International Journal of Robotics and Automation, Vol. 33, No. 5, 2018

COALITIONAL SKILL GAMES FOR SELF-INTERESTED ROBOTS WITH SVO

Ming-Lan Fu,* Hao Wang,* and Bao-Fu Fang*

Abstract

Robots in a multi-robot system cannot communicate with each other due to various reasons. The robots can only maximize their own individual performance without regard for the overall system performance. To guarantee high system revenue even when the robots are self-interested, an algorithm called the intermediary recruitment algorithm (IRA) is proposed. It imitates the operating mechanism of an intermediary recruitment market to allocate the tasks to the robots. The simulation results analyse the influence of the robots' social value orientation (SVO) on the system revenue, where SVO is used to measure the degree of self-interest, and the effectiveness of the IRA is also verified.

Key Words

Self-interested robot, coalitional skill games, social value orientation

1. Introduction

Multi-robot collaboration problems have been intensively studied by scholars in the relevant areas. They are applied in many areas such as the military, space/subsea exploration, and disaster relief [1]–[6]. Coalitional skill games (CSGs) were first proposed by Bachrach and Rosenschein [7], [8] as a restricted form of a coalition generating problem. The CSGs include three sets: the agent set, skill set, and task set. Each agent and each task have their respective skill sets. This ensures that a task can be accomplished by some agents. Each task has a utility value. The system revenue is defined as the sum of the utility of those tasks that can be completed. An important problem in CSGs is how to allocate tasks to agents to get the maximum system revenue, *i.e.*, the optimal coalition structure generation problem. If the agents in CSGs are robots, the problem is called a multi-robot coalitional skill game. The time complexity of other CSG problems such as computing the coalition value and checking whether an agent is veto has also been studied [7], [8]. The time complexity of optimal coalition structure generation problem is given in [9], and it is proved that in general CSGs, as well as in very restricted versions of optimal coalition structure generation problems, computing the optimal coalition structure is hard. Nevertheless, an algorithm to compute the optimal coalition structure was given [9], and it was proved to have polynomial time complexity if the number of tasks and the tree-width of the corresponding hyper graph were both bounded within a constant, but this restrictive condition was very strict. Other similar models include characteristic function games [10], the resource model and the service model [11], [12]. The characteristic function games assume that each coalition is assigned a value independent of the assigned task. Adams [12] proved that the coalition formation problem in both the resource model and service model, with m tasks, is NP-hard to solve exactly or to approximate within a factor of $O(m^{1-\varepsilon})$ for all $\varepsilon > 0$.

A multi-robot coalitional skill game is single task allocation problem model if the tasks only need one robot, *i.e.*, it is assumed that the tasks are indivisible and can be completed by one robot. Yet, in practical application, there exist many multi-robot coalition formation problems [13]–[16] where accomplishing the tasks requires more than one robot, which is the situation we are concerned with in this paper. The related task allocation problem was introduced in [13].

The abovementioned robots or agents are all cooperative, and their goals are to maximize system total revenue. Yet, in this paper, it is assumed that the robots are self-interested. This assumption has a realistic significance [17]–[20]. As far as we know, only a few studies exist which explore the cooperation of self-interested robots. In [14], [15], combinatorial bids theory was used to solve multi-robot coalition formation problems, and the robots could be considered to be self-interested. Cui *et al.* [17] proposed game theory-based negotiation for a multiple robot task allocation algorithm which assumed that all the robots were self-interested and always selected the task which could maximize their individual revenue. However, they also assumed that each task only needed one robot, *i.e.*, the problem solved in [17] was a single-agent task's allocation problem. Game theory is a powerful tool to study self-interested agents' strategy selection problems. Game theory-based learning algorithms include best response (BR), fictitious play (FP), sampled FP, and computationally efficient sampled FP (CESFP) algorithms [21].

* School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China; e-mail: fml685@163.com, jsjxwangh@hfut.edu.cn, baofufang@163.com
Corresponding author: Hao Wang
Recommended by Dr. Xiaonv Hu
(DOI: 10.2316/Journal.206.2018.5.206-0062)

In addition to assuming the robots are self-interested, this paper also assumes that all the robots have SVO [22], [23], which is used to assess a person's degree of self-interest. Based on the abovementioned assumptions, this paper defines a multiple self-interested robots CSGs model. By imitating the operating mechanisms of the intermediary recruitment market, the IRA is proposed to allocate the tasks to the robots. The robots' self-interested decision can guarantee high system revenue. The IRA provides a new thinking for resolving multi-agent coordination and cooperation problems. The final simulation results analyse the influence of the SVO of robots on the system performance, and the effectiveness of the IRA is also verified.

The remainder of this paper is structured as follows. Section 2 presents the formal problem definitions considered in this paper. Section 3 presents the basic thoughts of IRA and the convergence and time complexity are also presented. The simulation results in Section 4 show that our proposed algorithm can allocate tasks to self-interested robots effectively. Finally, the conclusion of our work is presented in Section 5.

2. Definition of the Problem Model

The formal definition of the problem model is given in Definition 1:

Definition 1 (Multiple Self-interested Robots Coalitional Skill Games Model, MSRCSGM). *There exist three sets in MSRCSGM: a set* $R=\{r_1,r_2,\dots,r_n\}$ *of robots, a set* $S=\{s_1,s_2,\cdots,s_l\}$ *of skills, and a set* $T=\{t_1,t_2,\cdots,t_m\}$ *of tasks, where* $n, l,$ *and* m *denote the total numbers of robots, skills, and tasks, respectively.* RS *is a matrix of* $n\times l$. $RS_{i,j}\in\{0,1\}$ *denotes whether the robot* r_i *owns skill* s_j. ST *is a matrix of* $l\times m$. $ST_{j,k}\in\{0,1\}$ *denotes whether task* t_k*'s accomplishment needs* s_j. $cost$ *is a matrix of* $n\times l$, *where* $cost_{i,j}\in R$ *denotes the cost spent by the robot* r_i *when it uses the skill* s_j ($cost_{i,j}=-1$ *when the robot* r_i *does not own* s_j). *Unless specifically stated otherwise, the value ranges of* $i, j,$ *and* k *are* $i\in\{1,2,\cdots,n\}$, $j\in\{1,2,\cdots,l\}$, *and* $k\in\{1,2,\cdots,m\}$. *Each task* $t_k\in T$ *has an associated utility* u_k *representing the value that completing the task is worth,* $TU=\{u_1,\cdots,u_m\}$. $RST(t)$ *is a matrix of* $n\times l$. $RST_{i,j}(t)\in T$ *denotes which task is selected by the robot* r_i *at time* t *and the skill* s_j *is used. It is assumed that all robots are allowed to select at most one task and only one skill is performed. The task* t_k *can be completed if and only if every needed skill is provided by some robots. The system revenue,* $totalU(t)$, *is defined as the sum of the utilities of the completed tasks.*

MSRCSGM assumes that the robots are self-interested and have SVO. The robot states are defined in Definition 2.

Definition 2 (Robot's state). *Let* $r_i(t)=<RS_i, \cos t_i, SVO_i, RST_i(t), ru_i(t), haveR_i(t)>$ *denote the states of robot* r_i *at time* t, *where* RS_i *denotes the ith row of* RS. $cost_i$ *denotes the ith row of cost.* SVO_i *denotes* r_i*'s SVO.* $RST_i(t)$ *denotes the ith row of* RST. $ru_i(t)$ *includes j elements, where* $ru_{i,j}(t)$ *denotes how much revenue the robot* r_i *can get if it uses the skill* s_j *at time t. If robot* r_i *is removed from the IRA at time t,* $haveR_i(t)=1$, *otherwise,* $haveR_i(t)=0$.

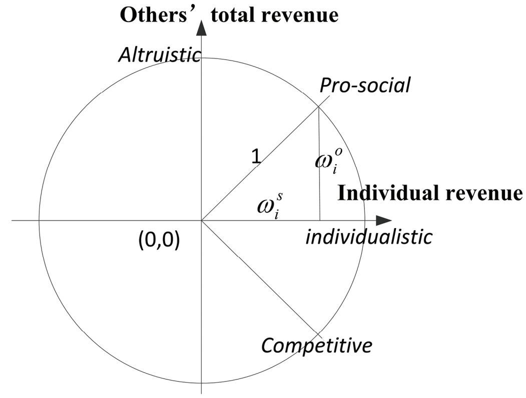

Figure 1. The SVO model.

Definition 3 (SVO). *The individual revenue weight and the others' revenue weight considered by one robot are shown in Fig. 1. The SVO of robot* r_i, SVO_i, *is defined in (1), where* ω_i^o *and* ω_i^s *denote the weights of the others' total revenue and the individual revenue considered by* r_i [22]:

$$SVO_i=\arctan\left(\frac{\omega_i^o}{\omega_i^s}\right) \tag{1}$$

Because ${\omega_i^o}^2+{\omega_i^s}^2=1$, (2) and (3) are established:

$$\omega_i^o=\sqrt{\frac{\tan^2(SVO_i)}{1+\tan^2(SVO_i)}} \tag{2}$$

$$\omega_i^s=\sqrt{\frac{1}{1+\tan^2(SVO_i)}} \tag{3}$$

Definition 4 (Skill's state). *The state of skill* s_j *at time* t *is denoted by a 5-tuple:* $s_j(t)=<RS_j, ST_j, RList_j(t), TList_j(t), successNum_j(t)>$, *where,* RS_j *denotes the jth column of RS.* ST_j *denotes the jth row of ST.* $RList_j(t)$ *is a list which contains the robots which intend to use skill* s_j *and the corresponding cost.* $RList_j(t)$ *is sorted by the costs from smallest to largest.* $LR_{i'}^j(t)$ *and* $LC_{i'}^j(t)$ *denote the* i'*th robot and its cost in* $RList_j(t)$. $TList_j(t)$ *is a list which contains the tasks that need* s_j *and the shares of utility distributed to* s_j *by the tasks.* $TList_j(t)$ *is sorted by the shares of utility from largest to smallest.* $LT_{k'}^j(t)$ *and* $LU_{k'}^j(t)$ *denote the* k'*th task and the corresponding share of utility in* $TList_j(t)$. *In* $RList_j(t)$ *and* $TList_j(t)$, *the tasks which are impossible to be completed and the tasks which possess all the needed skills are ignored. Pair the elements of* $RList_j(t)$ *and* $TList_j(t)$ *from the first pair. If the cost of the robot in the* $RList_j(t)$ *is smaller or equal to the share of utility in* $TList_j(t)$, *we can say that the pairing is successful. Let* $successNum_j(t)$ *denote the total number of successful pairings in skill* s_j *at time t.*

Definition 5 (**Task's state**). *The state of task* t_k *at time* t *is denoted by a 4-tuple:* $t_k(t) = <ST_k, TSN_k, u_k, haveT_k(t)>$*, where* ST_k *is the* k*th column of* ST. $TSN_k \in Z+$ *denotes the number of skills that are needed by task* t_k*.* $haveT_k(t) = 1$ *indicates that* t_k *has all the needed robots and is removed from the IRA.* $haveT_k(t) = -1$ *indicates that* t_k *is removed from the IRA without getting any robots.* $haveT_k(t) = 0$ *indicates that the task* t_k *is still in the IRA at time* t*.*

3. Basic Thoughts of the IRA

This section describes some basic thoughts of the IRA. The robots are regarded as interviewees. The skills are regarded as intermediaries and the tasks are regarded as companies that intend to recruit interviewees. The difference between the IRA and a real intermediary recruitment market is that a particular intermediary s_j in the IRA only receives the robots that possess the skill s_j. Robot r_i has skill s_j implies that robot r_i can send its resume to skill s_j. The robots are self-interested and have one or more skills, so the robots can send their resumes to more than one skill. Yet, when the IRA ends, the robots can only select one task. The task can gain corresponding utility when it obtains all the needed skills. The robot r_i is willing to select task t_k only if $ru_{i,j}(t) \geq \cos t_{i,j}$, where $ru_{i,j}(t)$ is the share of utility distributed to r_i by t_k. The task t_k distributes the u_k to all the needed skills. The IRA is described as follows:

Algorithm 1: IRA

Inputs: RS, ST, TU, SVO.

Outputs: *totalU* and RST.

Step 1. Initialization ($t = 0$). (1) Initialize TSN according to ST. (2) For $\forall 1 \leq j \leq l$, initialize the lists of $RList_j(0)$ and $TList_j(0)$ according to RS_j and ST_j. For $\forall 1 \leq i \leq n$, r_i and the $cost_{i,j}$ are included into $RList_j(0)$ if $RS_{i,j} = 1$. Sort $RList_j(0)$ by costs from smallest to biggest. For $\forall 1 \leq k \leq m$, t_k and the share of utility u_k/TSN_k are included into $TList_j(0)$ if $ST_{j,k} = 1$. Sort $TList_j(0)$ by the shares of utility from biggest to smallest. Pair the robot that has the i'th smallest cost with the task that has the i'th biggest share of utility. The pairing is successful if the cost is small or equal to the share of utility. Let $successNum_j(0)$ denote the total number of successful pairings. (3) For $r_i \in R$, compute the values of ω_i^o and ω_i^s according to SVO_i. (4) For $\forall 1 \leq k \leq m$ and $\forall 1 \leq j \leq l$, $TS_{k,j}(0) \leftarrow u_k/TSN_k$ if $ST_{j,k} = 1$. $TS_{k,j}(0) \leftarrow -1$, if $ST_{j,k} = 0$.

Step 2. Iterate ($t > 0$).

DO

$TS^{old} \leftarrow TS$;

FOR $k \in \{1, 2, \cdots, m\}$ and $have\ T_k(t) = 0$ **DO**

$adjustU(RS, ST, TU, TS(t), haveR(t), haveT(t), k)$;

END FOR

WHILE $\exists k_1 \in \{1, \cdots, m\}$ and $\exists j_1 \in \{1, \cdots, l\}$ make $TS^{old}_{k_1,j_1} \neq TS_{k_1,j_1}$ set up

Step 3. All the skills send confirmation messages to the successfully paired robots. If $r_i \in R(have\ R_i(t) = 0)$ is paired successfully in more than one skill, $ss_i(t)$ denotes the set of skills in which r_i is paired successfully, the skill $s_{\max} = \arg\max_{j \in ss_i(t)} (\omega_i^s \cdot ru_{i,j}(t) + \omega_i^o \cdot \sum_{r_{i'} \in R - \{r_i\}} IR_{i'}(i,j,t) - \mathrm{cost}_{i,j})$ is selected by r_i, where $IR_{i'}(i,j,t)$ denotes the maximal individual revenue that can be obtained by $r_{i'} \in R - \{r_i\}$ at time t if r_i selects the task recommended by s_j. Then, the rejected notices are sent to $T_{i,rej}(t)$, where denotes the set of tasks rejected by r_i.

Step 4. Let $T_u(t) = \{t_{\max}(t)\} \cup T_{i,rej}(t)$, for $\forall k' \in T_u(t)$, execute $adjustU(RS, ST, TU, TS(t), haveR(t), haveT(t), k')$. Execute Step 4 repeatedly until there is no $t_{k'} \in T_u(\mathrm{t}) - \{t_{\max}(t)\}$ that can improve its share of utility to robot ri.

Step 5. Go to Step 2. Execute Steps 2–5 until all the robots receive only one offer when Step 2 is over, and then go to Step 6.

Step 6. If t_k (s.t. $t_k \in T$ and $haveT_k(t) = 0$) has all the needed robots, $haveT_k(t) \leftarrow 1$ and $TS_{k,j}(t) \leftarrow -1$ for $\forall 1 \leq j \leq l$. If the robot r_i is paired successfully with t_k, $haveR_i(t) \leftarrow 1$. Remove t_k and all the robots paired with t_k from all the skills. If there is no task which has all the needed robots, task t_m is selected, where $t_m = \arg\max_{t_k \in T \wedge haveT_k(t) = 0} u_k/TSN_k$. A set of robots, which are needed to complete t_m and have the smallest sum of the costs, are removed.

Step 7. For any task t_k ($t_k \in T$ and $haveT_k(t) = 0$), if there exist some needed skills which do not have any robots at time t or $\sum_{s_{j'} \in S, ST_{j'k} = 1} \min_{1 \leq i \leq n, haveR_i = 0} (\cos t_{i,j'}) > u_k$, then remove t_k from all the skills and $TS_{k,j} \leftarrow -1$ ($s_j \in S$).

Step 8. Make sure that $r_i \in R(haveR_i(t) = 0)$ exists in $s_j(RS_{i,j} = 1)$, then go to Step 2, and execute Steps 2–8 repeatedly until $TS_{k,j} = -1$ for $\forall t_k \in T, \forall s_j \in S$. Compute $totalU(t)$ according to $RST(t)$, RS, ST, and TU. Output $totalU(t)$ and $RST(t)$.

In the sub-function $adjustU(RS, ST, TU, TS(t), haveR(t), haveT(t), k)$, the task $t_k \in T$ adjusts the distribution scheme of u_k to recruit all the needed robots.

The convergence of Step 2 can be demonstrated as follows:

For a particular task t_k and one of the needed skills $s_j(ST_{j,k} = 1)$ because $LU^j_{successNum_j(t)}$ will not decrease. If the smallest adjusting step-size is ± 1, the max number of adjusting steps of $LC^{\mathrm{j}}_{\max}$ is TU_k. If $TS_{k,j}(t) \leq LU^j_{successNum_j(t)}$, $TS_{k,j}(t)$ will not decrease. If $TS_{k,j}(t) > LU^j_{successNum_j(t)}$, $TS_{k,j}(t)$ will not change or decrease first to $LU^j_{successNum_j(t)}$, then increase with $LU^j_{successNum_j(t)}$. The max numbers of adjusting steps are TU_k. The total numbers of tasks and skills are finite, so Step 2 will end in finite steps.

If the smallest adjusting step-size is ± 1, the time complexity of the IRA is $O(n^2lm + nlm^2 + n^{\max_{1 \leq k \leq m} TSN_k})$.

Table 1
Average System Revenues of GGA with Different Values for CP and MP

	0.02	0.06	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0.02	263.4	295.7	304.1	300.7	294.2	296.0	280.4	233.4	198.3	173.3	168.5
0.06	256.7	289.1	297.2	298.9	297.7	289.6	279.9	233.8	203.8	173.9	162.0
0.1	263.1	293.3	301.9	295.3	303.4	293.7	280.0	236.1	197.8	176.8	166.8
0.15	266.2	292.9	293.3	298.2	293.6	292.5	280.2	228.8	195.9	175.1	161.3
0.2	265.5	286.1	299.8	298.2	300.4	292.9	282.1	227.1	203.7	175.3	168.6
0.25	266.8	292.6	298.8	297.1	297.7	295.1	274.2	238.5	201.5	180.7	165.8
0.3	260.1	289.5	296.0	298.0	297.4	295.6	276.5	235.2	195.3	177.0	166.7
0.35	263.6	293.4	305.6	300.3	297.4	295.3	281.4	232.0	197.0	176.4	166.5
0.4	264.5	297.1	299.3	299.4	302.7	296.6	280.5	233.8	199.4	176.7	167.6
0.45	261.3	292.3	298.9	298.5	299.0	295.5	280.8	238.9	200.3	171.7	166.8
0.5	261.6	288.4	295.4	300.7	301.7	289.9	276.7	241.8	200.1	175.1	162.3

The IRA is polynomial if $\max_{1\le k\le m} TSN_k$ is limited to a constant.

4. Simulation Results

In Simulation 1, the system revenues obtained by the IRA are compared to those of the other five algorithms. Of these five algorithms, there are three algorithms in which the robots are collectively rational. The simulation results indicate that the system revenue obtained by IRA is better than that of the five algorithms in most cases and its run time is also acceptable. Simulation 2 analysed the influences of the SVO of the self-interested robots on the system revenue. It shows that the system revenue is maximal when the SVOs of all the robots are 45°. The simulation environment comprised the following: CPU, Intel (R) Core (TM) i3-3240; internal memory capacity, 4.0G; dominant frequency, 3.40 GHz; operating system, Win 8, Lenovo G50.

Simulation 1. In this simulation, 10 groups of data denoted as 1.1–1.10 were generated by the following methods, the sizes of which were the same: $n=50$, $l=15$, $m=50$. Let $random(a,b)$ denote a random integer between a and b (a and b are included). The number of skills possessed by $r_i \in R$ was $random(1,5)$, where the skill possessed by $r_i \in R$ was $random(1,l)$. $TSN_k = random(1,4)$, $1 \le k \le m$, which skill was needed by $t_k \in T$ was $random(1,l)$. $SVO_i(1 \le i \le n) = 0°$. $cost_{i,j}(1 \le i \le n, 1 \le j \le l) = random(0,1)$. $u_k = TSN_k \times random(1,m)$. In this simulation, the system revenues and run time of the IRA are compared with that of the GGA, Service and Adams' algorithm (SAA) [12], combinatorial bids based algorithm (CBA) [14], CESFP, and Vig and Adams' algorithm (VAA) [24], where the agents in IRA, CBA, and CESFP are self-interested and the agents in GGA, SAA, and VAA are collectively rational.

To get the optimal crossover probability and mutation probability of GGA, a data set was generated with the following random method, the size of which was: $n=30$, $l=15$, $m=30$. The number of skills possessed by $r_i(r_i \in R)$ was $random(1,5)$, with the skill possessed by $r_i(r_i \in R)$ being $random(1,l)$. $TSN_k = random(1,4)$, $1 \le k \le m$, and which skill was needed by $t_k(t_k \in T)$ was $random(1,l)$. $SVO_i(1 \le i \le n) = 0°$. $cost_{i,j}(1 \le i \le n, 1 \le j \le l) = random(0,1)$. $u_k = TSN_k \times random(1,m)$. For this data set, with different values for the two parameters, the GGA ran 30 times, and the average system revenues are shown in Table 1. It can be found from the results that the largest system revenue was reached when the crossover probability (CP) was 0.35 and the mutation probability (MP) was 0.1. (In Table 1, rows represent the mutation probability values and columns represent the crossover probability values.) The other parameters of GGA were set as the following: the size of the population was 1,000, and the maximal number of iteration was 10,000. $\rho(t) = t^{-0.5}$ in CESFP. The maximal coalition size in VAA was 4, which is because the maximum number of skills needed by a particular task was 4 according to the generation method for the data sets.

Figure 2 shows the comparative results of the IRA and cooperative agent algorithms. Figure 3 shows the comparative results of the IRA and the self-interested agent algorithms. GGA, BR, and CESFP ran 100 times and the average system total revenues are computed. The run time for these six algorithms is shown in Table 2.

From Figs. 2 and 3, it can be concluded that the system revenues for the IRA are better than those of the other five algorithms in most cases. The system revenues for CESFP are the lowest, and this is partly because mixed strategies were adopted.

The run time for the GGA is the longest. The run time for the VAA is shorter than that of the GGA, which is because the max coalition size of the VAA is 4. Yet,

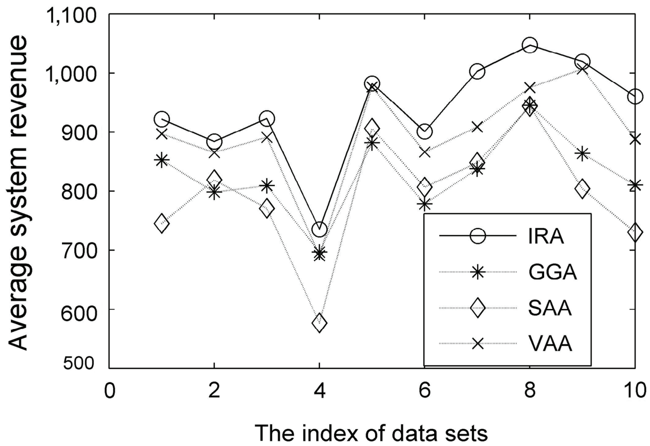

Figure 2. The average system revenues in the IRA, GGA, SAA, and VAA.

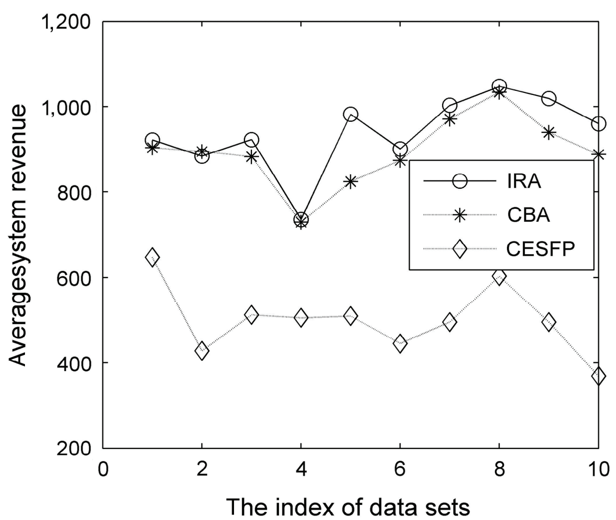

Figure 3. The system revenues for the IRA, CBA, and CESFP.

the run time for the VAA actually increases exponentially with the size of the maximal coalition. The run time for the IRA is not the best but it is acceptable.

Simulation 2. Ten data groups, denoted as 2.1 to 2.10, were generated with random methods the same as that described in Simulation 1. This simulation checked the influence of the robots' SVO on system revenue. For data sets 2.1–2.10, it was assumed that the SVO of all 30 robots were the same. The average system revenues of the 10 data sets are shown in Table 3.

From the results of Table 3, it can be concluded that the system revenues were small or even 0 in most cases when SVO of all the robots were negative. This is because the goal of the robots with negative SVO was to reduce the others' revenue to the greatest extent even if doing so would reduce its own individual revenue. The robots' SVO of 0° indicates that the robots are self-interested.

Table 2
Run Time for the Six Algorithms

	IRA (s)	GGA (s)	SAA (s)	VAA (s)	CESFP (s)	CBA (s)
1.1	11.1	306	0.15	32.2	8.2	3.7
1.2	13.0	334	0.16	28.1	7.7	3.3
1.3	31.1	331	0.15	28.2	7.8	1.6
1.4	3.1	284	0.16	30.8	7.1	1.5
1.5	18.4	337	0.16	27.3	7.6	2.8
1.6	12.0	321	0.15	28.1	7.8	1.7
1.7	5.5	331	0.16	26.7	8.2	1.8
1.8	6.9	307	0.15	29.1	8.4	1.7
1.9	5.2	315	0.16	27.6	8.5	1.4
1.10	20.9	347	0.16	27.9	8.2	2.0

Table 3
Average System Revenues (ASR) with Different SVO

SVO	−20°	−10°	0°	10°	20°	30°	45°	89°
ASR	0	3,230	9,753	9,835	9,917	9,960	10,001	9,964

When selecting a task, they only consider their individual revenue and ignore the others' revenue. The system revenues improved a little compared with the negative SVOs but are still not the best. The robots are pro-social if their SVOs are around 45°. When selecting a task, the pro-social robots consider both the others' revenue and their own individual revenue. In this case, the system revenue is much greater. The robots are altruistic if their SVOs are close to 90°. From Table 3, it can be seen that the system revenue is not the greatest if all the robots are altruistic.

5. Conclusion

This paper proposed an algorithm, the IRA, to resolve the coalition formation problem of multiple self-interested robots CSGs model in which the robots are self-interested. The IRA imitates the mechanisms of the intermediary recruitment market, and the tasks compete for the limited robots by distributing the revenue reasonably. The robots with SVO select the most satisfying tasks according to individual revenue and the others' revenue. The IRA can guarantee higher system revenue even if all the robots are self-interested. The simulation results verified the validity of the IRA. Further studies will mainly focus on four aspects: First, this paper only considered a static task distribution problem. Elango *et al.* [25] showed that task allocation problems can be divided into static and dynamic problems. There are many dynamic task allocation problems in practical applications. Thus, the next work could invest the reasonable distribution of the utility of the task in dynamic task allocation problems. Second, because of the inherent hardness in computing the optimal coalition structure, other restricted problems should be studied. Third, the IRA can be applied to many practical applications such as multi-robot pursuit-evasion problems [26], [27], and multi-robot patrolling [28], [29]. The last problem is to investigate how the personalities and emotions of the self-interested robots influence the task selection and the system revenue.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61175051, 61175033), the National High Technology Research and Development Program of China (863 Program) (Grant No. 2012AA011005), the Natural Science Foundation of Anhui Province (1308085QF108), and the Doctorate Personnel Special Fund of Hefei University.

References

[1] C. Yang, Z. Li, and J. Li, Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models, *IEEE Transactions on Cybernetics*, *43*(1), 2013, 24–36.

[2] L. Sabattini, C. Secchi, M. Cocetti, A. Levratti, *et al.*, Implementation of coordinated complex dynamic behaviors in multirobot systems, *IEEE Transactions on Robotics*, *31*(4), 2015, 1018–1032.

[3] S. Nardi, C. Della Santina, D. Meucci, and L. Pallottino, Coordination of unmanned marine vehicles for asymmetric threats protection, in *OCEANS* (IEEE, 2015), 1–7.

[4] C. Wang, F. Zhang, and D. Schaefer, Dynamic modeling of an autonomous underwater vehicle, *Journal of Marine Science & Technology*, *20*(2), 2015, 199–212.

[5] S. Nurmaini, B. To, and A.R. Thoharsin, Intelligent mobile olfaction of swarm robots, *IAES International Journal of Robotics and Automation*, *2*(4), 2013, 189–198.

[6] W. Wei and K. Curran, Indoor robot localisation with active RFID, *IAES International Journal of Robotics and Automation*, *1*(3), 2012, 137–144.

[7] Y. Bachrach and J.S. Rosenschein, Coalitional skill games, *Proc. 7th International Joint Conf. on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, 2008, 1023–1030.

[8] Y. Bachrach, D.C. Parkes, and J.S. Rosenschein, Computing cooperative solution concepts in coalitional skill games, *Artificial Intelligence*, *204*(9), 2013, 1–21.

[9] Y. Bachrach, R. Meir, K. Jung, and P. Kohli, Coalitional structure generation in skill games, *The National Conf. on Artificial Intelligence*, Atlanta, Georgia, USA, 2010, 703–708.

[10] T. Sandholm, K. Larson, M. Andersson, and O. Shehory, Coalition structure generation with worst case guarantees, *Artificial Intelligence*, *111*(1), 1999, 209–238.

[11] L. Vig and J.A. Adams, Coalition formation: From software agents to robots, *Journal of Intelligent and Robotic Systems*, *50*(1), 2007, 85–118.

[12] J.A. Adams, Coalition formation for task allocation: Theory and algorithms, *Autonomous Agents and Multi-Agent Systems*, *22*(2), 2011, 225–248.

[13] R. Zlot and A. Stentz, Complex task allocation for multiple robots, *Proc. 2005 IEEE Conf. on Robotics and Automation*, Barcelona, Spain, 2005, 1515–1522.

[14] L. Lin and Z. Zheng, Combinatorial bids based multi-robot task allocation method, *Proc. 2005 IEEE Conf. On Robotics and Automation*, Barcelona, Spain, 2005, 1145–1150.

[15] L. Vig, and J.A. Adams, Market-based multi-robot coalition formation, in *Distributed Autonomous Robotic Systems*, *7* (Japan: Springer International Publishing, 2006), 227–236.

[16] A.M. Khamis, A.M. Elmogy, and F.O. Karray, Complex task allocation in mobile surveillance systems, *Journal of Intelligent & Robotic Systems*, *64*(1), 2011, 33–55.

[17] R. Cui, J. Guo, and B. Gao, Game theory-based negotiation for multiple robots task allocation, *Robotica*, *31*(6), 2013, 923–934.

[18] G. Zlotkin and J.S. Rosenschein, Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains, *IEEE Transactions on Systems, Man and Cybernetics*, *21*(6), 1991, 1317–1324.

[19] H. Wang, C.W. Dong, and B.F. Fang, A novel multi pursuers-one evader game based on quantum game theory, *Information Technology Journal*, *12*(12), 2013, 2358–2365.

[20] S. Damer and M. Gini, Cooperation without exploitation between self-interested agents, *Intelligent Autonomous Systems*, 12 (Berlin, Heidelberg: Springer International Publishing, 2013), 553–562.

[21] B. Swenson, S. Kar, and J. Xavier, A computationally efficient implementation of fictitious play for large-scale games, *Signal Processing Conference IEEE*, 2015, 1043–1047.

[22] R.O. Murphy, K.A. Ackermann, and M. Handgraaf, Measuring social value orientation, *Judgement and Decision Making*, *6*(8), 2011, 771–781.

[23] Z. Zhang, F. Zhang, L. Huang, B. Yuan, *et al.*, Theories and measurement methods of social value orientation related to decision making, *Advances in Psychological Science*, *22*(1), 2014, 48–56.

[24] L. Vig and J.A. Adams, Multi-robot coalition formation, *IEEE Transactions on Robotics*, *22*(4), 2006, 637–649.

[25] M. Elango, S. Nachiappan, and M.K. Tiwari, Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms, *Expert Systems with Applications*, *38*(6), 2011, 6486–6491.

[26] Y. Tian and N. Sarkar, Game-based pursuit evasion for nonholonomic wheeled mobile robots subject to wheel slips, *Advanced Robotics*, *27*(14), 2013, 1087–1097.

[27] L.A. Zadeh, Stochastic finite-state systems in control theory, *Information Sciences*, *251*(4), 2013, 1–9.

[28] A. Kawamura and M. Soejima, Simple strategies versus optimal schedules in multi-agent patrolling, in *Algorithms and Complexity*, (Berlin: Springer International Publishing, 2015), 261–273.

[29] E. Hernández, J. Cerro, and A. Barrientos, Game theory models for multi-robot patrolling of infrastructures, *International Journal of Advanced Robotic Systems*, *10*(2), 2013, 463–474.

Biographies

Ming-Lan Fu was born in Sichuan, P.R. China, in 1981. She received her BE degree in network engineering and ME degree in pattern recognition and intelligent systems from Sichuan University of Science & Engineering. She is now a doctoral candidate at the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. Her current research includes multi-robot systems, game theory, and emotional intelligence.

Hao Wang was born in Jiangsu, P.R. China, in 1962. He received his BE degree in automatic control from Shanghai Jiao Tong University in 1984 and the ME and Ph.D. degrees in computer application technology from Hefei University of Technology. He is now a professor at the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His current research includes artificial intelligence and robots.

Bao-Fu Fang was born in Anhui, P.R. China, in 1978. He received his BE degree in welding technology and equipment and ME in computer software theory from Hefei University of Technology (HFUT) in 2000 and 2003, respectively, and his Ph.D. degree in computer application technology from Harbin Institute of Technology (HIT) in 2013. He joined the department of computer science and technology, School of Computer Science and Information Engineering, Hefei University of Technology, as a teaching assistant in 2000, and became assistant professor in 2004, associate professor in 2010, and a master's supervisor in 2011. His current research interests include multi-robot/agent systems, multi-robot task allocation, robotic vision, emotion/self-interested robotics, and machine learning.