UNIVERSITY OF BIRMINGHAM

A task-oriented heuristic for repairing infeasible solutions to overlapping coalition structure generation

Zhang, Guofu; Su, Zhaopin; Li, Miqing; Qi, Meibin; Jiang, Jianguo; Yao, Xin

Link to publication on Research at Birmingham portal

A Task-Oriented Heuristic for Repairing Infeasible Solutions to Overlapping Coalition Structure Generation

Guofu Zhang, *Member, IEEE,* Zhaopin Su, *Member, IEEE,* Miqing Li, Meibin Qi, Jianguo Jiang, and Xin Yao, *Fellow, IEEE*

Abstract—Overlapping coalition formation (OCF), which provides a natural framework for modeling scenarios where each agent can join and allocate their resources to several completely different coalitions at the same time, has become a very active topic in multi-agent systems (MAS). For OCF in resource-constrained and subadditive task oriented domains, an agent may not possess sufficient resources to meet the needs of multiple coalitions simultaneously. As a result, there may exist many potential resource conflicts among the rival overlapping coalitions. To tackle such situations, we first present a natural variation of the traditional OCF model and analyze the size of the solution space and the computational complexity of the overlapping coalition structure generation (OCSG) problem. Next, we develop a generic task-oriented heuristic (TOH) for individual repairs that can be used in binary meta-heuristic algorithms (MHAs) to generate overlapping coalitions in a parallel manner. Moreover, we show how the proposed TOH repairs a two-dimensional individual to resolve resource conflicts and discuss several basic properties. Finally, to evaluate the effectiveness of TOH, we compare it with the existing agent-oriented heuristic (AOH) for the OCSG problem. The empirical results demonstrate that TOH is of high efficiency and effectiveness in harsh environments with fierce competition over scarce resources.

Index Terms—Overlapping coalition formation, subadditive tasks, constrained resources, two-dimensional binary individual, heuristic.

I. INTRODUCTION

IN multi-agent systems (MAS), agents are computational entities such as software programs and robots that are able to accomplish a series of tasks autonomously within a given time bound [1], [2]. Usually, these agents have some kinds of abilities or possess a certain amount of resources. When agents encounter tasks that cannot be fulfilled independently,

G. Zhang, Z. Su, M. Qi, and J. Jiang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China e-mail: {zgf;szp;mbqi;jgjiang}@hfut.edu.cn.
M. Li is with the CERCIA, School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K. e-mail: M.Li.8@cs.bham.ac.uk.
X. Yao is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China, and the CERCIA, School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K. e-mail: xiny@sustc.edu.cn.

they can share their resources and work together by forming coalitions [3], [4]. Coalition formation, which is a fundamental and important form of interaction and cooperation, can make weak agents strong enough to achieve impossible goals. Then, the biggest challenges we face are which coalitions will form and which coalitions are the most beneficial to members' collaboration and the execution of tasks [5], [6].

In recent years, coalition formation has become a highly active topic in the field of MAS and has been successfully applied in operation research and computer science [7]–[9]. However, the traditional coalition formation concentrates on a very harsh term that each agent can only be a member of one coalition at any given time [6], which may result in a very low utilization rate of agents' resources [10]. In some practical scenarios, an agent in a complex system has the characteristic of diversity. That is, the agent can often be attributed to different categories or goals in which members are overlapping and interrelated [10], [11]. For example, in virtual organizations [12], [13] and wireless networks [14], [15], an agent with sufficient resources is able to apply for joining several completely different cooperative coalitions to earn more profits or achieve more objectives. The reason for this is that overlapping coalitions make it possible for agents to form a coalition for each task, without preventing other tasks from being satisfied [10]. Moreover, each agent only needs to make necessary negotiations and contracts with partners which actually work together rather than with all the other agents in the entire system [5]. To model these scenarios, overlapping coalition formation (OCF) is introduced to allow each agent to be one member of completely different coalitions and thus can contribute their resources to the coalitions at the same time [5], [10].

Additionally, in most of the literature on traditional coalition formation, it is assumed that there is no resource conflict in coalition structure generation [6]. That is, each agent's resources are unlimited and can fully meet the needs of all the possible coalitions. Although this assumption is suitable for some specific scenarios considered in the exiting work, strictly speaking, each agent's resources are limited in most of the practical applications. For example, the finite energy of each node in wireless sensor networks [16] is the most valuable resource and directly determines its lifetime, so one of the most crucial problems is how to schedule and balance nodes' energy consumption to conserve energy and increase the lifetime of the whole network. Another example is the

distribution of emergency relief supplies in disaster-emergency management [17]. In this domain, there are numerous reserve and dispatch points at different locations, and the focus is on allocating the finite emergency relief supplies of multiple reserve points to different dispatch points as soon as possible for maintaining life and improving health. The final example focuses on multi-camera surveillance systems [18], in which each camera has limited data processing capability, memory space, and view angle, and thus each camera can monitor and recognize few people only within its observation area.

In the above scenarios, when an agent with rare, but highly demanded resources has joined several completely different coalitions simultaneously, those coalitions will compete keenly for this agent. Once the agent is unable to meet the resource needs of so many coalitions, there may exist potential resource conflicts over the use of joint resources which are rare, but in high demand among overlapping coalitions. Then, those formed coalitions will be unstable and may disband, which may affect the task execution. This means that when an agent has limited resources, potential resource conflicts among rival overlapping coalitions need to be taken into account before forming coalitions to ensure the reliability and stability of the entire system [19]. Therefore, each agent in the process of OCF has to consider the following questions—Which coalitions will I join? How many resources can I separately contribute to every coalition that I have joined? Do I have sufficient resources to meet the requirements of those formed coalitions at the same time?

On the basis of the preceding considerations, to date, there have been a number of attempts in the literature to analyze computational questions surrounding OCF [20]–[23] and find efficient algorithms for solving OCF to maximize the sum of the profits for each task, namely, the social welfare [5], [10], [24]–[27]. Specifically, to avoid the possible resource conflicts among the competitive overlapping coalitions, Shehory and Kraus [5], [10], as well as Zhan et al. [27], serially allocated agents' resources or weights to form each coalition for the given tasks, implying that a task's request can be satisfied only after the previous coalitions for the corresponding tasks have been successfully formed. In addition, Zhang et al. [24], [26], as well as Lin and Hu [25], improved the binary particle swarm optimization (BPSO) [28] with the two-dimensional encoding and the corresponding heuristics to search for overlapping coalitions in a parallel manner. However, the existing work has the following drawbacks.

- It is not shown theoretically that the embedded heuristics could strengthen the exploration ability of the search algorithms;
- The search algorithms would not work, when agents' resources are not enough to accomplish all the given tasks;
- The proposed heuristics need to evaluate all the rows and columns in a two-dimensional binary individual, which results in that the heuristics are complicated and time-consuming, especially under the large problem size.

Against this background, this paper makes the following contributions to the state of the art in OCF:

- We proposed a natural variation of the traditional OCF in resource-constrained and subadditive task oriented domains. In the proposed OCF model, each agent's resources are limited and the potential resource conflicts among rival coalitions may take place often.
- We analyzed the size of the search space and the complexity of the key overlapping coalition structure generation (OCSG) problem in the proposed OCF model. It was found that the OCSG problem is NP-complete and the traditional deterministic search techniques may not be available within an acceptable time.
- We developed a generic task-oriented heuristic (TOH) for repairing infeasible solutions which can be used in meta-heuristic algorithms for the OCSG problem. Unlike the previous work, TOH only repairs each row rather than all the rows and columns in a two-dimensional binary encoding. By applying TOH, the binary meta-heuristic algorithms are able to assign tasks according to the current residual resources of agents. This is helpful to make meta-heuristic algorithms workable and available in a harsh environment where agents have no enough resources to satisfy all the tasks.

The remainder of this paper is structured as follows. Section II surveys the related work on OCF. Section III proposes an OCF model in resource-constrained and subadditive task oriented domains and analyzes the computational complexity of the key OCSG problem. After discussing the existing AOH [26] that can be used in binary meta-heuristic algorithms for solving OCSG, the generic TOH is presented and analyzed in Section IV. An empirical verification of the proposed TOH in comparison with AOH is shown in Section V, which is followed by conclusive remarks and potential directions for future work in Section VI.

II. LITERATURE REVIEW

OCF provides a natural framework for modeling scenarios where agents can allocate their resources to different coalitions for the given tasks at the same time. Thus far, much of the existing work concentrates on studying computational questions surrounding OCF. Chalkiadakis et al. [20] presented a model for cooperative games with overlapping coalitions and explored the issue of stability in this setting on the basis of the conservative, refined, and optimistic core. Thereafter, Zick et al. [21]–[23] proposed an arbitrated OCF model and discussed its stability according to the arbitrated core. Moreover, Zick et al. [21], [22] proved that the algorithmic complexity of the associated problems in the arbitrated OCF crucially depends on the amount of resources that each agent possesses, the maximum coalition size, and the pattern of interaction among agents. However, the above serial work does not show the size of the solution space and how to search through the solution space.

A different line of research focuses on algorithms for finding the optimal overlapping coalition structure to maximize the sum of the profits for each task [5], [10], [27], which is also known as the overlapping coalition structure generation (OCSG) problem. In this regard, Shehory and Kraus [5],

[10] considered a task-based setting in which coalitions may overlap and will be successively formed according to the precedence ordering of each task. Moreover, to reduce the complexity, they developed greedy algorithms with a specific limitation on the size of each coalition. However, given such a limitation, their algorithms cannot find the optimal or even suboptimal solutions and still need a large amount of memory requirements and time consumption [29], because of the repetitive search for each task.

Zhan et al. [27] proved some computational complexity results of the threshold task games [20], a specific case of OCF games, and then presented dynamic programming and greedy algorithms for serial generation of overlapping coalitions. However, they assumed that the value of a coalition with responsibility for a given task is known and fixed no matter which agents will form the coalition. This assumption may simplify the search algorithm but makes coalition formation less meaningless in practice. This is because for a task, no matter which agents form the corresponding coalition and how members cooperate, it is impossible for members to obtain more profits. As a result, each member wants itself to become a freeloader but others to work as much as possible, which may lead to the non-production or under-production of a public good, namely, Pareto inefficiency [30].

In addition, the serial generation of overlapping coalitions for the given tasks [5], [10], [27] also has the following potential disadvantages. First, once the serial algorithms [5], [10], [27] cannot find a feasible coalition to satisfy the resource needs of a task, the algorithms will fall into an infinite loop and the other unassigned tasks cannot be dealt with, even if those tasks could have been accomplished by agents. Next, the previously formed coalitions will inevitably influence the searching results of coalitions for the successive tasks. The reason for this is that some agents' resources have been kept for the previous coalitions' own and cannot be used by the successive tasks. Hence, the final overlapping coalition structure obtained by the serial algorithms [5], [10], [27] cannot be optimal and even suboptimal.

In fact, the coalition formation in task-based settings is just selecting a portion of agents to form coalitions for as many tasks as possible, which is a complex combinatorial optimization problem [31]. It is known that the total number of coalitions grows exponentially with the number of agents involved and most of the natural decision-making problems related to coalition formation are NP-complete [31]. As the search space complexity increases with the numbers of agents, tasks, and resources, the cost of the above deterministic search algorithms can increase hugely [29], making the search of solutions infeasible. Another way to tackle this class of problems is to find a suboptimal solution in a reasonable time. Moreover, in some cases the optimal solution to the problem with appropriate size may even be found [32], [33]. In such optimization methods, meta-heuristic algorithms (MHAs) have become powerful and popular for solving varieties of coalition formation problems [34]–[39]. MHAs can be defined as a set of nature-inspired and population-based stochastic optimization techniques through the reproduction of genera- tions [43], [44]. More specifically, see Fig. 1, the basic meta-

Input: The algorithmic parameters and the test instance
Output: The best found solution
1. Design a representation of an individual
2. Design a way of evaluating an individual
3. Create an initial population (usually at random)
4. Evaluate all the individuals
5. **repeat**
6.   Select some individuals to be parents
7.   Create offspring by evolutionary operations on the parents
8.   Evaluate the offspring
9.   Select some individuals to be replaced by the new better offspring
10. **until** A predefined criteria is met

Fig. 1. The psuedocode of the most basic meta-heuristic algorithm.

heuristic algorithm first designs a representation (or encoding) of an individual and initializes every individual on the basis of the representation to create the original population. Then, use a fitness function (usually the objective function of the given problem) to determine how good each individual is. After that, execute evolutionary operations on selected parents according to a series of methods (here is the major difference among MHAs), such as selection, crossover, mutation, and replacement, to modify and update each individual if necessary and create the next generation that contains the best solutions in the past. The algorithm repeats the above steps until a termination criterion is met, such as the preset generation number is reached or a solution with adequate objective function value is found.

Particularly, considering agents, coalitions, and tasks as a whole and dealing with OCSG in a parallel manner, we improved BPSO [28] with a two-dimensional binary encoding and presented a simple heuristic to repair infeasible individuals [24]. However, this heuristic is rather rough and hard to make each individual feasible.

Lin and Hu [25] argued that if a coalition is feasible for its task, the remaining resources of this coalition except the task's needs can be transferred to a selected agent. Then, they delegated this selected agent to join and help other infeasible coalitions. However, on the basis of their approach, one only knows the selected agent joins an infeasible coalition. It remains unknown that which members in the previous feasible coalition represented by the selected agent actually join the infeasible coalition. This makes the final coalition structure less practical.

After that, an agent-oriented heuristic (AOH) [26] was presented to deal with the potential resource conflicts when some coalitions compete each other for the same agents with insufficient resources. AOH can show how many resources each agent contributes to each coalition that it has joined. However, AOH can only work in a very ideal environment where agents have sufficient resources to accomplish all the possible tasks. Moreover, AOH either provides solutions that are of very low quality or has a very high computational complexity (see Section IV for more details).

To address these shortcomings, in this paper we aim to present a more efficient and effective heuristic for OCSG by dynamically tracking each agent's residual resources that may be available for infeasible coalitions.

III. Problem Description

This section formalizes OCF in resource-constrained and subadditive task oriented domains, which can be seen as a natural extension of the traditional OCF [20]. Let $A = \{a_1, \cdots, a_n\}$ denote a set of $n \in \mathbb{N}$ agents with bounded resources that have to cooperate to accomplish a set of possible tasks $T = \{t_1, \cdots, t_m\}$ ($m \in \mathbb{N}$).

Each task $t_i \in T$, $i \in \{1, \cdots, m\}$, requires $r \in \mathbb{N}$ types of resources $\mathbf{D}_i = \langle d_1^i, \cdots, d_r^i \rangle$, where $d_k^i \geq 0$, $k \in \{1, \cdots, r\}$, is a nonnegative integer and represents the amount of resource of type k required by task t_i. In addition, each $t_i \in T$ has a certain reward $\mu_i > 0$ that is a positive integer if t_i can be accomplished.

Each agent $a_j \in A$, $j \in \{1, \cdots, n\}$, is endowed with r types of original resources $\mathbf{B}_j = \langle b_1^j, \cdots, b_r^j \rangle$, where $b_k^j \geq 0$ is a nonnegative integer and denotes the amount of resource of type k owned by agent a_j. Besides, there is a coordination and communication cost $\pi_{jj^*} \geq 0$, $j^* \in \{1, \cdots, n\}$ and $j^* \neq j$, between any two different agents a_j and a_{j^*} [5], [10], [25], [26], [35], [37]. π_{jj^*} is a nonnegative integer, satisfying $\pi_{jj} = 0$ and $\pi_{jj^*} = \pi_{j^*j}$.

In OCF, an agent may contribute its resources to different tasks at the same time. In this way, each agent $a_j \in A$ has an actual resource contribution, $\mathbf{W}_{ji} = \langle w_1^{ji}, \cdots, w_r^{ji} \rangle$, for each $t_i \in T$, where $0 \leq w_k^{ji} \leq b_k^j$ is a nonnegative integer and denotes the amount of resource of type k contributed by a_j to t_i. Note that if a_j does not join t_i, $w_k^{ji} = 0$, $k \in \{1, \cdots, r\}$. It follows logically that each agent $a_j \in A$ has r types of residual resources $\mathbf{P}_j = \langle p_1^j, \cdots, p_r^j \rangle$, where $0 \leq p_k^j \leq b_k^j$ is the amount of the remaining resources of type k owned by a_j after it has distributed its original resources to its joined tasks. If a_j does not join any task, $p_k^j = b_k^j$, $k \in \{1, \cdots, r\}$; otherwise, $p_k^j = b_k^j - \sum_{i=1}^{m} w_k^{ji}$, $k \in \{1, \cdots, r\}$.

A coalition $C_i \subseteq A$ is a set of member agents with responsibility for a task $t_i \in T$. C_i also has r types of resources, $\mathbf{B}_{C_i} = \langle b_1^{C_i}, \cdots, b_r^{C_i} \rangle$, where $b_k^{C_i} \geq 0$ is the sum of resources that members contribute to C_i for performing t_i. That is, $b_k^{C_i} = \sum_{\forall a_j \in C_i} w_k^{ji} = d_k^i$, $k \in \{1, \cdots, r\}$. Note that if $\exists k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j < d_k^i$, it is impossible for C_i to perform t_i even if $C_i = A$, so we say that t_i is unachievable under the current residual resources of agents.

It should be noted that in subadditive task oriented domains, each task is assigned at most one coalition of agents to improve resource utilization and task execution efficiency [5], [40], [41]. However, sometimes, a coalition may carry out several different tasks at the same time if only it has rare, but highly demanded resources. Then, we say that the assigned overlapping coalitions to those different tasks are the same.

The value of a coalition C_i with responsibility for t_i is calculated by a characteristic function [5], [10], [25], [26], [34], [35], [37]

$$v(C_i) = \mu_i - \theta(C_i) \tag{1}$$

where $\theta(C_i) = \sum_{\forall a_j, a_{j^*} \in C_i, j < j^*} \pi_{jj^*}$ is the total intra-coalition coordinate and communication costs among members in C_i

for carrying out t_i. For example, if $C_i = \{a_1, a_2, a_3\}$, $\theta(C_i) = \pi_{12} + \pi_{13} + \pi_{23}$. Note that if t_i is not achievable, C_i is infeasible, and thus we say that $C_i = \emptyset$ and $v(C_i) = 0$. Moreover, it can be observed that $v(C_i)$ is a subadditive function in which smaller coalition size is better. The reason for this is that $\theta(C_i)$ grows exponentially with the number of members in C_i [5], [10], [25], [26], [35], [37]. Hence, for the given t_i, C_i with smaller size has a higher coalition value in subadditive task oriented domains [10], [40].

An overlapping coalition structure (OCS) $\{C_1, \cdots, C_m\}$ is composed of m possible overlapping coalitions, each of which is related to $t_1, \cdots, t_m$, respectively. As in the traditional OCF [5], [10], [25], [26], we consider the value of an OCS in terms of its social welfare. That is, the value of an OCS is the sum of the values of its containing m overlapping coalitions:

$$v(OCS) = \sum_{i=1}^{m} v(C_i) \tag{2}$$

In particular, $OCS = \{C_1, \cdots, C_m\}$ satisfies the following resource constraints.

$$\sum_{j=1}^{n} w_k^{ji} = d_k^i, k \in \{1, \cdots, r\}, i \in \{1, \cdots, m\}, C_i \neq \emptyset \tag{3}$$

$$\sum_{i=1}^{m} w_k^{ji} \leq b_k^j, k \in \{1, \cdots, r\}, j \in \{1, \cdots, n\} \tag{4}$$

where constraint set (3) ensures that each feasible C_i can satisfy the resource needs of t_i; constraint set (4) ensures that there is no resource conflict over the usage of agents' resources among m overlapping coalitions.

Let Π_A^T denote the set of possible overlapping coalition structures over A and T. Given the preceding terms, the problem of overlapping coalition structure generation (OCSG) in OCF is then to find an optimal overlapping coalition structure OCS^* to maximize the social welfare:

$$OCS^* = \arg \max_{OCS \in \Pi_A^T} v(OCS) \tag{5}$$

In summary, we describe the OCSG in OCF as the following constrained optimization problem.

$$\begin{cases} \text{Maximize:} \quad \sum_{i=1}^{m} v(C_i) \\ \text{Subject to:} \\ d_k^i = \sum_{j=1}^{n} w_k^{ji}, k \in \{1, \cdots, r\}, i \in \{1, \cdots, m\}, C_i \neq \emptyset \\ \sum_{i=1}^{m} w_k^{ji} \leq b_k^j, k \in \{1, \cdots, r\}, j \in \{1, \cdots, n\} \end{cases}$$

Given the above formulations, we investigate the size of the search space and the complexity of the OCSG problem in the proposed OCF model.

Theorem 1: The total number of the possible overlapping coalition structures is 2^{mn}.

Proof: It is known that there are in total $(2^n - 1)$ nonempty coalitions for the given n agents [6], [31]. In OCF, each agent may join these $(2^n - 1)$ coalitions at the same time. Accordingly, for each $t_i \in T$, there are 2^n possibilities: t_i is not handled or is assigned a coalition from the $(2^n - 1)$

nonempty coalitions. As there are m tasks in T, the number of the possible overlapping coalition structures is $(2^n)^m = 2^{mn}$. ∎

It is clear that the size of the search space Π_A^T in the proposed OCF model is extremely big. Ideally, all of the tasks in T may be assigned and completed. However, in the proposed OCF model the total available resources of all the agents in A are limited. When we attempt to maximize the social welfare, some tasks which have plenty of rewards may be facilitated but others hindered. Accordingly, in an overlapping coalition structure $OCS = \{C_1, \cdots, C_m\}$, it is possible that some coalitions may be the same or empty, and some agents may not be selected to join any coalition. Then, we can merge them altogether into one. For instance, given four agents $\{a_1, a_2, a_3, a_4\}$ and four tasks $\{t_1, t_2, t_3, t_4\}$. If a possible OCS for the four tasks is $\{\{a_1, a_2\}, \{a_1, a_2\}, \{a_2, a_4\}, \emptyset\}$, we know that coalitions $C_1 = \{a_1, a_2\}$ for t_1 and $C_2 = \{a_1, a_2\}$ for t_2 are the same, coalition $C_3 = \{a_2, a_4\}$ does t_3, and coalition $C_4 = \emptyset$ means that t_4 is not assigned. Here, a_3 is not selected by any coalition. In coalition structure generation [6], a coalition structure is feasible only if any coalition cannot be empty and the forming coalitions must cover all the given agents in A. That is, the union of all the nonempty coalitions in the OCS must be equal to A. To satisfy these constraints, we can repair the original $\{\{a_1, a_2\}, \{a_1, a_2\}, \{a_2, a_4\}, \emptyset\}$ into a feasible OCS $\{\{a_1, a_2\}, \{a_2, a_4\}, \{a_3\}\}$ in which $\{a_1, a_2\}$ carries out t_1 and t_2, $\{a_2, a_4\}$ executes t_3, and $\{a_3\}$ does nothing.

Theorem 2: There are at most $(m+1)$ coalitions in an OCS.

Proof: In $OCS = \{C_1, \cdots, C_m\}$, there are at most m different nonempty coalitions with a non-zero value for $t_1, \cdots, t_m$. Beyond that, if $\bigcup_{i=1}^{m} C_i \subset A$, some agents in A may not be included in $\bigcup_{i=1}^{m} C_i$. Then, according to the preceding terms, we have to form coalition C_{m+1} to contain these agents that do nothing and ensure the completeness of OCS. Note that the value of C_{m+1} is zero because it does nothing. As a result, there are at most $(m+1)$ different nonempty coalitions in a feasible OCS. ∎

Theorem 3: The problem of OCSG in the proposed OCF model is NP-complete.

Proof: It is known that the coalition structure generation problem for Threshold Task Games (TTGs), a subclass of OCF games, is NP-complete [20], [27]. In TTGs, each agent a_j has a weight ω_j. End each task t_i has a threshold τ_i and a utility u_i. Coalition C_i can do t_i and obtain u_i only if the sum of weights of members in C_i is bigger than or equal to the threshold τ_i of t_i. In the proposed OCF model, if we set $r = 1$ (i.e., there is only one type of resource) and the communication cost π_{jj^*} between any two different agents to be zero, the proposed OCF model just becomes a simple TTGs. Hence, it is clear that the problem of OCSG in the proposed OCF model is also NP-complete on the basis of the reduction principle [42]. ∎

Theorem 4: All of the tasks in T can be successfully assigned if and only if

$$\sum_{j=1}^{n} b_k^j \geq \sum_{i=1}^{m} d_k^i, k \in \{1, \cdots, r\} \quad (6)$$

Proof: We show the implications separately.

$\Longrightarrow$: In this case, all of the tasks in T can be successfully assigned, so each coalition C_i is feasible for each $t_i \in T$. Then, we know that the corresponding OCS for T satisfies (3) and (4). By (3), we can obtain

$$\sum_{i=1}^{m} \sum_{j=1}^{n} w_k^{ji} = \sum_{i=1}^{m} d_k^i, k \in \{1, \cdots, r\} \quad (7)$$

According to (4), we have

$$\sum_{j=1}^{n} \sum_{i=1}^{m} w_k^{ji} \leq \sum_{j=1}^{n} b_k^j, k \in \{1, \cdots, r\} \quad (8)$$

Taking (7) and (8) altogether, we prove the results in (6).

$\Longleftarrow$: Suppose for the sake of contradiction that not all tasks in T can be satisfied by agents in A. Without loss of generality, we assume that the previous $(m-1)$ tasks, $t_1, \cdots, t_{m-1}$, have been successfully assigned and satisfied by agents in A, but the last t_m cannot be satisfied by the residual resources of agents after they have responded the previous $(m-1)$ tasks. It means that for the previous $(m-1)$ tasks, we have

$$\sum_{i=1}^{m-1} \sum_{j=1}^{n} w_k^{ji} = \sum_{i=1}^{m-1} d_k^i, k \in \{1, \cdots, r\} \quad (9)$$

but for the last t_m,

$$\sum_{j=1}^{n} p_k^j < d_k^m, \exists k \in \{1, \cdots, r\} \quad (10)$$

$$p_k^j = b_k^j - \sum_{i=1}^{m-1} w_k^{ji}, k \in \{1, \cdots, r\} \quad (11)$$

Substituting (11) into (10), we can obtain

$$\sum_{j=1}^{n} b_k^j - \sum_{j=1}^{n} \sum_{i=1}^{m-1} w_k^{ji} < d_k^m, \exists k \in \{1, \cdots, r\} \quad (12)$$

By (9) and (12), we have

$$\sum_{j=1}^{n} b_k^j < \sum_{i=1}^{m} d_k^i, \exists k \in \{1, \cdots, r\}$$

which contradicts the initial condition (6), so we obtain that all of the tasks in T can be successfully assigned. ∎

Theorem 5: For $\forall C^* \subseteq A$ and $\forall T^* \subseteq T$, as long as

$$\sum_{a_j \in C^*} b_k^j \geq \sum_{t_i \in T^*} d_k^i, k \in \{1, \cdots, r\} \quad (13)$$

there certainly exists a feasible OCS.

Proof: By (13) and Theorem 4, we know that all of the tasks in T^* can be successfully assigned by agents in C^*. That is, for each $t_i \in T^*$, there is a corresponding feasible coalition $C_i \subseteq C^*$ such that $\bigcup_{\forall t_i \in T^*} C_i = C^*$. Moreover, there is no resource conflict among coalitions for T^*. As a result,
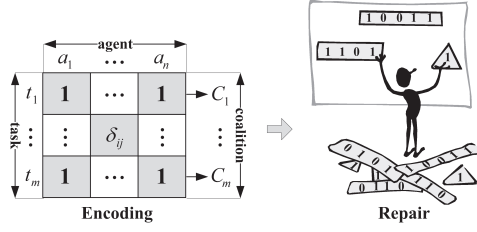
Fig. 2. An individual is represented by a two-dimensional binary encoding and needs to be repaired for feasibility before evaluation.

we can obtain a feasible overlapping coalition structure by merging the same coalitions and forming a new coalition for agents that do nothing. ∎

Theorem 6: If OCS^* is an optimal overlapping coalition structure and there is an unassigned task $t_i \in T$, it is impossible that

$$\sum_{j=1}^{n} p_k^j \geq d_k^i, k \in \{1, \cdots, r\} \quad (14)$$

Proof: For the sake of contradiction suppose (14) holds. We set $T^* = \{t_i\}$ and C^* is formed by agents in A that have residual resources. By (14), we obtain that C^* and T^* satisfy (13) according to the residual resources of members in C^*. Therefore, on the basis of Theorem 5, we have there certainly exists a feasible OCS for T^* with $\upsilon(OCS) > 0$. Without loss of generality, we assume $OCS = \{C^*\}$. Then, we have $\upsilon(OCS^* \cup OCS) > \upsilon(OCS^*)$, namely, OCS^* is not optimal, so this is a contradiction. ∎

IV. SOLVING THE OCSG PROBLEM

In this section, we present an efficient and effective heuristic that can be used to strengthen the exploration ability of binary MHAs for the OCSG problem. For the purpose of illustration, we start by introducing the classical two-dimensional binary encoding and recalling the existing agent-oriented heuristic for individual repairs. After that, we show a generic task-oriented heuristic for OCSG. Finally, we investigate several fundamental properties of the proposed heuristic.

A. Two-Dimensional Binary Encoding

To show the combinatorial optimization characteristics of the OCSG problem more intuitively, Fig. 2 depicts a simple two-dimensional binary encoding, in which each row represents a task (or the corresponding coalition) and each column denotes an agent. Let δ_{ij} denote a bit in this encoding. If $\delta_{ij} = 1$, a_j will take part in C_i with responsibility for t_i. If $\delta_{ij} = 0$, a_j will not join C_i. Each individual will be initialized at random according to this encoding scheme. Then, each individual represents a possible $OCS = \{C_1, \cdots, C_m\}$ and the fitness value of each individual can be calculated on the basis of (2). Note that there are m rows in the encoding for the reason that there are at most m coalitions with a non-zero value in the created OCS for the m given tasks.

For each individual, three important aspects should be considered to ensure the created OCS is feasible and has as

much social welfare as possible. First of all, as shown in (3), the endowed resources of each nonempty coalition need to be rich enough to carry out its task. Next, as shown in (4), there is no resource conflict over the usage of each agent's resources, even if an agent has joined several different coalitions at the same time. Finally, no unassigned task can be satisfied by the residual resources of agents, that is, the utilization rate of agents' resources needs to be maximized.

However, for each individual initialized randomly, we do not know whether any of the above conditions is satisfied. Furthermore, with the evolution of the MHAs, individuals also stay changing. Is the corresponding OCS in the created offspring individual feasible? If the embedded OCS in an individual is infeasible, the individual is just infeasible and its fitness value cannot be evaluated. A large amount of such infeasible individuals will consumedly debase the availability of population and weaken the performance of MHAs [43]. Hence, as shown in Fig. 2, before evaluating an individual we have to repair it to be feasible. We need to make sure that the repaired OCS satisfies the constraint sets (3) and (4). This thus raises an open question: how to repair an individual for feasibility and as much social welfare as possible?

To make individual repairs efficient and effective, we next discuss the details of the agent-oriented heuristic [26] in the following subsection, because of its relevance.

B. Agent-Oriented Heuristic

The agent-oriented heuristic (AOH) for generating an OCS [26] is based upon the workload that each agent contributes at least for each task. It proceeds in the following two stages. First, check each row and make its corresponding coalition feasible by randomly selecting available agents with the required resources to join each infeasible coalition. Second, check each agent to resolve potential resource conflicts. For each column j (or agent a_j), estimate the amount of resources (i.e., $\mathbf{W}_{ji}$) that each bit "1" in this column contributes at least to each task t_i. If $\forall k \in \{1, \cdots, r\}$, $w_k^{ji} = 0$, a_j is redundant for C_i, so eliminate a_j from C_i to reduce the size of C_i. If $\exists k \in \{1, \cdots, r\}$, $\sum_{i=1}^{m} w_k^{ji} > p_k^j$, a_j does not have sufficient residual resources to satisfy so many demands and a resource conflict occurs. Then, let a_j drop out of some coalitions until $\sum_{i=1}^{m} w_k^{ji} \leq p_k^j$, $\forall k \in \{1, \cdots, r\}$. Note that once a_j withdraws from a coalition C_i, C_i will be infeasible for t_i, so other available agents with required resources in the ith row need to be selected to join C_i until C_i can meet t_i. Finally, update the residual resources of a_j. Repeat the above steps until all the columns have been checked and repaired.

Fig. 3 provides the pseudo-code for the two stages of the AOH. Although AOH can resolve potential resource conflicts, it must evaluate all the rows and columns. From a more technical point of view, AOH is complicated and time-consuming, which can be verified by the below given complexity of the AOH.

Theorem 7: Given an arbitrary OCF instance, the worst case complexity of the AOH is $O(m \times n^2 \times r^2)$.

```
Input: A, T, B_1, ···, B_n, D_1, ···, D_m
Output: C_1, ···, C_m
 1. for i ∈ {1, ···, m} do
 2.     C_i ← ∅
 3.     b_k^{C_i} ← 0, k ∈ {1, ···, r}
 4. end for
 5. for j ∈ {1, ···, n} do
 6.     p_k^j ← b_k^j, k ∈ {1, ···, r}
 7. end for
 8. repeat
 9.     Select randomly an unchecked row i
10.     for j ∈ {1, ···, n} do
11.         if δ_{ij} = 1 then
12.             b_k^{C_i} ← b_k^{C_i} + p_k^j, k ∈ {1, ···, r}
13.         end if
14.     end for
15.     while ∃k ∈ {1, ···, r}, b_k^{C_i} < d_k^i do
16.         Select randomly an available agent a_j with the required resources
17.         δ_{ij} ← 1
18.         b_k^{C_i} ← b_k^{C_i} + p_k^j, k ∈ {1, ···, r}
19.     end while
20. until all rows have been checked
21. repeat
22.     Select randomly an unchecked column j
23.     for i ∈ {1, ···, m} do
24.         if δ_{ij} = 1 then
25.             b_k^{C_i} ← 0, k ∈ {1, ···, r}
26.             for j* ∈ {1, ···, n} do
27.                 if δ_{ij*} = 1 and j* ≠ j then
28.                     if ∃k ∈ {1, ···, r}, w_k^{j*i} > 0 then
29.                         b_k^{C_i} ← b_k^{C_i} + w_k^{j*i}, k ∈ {1, ···, r}
30.                     else
31.                         b_k^{C_i} ← b_k^{C_i} + p_k^{j*}, k ∈ {1, ···, r}
32.                     end if
33.                 end if
34.             end for
35.             w_k^{ji} ← d_k^i − b_k^{C_i}, k ∈ {1, ···, r}
36.             if ∀k ∈ {1, ···, r}, w_k^{ji} = 0 then
37.                 δ_{ij} ← 0
38.             end if
39.         end if
40.     end for
41.     while ∃k ∈ {1, ···, r}, Σ_{i=1}^{m} w_k^{ji} > p_k^j do
42.         Select randomly a row i with w_k^{ji} > 0 in the jth column
43.         δ_{ij} ← 0
44.         w_k^{ji} ← 0, k ∈ {1, ···, r}
45.         Recheck the ith row to ensure ∀k ∈ {1, ···, r}, b_k^{C_i} ≥ d_k^i
46.     end while
47.     p_k^j ← p_k^j − Σ_{i=1}^{m} w_k^{ji}, k ∈ {1, ···, r}
48. until all columns have been checked
49. for i ∈ {1, ···, m} do
50.     for j ∈ {1, ···, n} do
51.         if δ_{ij} = 1 then
52.             C_i ← C_i + a_j
53.         end if
54.     end for
55. end for
```

Fig. 3. The agent-oriented heuristic for individual repairs.

Proof: See Fig. 3, all the n columns need to be traveled to check whether there exist potential resource conflicts. First of all, there are at most m bits "1" in the jth column. When calculating each $\mathbf{W}_{ji}$, at total $(n-1)$ bits will be evaluated to register C_i's members and count r types of resources, so the number of operations required to estimate the total workload of a_j is $O(m \times n \times r)$. Next, it is clear that all the r types of resources must be checked to evaluate whether there exists

potential resource conflicts. If a resource conflict occurs, it is possible that $(m-1)$ bits "1" in the jth column will be reset to "0" to let a_j drop out of some coalitions. Once a bit $\delta_{i^*j} = 1$ is reset to "0", the i^*th row has to be rechecked and then at most $(n-1)$ agents and r types of resources will be recorded to ensure the feasibility of C_{i^*}, so the number of operations required to resolve potential resource conflicts over a_j is $O(r \times m \times n \times r)$. In brief, the worst case complexity of the AOH is $O(n \times (m \times n \times r + r \times m \times n \times r)) = O(m \times n^2 \times r^2)$ rather than $O(m \times n^2 \times r)$ suggested in [26]. ∎

In Fig. 3, it can be observed that to evaluate each agent a_j's resource contribution to every coalition, AOH has to recalculate each $\mathbf{B}_{C_i}$ which has been traveled and saved during row checks in advance. Moreover, once a_j exits from a coalition C_i, the the ith row must be checked again to ensure the feasibility of C_i. This may bring the following problems. First, once a conflict over a type of resource of a_j occurs, a_j has to exit from some coalitions, even if there is no conflict over other types of resources of a_j, which greatly reduces the utilization rate of a_j's resources. Second, it is possible that a_j is selected to join C_i again if it does have the required residual resources during the repairs of the successive columns. Then, the previous operation of a_j's exit from C_i becomes superfluous and meaningless. Last, and most importantly, if $\exists k \in \{1, ···, r\}, \sum_{j=1}^{n} b_k^j < \sum_{i=1}^{m} d_k^i$, a_j's exit from C_i may result in that it is impossible for C_i to be feasible even if the residual resources of all the available agents are considered. In this case, AOH would fall into an infinite loop and could not obtain an OCS.

The underlying cause for those drawbacks stated above is that AOH considers much about the two-dimensional binary encoding but ignores the inherent relationship among agents, coalitions, and tasks in the OCF model. In OCF, it is worth noting that once a coalition C_i with responsibility for t_i is formed, there may be a large number of feasible resource-allocation schemes for members in C_i to carry out t_i. However, the resource-allocation scheme is irrelative to $\upsilon(C_i)$ on the basis of (1), so whichever allocation scheme members choose, it will not change the value of C_i. Therefore, in the process of individual repairs, it would not be worth making the effort to spend so much time on estimating each agent's resource contribution for each task. Instead, we need to focus on maximizing the utilization rate of agents' resources and making the OCS satisfy the constraint sets (3) and (4). To this end, we present a generic task-oriented heuristic for individual repairs and try to overcome the above outlined disadvantages.

C. Task-Oriented Heuristic

As mentioned earlier, if a possible OCS imbedded in an individual is not repaired, we cannot know whether each coalition is feasible and whether each agent has sufficient resources to satisfy the needs of overlapping coalitions at the same time. Hence, to create a feasible OCS efficiently and effectively, we present a generic task-oriented heuristic (TOH) by dynamically tracking each agent's residual resources. The main idea is as follows. For each task t_i, if an agent a_j is chosen to join a coalition C_i with responsibility for t_i, we

```
Input: A, T, B₁, ⋯, Bₙ, D₁, ⋯, Dₘ
Output: C₁, ⋯, Cₘ
 1.  for i ∈ {1, ⋯, m} do
 2.      Cᵢ ← ∅
 3.      b_k^{Cᵢ} ← 0, k ∈ {1, ⋯, r}
 4.  end for
 5.  for j ∈ {1, ⋯, n} do
 6.      p_k^j ← b_k^j, k ∈ {1, ⋯, r}
 7.  end for
 8.  repeat
 9.      Select randomly an unchecked row i
10.      if ∃k ∈ {1, ⋯, r}, Σ_{j=1}^{n} p_k^j < d_k^i then
11.          for j ∈ {1, ⋯, n} do
12.              if δ_{ij} = 1 then
13.                  δ_{ij} ← 0
14.              end if
15.          end for
16.      else
17.          repeat
18.              Select randomly an unevaluated δ_{ij} = 1 in the ith row
19.              if ∀k ∈ {1, ⋯, r}, min{d_k^i − b_k^{Cᵢ}, p_k^j} = 0 then
20.                  δ_{ij} ← 0
21.              else
22.                  Cᵢ ← Cᵢ + a_j
23.                  w_k^{ji} ← min{d_k^i − b_k^{Cᵢ}, p_k^j}, k ∈ {1, ⋯, r}
24.                  p_k^j ← p_k^j − w_k^{ji}, k ∈ {1, ⋯, r}
25.                  b_k^{Cᵢ} ← b_k^{Cᵢ} + w_k^{ji}, k ∈ {1, ⋯, r}
26.              end if
27.          until all the bits "1" in the ith row are evaluated
28.          while ∃k ∈ {1, ⋯, r}, b_k^{Cᵢ} < d_k^i do
29.              Select randomly a bit with min{d_k^i − b_k^{Cᵢ}, p_k^j} > 0 in the ith row
30.              if δ_{ij} = 0 then
31.                  δ_{ij} ← 1
32.                  Cᵢ ← Cᵢ + a_j
33.              end if
34.              w_k^{ji} ← min{d_k^i − b_k^{Cᵢ}, p_k^j}
35.              p_k^j ← p_k^j − w_k^{ji}
36.              b_k^{Cᵢ} ← b_k^{Cᵢ} + w_k^{ji}
37.          end while
38.      end if
39.  until all the rows have been checked
```

Fig. 4. The task-oriented heuristic for individual repairs.

first maximize a_j's actual resource contribution to C_i and then update a_j's residual resources. After that, a_j can join other tasks only by using its residual resources. Once a_j has used up all the endowed resources, it will not respond any task.

A brief description of the main stages of the proposed TOH is displayed as follows. Check all the rows and evaluate the feasibility of each row (or coalition). For the ith row, we need to evaluate the availability of each agent for t_i. Specifically, if $\exists k \in \{1, \cdots, r\}, \sum_{j=1}^{n} p_k^j < d_k^i$, by Theorems 4 and 5 we know that it is impossible for C_i to be feasible, even if $C_i = A$. As a result, in the ith row, for each $\delta_{ij} = 1$, we set $\delta_{ij} \leftarrow 0$ to prevent any agent from carrying out t_i. If $\forall k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j \geq d_k^i$, it is certain that C_i can be repaired to be feasible for t_i. Then, for each $\delta_{ij} = 1$, we consider the following:

- If $\exists k \in \{1, \cdots, r\}$, $b_k^{C_i} < d_k^i$ and a_j has required resources, calculate $\mathbf{W}_{ji}$ (the amount of resources that a_j can contribute at most to t_i) according to

$$w_k^{ji} \leftarrow \min\{d_k^i - b_k^{C_i}, p_k^j\}, k \in \{1, \cdots, r\} \qquad (15)$$

and update a_j's residual resources, $p_k^j \leftarrow p_k^j - w_k^{ji}$, $k \in \{1, \cdots, r\}$.
- If $\exists k \in \{1, \cdots, r\}$, $b_k^{C_i} < d_k^i$ but a_j has no available resource for t_i, a_j is useless for C_i and then let a_j exit from C_i by setting $\delta_{ij} \leftarrow 0$.
- If $\forall k \in \{1, \cdots, r\}$, $b_k^{C_i} = d_k^i$, C_i has been feasible and a_j is superfluous for C_i, so we set $\delta_{ij} \leftarrow 0$ to exclude a_j from C_i and reduce the size of C_i.

After that, if $\exists k \in \{1, \cdots, r\}$, $b_k^{C_i} < d_k^i$ still holds, we evaluate the availability of bits "0" in the ith row and select available agents from $A - C_i$ with the required resources to join C_i until $\forall k \in \{1, \cdots, r\}$, $b_k^{C_i} = d_k^i$.

The pseudo-code of the TOH is presented in Fig. 4. It can be observed that before adjusting C_i for t_i, we first evaluate whether the residual resources of agents can satisfy the needs of t_i to avoid the useless repair operations and possible infinite loop that occurs in AOH. Beyond that, a_j will join C_i only if a_j has the available residual resources required by t_i, which can effectively prevent conflicts over agents' resources. Moreover, once a_j is evaluated to be available for t_i, a_j will do its utmost to contribute its residual resources to t_i. On the contrary, in AOH each agent a_j tries to contribute as few resources as possible to t_i. Therefore, TOH is good at maximizing the utilization rate of agents' resources, reducing the size of C_i, and thus increasing the value of C_i. Finally, it is clear that in the OCS repaired by TOH no unassigned task can be satisfied by the residual resources of agents, so by Theorem 6, we know that TOH can make the created OCS reach as close as possible to the optimal.

On the other hand, it should be noted that in TOH each infeasible coalition is repaired in a random manner. The reason is that MHAs are easy to fall into local optimum, especially dealing with huge amount of data [43]. Repairing agents and coalitions randomly in an individual can maintain diversity in the population to a certain extent and make the MHAs escape from the local optima and explore solutions as fully as possible [43].

In addition to the above, we now further analyze the difficulty of repairing an individual in TOH and whether the corresponding OCS can be repaired to be feasible. We have the following results.

Theorem 8: The worst case complexity of the TOH for individual repairs is $O(m \times n \times r)$.

Proof: See Fig. 4, in TOH at most m rows (or tasks) are checked. Once a row i is selected, we first need to evaluate whether the total residual resources of all the n agents can satisfy the needs of t_i. Because each type of resources needs to be checked, the number of operations required for the evaluation is $O(n \times r)$. If $\exists k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j < d_k^i$, there are at most n bits "1" that may be set to "0". If $\forall k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j \geq d_k^i$, there are at most n bits "1" that may be evaluated. For each $\delta_{ij} = 1$, we need to reuse all the r types of resources to estimate $\mathbf{W}_{ji}$. As a result, the number of operations required to evaluate bits "1" in the ith row is $O(n \times r)$. After that, if $\exists k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j < d_k^i$

d_k^i	μ_i	b_k^j	π_{jj*}
[0,100]	[50,1000]	[0,10]	[0,10]

still holds, we need to select at most n agents to help C_i. Since at most r types of resources should be checked, the number of operations required to select available agents is also $O(n \times r)$. To sum up, the worst case complexity of the TOH is $O(m \times (n \times r + n + n \times r + n \times r))$, namely, $O(m \times n \times r)$. ∎

Theorem 9: Any individual can be repaired to contain a feasible OCS by TOH.

Proof: We show the implications separately. We first prove that there certainly exists a feasible OCS in an individual after it is repaired. Next, we prove that the final OCS repaired by TOH is feasible.

In TOH t_i can be assigned only if the resource needs of t_i can be satisfied by the total residual resources of all the agents, namely, $\forall k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j \geq d_k^i$. For each feasible coalition C_i and each assigned task t_i in the repaired individual, we set $C^* = \bigcup_{i=1}^{m} C_i$ and $T^* = \bigcup_{i=1, C_i \neq \emptyset}^{m} \{t_i\}$. Then, it is clear that C^* and T^* satisfy (13). According to Theorem 5, we obtain that there certainly exists a feasible OCS on the basis of TOH.

On the other hand, as long as $\forall k \in \{1, \cdots, r\}$, $\sum_{j=1}^{n} p_k^j \geq d_k^i$, C_i will be formed to carry out t_i; otherwise, $C_i = \emptyset$. Additionally, each agent a_j will not join C_i unless a_j has the residual resources required by t_i. Hence, according to (15), we can easily obtain that each feasible coalition C_i for t_i in an individual satisfies (3) and each a_j that has joined tasks satisfies (4). That is, the repaired OCS is feasible. ∎

V. EMPIRICAL RESULTS

To verify the proposed TOH, we now present empirical results against AOH on the basis of three popular binary MHAs: genetic algorithm (GA) [35], BPSO [39], and binary differential evolution (BDE) [45]. To make the comparisons as fair as possible, we adopt the basic algorithmic parameters of GA, BPSO, and BDE that were recommended in [35], [39], [45]. In common, the population size is 30 and the maximal generation number is 500. Additionally, the crossover and mutation probabilities in GA are 0.9 and 0.1, respectively; in BPSO, the two learning factors are both set to 2.0 and the maximal velocity of each particle is 5.0; in BDE, the crossover and mutation rates are 0.25 and 1.0, respectively.

Particularly, we test TOH and AOH in the following three classical experimental environments:

- The relaxed environment where the sum of the available resources of all agents is more than the sum of the required resources of all tasks, namely, $\sum_{j=1}^{n} b_k^j > \sum_{i=1}^{m} d_k^i$, $k \in \{1, \cdots, r\}$.

Instance	GA		BPSO		BDE	
	AOH	TOH	AOH	TOH	AOH	TOH
No.1	0.39±0.01	0.15±0.01	0.37±0.03	0.16±0.02	0.43±0.03	0.14±0.01
No.2	0.47±0.04	0.17±0.02	0.40±0.04	0.18±0.02	0.43±0.03	0.15±0.02
No.3	0.45±0.03	0.15±0.01	0.40±0.04	0.17±0.02	0.39±0.03	0.15±0.01
No.4	0.50±0.04	0.15±0.02	0.47±0.04	0.21±0.02	0.47±0.04	0.17±0.02
No.5	0.49±0.04	0.15±0.01	0.46±0.05	0.20±0.02	0.45±0.03	0.16±0.02
No.6	0.51±0.04	0.16±0.01	0.44±0.03	0.18±0.02	0.45±0.03	0.18±0.02
No.7	0.54±0.04	0.16±0.01	0.44±0.04	0.21±0.02	0.48±0.04	0.20±0.02
No.8	0.49±0.04	0.16±0.01	0.45±0.05	0.19±0.01	0.45±0.03	0.19±0.02
No.9	0.52±0.05	0.17±0.01	0.45±0.04	0.20±0.02	0.49±0.05	0.16±0.01
No.10	0.48±0.04	0.14±0.01	0.40±0.04	0.16±0.02	0.43±0.03	0.18±0.02
No.11	0.49±0.04	0.17±0.02	0.44±0.04	0.18±0.02	0.46±0.04	0.19±0.02
No.12	0.48±0.03	0.15±0.01	0.35±0.04	0.17±0.01	0.38±0.04	0.19±0.02
No.13	0.52±0.04	0.16±0.01	0.45±0.04	0.20±0.02	0.48±0.04	0.18±0.01
No.14	0.51±0.04	0.17±0.01	0.41±0.04	0.18±0.03	0.43±0.03	0.19±0.02
No.15	0.52±0.04	0.15±0.02	0.41±0.04	0.19±0.02	0.45±0.03	0.20±0.02
No.16	0.46±0.04	0.16±0.01	0.40±0.03	0.18±0.02	0.39±0.03	0.19±0.02
No.17	0.51±0.04	0.16±0.02	0.42±0.04	0.19±0.02	0.46±0.03	0.20±0.02
No.18	0.56±0.04	0.16±0.01	0.44±0.04	0.19±0.02	0.51±0.03	0.21±0.03
No.19	0.52±0.04	0.16±0.01	0.44±0.04	0.18±0.02	0.45±0.03	0.19±0.02
No.20	0.53±0.04	0.15±0.01	0.39±0.03	0.18±0.02	0.45±0.03	0.20±0.02
No.21	0.53±0.04	0.16±0.01	0.40±0.04	0.17±0.02	0.41±0.04	0.20±0.02
No.22	0.49±0.04	0.16±0.01	0.39±0.04	0.19±0.02	0.50±0.01	0.19±0.02
No.23	0.54±0.04	0.15±0.01	0.49±0.04	0.21±0.02	0.57±0.02	0.20±0.02
No.24	0.53±0.04	0.16±0.01	0.45±0.04	0.13±0.01	0.53±0.02	0.20±0.02
No.25	0.54±0.05	0.17±0.01	0.47±0.04	0.15±0.01	0.54±0.02	0.19±0.02
No.26	0.57±0.05	0.16±0.01	0.46±0.04	0.16±0.01	0.62±0.01	0.22±0.03
No.27	0.51±0.04	0.16±0.01	0.39±0.03	0.12±0.01	0.47±0.02	0.19±0.02
No.28	0.51±0.04	0.16±0.01	0.44±0.04	0.12±0.01	0.51±0.03	0.19±0.02
No.29	0.52±0.04	0.15±0.01	0.42±0.03	0.13±0.02	0.57±0.03	0.21±0.02
No.30	0.53±0.04	0.15±0.01	0.46±0.04	0.13±0.01	0.51±0.02	0.20±0.02

- The tight environment in which the sum of the available resources of all agents is just equal to the sum of the required resources of all tasks, namely, $\sum_{j=1}^{n} b_k^j = \sum_{i=1}^{m} d_k^i$, $k \in \{1, \cdots, r\}$.
- The harsh environment in which the sum of the available resources of all agents is smaller than the sum of the required resources of all tasks, namely, $\sum_{j=1}^{n} b_k^j < \sum_{i=1}^{m} d_k^i$, $k \in \{1, \cdots, r\}$.

Following the studies in [26], [35], [39], we generate every OCF instance randomly in the above three experimental environments according to the range of parameters listed in Table I. Beyond that, each OCF instance is tested repeatedly for 30 independent runs with different random seeds on a PC with Intel 2.50 GHz CPU and 10 GB of RAM.

A. Impact of the Repair Operations

In the process of individual repairs, setting bits "0" to "1" (i.e., selecting available agents to join a coalition C_i) will increase the size of C_i, bring extra communication cost, and thus decrease the value of C_i. In contrast, setting bits "1" to "0" (i.e., winkling unavailable agents out of C_i) will reduce the size of C_i and thus increase the value of C_i. It is clear that the repair operations will affect the value of the created

(a) In the relaxed environment

(b) In the tight environment
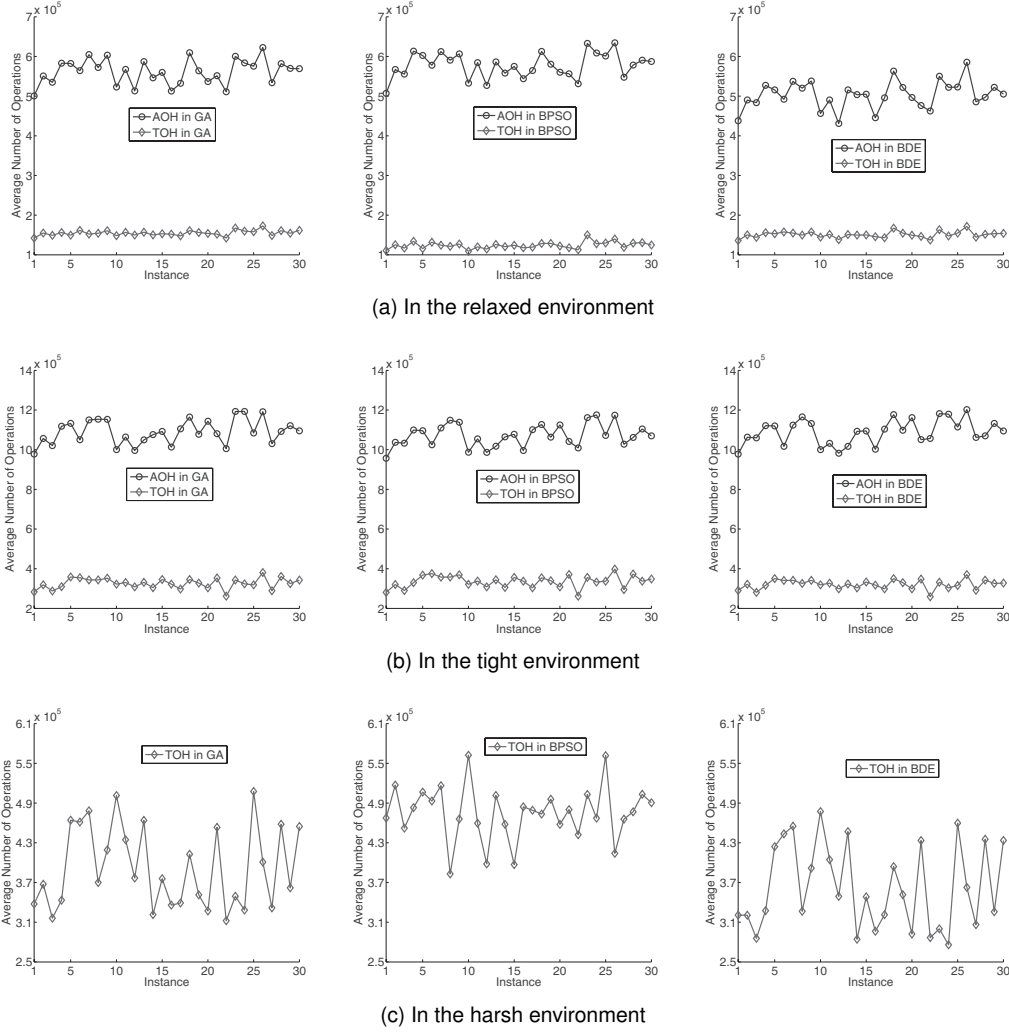

(c) In the harsh environment

Fig. 5. The average numbers of the repair operations executed by TOH and AOH.

OCS. Therefore, in the first experiment we analyze the impact of the repair operations in TOH and AOH.

Now, in terms of 20 agents (i.e., $n = 20$), 5 tasks (i.e., $m = 5$), and 5 resources (i.e., $r = 5$), we generated 30 different test instances randomly on the basis of the pre-defined intervals in Table I, considering the relaxed, the tight, and the harsh environments, respectively. Moreover, for the generated instances, there are more than two levels of the relaxed and harsh environments.

Fig. 5 shows the average numbers of the repair operations executed by TOH and AOH in GA, BPSO, and BDE. It can be observed that in both the relaxed and tight environments, the numbers of the repair operations required by TOH are much smaller than those required by AOH. In the harsh environment, the possible resource conflicts occur more often, so AOH fell easily into an infinite loop, while TOH executed less than 600,000 repair operations in each search algorithm. Like what we had initially expected, on average, TOH required far fewer repair operations in all the 90 OCF instances. The reason is that on the basis of (15), TOH can make each agent contribute the most amounts of its resources to a coalition, which reduces the chance of other agents joining the coalition.

On the contrary, AOH makes each agent contribute the least amounts of its resources to a coalition, so AOH has to select a number of other available agents to join the coalition for satisfying the corresponding task.

The above observations also imply the efficiency of TOH in comparison with AOH. Specifically, Tables II-IV show the mean run time ($\pm 95\%$ confidence interval of the mean [46], in seconds) consumed by TOH and AOH with different instances in the three environments. Note that here for fairness, we only count the total time consumed by TOH and AOH rather than the total running time of the search algorithm. As can be seen from Tables II-IV, TOH consumed less time than AOH on all the 90 OCF instances. Note that in the harsh environment AOH fell into an infinite loop, because no available agent could be selected to join an infeasible coalition when some members exited from the coalition. In contrast, TOH has a clear way to avoid impossible tasks by estimating the total amount of the residual resources of all the agents.

Tables V-VII show the coalition structure values (mean and standard deviation) obtained by TOH and AOH with different instances in the three environments. It can be found that the coalition structure values obtained by TOH are higher than

TABLE III
MEAN RUN TIME (±95% CONFIDENCE INTERVAL OF THE MEAN, IN SECONDS) CONSUMED BY TOH AND AOH WITH DIFFERENT INSTANCES IN THE TIGHT ENVIRONMENT

Instance	GA		BPSO		BDE	
	AOH	TOH	AOH	TOH	AOH	TOH
No.1	0.99±0.07	0.21±0.01	0.87±0.06	0.18±0.02	0.78±0.02	0.27±0.02
No.2	1.08±0.07	0.20±0.01	1.04±0.07	0.18±0.02	0.84±0.02	0.24±0.02
No.3	0.99±0.07	0.17±0.02	0.94±0.07	0.17±0.01	0.82±0.02	0.24±0.02
No.4	1.09±0.08	0.21±0.02	1.05±0.07	0.18±0.01	0.94±0.06	0.25±0.02
No.5	1.11±0.08	0.20±0.01	1.01±0.08	0.20±0.02	1.02±0.07	0.29±0.02
No.6	1.05±0.07	0.20±0.01	0.94±0.06	0.20±0.02	0.93±0.07	0.26±0.02
No.7	1.08±0.08	0.19±0.02	1.07±0.08	0.20±0.02	1.10±0.08	0.28±0.02
No.8	1.14±0.07	0.20±0.01	0.96±0.06	0.20±0.02	1.07±0.08	0.25±0.02
No.9	1.09±0.08	0.19±0.01	1.07±0.08	0.18±0.01	1.15±0.08	0.24±0.02
No.10	0.95±0.07	0.19±0.01	0.93±0.08	0.19±0.02	0.96±0.07	0.29±0.03
No.11	1.07±0.08	0.27±0.02	0.97±0.07	0.18±0.01	1.05±0.08	0.26±0.03
No.12	0.97±0.07	0.26±0.03	0.94±0.07	0.18±0.01	0.95±0.07	0.23±0.02
No.13	1.05±0.08	0.25±0.03	0.99±0.07	0.19±0.01	1.06±0.08	0.26±0.03
No.14	1.03±0.08	0.26±0.03	0.99±0.08	0.19±0.01	1.09±0.08	0.24±0.03
No.15	1.06±0.08	0.26±0.02	1.06±0.08	0.17±0.02	1.03±0.08	0.27±0.03
No.16	0.97±0.07	0.27±0.03	0.90±0.06	0.20±0.02	0.94±0.07	0.26±0.04
No.17	1.07±0.08	0.29±0.02	1.02±0.07	0.18±0.02	1.14±0.08	0.26±0.02
No.18	1.14±0.08	0.27±0.02	1.07±0.08	0.18±0.02	1.12±0.08	0.25±0.03
No.19	1.07±0.07	0.28±0.02	1.01±0.07	0.19±0.01	1.08±0.07	0.29±0.03
No.20	1.05±0.08	0.28±0.03	1.10±0.07	0.18±0.01	1.11±0.08	0.26±0.03
No.21	1.05±0.07	0.26±0.02	0.99±0.07	0.18±0.01	1.04±0.08	0.27±0.03
No.22	0.98±0.09	0.25±0.03	0.99±0.07	0.17±0.01	1.03±0.07	0.23±0.02
No.23	1.12±0.08	0.29±0.03	1.09±0.08	0.18±0.02	1.16±0.08	0.25±0.02
No.24	1.05±0.08	0.26±0.02	1.08±0.07	0.17±0.01	1.08±0.08	0.27±0.03
No.25	1.05±0.07	0.31±0.03	1.02±0.07	0.19±0.01	1.07±0.08	0.26±0.02
No.26	1.21±0.09	0.30±0.03	1.15±0.08	0.19±0.02	1.16±0.09	0.29±0.02
No.27	0.99±0.08	0.28±0.03	0.93±0.08	0.17±0.01	1.08±0.07	0.26±0.02
No.28	1.06±0.08	0.27±0.03	1.01±0.07	0.20±0.02	1.08±0.08	0.26±0.02
No.29	1.14±0.09	0.28±0.02	1.04±0.07	0.19±0.01	1.13±0.08	0.26±0.03
No.30	1.00±0.07	0.29±0.02	0.99±0.08	0.19±0.01	1.09±0.08	0.27±0.03

TABLE IV
MEAN RUN TIME (±95% CONFIDENCE INTERVAL OF THE MEAN, IN SECONDS) CONSUMED BY TOH WITH DIFFERENT INSTANCES IN THE HARSH ENVIRONMENT

Instance	GA	BPSO	BDE
	TOH	TOH	TOH
No.1	0.22±0.02	0.14±0.01	0.24±0.02
No.2	0.21±0.02	0.15±0.01	0.20±0.02
No.3	0.23±0.02	0.14±0.02	0.19±0.02
No.4	0.24±0.03	0.15±0.01	0.21±0.02
No.5	0.21±0.02	0.16±0.01	0.18±0.02
No.6	0.23±0.03	0.15±0.01	0.24±0.02
No.7	0.20±0.02	0.15±0.01	0.19±0.03
No.8	0.18±0.02	0.14±0.02	0.17±0.02
No.9	0.20±0.02	0.12±0.01	0.20±0.02
No.10	0.22±0.02	0.15±0.01	0.19±0.02
No.11	0.22±0.02	0.16±0.01	0.19±0.02
No.12	0.21±0.02	0.14±0.01	0.18±0.01
No.13	0.23±0.03	0.15±0.01	0.20±0.02
No.14	0.21±0.02	0.14±0.01	0.17±0.01
No.15	0.21±0.02	0.15±0.01	0.19±0.01
No.16	0.19±0.02	0.14±0.01	0.17±0.01
No.17	0.22±0.02	0.15±0.01	0.18±0.02
No.18	0.20±0.02	0.12±0.01	0.17±0.01
No.19	0.24±0.03	0.17±0.01	0.20±0.02
No.20	0.23±0.02	0.13±0.01	0.17±0.02
No.21	0.23±0.03	0.14±0.01	0.21±0.02
No.22	0.19±0.02	0.15±0.01	0.20±0.02
No.23	0.22±0.02	0.16±0.02	0.19±0.02
No.24	0.16±0.02	0.11±0.01	0.16±0.02
No.25	0.21±0.02	0.15±0.01	0.20±0.02
No.26	0.19±0.02	0.13±0.01	0.18±0.01
No.27	0.24±0.02	0.14±0.01	0.21±0.02
No.28	0.24±0.02	0.14±0.01	0.20±0.02
No.29	0.24±0.02	0.15±0.02	0.21±0.03
No.30	0.21±0.02	0.14±0.01	0.21±0.02

those obtained by AOH on all the 90 OCF instances. Moreover, we used the Wilcoxon rank-sum test [47] at a 0.05 significance level to measure the significance of the differences between the results obtained by TOH and AOH. We found that all the differences between TOH and AOH are highly significant in both the tight and the harsh environments. Particularly, in the harsh environment TOH still obtained considerable coalition structure values, while AOH failed in exploration for the reason that all the AOH entries are not available.

To summarize, in the relaxed environment TOH is slightly better than AOH, but in both the tight and the harsh environments, TOH significantly outperforms AOH on the runtime and the coalition structure value. The repair operations in TOH are simple and effective, can present clear heuristic information, and enable GA, BPSO, and BDE to explore high-value overlapping coalitions quickly.

B. Impact of the Problem Size

AOH does not work in the harsh environment. Hence, here we consider the impact of the problem size (i.e., different numbers of agents, tasks, and resources) to further show the differences between TOH and AOH in both the relaxed and the tight environments. Theorem 7 shows that AOH is more sensitive to the values of n and r than the value of m. In addition, Theorem 8 implies that TOH is not sensitive to

the values of m, n, and r. To verify the theoretical results in Theorems 7 and 8 and discuss the potential application scenarios of the AOH and TOH, in this section we tested the impact of one factor with the other factors fixed and taken small values. This test method was used in [26], [29], [39] to reduce the interference of the other factors over the experimental results, when one factor was evaluated. The reason is that if all the testing instances are generated with random values of m, n, and r, it is not known which factors contribute to the change of the experimental results.

1) Different numbers of agents: First of all, we evaluated the performance of the TOH and AOH with different numbers of agents (i.e., n, ranging from 30 to 100) and the fixed numbers of tasks and resources (i.e., $m = 5$ and $r = 5$).

Fig. 6 shows the average time consumption (in seconds) of TOH and AOH for different n. It can be found that in both the relaxed and the tight environments, TOH is much faster than AOH and the gap between TOH and AOH is greatly widening with the increase of n. For example, in the relaxed environment TOH took less than 38% and 43% of the time consumed by AOH when $n = 30$ and $n = 100$, respectively. In the tight environment, TOH took less than only 20% and 11% of the time when $n = 30$ and $n = 100$, respectively. The above observations imply that AOH is very sensitive to n in the tight environment, which coincides with the results in Theorem 7.

TABLE V
COALITION STRUCTURE VALUES (MEAN AND STANDARD DEVIATION) OBTAINED BY TOH AND AOH WITH DIFFERENT INSTANCES IN THE RELAXED ENVIRONMENT

Instance	GA		BPSO		BDE	
	AOH	TOH	AOH	TOH	AOH	TOH
No.1	1571(15)	1608(6)	1584(20)	1612(9)	1593(10)	1602(6)
No.2	1380(15)	1428(7)	1384(22)	1436(11)	1402(10)	1419(6)
No.3	1277(13)	1323(6)	1283(18)	1327(6)	1294(9)	1313(6)
No.4	1377(15)	1423(9)	1380(21)	1436(12)	1398(11)	1415(8)
No.5	1398(17)	1464(9)	1392(26)	1463(9)	1416(10)	1447(10)
No.6	1513(16)	1578(10)	1514(18)	1582(9)	1539(11)	1567(6)
No.7	1346(18)	1413(8)	1353(19)	1422(9)	1379(13)	1403(6)
No.8	1313(17)	1360(10)	1311(23)	1370(10)	1335(14)	1350(8)
No.9	1102(19)	1169(10)	1112(23)	1176(10)	1134(12)	1156(8)
No.10	1266(12)	1304(6)	1279(21)	1312(6)	1285(9)	1297(6)
No.11	1123(15)	1181(10)	1131(22)	1196(10)	1153(11)	1171(7)
No.12	1155(13)	1190(8)	1160(19)	1200(6)	1176(8)	1185(9)
No.13	1196(23)	1249(7)	1203(23)	1261(7)	1213(12)	1243(8)
No.14	1304(15)	1352(9)	1303(28)	1355(8)	1319(15)	1339(8)
No.15	1319(13)	1367(8)	1321(27)	1382(14)	1342(14)	1360(10)
No.16	1201(16)	1241(8)	1200(23)	1249(9)	1220(10)	1231(7)
No.17	1189(14)	1232(9)	1179(25)	1240(8)	1197(10)	1221(8)
No.18	1318(24)	1388(10)	1324(25)	1395(9)	1342(14)	1367(8)
No.19	1480(20)	1538(9)	1488(28)	1546(8)	1497(15)	1525(9)
No.20	1199(15)	1233(8)	1207(25)	1241(9)	1212(10)	1221(8)
No.21	1425(17)	1471(7)	1434(17)	1479(7)	1448(9)	1467(5)
No.22	1466(13)	1493(7)	1465(16)	1505(5)	1474(8)	1487(7)
No.23	1193(19)	1253(7)	1187(23)	1267(11)	1209(13)	1246(9)
No.24	1310(16)	1369(11)	1313(21)	1378(8)	1331(16)	1354(8)
No.25	1346(18)	1394(7)	1343(21)	1403(7)	1367(9)	1383(7)
No.26	1465(23)	1531(12)	1470(28)	1545(12)	1484(16)	1518(11)
No.27	1690(11)	1720(5)	1700(16)	1730(5)	1705(8)	1714(5)
No.28	1210(18)	1275(8)	1222(20)	1281(7)	1241(11)	1267(7)
No.29	1483(20)	1544(9)	1482(25)	1550(8)	1503(15)	1532(9)
No.30	1259(17)	1301(7)	1258(26)	1311(7)	1277(9)	1294(8)

TABLE VI
COALITION STRUCTURE VALUES (MEAN AND STANDARD DEVIATION) OBTAINED BY TOH AND AOH WITH DIFFERENT INSTANCES IN THE TIGHT ENVIRONMENT

Instance	GA		BPSO		BDE	
	AOH	TOH	AOH	TOH	AOH	TOH
No.1	1218(46)	1370(15)	1224(46)	1374(19)	1206(28)	1342(19)
No.2	990(55)	1183(20)	964(42)	1178(27)	961(37)	1137(18)
No.3	966(44)	1093(12)	928(60)	1092(22)	906(31)	1065(11)
No.4	941(48)	1129(17)	926(67)	1118(32)	911(45)	1092(24)
No.5	913(56)	1164(22)	882(75)	1169(22)	884(38)	1147(18)
No.6	1110(29)	1318(23)	1106(63)	1306(21)	1135(34)	1302(17)
No.7	832(57)	1141(16)	802(68)	1122(25)	816(47)	1106(23)
No.8	788(42)	1039(17)	728(79)	1025(23)	770(41)	1011(18)
No.9	601(56)	875(18)	573(89)	859(25)	599(45)	855(25)
No.10	916(49)	1077(15)	882(72)	1081(25)	894(45)	1050(19)
No.11	708(41)	949(23)	680(72)	942(38)	704(51)	916(18)
No.12	822(54)	974(23)	797(54)	976(21)	821(39)	949(10)
No.13	827(53)	1035(23)	838(72)	1022(25)	844(45)	1011(15)
No.14	808(50)	1064(24)	774(51)	1065(42)	773(60)	1013(23)
No.15	845(47)	1075(22)	817(100)	1069(27)	785(47)	1051(16)
No.16	811(41)	987(15)	775(52)	975(22)	800(33)	961(13)
No.17	644(42)	883(24)	601(40)	878(25)	599(30)	847(19)
No.18	783(64)	1105(28)	750(70)	1096(39)	761(61)	1059(29)
No.19	1055(42)	1218(21)	1031(68)	1208(28)	1026(66)	1177(23)
No.20	623(54)	895(22)	568(75)	896(32)	575(47)	871(21)
No.21	934(48)	1208(22)	930(62)	1198(28)	930(36)	1176(16)
No.22	1133(35)	1264(15)	1097(35)	1264(27)	1078(32)	1224(15)
No.23	592(48)	942(18)	547(88)	926(20)	554(55)	919(17)
No.24	744(51)	1074(20)	698(56)	1061(28)	708(38)	1039(19)
No.25	980(45)	1131(17)	935(58)	1123(17)	934(41)	1115(19)
No.26	869(44)	1161(24)	836(87)	1154(35)	821(47)	1133(28)
No.27	1384(34)	1493(21)	1351(53)	1502(22)	1350(32)	1469(17)
No.28	728(47)	1010(19)	748(78)	998(20)	753(52)	995(15)
No.29	953(64)	1237(21)	919(71)	1230(31)	905(46)	1203(24)
No.30	838(44)	1041(17)	778(69)	1033(20)	807(55)	1020(17)

This is because in the tight environment, AOH has to check each column (which represents an agent) to resolve intense resource conflicts, and thus with the increase of n, a large amount of repeated and redundant operations are produced, which progressively increases the time consumption of the AOH.

Fig. 7 illustrates the average coalition structure values obtained by TOH and AOH for different n. As can be seen, with the increase of n, the difference between TOH and AOH in the relaxed environment is slight. However, in the tight environment, the coalition structure value obtained by TOH is much higher than that obtained by AOH, and the coalition structure values obtained by TOH or AOH have shown a sharp decline with the increase of n. Moreover, when $n \geq 80$, AOH failed in exploring feasible coalitions. The reason for this is that in the tight environment all the agents have to join coalitions to accomplish all the given tasks, and with the increase of n, each coalition contains a large number of members, which results in high communication cost and thus reduces the coalition structure value greatly. Additionally, in the relaxed environment the coalition structure values given by both TOH and AOH decreased slowly with the increase of n. An explanation is that the number of possible coalitions is $2^n - 1$ [31], and with the increase of n, the search abilities of GA, BPSO, and BDE may weaken gradually in high-

dimensional spaces [43].

2) Different numbers of tasks: Here, given fixed numbers of agents and resources (i.e., $n = 20$ and $r = 5$), we tested the performance of the TOH and AOH for different numbers of tasks (i.e., m, ranging from 10 to 45).

Fig. 8 depicts the average time consumption (in seconds) of the TOH and AOH for different m. It can be observed that although TOH is always faster than AOH, the difference between TOH and AOH increases slowly. For example, in the relaxed environment TOH took less than 65% and 57% of the time consumed by AOH when $m = 10$ and $m = 45$, respectively. In the tight environment, TOH took less than 36% and 52% of the time when $m = 10$ and $m = 45$, respectively. The above observations reflect that neither of TOH and AOH is sensitive to m. The reason is that in both TOH and AOH checking each row is relatively simple and the potential resource conflicts occur only in each column.

Fig. 9 shows the average coalition structure values obtained by TOH and AOH for different m. It can be found that with the increase of m, the average coalition structure values obtained by both TOH and AOH soar. This is because the increase of m brings more rewards, but the numbers of agents and resources are fixed and thus the total communication cost is bounded, which results in the continuous increase of the coalition structure value. Note that in the tight environment the

TABLE VII
COALITION STRUCTURE VALUES (MEAN AND STANDARD DEVIATION)
OBTAINED BY TOH WITH DIFFERENT INSTANCES IN THE HARSH
ENVIRONMENT

Instance	GA TOH	BPSO TOH	BDE TOH
No.1	762(11)	763(13)	733(18)
No.2	642(15)	644(20)	618(16)
No.3	688(12)	687(18)	665(10)
No.4	672(15)	664(18)	630(13)
No.5	576(16)	570(21)	551(18)
No.6	822(14)	815(16)	790(17)
No.7	651(21)	642(21)	608(20)
No.8	526(7)	523(8)	517(8)
No.9	363(21)	372(26)	353(19)
No.10	613(23)	595(20)	570(20)
No.11	432(36)	427(35)	393(17)
No.12	502(15)	496(19)	482(15)
No.13	575(22)	545(25)	529(21)
No.14	736(13)	739(11)	720(10)
No.15	540(16)	534(15)	517(14)
No.16	550(11)	547(15)	528(13)
No.17	588(18)	590(23)	551(14)
No.18	437(24)	451(37)	421(24)
No.19	658(15)	649(21)	610(20)
No.20	535(11)	535(14)	511(12)
No.21	629(16)	617(22)	604(16)
No.22	841(11)	847(11)	827(11)
No.23	561(17)	549(22)	534(12)
No.24	485(4)	485(5)	480(5)
No.25	598(27)	584(32)	559(22)
No.26	513(13)	513(14)	492(14)
No.27	876(21)	875(16)	854(13)
No.28	445(11)	446(19)	431(9)
No.29	704(22)	698(25)	671(19)
No.30	531(20)	511(22)	479(22)

coalition structure value obtained by TOH is higher than that obtained by AOH. In the relaxed environment the difference between TOH and AOH is slight when m is reasonably small. However, with the increase of m, the difference also increases. An explanation is that when m is small, each task can be easily satisfied by agents' sufficient resources, so the effect of the repair heuristic is limited. On the other hand, the increase of m reduces the gap between agents' total resources and tasks's required resources, while the exploration ability of TOH in the harsh situation is stronger than that of AOH.

3) Different numbers of resources: Finally, given the fixed numbers of agents and tasks (i.e., $n = 20$ and $m = 5$), we analyzed the performance of the TOH and AOH for different numbers of resources (i.e., r, ranging from 10 to 45).

Fig. 10 illustrates the average time consumption (in seconds) of the TOH and AOH for different r. As can be seen from Fig. 10, whether in the relaxed or tight environment, TOH is much faster than AOH and the gap between TOH and AOH keeps widening with the increase of r. For example, in the relaxed environment TOH took less than 37% and 29% of the time consumed by AOH when $r = 10$ and $r = 45$, respectively. In the tight environment, TOH took less than only 21% and 18% of the time when $r = 10$ and $r = 45$, respectively. The above observations indicate that AOH is very sensitive to r, which is in line with the computational complexity results in Theorem 7.

Fig. 11 shows the average coalition structure values obtained by TOH and AOH for different r. It can be observed that with the increase of r, the difference between TOH and AOH increases greatly in both the relaxed and the tight environments. Particularly, in the tight environment when $r \geq 30$, AOH failed in exploring feasible coalitions, but TOH still obtained high coalition structure values. The reason for this is that in AOH each agent has to use as few resources as possible when it is selected to join a coalition, and thus AOH needs to select many alternative members to satisfy the corresponding task. On the contrary, in TOH each agent is expected to do its utmost to contribute its residual resources to a coalition. Accordingly, with the increase of r, a coalition obtained by AOH will contain more members, which results in a lot of extra commutation cost and thus a sharp decrease of the coalition structure value.

On the whole, the reliability and effectiveness of the AOH and TOH are validated from the empirical results on the basis of different numbers of agents, tasks, and resources in both the relaxed and the tight environments. We find that the coalition structure value in OCF is closely related to the numbers of agents and tasks, but has little relation with the number of resources.

The proposed TOH is feasible and efficient no matter how n, m, and r change, which justifies the utility of TOH in practice, especially in large-scale domains. Specifically, when the number of agents or resources is big, TOH is far more effective than AOH no matter whether on the coalition structure value or on the time consumption. This indicates that TOH can support stronger heuristic information to guide the evolution of the MHAs. On the contrary, AOH is sensitive to n and r. When there are too many agents or resources, the performance of the AOH will decrease much. Yet, one bright spot for AOH is that it was affected insignificantly by the growing numbers of tasks. Accordingly, AOH may be serviceable in special scenarios where there are a large number of tasks but few agents and resources.

VI. CONCLUSION

Coalition structure generation is one of the most important challenging problems in multi-agent systems [6]. Apart from the traditional non-overlapping coalition formation, very little work exists on generation of overlapping coalitions. This is because the search space of overlapping coalitions is far larger than that of non-overlapping coalitions [20]. In this paper, we discussed OCF in resource-constrained and subadditive task oriented domains and examined the complexity of the overlapping coalition structure generation (OCSG) problem. After that, to uncover the overlapping coalition structure, we proposed a generic task-oriented heuristic (called TOH) which can be imbedded in binary meta-heuristic algorithms. Through extensive comparative experiments on the basis of GA, BPSO, and BDE in the relaxed, tight, and harsh environments with varied numbers of agents, tasks, and resources, respectively, the proposed TOH is shown to be promising in solving the OCSG problem.
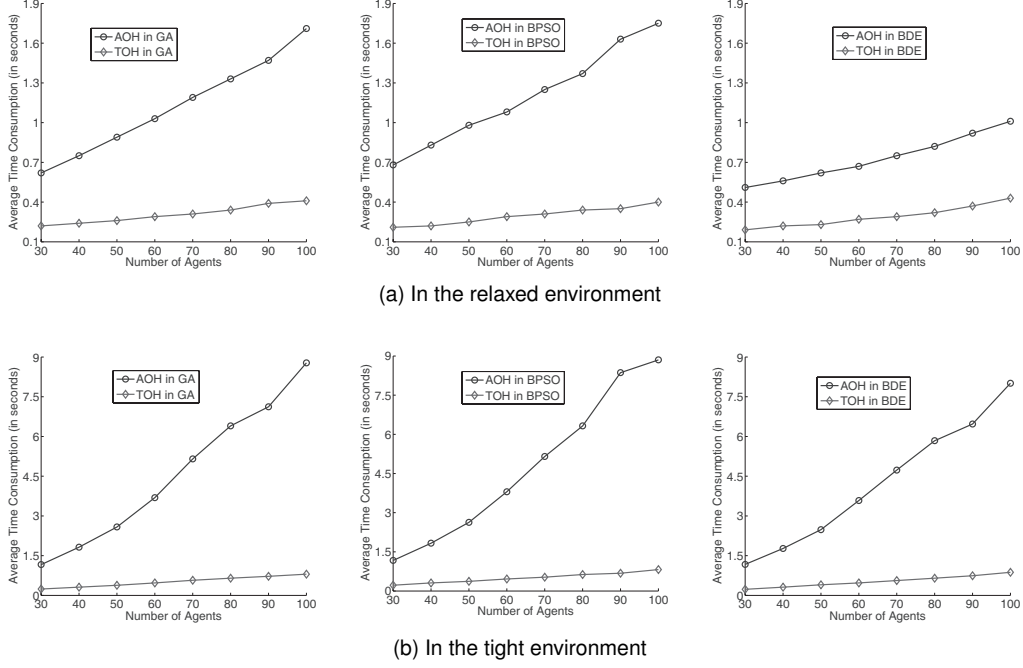
Fig. 6. The average time consumption (in seconds) of the TOH and AOH with different numbers of agents.
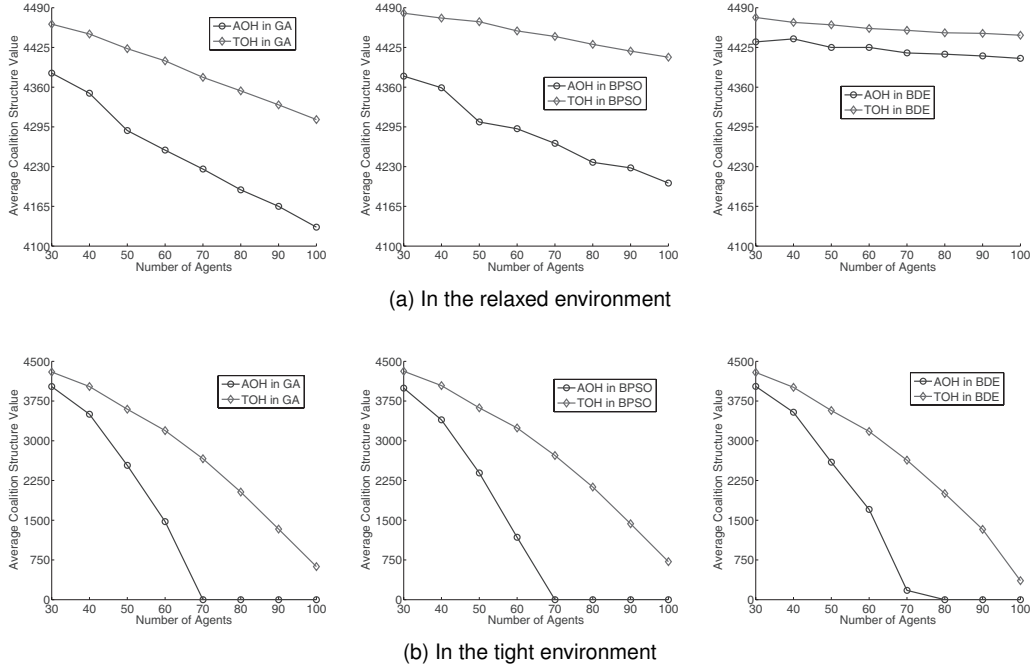

Fig. 7. The average coalition structure values obtained by TOH and AOH with different numbers of agents.

However, although TOH based GA, BPSO, and BDE return high-value overlapping coalitions quickly, they do not provide any guarantee on finding the optimal. This leads two main avenues in our future work. On one hand, there is a need to verify whether the proposed TOH is able to help GA, BPSO, and BDE establish a bound from the optimal. This can borrow the analytical method in [32] to show the approximation performance of TOH based GA, BPSO, and BDE in terms of how close the overlapping coalition structure obtained is to the optimal. On the other hand, perhaps one of the most promising directions is to combine the proposed TOH with the state-of-the-art dynamic programming algorithm for non-overlapping coalition structure generation which guarantees to find the optimal in $O(3^n)$ time [6].

(a) In the relaxed environment

(b) In the tight environment

Fig. 8. The average time consumption (in seconds) of the TOH and AOH with different numbers of tasks.

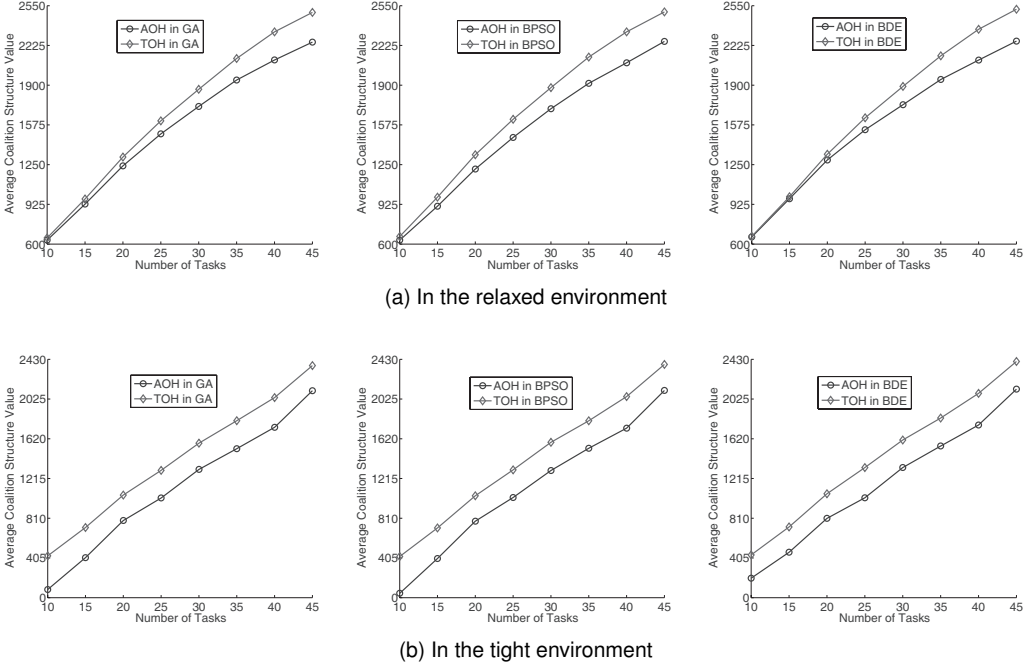(a) In the relaxed environment

(b) In the tight environment
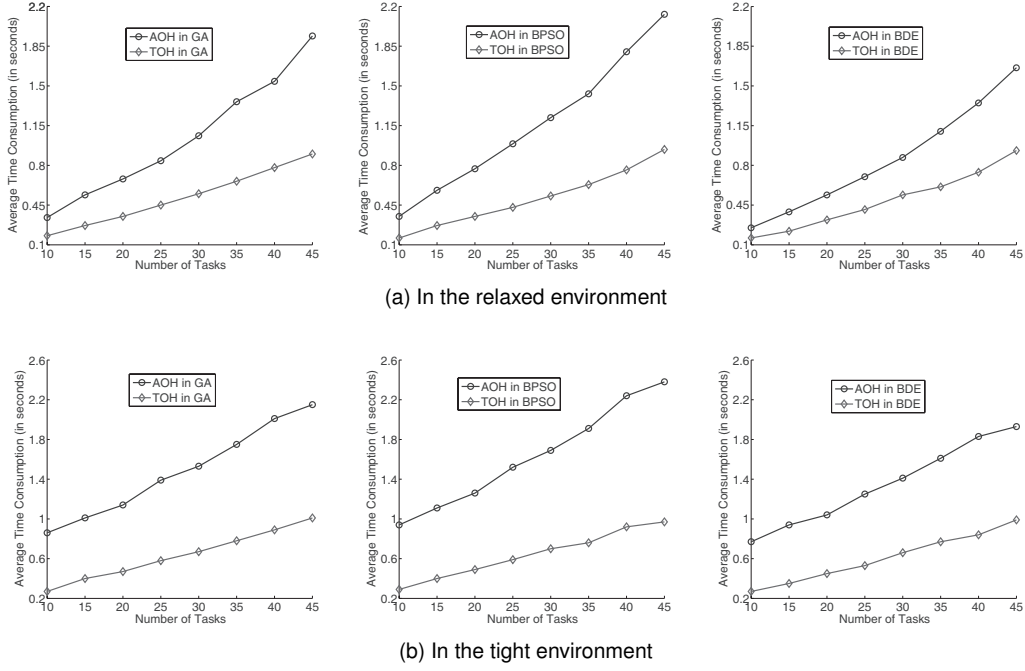
Fig. 9. The average coalition structure values obtained by TOH and AOH with different numbers of tasks.

REFERENCES

[1] M. Bristow, L. Fang, and K. W. Hipel, "Agent-based modeling of competitive and cooperative behavior under conflict," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 44, no. 7, pp. 834–850, 2014.

[2] V. Loia and A. Vaccaro, "Decentralized economic dispatch in smart grids by self-organizing dynamic agents," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 44, no. 4, pp. 397–408, 2014.

[3] K. T. Seow, K. M. Sim, and S. Y. C. Kwek, "Coalition formation for resource coallocation using BDI assignment agents," *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, vol. 37, no. 4, pp. 682–693, 2007.

[4] X. Liang and Y. Xiao, "Studying bio-inspired coalition formation of robots for detecting intrusions using game theory," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 40, no. 3, pp. 683–693, 2010.

[5] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artif. Intell.*, vol. 101, no. 1–2, pp. 165–200, 1998.

[6] T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings, "Coalition structure generation: a survey," *Artif. Intell.*, vol. 229, pp. 139–174, 2015.

[7] M. Owliya, M. Saadat, G. G. Jules, M. Goharian, and R. Anane, "Agent-based interaction protocols and topologies for manufacturing task allocation," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 43, no. 1, pp. 38–52, 2013.

[8] M. J. Lanham, G. P. Morgan, and K. M. Carley, "Social network modeling and agent-based simulation in support of crisis de-escalation,"
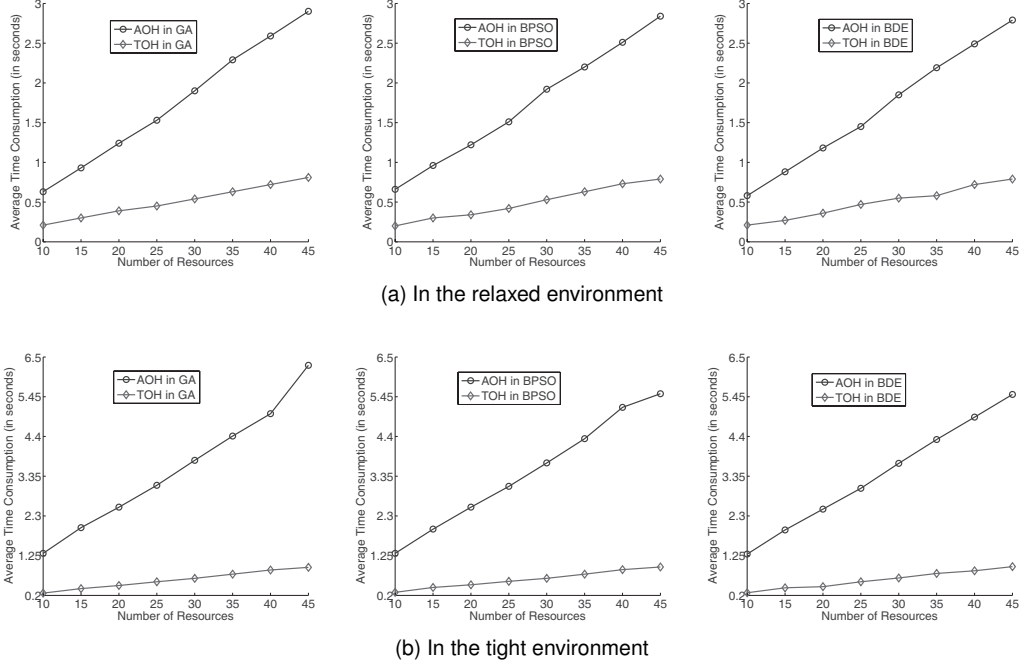
(a) In the relaxed environment

(b) In the tight environment

Fig. 10. The average time consumption (in seconds) of the TOH and AOH with different numbers of resources.

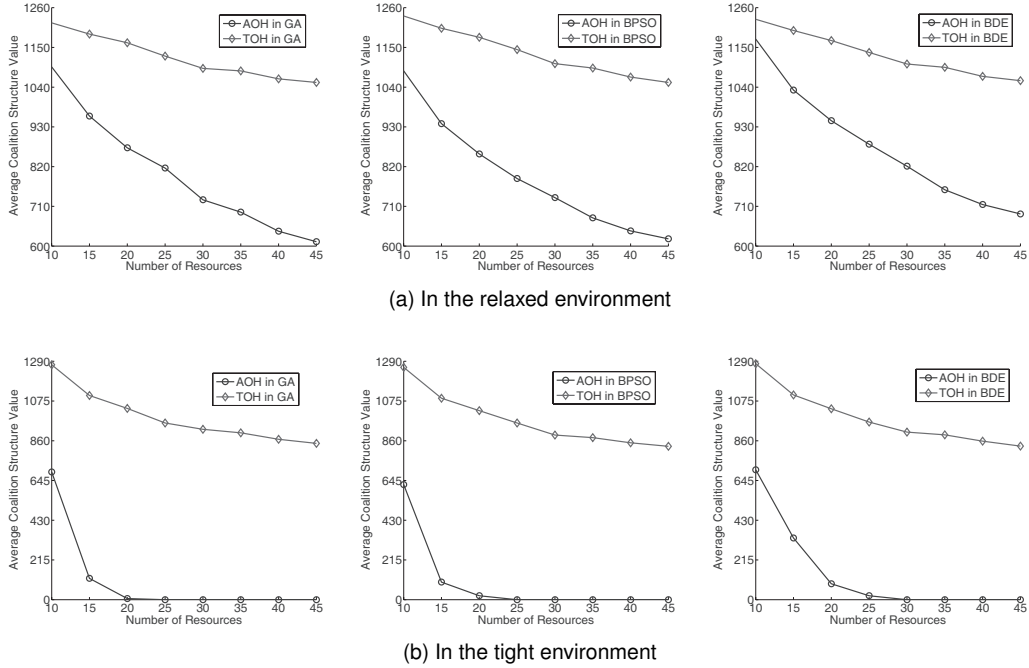(a) In the relaxed environment

(b) In the tight environment

Fig. 11. The average coalition structure values obtained by TOH and AOH with different numbers of resources.

IEEE Trans. Syst. Man Cybern. Syst., vol. 44, no. 1, pp. 103–110, 2014.

[9] G. Jules and M. Saadat, "Agent cooperation mechanism for decentralized manufacturing scheduling," *IEEE Trans. Syst. Man Cybern. Syst.*, DOI: 10.1109/TSMC.2016.2578879, 2016.

[10] O. Shehory and S. Kraus, "Formation of overlapping coalitions for precedence-ordered task-execution," in *Proceedings of the 2nd Int. Conf. Multi-Agent Syst.*, Kyoto, Japan, December 9–13, pp. 330–337, 1996.

[11] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[12] G. Zhang, J. Jiang, Z. Su, M. Qi, and H. Fang, "Searching for overlapping coalitions in multiple virtual organizations," *Inform. Sci.*, vol. 180, no. 17, pp. 3140–3156, 2010.

[13] Z. He, T. C. E. Cheng, J. Dong, and S. Wang "Evolutionary location and pricing strategies in competitive hierarchical distribution systems: A spatial agent-based model," *IEEE Trans. Syst Man Cybern. Syst.*, vol. 44, no. 7, pp. 822–833, 2014.

[14] C. Schreiber and K. M. Carley, "Validating agent interactions in construct against empirical communication networks using the calibrated grounding technique," *IEEE Trans. Syst Man Cybern. Syst.*, vol. 43, no. 1, pp. 208–214, 2013.

[15] Y. Xiao, K. C. Chen, C. Yuen, Z. Han, and L. A. Dasilva, "A Bayesian overlapping coalition formation game for device-to-device spectrum sharing in cellular networks," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 7, pp. 4034–4051, 2015.

[16] D. Wu, Y. Cai, L. Zhou, and J. Wang, "A cooperative communication

scheme based on coalition formation game in clustered wireless sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 3, pp. 1190–1200, 2012.

[17] C. Liu, Q. Zeng, H. Duan, M. Zhou, F. Lu, and J. Cheng, "E-net modeling and analysis of emergency response processes constrained by resources and uncertain durations," *IEEE Trans. Syst Man Cybern. Syst.*, vol. 45, no. 1, pp. 84–96, 2015.

[18] J. Niño-Castañeda, A. Frías-Velázquez, N. B. Bo, M. Slembrouck, J. Guan, G. Debard, B. Vanrumste, T. Tuytelaars, and W. Philips, "Scalable semi-automatic annotation for multi-camera person tracking," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2259–2274, 2016.

[19] M. Wooldridge and P. E. Dunne, "On the computational complexity of coalitional resource games," *Artif. Intell.*, vol. 170, no. 10, pp. 835–871, 2006.

[20] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings, "Cooperative games with overlapping coalitions," *J. Artif. Intell. Res.*, vol. 39, pp. 179–216, 2010.

[21] Y. Zick and E. Elkind, "Arbitrators in overlapping coalition formation games," in *Proceedings of the 10th Int. Conf. Auton. Agent Multi-Agent Syst.*, Taipei, Taiwan, May 2–6, pp. 55–62, 2011.

[22] Y. Zick, G. Chalkiadakis, and E. Elkind, "Overlapping coalition formation games: charting the tractability frontier," in *Proceedings of the 11th Int. Conf. Auton. Agent Multi-Agent Syst.*, Valencia, Spain, June 4–8, pp. 787–794, 2012.

[23] Y. Zick, E. Markakis, and E. Elkind, "Arbitration and stability in cooperative games with overlapping coalitions," *J. Artif. Intell. Res.*, vol. 50, pp. 847–884, 2014.

[24] G. Zhang, J. Jiang, N Xia, and Z. Su, "Particle swarms cooperative optimization for coalition generation problem," in *Proceedings of the 6th Int. Conf. Simulat. Evol. Learn.*, Hefei, China, October 15–18, pp. 166–173, 2006.

[25] C. Lin and S. Hu, "Multi-task overlapping coalition parallel formation algorithm," in *Proceedings of the 6th Int. Joint Conf. Auton. Agent Multi-Agent Syst.*, Honolulu, HI, USA, May 14–18, pp. 1260–1262, 2007.

[26] G. Zhang, J. Jiang, C. Lu, Z. Su, H. Fang, and Y. Liu, "A revision algorithm for invalid encodings in concurrent formation of overlapping coalitions," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2164–2172, 2011.

[27] Y. Zhan, J. Wu, C. Wang, and J. Xie, "On the complexity and algorithms of coalition structure generation in overlapping coalition formation games," in *Proceedings of the 24th Int. Conf. Tools Artif. Intell.*, Athens, Greece, Novermber 7–9, pp. 868–873, 2012.

[28] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Procedings of the IEEE Int. Conf. Syst. Man Cybern.*, Orlando, FL, USA, October 12–15, pp. 4104–4108, 1997.

[29] T. Rahwan and N. R. Jennings, "An algorithm for distributing coalitional value calculations among cooperating agents," *Artif. Intell.*, vol. 171, no. 8–9, pp. 535–567, 2007.

[30] J. E. Cohen, "Cooperation and self-interest: Pareto-inefficiency of Nash equilibria in finite random games," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 95, no. 17, pp. 9724–9731, 1998.

[31] T. W. Sandhlom and V. R. T. Lesser, "Coalitions among computationally bounded agents," *Artif. Intell.*, vol. 94, no. 1–2, pp. 99–137, 1997.

[32] Y. Yu, X. Yao, and Z. Zhou, "On the approximation ability of evolutionary optimization with application to minimum set cover," *Artif. Intell.*, vol. 180–181, pp. 20–33, 2012.

[33] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Grammatical evolution hyper-heuristic for combinatorial optimization problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 840–861, 2013.

[34] S. Sen and P. Dutta, "Search for optimal coalition structures," in *Proceedings of the 4th Int. Conf. Multiagent Syst.*, Boston, MA, USA, July 10–12, pp. 286–292, 2000.

[35] J. Yang and Z. Luo, "Coalition formation mechanism in multi-agent systems based on genetic algorithms," *Appl. Soft Comput.*, vol. 7, no. 2, pp. 561–568, 2007.

[36] B. Li, H. Yu, Z. Shen, L. Cui, and V. R. T. Lesser, "An evolutionary framework for multi-agent organizations," in *Proceedings of the IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Techn.*, Singapore, December 6–9, pp. 35–38, 2015.

[37] N. Xia, J. Jiang, and C. Yu, "Solution to agent coalition problem using improved ant colony optimization algorithm," in *Proceedings of the IEEE/WIC/ACM Int. Conf. Intell. Agent Techn.*, Beijing, China, September 20–24, pp. 475–478, 2004.

[38] H. Keinänen, "Simulated annealing for multi-agent coalition formation," in *Proceedings of the 3rd KES Int. Symp. Agent Multi-Agent Syst. Techn. Appl.*, Uppsala, Sweden, June 3–5, pp. 30–39, 2009.

[39] G. Zhang, R. Yang, Z. Su, F. Yue, Y. Fan, M. Qi, and J. Jiang, "Using binary particle swarm optimization to search for maximal successful coalition," *Appl. Intell.*, vol. 42, no. 2, pp. 195–209, 2015.

[40] G. Zlotkin and J. S. Rosenschein, "Coalition, cryptography, and stability: mechanisms for coalition formation in task oriented domains," in *Proceedings of the 12th National Conf. Artif. Intell.*, Washington, USA, July 31–August 4, pp. 432–437, 1994.

[41] V. D. Dang and N. R. Jennings, "Coalition structure generation in task-based settings," in *Proceedings of the 17th Europ. Conf. Artif. Intell.*, Riva del Garda, Italy, August 28–September 1, pp. 210–214, 2006.

[42] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, New York, USA: Cambridge University Press, 2009.

[43] R. Sarker, M. Mohammadian, and X. Yao, *Evolutionary Optimization*, New York, USA: Springer Science & Business Media, 2002.

[44] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, 2015.

[45] G. Pampara, A. P. Engelbrecht, and N. Franken, "Binary differential evolution," in *Proceedings of the IEEE Int. Conf. Evol. Comput.*, Vancouver, BC, Canada, July 16–21, pp. 1873–1879, 2006.

[46] D. G. Altman, D. Machin, T. N. Bryant, M. J. Gardner, *Statistics with Confidence: Confidence Intervals and Statistical Guidelines*, London, England: BMJ Books, 2000.

[47] P. Sprent and N. C. Smeeton, *Applied Nonparametric Statistical Methods*, London, England: Chapman and Hall/CRC Press, 2000.

Guofu Zhang (M'11) received the B.S. and Ph.D. degrees in computer science from Hefei University of Technology, Hefei, China, in 2002 and 2008, respectively.

He is currently an Associate Professor with School of Computer and Information, Hefei University of Technology. His current research interests include multi-agent systems, evolutionary computation, and search-based software engineering.

Zhaopin Su (M'14) received the B.S. and Ph.D. degrees in computer science from Hefei University of Technology, Hefei, China, in 2003 and 2008, respectively.

She is currently an Associate Professor with School of Computer and Information, Hefei University of Technology. Her current research interests covers evolutionary computation, disaster emergency decision-making, and multimedia security.

Miqing Li received the Ph.D. degree in computer science from the Department of Computer Science, Brunel University London, UK in 2015. He is currently a research fellow in CERCIA, School of Computer Science, University of Birmingham, UK.

His research focuses on evolutionary multi- and many-objective optimisation, including algorithm design, performance assessment, test problem construction and visualisation, and their application in diverse fields.

Meibin Qi received the B.S. degree in electronic engineering from Chongqing University, Chongqing, China, in 1991, and the M.S. and Ph.D. degrees in electronic engineering from Hefei University of Technology, Hefei, China, in 2001 and 2007.

He is currently a Professor with the School of Computer and Information, Hefei University of Technology. His current research interests include digital signal processing, multi-agent systems, and intelligent visual surveillance.

Jianguo Jiang received the M.S. degree in computer science from Hefei University of Technology, Hefei, China, in 1989.

He is currently a Professor with the School of Computer and Information, Hefei University of Technology, China. His current research interests include automatic control, image processing, and multi-agent systems.

Xin Yao (M'91-SM'96-F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is currently a Chair Professor of Computer Science at the Southern University of Science and Technology, Shenzhen, China, and a part-time Professor of Computer Science at the University of Birmingham, UK. He is an IEEE Fellow, and a Distinguished Lecturer of IEEE Computational Intelligence Society (CIS). His major research interests include evolutionary computation, ensemble learning, and their applications in software engineering. His research won the 2001 IEEE Donald G. Fink Prize Paper Award, 2010, 2016, and 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Awards, 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist), 2011 IEEE Transactions on Neural Networks Outstanding Paper Award, and many other best paper awards. He received the prestigious Royal Society Wolfson Research Merit Award in 2012 and the IEEE CIS Evolutionary Computation Pioneer Award in 2013. He was the the President (2014-15) of IEEE CIS, and the Editor-in-Chief (2003-08) of IEEE Transactions on Evolutionary Computation.