Task Allocation for Undependable Multiagent Systems in Social Networks

Yichuan Jiang, *Senior Member*, *IEEE*, Yifeng Zhou, and Wanyuan Wang

Abstract—Task execution of multiagent systems in social networks (MAS-SN) can be described through agents' operations when accessing necessary resources distributed in the social networks; thus, task allocation can be implemented based on the agents' access to the resources required for each task and aimed to minimize this resource access time. Currently, in undependable MAS-SN, there are deceptive agents that may fabricate their resource status information during task allocation but not really contribute resources to task execution; although there are some game theory-based solutions for undependable MAS, but which do not consider minimizing resource access time that is crucial to the performance of task execution in social networks. To achieve dependable resources with the least access time to execute tasks in undependable MAS-SN, this paper presents a novel task allocation model based on the negotiation reputation mechanism, where an agent's past behaviors in the resource negotiation of task execution can influence its probability to be allocated new tasks in the future. In this model, the agent that contributes more dependable resources with less access time during task execution is rewarded with a higher negotiation reputation, and may receive preferential allocation of new tasks. Through experiments, we determine that our task allocation model is superior to the traditional resources-based allocation approaches and game theory-based allocation approaches in terms of both the task allocation success rate and task execution time and that it usually performs close to the ideal approach (in which deceptive agents are fully detected) in terms of task execution time.

Index Terms—Social networks, multiagent systems, task allocation, load balancing, undependable, deceptive agents

1 INTRODUCTION

In contemporary social applications of large-scale multiagent systems, agents can be organized into social networks [1], [2], [3], [4], [5], which is called multiagent systems in social networks (MAS-SN). MAS-SN differs from the common MAS in that the former takes the social networks to organize the agents' behaviors and the social relations between agents as well as the social mechanisms can influence the coordination of agents. In reality, undependable MAS-SN may be seen due to the openness and heterogeneity of MAS-SN [6], [7], where agents may take undependable actions [3], [8].

Without loss of generality, task execution in MAS-SN can be described through agents' operations when accessing necessary resources distributed in the social networks [3], [9], [10]. Thus, this paper focuses on describing the characteristics of agents according to their behaviors during resource negotiation in task allocation and execution. A truthful agent is *dependable* that it provides real resource status information in task allocation and fully contributes its free resources to the execution of the task, and a deceptive agent is *undependable* that it fabricates its resource status information in task allocation but does not fully contribute its resources to the execution of the task.

Task allocation is often implemented based on the accessibility of required resources, and one of the main objectives is to minimize resource access time [9]. However, in undependable MAS-SN there may be deceptive agents that do not provide dependable resource access in task allocation and execution; thus, the task allocation should *guarantee dependable resource access as well as minimize resource access time*. To measure the probability that the tasks will obtain dependable resources from the agents with the least access time in undependable MAS-SN, this paper presents the concept of *negotiation reputation* in task allocation, where an agent's past behaviors during the resource negotiation of task execution can influence its probability of being allocated new tasks in the future.

The related work can be categorized into two types: resource-based task allocation and game theory-based task allocation. Much of the research on resource-based task allocation in common MAS [9], [10], [11], [12] mainly aims to optimize the resource access but seldom systematically addresses the problem of undependable resource access. On the other hand, although there are some studies based on game theory and mechanism design considering the deception of agents in common MAS [13], [14], [15], they do not consider the objective of minimizing resource access time in network structure that is crucial to the performance of task execution in MAS-SN.

Therefore, this paper's main contribution is to present systematic research on task allocation in undependable MAS-SN for the first time with the aims of both *achieving dependable resource access and minimizing the tasks' necessary resource access time*. Moreover, because our approach is implemented based on negotiation reputation that is simply relying on agents' experiences of executing tasks, it avoids causing heavy costs for the computing and communication of agents. Through experiments, we determine that our model is superior to the traditional resources-based and

• *The authors are with the Key Laboratory of Computer Network and Information Integration of State Education Ministry, School of Computer Science and Engineering, Southeast University, Jiangning District, Nanjing 211189, China; and the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: yjiang@seu.edu.cn, {yfzhouseu, wywang.seu}@gmail.com.*

Manuscript received 1 Dec. 2011; revised 22 July 2012; accepted 13 Aug. 2012; published online 24 Aug. 2012.
Recommended for acceptance by D. Turgut.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-12-0874.
Digital Object Identifier no. 10.1109/TPDS.2012.249.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

1045-9219/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

game theory-based allocation approaches in terms of both the task allocation success rate and task execution time in MAS-SN.

The remainder of this paper is organized as follows: In Section 2, we compare our work with the related work on the subject; in Section 3, we present the problem description; in Section 4, we describe the agent characteristics in undependable MAS-SN; in Section 5, we propose the task allocation model; in Section 6, we provide the experimental results; finally, we discuss and conclude our paper in Section 7.

2 Related Work

2.1 Task Allocation Based on Resources

The previous works on the subject can be categorized into centralized and distributed approaches according to the allocation control mechanism. Centralized approaches use a centralized controller that should know the status information of the whole system in real time [16]. In a distributed approach, there is no centralized controller [17]; for example, An et al. presented an optimization method for multiresource negotiation in task allocation [9]. Another typical distributed method is the contract net, a well-known task-sharing protocol in which each agent in a network can be a manager or a contractor at different times or for different tasks [3].

Jiang and Jiang provided a spectrum between a totally centralized approach and a totally decentralized approach to task allocation [10]: the centralized heuristic is utilized to control the overall status information, and the distributed heuristic is utilized to achieve the flexibility of task allocation. Our study substantially extends the architecture in [10] by taking into account the agents' negotiation reputations during task allocation, where tasks can obtain *dependable* resources in the least access time.

Moreover, to deal with some real-world applications in which agents may fail in their endeavors, Ramchurn et al. presented a mechanism taking the trust between agents when allocating tasks into account [18]. However, their mechanism did not adopt any measures to punish the deceptive behaviors of agents in task allocation. In comparison, our study presents a social mechanism of reward and punishment to encourage truthful behaviors and restrain the deceptive behaviors.

2.2 Task Allocation Based on Game Theory and Mechanism Design

Game theory and mechanism design are often used in task allocation for heterogeneous MAS, which considers a situation in which each agent optimizes its own performance independently of the others and they all eventually reach equilibrium [14], [15].

There are some studies that can deal with the undependable MAS to a certain extent. For example, a representative work is that Weerdt et al. presented a mechanism design approach that can incentivize self-interested agents to report their private information correctly [3]; Zlotkin and Rosenschein presented a negotiation mechanism that can deal with the incomplete information and deception in undependable MAS [13]; Ephrati and Rosenschein introduced a multiagent planning technique that can be made resistant to untruthful agents [19]; Shehory and Kraus presented algorithms for task allocation among agents via coalition formation, which can motivate the agents to act to maximize the benefits of the system as a whole so that the deceptive problem can also be addressed to a certain extent [20]. However, those approaches do not consider the objective of minimizing resource access time in social networks that is crucial to the performance of task execution in MAS-SN.

2.3 Task Allocation in Networked Multiagent Systems (N-MASs)

MAS-SN is, in fact, a kind of N-MAS [5]. The related work on task allocation in N-MAS can be categorized into two types.

The first type of approach is implemented by *satisfying* the constraints of the network structure. Weerdt et al. investigated distributed task allocation in social networks and developed an algorithm based on the contract-net protocol [21]. Jiang and Li provided a method that takes both the resources and the network locality of the agents into account [22].

Another type of approach is to implement task allocation by *adjusting* network structures to achieve a better performance. Kota et al. presented a decentralized approach to structural adaptation by explicitly modeling problem-solving agent organizations [23]. Their approach enables agents to modify their structural relations to achieve a better allocation of tasks; and agents can set the edge weights to either 0 (disconnected) or 1 (connected) for task allocation.

The works described above did not assume the existence of deceptive agents in the network. In comparison, our study considers the sociality characteristics of agents (truthful or deceptive) and implements task allocation by adjusting the interaction relation weights of the social network according to their histories of task execution. Moreover, our study adjusts the interaction relation weight values within the closed interval of [0, 1], differently from the approach in [23], which adjusts the weight values only as either 0 or 1 (the interaction relation is either disconnected or connected).

3 Problem Description

3.1 Formalization of Task Allocation in MAS-SN

This paper investigates the task allocation in MAS-SN, where the agents are connected in a social network and tasks arrive at the agents distributed over the network [3]. Now, we give our definition on task allocation in MAS-SN, shown as follows.

Definition 1. *Given a MAS-SN, $<A, E>$, where A is the set of agents, and $\forall <a_i, a_j> \in E$ indicates the existence of a social relation between agent a_i and a_j. It is assumed that the set of resources in agent a_i is R_{ai}, and the set of resources required by task t_j is R_{tj}. If the set of tasks is $T = \{t_1, t_2, \ldots, t_m\}$, the task allocation in MAS-SN can be defined as the mapping of task $\forall t_j \in T$, $1 \leq j \leq m$, to a set of agents, A_{tj}, which can satisfy the following situations:*

1. *The resource requirements of t_j can be satisfied, i.e., $R_{tj} \subseteq \cup_{\forall a_x \in A_{tj}} R_{ax}$;*
2. *The predefined objective can be achieved by the task execution of A_{tj}.*

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

3. *The agents in A_{tj} can execute the allocated tasks under the constraint of social network, for example, $\forall a_x, a_y \in A_{tj}$, $N_{xy} \subseteq E$, where N_{xy} denotes the negotiation path between a_x and a_y.*

3.2 Objective of Task Allocation in MAS-SN

We adopt manager/contractor architecture that integrating the centralized and distributed heuristics in task allocation: An agent is allocated as a manager for a task using a centralized heuristic, and if the manager lacks the necessary resources it negotiates with other agents for contractors using a distributed heuristic [3], [10].

In MAS-SN, the coordination of agents is significantly influenced by the social network structures [8]; therefore, the task execution of MAS-SN (which is implemented by the coordination of agents) is also significantly influenced by the social network structures [3].

One of the main goals of task allocation is to minimize the task execution time [9], [10], [11], [20], [24], [25]. Task execution in MAS-SN can be described as agents' operations when accessing the necessary resources distributed in the social networks [3], [9], [10]. Therefore, *to reduce the execution time of a task, one of the key problems is to reduce the time used accessing the resources necessary for the task*. Obviously, the communication time between allocated agents is crucial to the resource access time of the task [24], [28]; moreover, it may take time to wait for agents' resources if those resources are being accessed by other tasks [11], [25]. Therefore, t*o reduce the execution time of a task, we can reduce the utilities of resource access time that include two factors: the communication time between the manager agent and contractor agents in the social network, and the task's waiting time for resources at the agents* [26].

Let the manager agent for task t be a_t and the set of manager and contractor agents for t be A_t. It is assumed that a resource can be accessed by only one task at a time; if the allocated agents are executing the task, the new task will have to wait until all the required resources are free. Let $£_t$ be the resource access time of task t, $C(a_t, a_j)$ be the communication time between a_t and a_j, and W_{tj} be the waiting time of task t at agent a_j. *The objective of task allocation* is then to select the agent set A_t to minimize $£_t$:

$$£_t = \sum_{\forall a_j \in A_t, a_j \neq a_t} C(a_t, a_j) + \sum_{\forall a_j \in A_t} W_{tj}, \tag{1}$$

under the constraint that the communication of agents is implemented through the social network and each resource can be accessed by only one task at a time. The minimization of $£_t$ can be described as the reduction of the utilities of $C(a_t, a_j)$ and W_{tj}. To reduce $C(a_t, a_j)$, a_t will seek the nearest possible contractor agents; to reduce W_{tj}, we can perform load balancing in task allocation.

3.3 New Problems in *Undependable* MAS-SN and Our Solutions

In undependable MAS-SN, if manager agent a_t is deceptive and cannot provide the desired resources, it will take more time to seek the missing resources by negotiating with other agents, which requires more $C(a_t, a_j)$. On the other hand, if a contractor agent, a_j, is deceptive and cannot provide the desired resources, manager agent a_t will try again to seek more other contractor agents for the missing resources; more $C(a_t, a_j)$ will be wasted in this case as well. More seriously, if the task cannot obtain the necessary resources, a new task allocation may be implemented, which will waste even more time.

Therefore, to reduce task execution time in *undependable* MAS-SN, we should *guarantee dependable resource access as well as minimize resource access time*. By extending the objective in Section 3.2, our objectives for task allocation in *undependable* MAS-SN can be described as follows:

- Improve the probability that tasks can obtain dependable resource access and be executed successfully (*Objective 1*).
- Reduce the communication time between the manager agent and contractor agents in social networks *(Objective 2)*.
- Reduce the tasks' waiting time for required resources at the agents *(Objective 3)*.

  To achieve the above three objectives, in this paper we adopt three measures:
- *A negotiation reputation-based allocation mechanism and a reward/punishment mechanism* are designed to ensure that the truthful agents have a higher likelihood of obtaining tasks than the deceptive agents, i.e., the truthful agents have higher probabilities to be assigned as manager or contractor agents for tasks *(for Objective 1)*.
- *A resource negotiation mechanism* (i.e., the allocation mechanism of contractor agents) is designed to ensure that the truthful agents with less communication distance have higher probabilities to be selected as contractors *(for Objectives 1 and 2)*.
- *A load balancing mechanism* is designed in the allocation of multiple tasks to alleviate the waiting time at heavy-burdened agents (*for Objective 3*).

4 Agents in Undependable MAS-SN

We are mainly concerned with the behavior strategies of agents when dealing with the resources necessary for tasks. The behavior strategies of the agents include two aspects: *those in task allocation and those in task execution*.

Definition 2. *If there are m resource types in the social network, the resource status of an agent can be described as $\{n_1r_1, n_2r_2, \ldots, n_mr_m\}$, which denotes that such agent owns resource $r_j (1 \leq j \leq m)$ with the amount of n_j.* Then, the resource strategy of an agent within a MAS-SN *is described as a tuple $<R_a, R_c>$, where*

1. *R_a: the resource status information reported by the agents during task allocation.*
2. *R_c: the set of resources actually contributed by agents during task execution.*

Let there be an agent, a_i. The behaviors of agent a_i in MAS-SN can be described as follows:

1. *In task allocation*, a_i reports its resource status information, R_{ai}, when it wants to assume the role of manager agent for a task or is negotiated by a manager agent.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

2. *In task execution*, a_i contributes some of its resources, R_{ci}, to execute the allocated task (now a_i acts as either manager or contractor agent for the task).
3. *After task execution*:

   a. a_i obtains certain rewards if the allocated task is executed successfully; the amount of rewards obtained by a_i is determined by a_i's role and real resource contribution during task execution.

   b. a_i pays certain penalties if the allocated task is executed unsuccessfully; the amount of penalties paid by a_i is determined by a_i's role and degree of nonfeasance in resource contribution during task execution.

We now distinguish the truthful and deceptive agents as follows.

Definition 3. An agent is truthful *if it satisfies both of the following conditions: 1) it reports its real resource status information to the system or other negotiating agents in task allocation; 2) it contributes all of its free resources if the allocated task requires it to do so, i.e., if the set of resources required by task t is R_t, then a_i can contribute all resources in $R_{ai} \cap R_t$ if t requires it to do so. Therefore, if the allocated agents for a task are all truthful, the task can obtain dependable resources for successful execution.*

Definition 4. An agent is deceptive *if it satisfies one of the following conditions:*

1. *It fabricates its resource status information during task allocation, which can be described as follows:*
   Let there be an agent, a_i; the real resources owned by a_i are R_{ai}. When the controller or manager agent inquires into the resource status of a_i, and the reported resources status of a_i is MR_{ai}, if $MR_{ai} \neq R_{ai}$, a_i is undependable for its resource status information.
2. *It does not contribute all its free resources if the allocated task requires it to do so, which can be described as follows.*
   Let task t require some resources from a_i, which are denoted as R^t_{ai}. The set of resources that a_j really contributes to task t is $R^{t\prime}_{ai}$. If $(R^t_{ai} - R^{t\prime}_{ai}) \cap R_{a_i} \neq \phi$, a_i is undependable in resource contribution.

Therefore, if any allocated agents for a task are deceptive, the task may not obtain dependable resources and cannot be executed successfully.

5 Task Allocation Model

5.1 Negotiation Reputation

An agent's reputation refers to other agents' opinions of that agent [2]. In MAS-SN, agents negotiate for resources in social networks to execute tasks; thus, the reputation of an agent can be determined by other agents' negotiation histories with that agent regarding the resources for executing tasks. In this paper, we present the concept of *negotiation reputation*; the higher the negotiation reputation of an agent is, the more likely a dependable resource contribution from that agent can be achieved.

In a MAS-SN, each edge $<a_i, a_j>$ is associated with a weight $w(a_i, a_j)$, $0 \leq w(a_i, a_j) \leq 1$, which represents the *negotiation strength* between a_i and a_j. The interaction relation weight between two agents can be adapted according to the two agents' past negotiations in task execution; for example, if a_i obtained a resource contribution from a_j, then $w(a_i, a_j)$ will be gained, and vice versa.

We can represent the weighted MAS-SN with an adjacent matrix: $W = [w(a_i, a_j)]$, $1 \leq i, j \leq n$, $0 \leq w(a_i, a_j) \leq 1$, where n denotes the number of agents, $w(a_i, a_j) = 0$ shows that there are no immediate interaction relations between a_i and a_j, and $w(a_i, a_j) \neq 0$ denotes that there is an immediate interaction relation between a_i and a_j with the negotiation strength of $w(a_i, a_j); w(a_i, a_i) = 1$.

Two agents may have no immediate interaction relations between them, but they can also negotiate through other intermediate agents (the agents within the path between those two agents in the social network) [27]. Therefore, we present the following definitions.

Definition 5. Cumulative negotiation strength. *Let there be a path between a_i and a_j, P_{ij}, in the social network. The cumulative negotiation strength between a_i and a_j along P_{ij} is the product of the negotiation strengths for all immediate interaction relations within P_{ij}:*

$$\prod_{\forall <a_m, a_n> \in P_{ij}} w(a_m, a_n). \tag{2}$$

Definition 6. The negotiation path between two agents, a_i *and a_j, is the path between them that has the maximum cumulative negotiation strength, denoted as NP_{ij}.*

We use Algorithm 1 to compute the cumulative strengths of the negotiation paths between all agents, denoted as a matrix: $CN = [cn_{ij}]$.

Algorithm 1. Compute the cumulative strengths of negotiation paths between all agents.
$CN = W$;
for $(k = 1; k <= n; k++)$
  for $(i = 1; i <= n; i++)$
    for $(j = 1; j <= n; j++)$
      if $cn_{ij} < cn_{ik} \times cn_{kj}$, ***then***: $cn_{ij} = cn_{ik} \times cn_{kj}$;
Output CN.

To measure the probability that an agent will contribute dependable resources, we present the concept of negotiation reputation as follows.

Definition 7. Negotiation reputation. *The negotiation reputation of a_i in the social network can be defined as follows:*

$$\lambda_i = \left(\sum_{\forall a_j \in (A - \{a_i\})} cn_{ji} \right) \Big/ (|A| - 1), \tag{3}$$

where A is the set of all agents in the social network.

Now, we present an example for illustrating cumulative negotiation strengths and negotiation reputations, shown as Section A1 in the Appendix, which can be found online in the CSDL at http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.249.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

5.2 Task Allocation Mechanism

5.2.1 Introduction of Manager/Contractor Architecture

In the manager/contractor architecture [3], [10], a manager agent is allocated to a task using a centralized heuristic and then negotiates with the other contractors for resource assistance using a distributed heuristic.

The task allocation process based on manager/contractor architecture can be described as follows: A task may be first allocated to one agent, which takes charge of the execution of the task (we call this agent the *manager agent*). When the manager agent lacks the necessary resources to execute the allocated task, it negotiates with other agents in the social network; if other agents have the required resources (we call the agents that provide resources to the manager agent *contractor agents*), the manager and contractor agents will work together to execute the task.

5.2.2 Allocation to Manager Agent

To measure an agent's dependable access to one kind of resource, we offer the following definition.

Definition 8. *Let $n_j(k)$ denote the amount of resource r_k owned by agent a_j.* The estimated resource enrichment factor of agent a_i *for resource r_k is then defined as follows:*

$$\Phi_i(k) = \sum_{\forall a_j \in (A-\{a_i\})} \left(\lambda_j \cdot n_j(k) \cdot \left(cn_{ij} \middle/ \sum_{\forall a_j \in (A-\{a_i\})} cn_{ij}\right) \right) + \lambda_i \cdot n_i(k), \quad (4)$$

where λ_j is used to denote that the negotiation reputation of agent a_j (which is determined by the negotiation history between a_i and a_j and the negotiation histories between all other agents and a_j) can influence the probability that a_i will obtain dependable resources from a_j. Thus, even if there are few negotiation histories showing that a_i received resource from a_j (i.e., the negotiation strength from a_i to a_j is low), the probability that a_i can dependably obtain resources from a_j may be high when λ_j is high. Therefore, it is more likely to obtain dependable resources from the agent with the highest negotiation reputation.

Theorem 1. *It is assumed that task t requires resource r_k, and the reputation values are correct. Let there be two agents, a_1 and a_2; $P_i(t-k)$ denotes the probability that task t can obtain dependable resource r_k from agent a_i. Therefore, $\Phi_1(k) > \Phi_2(k) \Rightarrow P_1(t-k) > P_2(t-k)$.*

Proof. The proof can be seen in the Appendix. □

According to Theorem 1, the higher $\Phi_i(k)$ is, the higher the dependable access of a_i to resource r_k is. Therefore, we can base allocation of manager agent on the estimated resource enrichment factor. *So our allocation of manager agent satisfies Objective 1 of task allocation.*

5.2.3 Allocation to Contractor Agents

If the manager agent lacks the necessary resources to execute an allocated task, it may negotiate with others in the social network for assistance. To minimize the communication time between the manager agent and contractor agents during task execution, we presented a contextual resource negotiation model in our previous work, in which the manager agent negotiates with the contextual agents from locations near and far in the network until all requested resources are satisfied [10].

However, the above model only considers the communication distance between the manager agent and contractor agents, not the characteristics of contractor agents. In undependable MAS-SN, if a deceptive agent with a lower communication distance is selected by the manager agent as a contractor agent, the manager agent may not obtain dependable resources in task execution. Therefore, it may be better for the manager agent to select another agent that is truthful but has a relatively higher communication distance.

Therefore, we now combine the factors of communication distance and negotiation reputation and make a tradeoff between them.

Definition 9. *Let a_t be the manager agent for task t. It is assumed that a_j will be negotiated by a_t for resource assistance.* The negotiation value of a_j for t *is as follows:*

$$V_j(t) = \alpha \cdot (1/d_{tj}) + (1-\alpha) \cdot \left(\lambda_j(|R_{a_j} \cap \overline{R_t}|/|\overline{R_t}|)\right), \quad (5)$$

where α is a parameter, λ_j is the negotiation reputation of a_j, d_{tj} is the communication distance between a_t and a_j, $\overline{R_t}$ is the set of resources for t that are currently lacking. The difference between λ_j and $V_j(t)$ is as follows: λ_j is the opinion of all other agents toward a_j, but $V_j(t)$ is only the opinion of a_t to a_j for task t; λ_j can influence $V_j(t)$ to some degree.

According to Definition 9, the negotiation value of an agent is determined by 1) the communication cost, d_{tj}, for Objective 2 of task allocation; 2) the resource satisfaction degree of the agent, $|R_{a_j} \cap \overline{R_t}|/|\overline{R_t}|$, which also addresses Objective 2 because it can reduce the number of contractor agents; and 3) the negotiation reputation of an agent, λ_j, which addresses Objective 1 of task allocation. *Thus, the negotiation value is designed by considering Objectives 1 and 2 of task allocation.*

The manager agent selects the contractor agents according to their negotiation values arranged in descending order, as shown in Algorithm 2.

Algorithm 2. Allocation of contractor agents.

/* a_t: the manager agent; A: the set of all agents; R_{at}: resources owned by a_t; R_t: requested resources for t; $\overline{R_t}$: the set of currently lacking resources for t */

1) $\overline{R_t} = R_t - R_{at}; A' = A - \{a_t\}; A_t = \{a_t\}; b1 = 0; b2 = 0;$
2) **If** $\overline{R_t} == \{\}$, **then** $b1 = 1$;
3) **While** $((b1 == 0)$ and $(b2 == 0))$ **do**:
   3.1) $maxvalue = 0$; $b2 = 1$;
   3.2) $\forall a_j \in A'$:
      3.2.1) $V_j(t) = \alpha \cdot (1/d_{tj}) + (1-\alpha) \cdot (\lambda_j(|R_{a_j} \cap \overline{R_t}|/|\overline{R_t}|))$
      3.2.2) **If** $V_j(t) > maxvalue$, **then**:
         $maxvalue = V_j(t)$; $sa = a_j$; $b2 = 0$;
   3.3) **If** $(b2 == 0)$, **then**:
      $A_t = A_t \cup \{sa\}$; $\overline{R_t} = \overline{R_t} - R_{sa}$; $A' = A' - \{sa\}$;
   3.4) **If** $\overline{R_t} == \{\}$, **then** $b1 = 1$;
4) **If** $(b1 == 1)$, **then Return** (A_t);
   else Return (False);
5) **End**.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

Theorem 2. *Let the manager agent for task t be a_t and the set of allocated agents using Algorithm 2 be A_t. It is, then, assumed that there is another agent set, A^*, that can also satisfy all the resources in $\overline{R_t}$. Thus, we have*

$$\left(\forall A^* \wedge \left(\overline{R_t} - \bigcup_{\forall a_j \in (A^* - \{a_t\})} R_{a_j} = \phi\right)\right) \Rightarrow \left(\sum_{\forall a_j \in (A_t - \{a_t\})} V_j(t) \geq \sum_{\forall a_j \in (A^* - \{a_t\})} V_j(t)\right).$$

Proof. The proof can be seen in the Appendix. □

From Theorem 2, Algorithm 2 can find the contractor agents with the maximum negotiation values, satisfying Objectives 1 and 2 of task allocation.

5.2.4 Load Balancing

As noted above, if an agent has a higher estimated resource enrichment factor or negotiation value, it may act as the manager or contractor agent for more tasks. However, if too many tasks are crowded on certain agents with high estimated resource enrichment factors or negotiation values, the tasks will require much more time to wait for the necessary resources [10], [11], [26]. More importantly, the problem of waiting time may outweigh the advantage of the time saved by accessing resources at the allocated agents; therefore, we should now apply load balancing to the task allocation.

Let the allocated agent set of task t be A_t and the resources required by t be R_t; the team of tasks that queue for resource r_k of agent a_i can be denoted as Q_{ik}, the processing rate of a_i is v_i and the size of Q_{ik} is s_{ik}. We should perform load balancing when the number of queuing tasks is too large, including load balancing for the manager agent and contractor agents.

First, we can modify the definition of e*stimated resource enrichment factor* in (4) as follows:

$$\Phi_i^*(k) = \psi(s_{ik}/(v_i \cdot \lambda_i)) \cdot \Phi_i(k), \tag{6}$$

where ψ is an attenuation function, $0 \leq \psi \leq 1$, the value of $\psi(s_{ik}/(v_i\lambda_i))$ decreases monotonically from 1 to 0 with the increase of $s_{ik}/(v_i\lambda_i)$.

Second, we can modify the definition of negotiation value in (5) to perform load balancing for contractor agents. Assume now that the set of current missing resources for t is $\overline{R_t}$; the set of resources that task t currently requires from a_j is $R_{a_j} \cap \overline{R_t}$. The waiting time of task t at a_j will be determined by the maximum length of the queues for resources in $R_{a_j} \cap \overline{R_t}$. Equation (5) can then be modified as follows:

$$V_j^*(t) = \psi\left(\max_{\forall r_k \in (R_{a_j} \cap \overline{R_t})} \left(s_{jk}/(v_j \cdot \lambda_j)\right)\right) \cdot V_j(t). \tag{7}$$

Obviously, load balancing can reduce the waiting time at agents for tasks' required resources, which satisfies Objective 3 of task allocation.

5.3 Adjustment after Task Execution

After a task is allocated and executed, the system will make adjustments according to the task execution results, which include two aspects: reward and punishment for allocated agents and adjustment for the weights of interaction relations with allocated agents. Moreover, if the task is executed unsuccessfully, a new allocation should be implemented.

5.3.1 Reward and Punishment Mechanism

After task t is executed successfully, the allocated agents A_t will achieve certain rewards from the system. On the other hand, if task t is executed unsuccessfully, the agents in A_t should pay certain penalties.

1) *Negotiation reputation reward.*

After task t is executed successfully, a reward, $r_t, 0 \leq r_t \leq 1$, will be distributed to the agents in A_t according to their roles in and their real resource contributions to the execution of t. First, a part of the utilities of r_t should be awarded to a_t for its role of manager agent. The remaining utilities of r_t are then awarded to all the agents in A_t according to their real resource contributions.

Definition 10. Reward for the role of manager agent. *Let the total utility of the reward for executing task t be r_t, $0 \leq r_t \leq 1$. We can now set a coefficient for rewarding the manager agent, ω, $0 \leq \omega \leq 1$; therefore, the manager agent will be rewarded with the utilities of ωr_t for its role of manager agent in the execution of task t.*

Definition 11. Reward for real resource contribution. *After the reward for the role of manager agent is given, the $(1-\omega)r_t$ will be distributed according to each agent's real resource contribution, as follows:*

$$\forall a_j \in A_t : r_{tj} = (1-\omega)r_t\left(|R_{a_j}^t|/|R_t|\right), \tag{8}$$

where r_{tj} is the reward obtained by agent a_j for its real resource contribution to the execution of task t; A_t is the set of agents that executed task t; $R_{a_j}^t$ is the set of resources that a_j really contributed to task t and R_t is the set of all resources required by task t.

Therefore, the manager agent a_t can obtain the reward of the utilities of $(\omega r_t + r_{tt})$; each contractor agent, a_k, will obtain the reward of the utilities of r_{tk}. Finally, the negotiation reputations of all the agents in A_t will be improved according to their rewards obtained, as follows:

$$\begin{aligned} a_t : \lambda_t &= \lambda_t \cdot (1 + \omega \cdot r_t + r_{tt}) \\ &= \lambda_t \cdot \left(1 + \omega \cdot r_t + (1-\omega)r_t \cdot \left(|R_{a_t}^t|/|R_t|\right)\right), \\ \forall a_k &\in (A_t - \{a_t\}) : \lambda_k = \lambda_k \cdot (1 + r_{tk}) \\ &= \lambda_k \cdot \left(1 + (1-\omega)r_t \cdot \left(|R_{a_k}^t|/|R_t|\right)\right). \end{aligned} \tag{9}$$

2) *Negotiation reputation punishment.*

If task t is executed unsuccessfully, the manager agent and contractor agents should be punished for their roles in and real resource contributions to the task execution.

Definition 12. Penalty for the role of manager agent. *We now set the total utility of the penalty for unsuccessfully executing task t to be equal to the reward for successfully executing task t, which is also r_t. Initially, the manager agent a_t will pay a penalty for its role of manager agent, ωr_t.*

Definition 13. Penalty for the nonfeasance of resource contribution. *After the penalty for the role of manager agent is given, the remaining penalty, $(1-\omega)r_t$, will be distributed as follows.*

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

Let $R^t_{a_j}$ be the set of real resources that a_j contributed to task t which can be achieved by a centralized heuristic; $R_{a_j} \cap R_t$ is the set of resources that a_j can contribute to task t. We will now punish a_j according to its degree of nonfeasance to resource contribution during task execution.

$$\forall a_j \in A_t : p_{tj} = (1-\omega) r_t \cdot \left(1 - \left(|R^t_{a_j}| / |R_t \cap R_{a_j}|\right)\right), \quad (10)$$

where p_{tj} is the penalty that agent a_j should pay for its nonfeasance to resource contribution in executing task t.

Therefore, the manager agent a_t should now pay the penalty with the utilities of $(\omega r_t + p_{tt})$; each contractor agent, a_k, should pay the penalty with the utilities of p_{tk}. Finally, the negotiation reputations of all the agents in A_t will be reduced according to their penalties paid. $\forall a_j \in A_t$, if a_j's negotiation reputation is λ_j, we can reduce λ_j as follows:

$$\begin{aligned} a_t : \lambda_t &= \lambda_t \cdot (1 - \omega r_t - p_{tt}) \\ &= \lambda_t \cdot \left(1 - \omega r_t - (1-\omega) r_t \left(1 - |R^t_{a_t}| / |R_t \cap R_{a_t}|\right)\right), \\ \forall a_k &\in (A_t - \{a_t\}) : \lambda_k = \lambda_k \cdot (1 - p_{tk}) \\ &= \lambda_k \cdot \left(1 - (1-\omega) r_t \left(1 - |R^t_{a_k}| / |R_t \cap R_{a_k}|\right)\right). \end{aligned} \quad (11)$$

From above, the reward of an agent varies directly as that agent' real resource contribution in task execution; and the penalty varies inversely as that agent's real resource contribution in task execution. *Therefore, our reward and punishment mechanism can encourage agents to act as truthful ones.*

5.3.2 Adjustment for Interaction Relation Weights

In the reward and punishment mechanism, only the negotiation reputations of the allocated agents are influenced. The negotiations among the allocated agents are implemented along the social network; therefore, the weights of the interaction relations related to the allocated agents should also be adjusted after task execution.

1) *Interaction relations within the negotiation paths.*

Let the set of allocated agents for t be A_t, where a_t is the manager agent. The negotiation paths for task t are then those between a_t and all agents in $A_t - \{a_t\}$.

The adjustment for the weights of the interaction relations within the negotiation paths should be implemented according to the extent of adjustment of the negotiation reputations of a_t and a_j. We now present the algorithm for the adjustment of the weights of the interaction relations within the negotiation paths, shown as Algorithm 3.

Finally, the intermediate agents within the negotiation paths will be influenced because they provide communication relay services for the negotiations between the manager agent and contractor agents. *Therefore, the weight adjustment mechanism encourages agents not only to contribute resources to task execution but also to provide communication relay services for other agents' task execution.*

2) *Interaction relations associated with allocated agents.*

After the negotiation reputations of agents in A_t are adjusted, we should adjust the negotiation strengths between the agents in A_t and their neighbors. Moreover, if the weight of an interaction relation has already been changed by Algorithm 3, it does not need to be changed again at this point, i.e., the weight of an interaction relation can only be adjusted once for each task.

Algorithm 3. Adjustment for the weights of interaction relations within the negotiation paths.

$\forall a_j \in (A_t - \{a_t\})$:
  $\forall \langle a_x, a_y \rangle \in NP_{tj}$:
    If t is finished successfully, **then**:
      $w(a_x, a_y) = \min(1, w(a_x, a_y) \cdot (1 + ((\omega r_t + r_{tt}) + r_{tj})/2))$;
    Else: $w(a_x, a_y) = \max(0, w(a_x, a_y) \cdot (1 - ((\omega r_t + p_{tt}) + p_{tj})/2))$.

Algorithm 4. Adjusting the weights of the interaction relations immediately associated with allocated agents.

/* λ_x is the negotiation reputation of a_x before executing task t; λ'_x is that of a_x after executing task t; a_t is the manager agent; $\forall a_x \in A_t$; the set of a_x's immediate neighbors in the MAS-SN is N_x */

$\forall a_x \in A_t$:
  $\forall a_y \in N_x$:
- $b = 0$;
- $\forall a_j \in (A_t - \{a_t\})$: **if** $\langle a_y, a_x \rangle \in NP_{tj}$ **then** $b = 1$;
- **If** $b == 0$, **then**: $w(a_y, a_x) = w(a_y, a_x) \cdot (1 + (\lambda'_x - \lambda_x)/\lambda_x)$

5.3.3 Reallocation of Tasks

If task t is executed unsuccessfully, we will perform a reallocation of the task, including two aspects:

- If the manager agent does not provide the desired resources in task execution, the system will then make a new allocation of both the manager agent (the deceptive manager agent is excluded in the new allocation) and contractor agents.
- If a contractor agent does not provide the desired resources in task execution, only a new allocation of contractor agents is necessary (the deceptive contractor agent is excluded in the new allocation).

6 Experimental Validation and Analyses

We perform our experiments based on our proposed simulation platform for MAS-SN. Without loss of generality, we use three criteria to satisfy the different resource requirements of tasks [10] in our experiments: 1) First required resource-first satisfy (FRFS); 2) most important resource-first satisfy (MIFS); and 3) all resources-averagely satisfy (AAS). We compare our model with the following four approaches:

- *Self-owned resource-based task allocation model (SR model).* The selection of the manager agent and contractor agents is based on the amount of the agents' stated resources.
- *Self-owned resource-based task allocation model with redundant resources (R-SR model).* Each task can be allocated more redundant resources to avoid the problem of undependable allocated resources.
- *Game theory-based task allocation model (Game theory model).* The manager agent uses game theory approach to seek contractor agents considering the possibility of deceptive behaviors; meanwhile, the deceptive contractor agents also estimate the actions

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

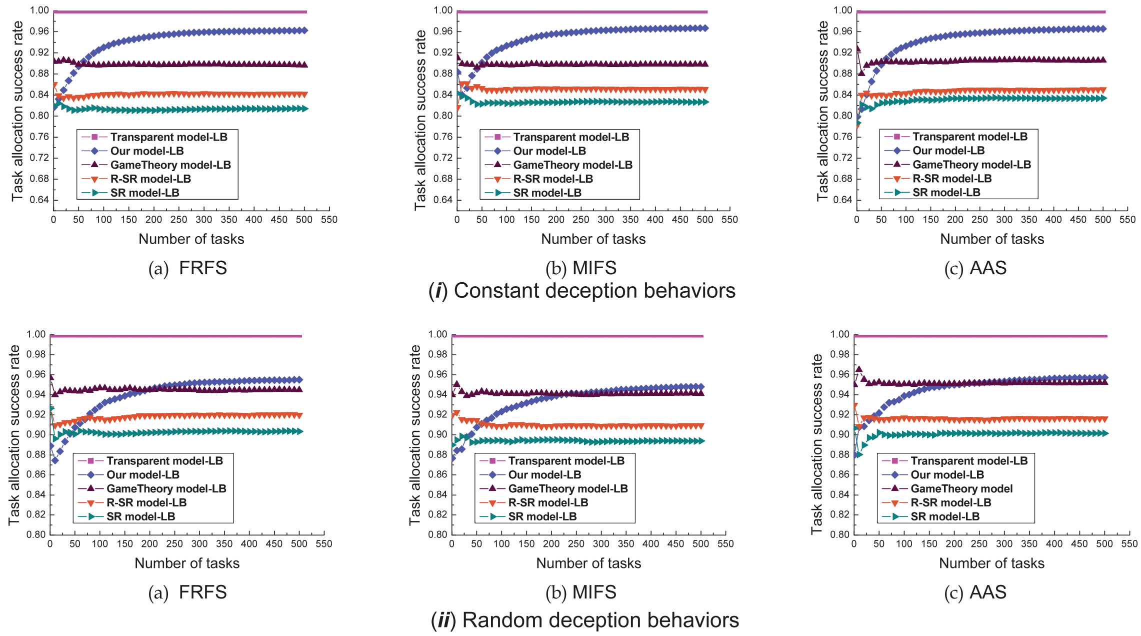


Fig. 1. Experiments on success rate of task allocation (LB means the load balancing is considered).

of manager agents and use game theory approach to provide resource assistance.

- *Ideal task allocation model in which all deceptive agents can be detected (Transparent model)*. The ideal method is that all deceptive agents can be detected, and tasks are only allocated to truthful agents.

To validate our model, we perform a series of experiments based on small world social network model [29] and further test our model on several typical networks in the Appendix. The introduction of experiment environment and setup can been seen in the Appendix. Five kinds of experiments have been performed:

1. tests of the success rate of tasks (*primarily for Objective 1*);
2. tests of the execution time of tasks (*primarily for Objective 2, partly for Objective 1*);
3. tests of the load balancing of tasks (*primarily for Objective 3*);
4. tests on the effects of MAS-SN structures and parameters in the model on task allocation performance (*to test the generality of our model*); and
5. tests of the situation where reputations and interaction relation weights may be manipulated by deceptive agents (*to test the robustness of our model*).

The first three kinds of tests can be seen in Sections 6.1, 6.2, and 6.3, respectively; the last two kinds of tests are shown in the Appendix for the space limitation.

6.1 Tests of the Success Rate of Tasks

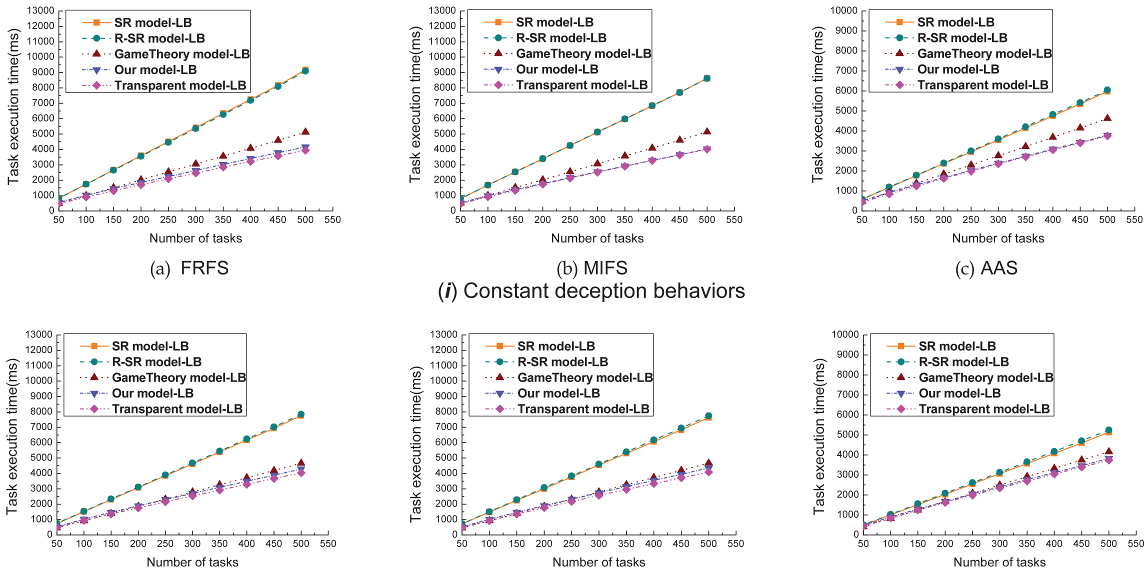
The success rate of a task is computed as follows: If a task can only be executed successfully after the nth allocation, the success rate of that task is $1/n$. The total success rate of a set of tasks is then the mean of the success rates of all tasks.

From the experimental results in Fig. 1, we conclude the following:

1. In all the experiments, the task allocation success rate of our model varies directly with the number of allocated tasks and can eventually reach more than 90 percent, showing that our model achieves a higher dependability of resource access, especially for a large number of tasks.
2. In all the experiments, the transparent model can detect all deceptive agents and all tasks can be allocated to truthful agents; therefore, its success rate is always 1. The R-SR model always outperforms SR because the presence of some redundant resources can improve the probability that the tasks will obtain dependable resources in execution.
3. In the early stages of each experiment, the SR (R-SR) and game theory models outperform our model; however, as the number of tasks increases our model performs better. The reason is that the global reputation mechanism is not conducted well in the early stages.
4. The success rate of our model for constant deception behaviors slightly outperforms the success rate for random deception behaviors; this may be because the effect of our reward and punishment mechanism is weakened when deceptive agents randomly adopt varying deceptive behaviors.
5. The performance gap between our model and game theory model for the random deception behaviors is smaller than the one for constant deception behaviors; the reason is that game theory model performs better for random deception behaviors than for constant ones, but our model performs better for constant deception behaviors than for random ones.

In conclusion, our model improves the probability that tasks will obtain dependable resources in social networks more effectively than the previous resource-based approaches and game theory-based approaches, especially when the number of tasks is high or the deception

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.



(a) FRFS (b) MIFS (c) AAS

(*i*) Constant deception behaviors

(a) FRFS (b) MIFS (c) AAS

(*ii*) Random deception behaviors

Fig. 2. Experiments on execution time of tasks. (LB means the load balancing is considered).

behaviors are constant. Thus, Objective 1 of task allocation in undependable MAS-SN can be validated.

6.2 Tests of the Execution Time of Tasks

The execution time of a task is the length of time between the task's arrival at the MAS-SN and its successful execution; the total execution time of a set of tasks is the sum of the execution times of all the tasks.

From the experimental results in Fig. 2, we determine the following:

1. Our model outperforms the SR(R-SR) and game theory models in all experiments, showing that our model can effectively reduce the necessary communication time for accessing the tasks' required resources.
2. Our model's performance is close to that of the transparent model; therefore, our model effectively improves the probability that the truthful agents will receive tasks.
3. The SR and R-SR models perform poorly because they do not consider the deceptive agents at all; therefore, frequent reallocation may be necessary.
4. The randomness of deception behaviors does not have an obvious effect on task execution time, which shows the generality of our model on reducing the communication time for accessing the tasks' required resources.

In conclusion, our model effectively reduces the communication time necessary for accessing the tasks' required resources in social networks better than the previous resource-based approaches and game theory-based approaches, especially when the number of tasks is high. Moreover, our model performs close to the ideal approach in terms of task execution time. Therefore, Objective 2 and partial Objective 1 of task allocation in undependable MAS-SN can be validated.

6.3 Tests of the Load Balancing of Tasks

We now validate our load balancing mechanism by conducting a comparison between our model with load balancing (Our model-LB) and our model without load balancing (Our model).

From the experimental results in Fig. 3, we conclude the following:

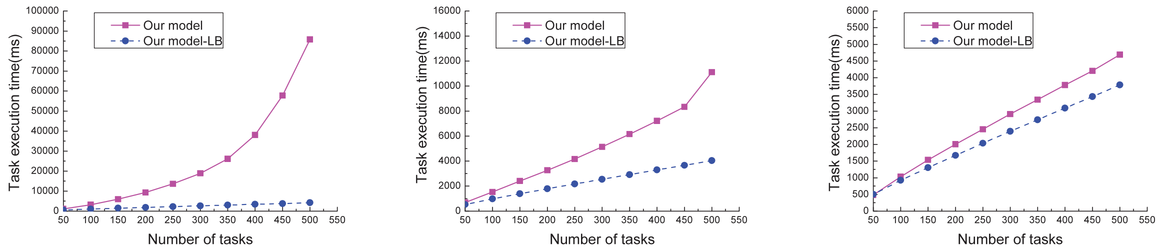
1. Our model-LB always outperforms our model without load balancing, especially when the number of tasks is higher, showing that our load balancing mechanism can effectively reduce the tasks' waiting time for resources at the agents.
2. The effect of our load balancing in AAS is weaker than the one in FRFS and MIFS; the potential reason is that AAS itself averages the overall required resources so that the unbalanced situation of task allocation are eased, thus now the effect of load balancing is not very obvious.

In conclusion, the experimental results prove that our load balancing mechanism can effectively reduce tasks' waiting time for resources at agents, especially when the number of tasks is high. Therefore, Objective 3 of task allocation in undependable MAS-SN can be validated.

7 Conclusions and Discussion

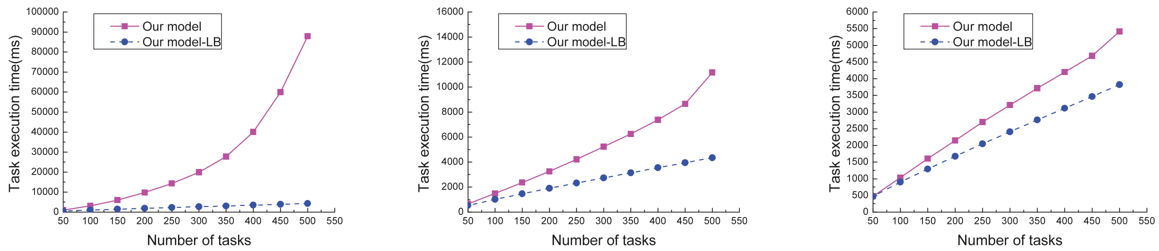
In undependable MAS-SN, deceptive agents may fabricate their resource status information during task allocation but not really contribute resources during task execution. To solve this problem, this paper presents task allocation objectives as follows: 1) improve the probability that tasks will obtain dependable resource access for successful execution; 2) reduce the communication time to access tasks' required resources in social networks; and 3) reduce the tasks' waiting time for the required resources.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.



(a) FRFS (b) MIFS (c) AAS

(*i*) Constant deception behaviors



(a) FRFS (b) MIFS (c) AAS

(*ii*) Random deception behaviors

Fig. 3. Task execution time comparison between our model with load balancing and our model without load balancing.

To achieve the first objective, a negotiation reputation-based allocation mechanism and a reward/punishment mechanism are designed to ensure that the truthful agents have higher probabilities of receiving tasks; to achieve the first and second objectives, a resource negotiation mechanism is designed to ensure that truthful agents with smaller communication distances have higher probabilities of being designated contractor agents; to achieve the third objective, load balancing is adopted in task allocation so that the problem of waiting time at heavy-burdened agents can be alleviated.

Through a series of experiments, our model can be validated for achieving the three objectives. Moreover, through the experiments of our model on typical networks in the appendix, it can be found that our model can perform better on typical social network structures.

In this paper, we mainly concern about the minimization of access time for dependable resources in social networks. However, in reality there are more other costs for executing tasks; therefore, in the future, we will consider situations where other costs can be minimized. Moreover, our future work will extend our model to dynamic topology social networks.

Acknowledgments

This work was supported by the NSFC (Nos. 61170164, 60803060), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20090092110048), the Program for Distinguished Talents of Six Domains in Jiangsu Province (No. 2011-DZ023), the FSSP (No. 20110092110053), the Outstanding Young Scholars of Southeast University (No. Seujq201111), and the Funds for Distinguished Young Scholars of Jiangsu Province (BK2012020). The authors wish to thank Mr. J. Jiang for the design of reputation mechanism.

References

[1] J. Sabater and C. Sierra, "Reputation and Social Network Analysis in Multi-Agent Systems," *Proc. First Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '02)*, pp. 475-482, July 2002.

[2] J.M. Pujol, R. Sangüesa, and J. Delgado, "Extracting Reputation in Multi Agent Systems by Means of Social Network Topology," *Proc. First Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '02)*, pp. 467-474, July 2002.

[3] M.M. Weerdt, Y. Zhang, and T. Klos, "Multiagent Task Allocation in Social Networks," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 1, pp. 46-86, 2012.

[4] S. Abdallah and V. Lesser, "Multiagent Reinforcement Learning and Self-Organization in a Network of Agents," *Proc. Sixth Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '07)*, pp. 172-179, May 2007.

[5] Y. Jiang, J. Hu, and D. Lin, "Decision Making of Networked Multiagent Systems for Interaction Structures," *IEEE Trans. Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1107-1121, Nov. 2011.

[6] J. Jiang and Z. Xia, "Cluster Partition-Based Communication of Multiagents: The Model and Analyses," *Advances in Eng. Software*, vol. 42, no. 10, pp. 807-814, 2011.

[7] J. Jiang and X. Xia, "Prominence Convergence in the Collective Synchronization of Situated Multi-Agents," *Information Processing Letters*, vol. 109, no. 5, pp. 278-285, 2009.

[8] H. Ohtsuki, C. Hauert, E. Lieberman, and M.A. Nowak, "A Simple Rule for The Evolution of Cooperation on Graphs and Social Networks," *Nature*, vol. 441, pp. 502-505, May 2006.

[9] B. An, V. Lesser, and K.M. Sim, "Strategic Agents for Multi-Resource Negotiation," *J. Autonomous Agents and Multi-Agent Systems*, vol. 23, no. 1, pp. 114-153, 2011.

[10] Y. Jiang and J. Jiang, "Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 5, pp. 641-653, May 2009.

[11] J. Liu, X. Jin, and Y. Wang, "Agent-Based Load Balancing on Homogeneous Minigrids: Macroscopic Modeling and Characterization," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 7, pp. 586-598, July 2005.

[12] S. Kraus and T. Plotkin, "Algorithms of Distributed Task Allocation for Cooperative Agents," *Theoretical Computer Science*, vol. 242, nos. 1/2, pp. 1-27, 2000.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.

[13] G. Zlotkin and J.S. Rosenschein, "Incomplete Information and Deception in Multi-Agent Negotiation," *Proc. 12th Int'l Joint Conf. Artificial Intelligence (IJCAI '91)*, pp. 225-231, Aug. 1991.

[14] S. Penmatsa and A.T. Chronopoulos, "Game-Theoretic Static Load Balancing for Distributed Systems," *J. Parallel and Distributed Computing*, vol. 71, no. 4, pp. 537-555, 2011.

[15] D. Grosu and A.T. Chronopoulos, "Noncooperative Load Balancing in Distributed Systems," *J. Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1022-1034, 2005.

[16] S. Fjuita and V.R. Lesser, "Centralized Task Distribution in the Presence of Uncertainty and Time Deadlines," *Proc. Second Int'l Conf. Multiagent Systems (ICMAS '96)*, pp. 87-94, Dec. 1996.

[17] S. Kraus, O. Shehory, and G. Taase, "Coalition Formation with Uncertain Heterogeneous Information," *Proc. Second Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '03)*, July 2003.

[18] S.D. Ramchurn, C. Mezzetti, A. Giovannucii, J.A. Rodriguez-Aguilar, R.K. Dash, and N.R. Jennings, "Trust-Based Mechanisms for Robust and Efficient Task Allocation in the Presence of Execution Uncertainty," *J. Artificial Intelligence Research*, vol. 35, no. 1, pp. 119-159, 2009.

[19] E. Ephrati and J.S. Rosenschein, "Multi-Agent Planning as Search for a Consensus That Maximizes Social Welfare," *Lecture Notes in Computer Science*, vol. 830, pp. 207-226, Springer, 1994.

[20] O. Shehory, S. Kraus, "Methods for Task Allocation via Agent Coalition Formation," *Artificial Intelligence*, vol. 101, nos. 1/2, pp. 165-200, 1998.

[21] M. de Weerdt, Y. Zhang, and T. Klos, "Distributed Task Allocation in Social Networks," *Proc. Sixth Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '07)*, pp. 500-507, May 2007.

[22] Y. Jiang and Z. Li, "Locality-Sensitive Task Allocation and Load Balancing in Networked Multiagent Systems: Talent versus Centrality," *J. Parallel and Distributed Computing*, vol. 71, no. 6, pp. 822-836, 2011.

[23] R. Kota, N. Gibbins, and N.R. Jennings, "Decentralised Approaches for Self-Adaptation in Agent Organizations," *ACM Trans. Autonomous and Adaptive Systems*, vol. 7, no. 1, pp. 1-28, 2012.

[24] K.P. Chow and Y.K. Kwok, "On Load Balancing for Distributed Multiagent Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 8, pp. 787-801, Aug. 2002.

[25] J. Xu, A.Y.S. Lam, and V.O.K. Li, "Chemical Reaction Optimization for Task Scheduling in Grid Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1624-1631, Oct. 2011.

[26] Y. Jiang and Z. Huang, "The Rich Get Richer: Preferential Attachment in the Task Allocation of Cooperative Networked Multiagent Systems with Resource Caching," *IEEE Trans. Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 5, pp. 1040-1052, Sept. 2012.

[27] Y. Jiang, J. Jiang, and T. Ishida, "Compatibility between The Local and Social Performances of Multi-Agent Societies," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4443-4450, 2009.

[28] B. Hong and V.K. Prasanna, "Adaptive Allocation of Independent Tasks to Maximize Throughput," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1420-1435, Oct. 2007.

[29] M.E.J. Newman and D.J. Watts, "Renormalization Group Analysis of The Small-World Network Model," *Physics Letters A*, vol. 263, nos. 4-6, pp. 341-346, 1999.

Yichuan Jiang received the PhD degree in computer science from Fudan University, Shanghai, China, in 2005. He is currently a full professor and the director of the Laboratory for Complex Systems and Social Computing, School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include multiagent systems, social networks, social computing, and complex distributed systems. He has published more than 70 scientific articles in refereed journals and conference proceedings, such as TPDS, TSMC-A, TSMC-C, JPDC, IJCAI, and AAMAS. He is a member of editorial board of Advances in Internet of Things, an editor of *International Journal of Networked Computing and Advanced Information Management*, an editor of Operations Research and Fuzziology, and a member of the editorial board of Chinese Journal of Computers. He was awarded the New Century Excellent Talents in University of State Education Ministry of China, the Best Paper Award from PRIMA2006, and the Nomination Award for the National Excellent Doctoral Dissertation of China. He is also the recipient of the HwaYing Young Scholar Award in 2008 from The HwaYing Education and Culture Foundation, and the Young Distinguished Professor in Southeast University. He is a senior member of the IEEE, the CCF, and the CIE.

Yifeng Zhou received the BE degree in electronic engineering from Southeast University, Nanjing, China, 2007. He is currently working toward the PhD degree at the Laboratory for Complex Systems and Social Computing, School of Computer Science and Engineering, Southeast University. His main research interests include social networks and multiagent systems.

Wanyuan Wang received the BS degree in information and computing science from Nanjing University of Aeronautics and Astronautics, China, 2011. He is currently working toward the PhD degree at the Laboratory for Complex Systems and Social Computing, School of Computer Science and Engineering, Southeast University. His main research interests include social networks and multiagent systems.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on May 08,2025 at 07:32:03 UTC from IEEE Xplore. Restrictions apply.