of the problem.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

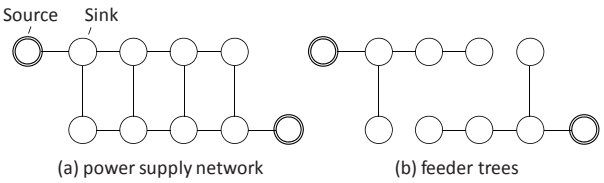


Fig. 1. Example of feeder tree

A. Modeling of Nodes

To represent the requirements of node i, we use the following parameters:

- P_i^{ch}: Hard requirement for consumption of resource.
- P_i^{cs}: Soft requirement for consumption of resource.
- P_i^{g}: Maximum amount of resource that can be supplied.

P_i^{ch} and P_i^{cs} represent requirements of resource. P_i^{ch} is a hard requirement that must be satisfied. It models the baseline of the consumption. We prefer not to use dedicated constraints to represent the hard requirement. Instead, the constraints are implicitly contained in other expressions. In contrast to P_i^{ch}, requirement P_i^{cs} may not be satisfied. It models selectable consumption of resource. To represent an incentive to consume P_i^{cs}, we define a negative cost (i.e. utility) value.

P_i^g represents the amount of i's resource. The resource can partially be consumed by itself, supplied to other nodes via links, or abandoned. As an incentive to supply the resource, a utility is defined.

Amounts of resource that are consumed and supplied by i are represented using variables as follows.

- p_i^c: Amount of consumption.
- p_i^{gi}: Amount of supplement to itself.
- p_i^{li}: Amount of supplement from other nodes.
- p_i^{lo}: Amount of supplement to other nodes.
- p_i^{gw}: Amount of resource unused.

Each value takes a positive value. p_i^c represents i's consumption. Its domain is defined as $[P_i^{ch}, P_i^{ch} + P_i^{cs}]$. As shown above, P_i^{ch} always has to be consumed because it is a hard requirement. Additionally, we assume P_i^{cs} can be divided.

p_i^{gi} represents i's supplement to itself. Clearly, p_i^{gi} takes a value from $[0, \min(P_i^g, P_i^{ch} + P_i^{cs})]$. There are no costs or utilities for p_i^{gi} because the resource is supplied by itself.

p_i^{li} and p_i^{lo} represent amounts of imported or exported resource. To represent purchase of the resource, a cost is defined for p_i^{li}. Similarly, a utility is defined for p_i^{lo} to represent selling.

p_i^{gw} represents wasted resource. The wasted resource can be caused by limitations in transferring the resource.

Those variables have several dependencies. p_i^{li} and p_i^{lo} are constrained not to take a nonzero value at the same time:

$$\neg(p_i^{li} > 0 \wedge p_i^{lo} > 0) \tag{1}$$

That means that each node chooses importing or exporting. p_i^c is equal to total supply for i. That is represented as:

$$p_i^c = p_i^{gi} + p_i^{li} \tag{2}$$

Summation of supplies and the wasted resource is equal to the maximum amount of resource:

$$p_i^{gi} + p_i^{lo} + p_i^{gw} = P_i^g \tag{3}$$

As described above, there are no costs for p_i^{gi}. Therefore, own resource P_i^g is supplied for the requirement p_i^c as much as possible.

By these dependencies, values of the variables are categorized based on P_i^g and p_i^c:

- $P_i^g > p_i^c$: $p_i^c = p_i^{gi}$, $p_i^{li} = 0$, $p_i^{lo} + p_i^{gw} = P_i^g - p_i^c$
- $P_i^g < p_i^c$: $p_i^c = P_i^g + p_i^{li}$, $p_i^{lo} = 0$, $p_i^{gw} = 0$
- $P_i^g = p_i^c$: $p_i^{li} = 0$, $p_i^{lo} = 0$, $p_i^{gw} = 0$

The third case can be generalized with other cases. In the optimization method, we indirectly determine the supplement/consumption of the resource through potential values of the resource.

B. Source node

The source node supplies or acquires the resource. We assume that the source node has a sufficiently large capacity for resource. The amount of the resource supplied or acquires is defined by the following parameters.

- $P^{s\perp}$, $P^{s\top}$: minimum and maximum amounts of resource.

$P^{s\perp}$ can take a negative value. In that case, the source node can absorb the resource.

C. Modeling of transferring resource

The transferring of resource is represented as a simple model motivated by power supply lines. In the model, we use the following parameters and variables,

- p_i: an amount of the resource coming from node i.
- v_i: the potential to transfer the resource.
- $V_i^{\perp}, V_i^{\top}$: the minimum and maximum values of v_i.
- $G_{i,j}$: a parameter that defines an amount of resource transferred through the link between node i and j.
- $V_{i,j}^{dif\top}$: a parameter that defines the maximum amount of resource transferred through the link between node i and j.

p_i represents the amount of resource coming from node i. When p_i represents a consumption of the resource, it takes a negative value. Using variables of node i, the value of p_i is defined as:

$$p_i = p_i^{lo} - p_i^{li} \tag{4}$$

v_i represents the potential to transfer the resource. The value of v_i must not exceed $V_i^{\perp}$ and $V_i^{\top}$. The current coming from node i is represented as p_i/v_i.

$G_{i,j}$ defines an amount of resource transferred through the link between node i and j. The current moving from j to i is represented as $G_{i,j} \cdot (v_j - v_i)$. Summation of all coming currents is zero.

$V_{i,j}^{dif\top}$ defines the maximum amount of resource transferred through the link between nodes i and j. $|v_j - v_i|$ must not exceed $V_{i,j}^{dif\top}$.

D. Costs and utilities

When an amount of resource are imported or exported, cost and utility are calculated. They are used to evaluate the assignment of the resource. Parameters for the cost and the utility are as follows.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- w_i^{inp} : cost value for a unit amount of the resource imported to node i.
- w_i^{exp} : utility value for a unit amount of the resource exported from node i.
- w_i^{utl} : utility value for a unit amount of the resource consumed for a part of P_i^{cs} of node i.

Those parameters are multiplied to an amount of relative resource. Then, the cost and utility values are combined.

III. Formalization as DCOP

To determine the amount of the supply/consumption of the resource, we formalize the problem as a distributed constraint optimization problem (DCOP). In the following, the definition of DCOP and a formalization of the resource sharing problem are shown.

A. Distributed constraint optimization problem

Here is the fundamental definition of distributed constraint optimization problems. A problem is defined by set A of agents, set X of variables, set D of domains of variables, set C of binary constraints, and set F of binary functions.

Agent i has its own variable x_i that takes a value from discrete finite domain D_i. The value of x_i is controlled by agent i. Constraint $c_{i,j}$ represents the relationship between x_i and x_j. The cost of assignment $\{(x_i, d_i), (x_j, d_j)\}$ is defined by binary function $f_{i,j}(d_i, d_j)$. The goal is to find global optimal solution $\mathcal{A}$ that minimizes the global cost function: $\sum_{f_{i,j} \in F,\ \{(x_i,d_i),(x_j,d_j)\} \subseteq \mathcal{A}} f_{i,j}(d_i, d_j)$.

Agent i knows the constraints and the cost functions that are related to x_i. The search process to find the optimal solution is represented as a distributed algorithm based on message communication between agents. In the above descriptions, binary cost functions are used. They can be generalized to n-ary functions including unary functions.

B. Formalization of problem

We optimize each value v_i of potential, which is mapped into variable x_i in DCOP. The supply/consumption of the resource is indirectly determined through v_i. While the value of v_i is continuous, we assume it takes a value from discrete values. Therefore, the solution represents approximate values. Following parameters are used to define the relationship between v_i and x_i.

- v_i^{unit}: unit quantity of the value of v_i.
- $x_i^{\perp}, x_i^{\top}$: minimum and maximum value of x_i.

Using the above parameters, v_i is represented as follows:

$$v_i = 1 + v_i^{unit} \cdot x_i \tag{5}$$

where value 1 is the standard value of v_i. x_i takes an integer value from $[x_i^{\perp}, x_i^{\top}]$. $x_i^{\perp}, x_i^{\top}$ are determined considering the permissible range of v_i. To improve accuracy of solution, v_i^{unit} should take as small a value as possible. On the other hand, the small value of v_i^{unit} needs a large number of discrete values of x_i.

When values of x_i and neighborhood variables are given, corresponding values of potential are determined. Using pairs of v_i and v_j of each neighborhood j, coming current excluding the current of node i itself is computed as follows:

$$c_i = \sum_{j \in \text{ neighborhood nodes of } i} G_{i,j} \cdot (v_j - v_i) \tag{6}$$

The amount p_i of resource supplied from node i is computed as $-v_i \cdot c_i$. By the condition shown in Expressions 1 and 4, p_i^{lo} and p_i^{li} are determined from p_i. If p_i is a positive value, $p_i = p_i^{lo}$. Otherwise $p_i = -p_i^{li}$. On the other hand, p_i is represented as follows:

$$p_i = P_i^g - p_i^c - p_i^{gw} = P_i^g - (P_i^{ch} + k_i^{cs} \cdot P_i^{cs}) - p_i^{gw} \tag{7}$$

where k_i^{cs} is a coefficient that takes a value from $[0, 1]$. In the case where $P_i^g \leq p_i^c$, p_i^{gw} equals zero. Otherwise, p_i^{gw} can take a non-zero value. We assume that each node wastes its own resource only if it can neither be consumed nor exported. By this assumption, Expression 7 is categorized as the following cases:

- $p_i \leq 0$: $p_i = P_i^g - (P_i^{ch} + k_i^{cs} \cdot P_i^{cs})$, $p_i^{gw} = 0$
- $p_i > 0$:
  - $p_i < P_i^g - (P_i^{ch} + P_i^{cs})$: $k_i^{cs} = 1$, $p_i^{gw} = P_i^g - (P_i^{ch} + P_i^{cs}) - p_i$
  - otherwise: $p_i = P_i^g - (P_i^{ch} + k_i^{cs} \cdot P_i^{cs})$, $p_i^{gw} = 0$

As a result, a value of $k_i^{cs} \cdot P_i^{cs}$ is determined from p_i.

Using cost values and utilities of the resource, a cost function f_i^{cost} for node i is defined. f_i^{cost} is represented by a combination of cost/utility functions. The cost/utility for importing/exporting the resource is shown as:

$$f_i^{inp}(p_i^{li}) = w_i^{inp} \cdot p_i^{li} \tag{8}$$

$$f_i^{exp}(p_i^{lo}) = -w_i^{exp} \cdot p_i^{lo} \tag{9}$$

The utility for $k_i^{cs} \cdot P_i^{cs}$ is shown as follows:

$$f_i^{utl}(k_i^{cs} \cdot P_i^{cs}) = \\ -w_i^{utl} \cdot (\lfloor k_i^{cs} \cdot P_i^{cs} / (P_i^{cs}/k^{csunit}) \rfloor \cdot (P_i^{cs}/k^{csunit})) \tag{10}$$

where k^{csunit} is a parameter that defines steps of the utility. The steps of the utility value represent units of requirements.

Two types of constraints are represented as cost functions. The constraint for the limitation of the difference between v_i and v_j is defined as follows:

$$f_{i,j}^{vdif}(v_i, v_j) = \begin{cases} 0 & |v_i - v_j| \leq V_{i,j}^{dif\top} \\ \infty & \text{otherwise} \end{cases} \tag{11}$$

The constraint for the limitation of the value of p_i is defined as follows:

$$f_i^p(p_i) = \begin{cases} 0 & p_i \text{ satisfies the limitation} \\ \infty & \text{otherwise} \end{cases} \tag{12}$$

where the limitation of p_i is categorized as

- $P_i^g - (P_i^{ch} + P_i^{cs}) > 0$: $0 \leq p_i \leq P_i^g - P_i^{ch}$
- otherwise: $P_i^g - (P_i^{ch} + P_i^{cs}) \leq p_i \leq P_i^g - P_i^{ch}$

In this work, we define f_i^{cost} as summation of $f_i^{inp}, f_i^{exp}, f_i^{utl}, f_{i,j}^{vdif}$ and f_i^p. The optimal assignment is defined as a solution that minimizes the summation of all cost/utility functions.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

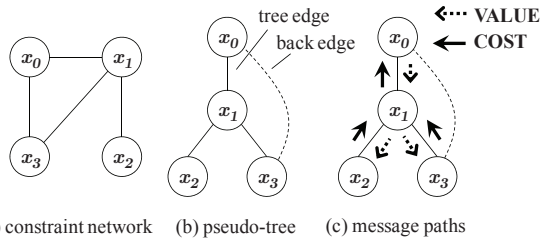


Fig. 2. Pseudo-tree and messages

IV. Solution method

We apply a variation of dynamic programming to the proposed problem. The solution method is based on the pseudo-tree that is a graph structure on constraint networks. To handle properties of the resource, representation and computation of conventional methods are modified.

In the following, the pseudo-tree and conventional computation based on the pseudo-tree are shown. Then modification of the computation, preprocessing and optimization processing are shown.

A. Pseudo-tree

A pseudo-tree [3], [8], [9], which is a graph structure that defines a partial order on variables, is based on a spanning tree of the constraint network. A typical pseudo-tree is generated using a depth-first traversal of a constraint network. For example, the pseudo-tree in Figure 2 (b) is generated from the constraint network in Figure 2 (a).

In the pseudo-tree, the edges of the original constraint network are categorized into either tree edges or back edges. The tree edges are the edges of the spanning tree. The other edges are back edges. The tree edges represent the partial order relation between the two variables. We consider the tree edges of the pseudo-tree the edges of the corresponding spanning tree. Also, nodes, variables, and agents may not be strictly distinguished. The following notations are used.

- $prnt_i$: parent variable of x_i.
- $Chld_i$: set of child variables of x_i.
- Nbr_i^u: partial set of ancestor variables of x_i. The variables in Nbr_i^u are related to x_i by constraints.
- Nbr_i^l: partial set of descendant variables of x_i. The variables in Nbr_i^l are related to x_i by constraints.
- $PsdPrnt_i$: partial set of ancestor variables of x_i. Let x_k denote a variable in $PsdPrnt_i$. For at least one variable x_j that is contained in the pseudo-tree rooted at x_i, x_k has relationship $x_k \in Nbr_j^u$.

No back edge exists between different subtrees. By employing this property, search processing can be performed in parallel.

In this work, we focus on feeder trees. Therefore, their pseudo-trees are true trees, which is the most simple case of the pseudo-tree. For generality, we show the computation for feeder networks that contain cycles.

B. Computation based on the pseudo-trees

We outline cost computation using pseudo-trees [2], [3]. Below, we assume that agents have already received both variables' values and cost values from other agents. Agent i's computation is based on partial solution s_i of $PsdPrnt_i$. s_i is called *context*.

Local cost $\delta_i(s_i \cup \{(x_i, d)\})$ for context s_i and value d of variable x_i are defined as follows.

$$\delta_i(s_i \cup \{(x_i, d)\}) = \sum_{(x_j, d_j) \in s_i,\ j \in Nbr_i^u} f_{i,j}(d, d_j) \qquad (13)$$

Optimal cost $g^*(s_i)$ for context s_i and the subtree routed at x_i are recursively defined as follows.

$$g_i^*(s_i) = \min_{d \in D_i} g_i(s_i \cup \{(x_i, d)\}) \qquad (14)$$

$$\begin{aligned} g_i(s_i \cup \{(x_i, d)\}) \quad &= \quad \delta_i(s_i \cup \{(x_i, d)\}) \\ + \sum_{j \in Chld_i} g_j^*(s_j) \quad &\text{s.t.} \quad s_j \subseteq (s_i \cup \{(x_i, d)\}) \end{aligned} \qquad (15)$$

In the above, binary cost function $f_{i,j}$ can be generalized to n-ary cost functions including unary functions.

When globally optimal cost $g_r^*(\phi)$ is computed for root variable x_r, r determines the optimal assignment of its variable. Similarly, an optimal solution for the rest of the problem can be computed in a top-down manner.

Dynamic programming [3] computes the optimal cost values of subtrees from leaf agents to a root agent. Then, the optimal assignments are decided from a root agent to leaf agents. Although there is no iterative processing, the size of the memory and the messages are exponential to the induced-width of the pseudo-trees because each agent i simultaneously computes $g^*(s_i)$ for all assignments of the variables contained in $PsdPrnt_i$. Moreover, the size of the domain of variables also increases the search space.

In this work, we focus on feeder trees. Therefore, induced-width is not critical. On the other hand, the size of the domain affects the size of the table of $g^*(s_i)$.

C. Modification of computation

In the conventional methods, the priorities of variables are defined by the pseudo-tree. Based on the priorities, values of cost functions are summed up. When cost functions are evaluated in a node i, only values of $PsdPrnt_i$ and x_i are considered.

In the formalization shown in III-B, values of the variables are mapped to values of the potentials. Then values of functions are computed. Although values of several functions can be computed using values of $PsdPrnt_i$ and x_i, functions that consider values of p_i need values of variables in Nbr_i^l. Note that p_i is computed based on c_i shown in Expression 6. c_i is computed using values of all neighborhood nodes of i.

To propagate the value of Nbr_i^l, another type of context t_{i,s_j} is used. We call the context *lower-context*. A set of variables is introduced to define t_{i,s_j}:

- $ExtNbr_i$: a set of i's descendant nodes that have a link to i or i's ancestor nodes. $ExtNbr_i$ contains variables in $Chld_i$.

In the case of tree, $ExtNbr_i$ only contains variables in $Chld_i$. t_{i,s_j} is defined for a subset $ExtNbr_{i,j}$ of $ExtNbr_i$. $ExtNbr_{i,j}$ only contains variables in a subtree rooted at i's child node j. t_{i,s_j} also depends on s_j that is a context of i's child node j. Note that $ExtNbr_{prnt_i} \subseteq ExtNbr_i \cup \{x_i\}$. t_{prnt_i,s_i} is computed from each set of t_{i,s_j} that relates

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

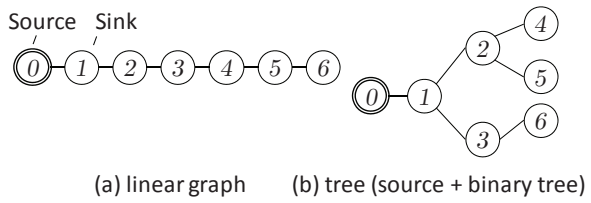


Fig. 3. Example problems (in the case of 7 nodes)

with t_{prnt_i,s_i}. In the computation, assignments of variables in $ExtNbr_i \cup \{x_i\} \backslash ExtNbr_{prnt_i}$ are removed. Then an assignment of x_i is added. The table of $g_i^*(s_i)$ is generalized to $(g_i^*(s_i,), t_{prnt_i,s_i})$.

Expression 13 is modified as follows. Using assignments of the related variables, values of cost/utility functions that are contained in f_i^{cost} are calculated.

$$\begin{array}{l} \delta_i(s_i \cup \{(x_i,d)\}) = f_i^{cost}(\\ \{(x_i,d)\} \cup \\ \bigcup_{(x_j,d_j) \in s_i,\ j \in Nbr_i^u} \{(x_j,d_j)\} \cup \\ \bigcup_{(x_k,d_k) \in t_{i,s_h}, h \in Chld_i,\ k \in Nbr_i^l} \{(x_k,d_k)\}) \end{array} \quad (16)$$

Additionally, the proposed formalization employs negative cost (i.e. utility) values. Although several pruning methods that based on monotonicity of the cost values have to be modified in such case, we simply apply a solver that does not employ the pruning.

D. Algorithm

Basically, the algorithm resembles the conventional dynamic programming method [3]. It computes optimal cost values and an optimal solution in a distributed manner. Because the distributed processing is relatively easy, we show brief explanations of the algorithm. The processing consists of the following phases.

- Pre-processing: a pseudo-tree is generated using a graph traversal in a distributed manner. A basic method is distributed depth first graph traversal. In the processing, sets of relative variables (i.e. $prnt_i$, $PsdPrnt_i$, etc.) are computed.
- COST propagation: Each leaf node i computes the table of $(g_i^*(s_i), t_{prnt_i,s_i})$. Then, i sends the table to $prnt_i$ using a COST message [1]. Other nodes j similarly compute the table of $(g_j^*(s_j), t_{prnt_j,s_j})$ and propagate it when COST messages are received from all nodes that have variables in $Chld_j$. As a result, the root node of the pseudo-tree computes the global optimal cost value.
- VALUE propagation: Root i determines optimal assignment of x_i based on the global optimal cost. Then, i composes the optimal assignment s_j^* for each node j that has a variable in $Chld_i$. s_j^* is sent to j using a VALUE message. The other node k similarly computes its optimal assignment and composes the optimal assignment s_l^* for each child l when a VALUE message is received from $prnt_k$. Then, k sends s_l^* to l.

Paths of the messages are shown in Figure 2(c).

[1]In this work, we prefer to use COST instead of UTIL, which is used in conventional works, because we focus on minimizing problems.

V. Experiments

As the first result, we experimentally evaluated behaviors of the proposed model. Two types of networks shown in Figure 3 were used. The problems were generated using the following parameters that were determined preliminary experiments.

- n: the number of nodes including one source and multiple sinks is 11 or 21.
- $P^{s\perp}$, $P^{s\top}$: the minimum and maximum amount of the resource are 0 and 3.
- k^{csunit}: the parameter that defines steps of utilities is 100.
- $w_i^{inp}, w_i^{exp}, w_i^{utl}$: the parameters of costs/utilities for importing/exporting resource are set based on several ratios.
- $loss_{i,j}^{\top}$: the maximum amount of a loss of resource in the link between nodes i and j. We estimated that this parameter is 0.0025.
- $G_{i,j}$: the parameter for the resource that is transferred through the link between nodes i and j is 435.
- $V_{i,j}^{dif\top}$: the parameter for the maximum amount of resource that is transferred through the link between nodes i and j is 0.0075.
- v_i^{unit}: the unit quantity of the value of v_i is 0.0001.
- $x_i^{\perp}, x_i^{\top}$: the minimum and maximum values of x_i are -500 and 500.
- P_i^{ch}, P_i^{cs}, P_i^g: the hard/soft requirement for consumption of resource, and the maximum amount of resource that can be supplied are determined as below.

While we evaluate an ideal model, we chose the scale of several parameters based on a type of actual power supply networks.

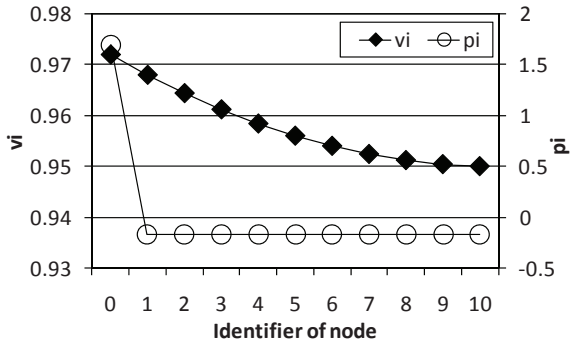
P_i^{ch}, P_i^{cs} and P_i^g are determined as follows. First, an amount of resource P^{call} is computed as $P^{s\top} -$ (number of sinks) $\cdot\ loss_{i,j}^{\top}$. Then an amount of resource P^{csnk} is computed by P^{call}/(number of sinks). P_i^{ch} and P_i^{cs} are determined so that $P_i^{ch} + P_i^{cs} = P^{csnk}$. Here we chose ratio $P_i^{ch} : P_i^{cs} = 1 : 1$. If node i have distributed source, P_i^g takes a value such that $P_i^g = P_i^{ch} + P_i^{cs}$. Otherwise, the value of P_i^g is zero.

In the case that several sinks have distributed sources, the distributed sources are randomly placed with uniform distribution. The results are averaged for twenty instances in the case of problems.

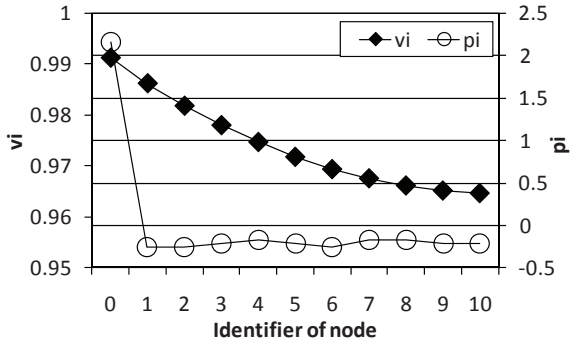
Figure 4 shows the results in the case of linear networks that have no distributed sources. Only the source node that has identifier 0 exports the resource. On the other hand, other nodes import the resource. Therefore, potential v_i monotonously decreases. Because we did not fix the standard value of the potential, only difference of v_i is important. p_i represents the amount of resource coming from node i.

In Figure 4 (a), there are no utilities to consume soft requirements of the resource. Therefore, each sink only intends to consume hard requirements. On the other hand, in Figure 4 (b), the cost and the utility for the soft requirements take same value. In such case, the result partially depends on biases of the solver. Figure 4 (c) shows the case that the utility exceeds the cost. Each node intends to consume the hard and soft requirements of the resource.

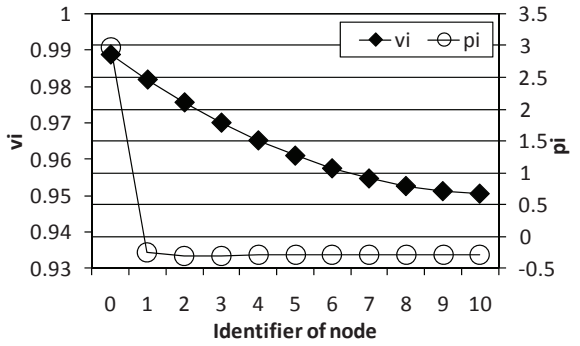
ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) $w^{inp} : w^{utl} = 1 : 0$

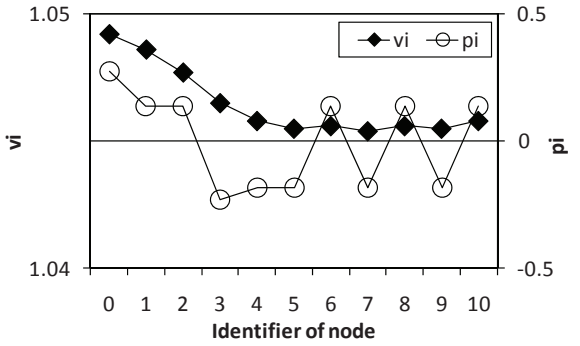


(b) $w^{inp} : w^{utl} = 1 : 1$

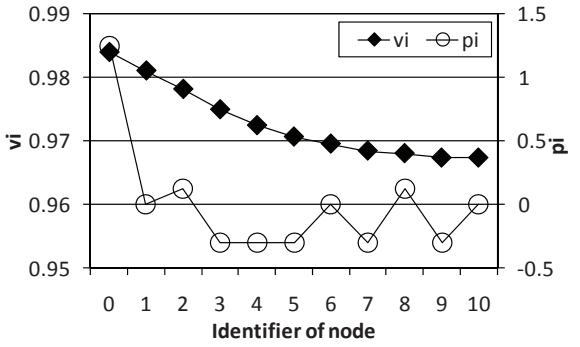


(c) $w^{inp} : w^{utl} = 1 : 1.5$

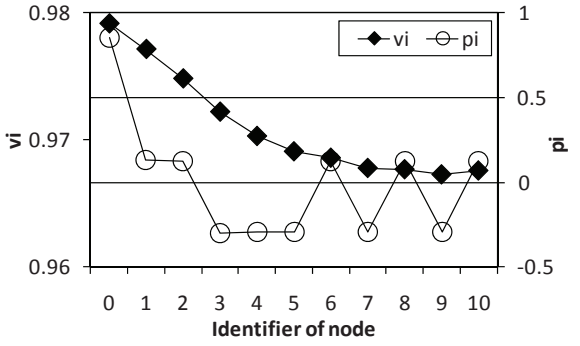
Fig. 4. $n = 11$, linear graph, no distributed sources



(a) $w^{inp} : w^{utl} : w^{exp} = 1 : 1 : 1.5$



(b) $w^{inp} : w^{utl} : w^{exp} = 1 : 1.5 : 1.5$



(c) $w^{inp} : w^{utl} : w^{exp} = 1 : 1.5 : 2$

Fig. 5. $n = 11$, linear graph, the ratio of the number of distributed sources = 0.5

Figure 5 shows the results in the case of linear networks that have distributed sources. In the results, sinks of identifiers 1, 2, 6, 8 and 10 have the distributed sources. p_i of these nodes takes positive value when the nodes export the resource. In Figure 5 (a), utilities for exporting the resource is relatively large. Therefore, all sinks that have the distributed sources export the resource. Because of the passing of the resource, v_i does not monotonously decrease around several sinks.

Figure 5 (b), shows the case that utilities for the exporting and soft requirements are same value. Since the exporting of the resource is not highly affected by the utilities, behaviors of the sinks with the distributed sources are different.

In Figure 5 (c), the utility of the exporting of the resource takes the most large value. Therefore, it affects the sinks with the distributed sources. On the other hand, in the sinks without the distributed sources, the utility of the soft requirements mainly affects.

Table I shows the results of p_i in linear networks that contain 11 nodes. The results correspond to the results shown in Figure 4. In the result of Table I (a), the value of row '1 : 1.5' and column 'source' should be nearly the maximum value 3 of the range of p_i. Similarly, row '1 : 1.5' and column 'sink, ave.' should be the minimum value -0.2975. Row '1 : 0' and column 'sink, ave.' should be the maximum value -0.1488. A reason of the difference of the results and the theoretical value is the discrete value of v_i. Therefore, the proposed model represents boundaries of feasible solutions. As addressed in Section VI, there are several opportunities to reduce the difference. Table I (b) shows the result in the case that the network contains distributed sources. The results almost correspond to the results shown in Figure 5.

Table II shows the results in tree networks that contain 11 nodes. The results resemble the case of the linear networks. Table III shows the results in linear networks that contain 21

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
p_i ($n = 11$, LINEAR GRAPH)

(a) no distributed source

node [range of p_i]	source [0, 3]	sink w/o dist. source [-0.2975, -0.1488]			
$w^{inp} : w^{utl}$	ave.	ave.	min.	max.	max. diff.
1 : 0	1.6913	-0.1664	-0.1684	-0.1653	0.0031
1 : 1	2.1559	-0.2117	-0.2574	-0.1681	0.0893
1 : 1.5	2.9682	-0.2887	-0.2971	-0.2563	0.0408

(b) the ratio of the number of distributed source = 0.5

node [range of p_i]	source [0, 3]	sink w/o dist. source [-0.2975, -0.1488]				sink w. dist. source [0, 0.1488]			
$w^{inp} : w^{utl} : w^{exp}$	ave.	ave.	min.	max.	max. diff.	ave.	min.	max.	max. diff.
1 : 1.5 : 0	0.3926	-0.2148	-0.2507	-0.1909	0.0598	0.1366	0.1364	0.1368	0.0004
1 : 1.5 : 1.5	1.2160	-0.2952	-0.2972	-0.2938	0.0033	0.0548	0.0000	0.1223	0.1223
1 : 1.5 : 2	0.8509	-0.2953	-0.2968	-0.2943	0.0024	0.1267	0.1263	0.1272	0.0010

TABLE II
p_i ($n = 11$, TREE)

(a) no distributed source

node [range of p_i]	source [0, 3]	sink w/o dist. source [-0.2975, -0.1488]			
$w^{inp} : w^{utl}$	ave.	ave.	min.	max.	max. diff.
1 : 0	1.6669	-0.1656	-0.1660	-0.1653	0.0007
1 : 1	2.4592	-0.2435	-0.2946	-0.1683	0.1263
1 : 1.5	2.9902	-0.2955	-0.2969	-0.2948	0.0021

(b) the ratio of the number of distributed source = 0.5

node [range of p_i]	source [0, 3]	sink w/o dist. source [-0.2975, -0.1488]				sink w. dist. source [0, 0.1488]			
$w^{inp} : w^{utl} : w^{exp}$	ave.	ave.	min.	max.	max. diff.	ave.	min.	max.	max. diff.
1 : 1.5 : 0	0.4037	-0.2172	-0.2323	-0.1956	0.0367	0.1367	0.1366	0.1368	0.0002
1 : 1.5 : 1.5	1.3611	-0.2964	-0.2971	-0.2960	0.0011	0.0259	0.0000	0.0912	0.0912
1 : 1.5 : 2	0.8503	-0.2964	-0.2969	-0.2960	0.0009	0.1272	0.1271	0.1274	0.0003

TABLE III
p_i ($n = 21$, LINEAR GRAPH)

(The ratio of the number of distributed source = 0.5)

node [range of p_i]	source [0, 3]	sink w/o dist. source [-0.1475, -0.0738]				sink w. dist. source [0, 0.0738]			
$w^{inp} : w^{utl} : w^{exp}$	ave.	ave.	min.	max.	max. diff.	ave.	min.	max.	max. diff.
1 : 1.5 : 0	0.6587	-0.1103	-0.1338	-0.0900	0.0438	0.0452	0.0449	0.0455	0.0006
1 : 1.5 : 1.5	1.1248	-0.1346	-0.1363	-0.1335	0.0028	0.0240	0.0000	0.0452	0.0452
1 : 1.5 : 2	0.9125	-0.1349	-0.1364	-0.1341	0.0023	0.0450	0.0447	0.0455	0.0007

nodes. Although there are errors between the discrete values and the theoretical range of values, the results resemble the case of 11 nodes.

VI. DISCUSSION

In this work, we defined a cooperative model motivated by power supply networks. Although the proposed model is different from the common formalization of power flow calculation, we preferred an intuitive representation of the problem that contains the power flow. An approximate method that employs active power values and phases of potentials resembles the power flow calculation of the proposed model. In related works, the power flow calculation is represented using more simple model [6]. The necessity of the detailed model of the power flow will be depend on purposes.

For the distributed constraint optimization problem, we used discrete potential value of the resource. The discrete value increases search space and decreases accuracy. Additional methods that reduce search spaces or numerical computation techniques that interpolate values [10] can be considered to overcome this problem.

The proposed model minimizes total cost of the system. As a more practical criteria, a kind of fairness among agents can be considered.

While we applied a dynamic programming method, other solvers based on pseudo-trees [2] can be modified for the proposed problem. There are opportunities to reduce the search when solvers which use pruning, optimistic search strategy and bounded errors are applied.

VII. CONCLUSION

We proposed a distributed cooperative model motivated by power supply networks. The model was represented as a distributed constraint optimization problem, and a conventional algorithm was modified to be applied to the

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

problem. Behaviors of the proposed model and the solver were experimentally evaluated.

Efficient methods that reduce search space and improve accuracy, theoretical analysis of the model and application to more practical problems will be included in future works.

REFERENCES

[1] R. Mailler and V. Lesser, "Solving distributed constraint optimization problems using cooperative mediation," in *3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004, pp. 438–445.

[2] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo, "Adopt: Asynchronous distributed constraint optimization with quality guarantees," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 149–180, 2005.

[3] A. Petcu and B. Faltings, "A scalable method for multiagent constraint optimization," in *9th International Joint Conference on Artificial Intelligence*, 2005, pp. 266–271.

[4] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg, "Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 55–87, 2005.

[5] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham, "Taking dcop to the real world: Efficient complete solutions for distributed multi-event scheduling," in *3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004, pp. 310–317.

[6] A. Kumar, B. Faltings, and A. Petcu, "Distributed constraint optimization with structured resource constraints," in *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 923–930.

[7] S. ThiAcbaux and M. odile Cordier, "Supply restoration in power distribution systems - a benchmark for planning under uncertainty," in *In Pre-Proceedings of the 6th European Conference on Planning (ECP-01*, 2001, pp. 525–532.

[8] E. C. Freuder, "A sufficient condition for backtrack-bounded search," *Journal of the ACM*, vol. 32, no. 14, pp. 755–761, 1985.

[9] T. Schiex, "A note on csp graph parameters," *Technical report 1999/03, INRA*, 1999.

[10] T. Voice, R. Stranders, A. Rogers, and N. R. Jennings, "A hybrid continuous max-sum algorithm for decentralised coordination," in *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, 2010, pp. 61–66.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)