

Article



Task Assignment of the Improved Contract Net Protocol under a Multi-Agent System

Jiarui Zhang ^{1,*}, Gang Wang ² and Yafei Song ²

- ¹ Graduate College, Air Force Engineering University, Xi'an 710051, China
- ² Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China;
- sharesunny@163.com (G.W.); sharesunny12@163.com (Y.S.)
- * Correspondence: zjr888@stu.xjtu.edu.cn; Tel.: +86-153-8450-1388

Received: 18 January 2019; Accepted: 25 March 2019; Published: 1 April 2019



Abstract: Background: The existing contract net protocol has low overall efficiency during the bidding and release period, and a large amount of redundant information is generated during the negotiation process. Methods: On the basis of an ant colony algorithm, the dynamic response threshold model and the flow of pheromone model were established, then the complete task allocation process was designed. Three experimental settings were simulated under different conditions. Results: When the number of agents was 20 and the maximum load value was $L_{max} = 3$, the traffic and run-time of task allocation under the improved contract net protocol had advantages over the dynamic contract net and classical contract net protocols in terms of both traffic and run-time. Setting up the number of agents, tasks and L_{max} to improve the task allocation under the contract net not only minimizes the number of errors, but also the task completion rate reaches 100%. Conclusions: The improved contract net protocols. Furthermore, the algorithm can achieve better assignment results and can re-forward all erroneous tasks.

Keywords: multi-agent system; contract net protocol; task assignment; response threshold; pheromone

1. Introduction

With the rapid development of information technology and distributed system theory, multi-agent systems (MAS) have attracted widespread attention from researchers. Agents can sense their own and external state changes, and can generate corresponding autonomous behavior.

The application of the multi-agent system concept to the military field is the key to future war systems. Each combat unit connected to the network can be regarded as an agent. Due to complex and volatile war environments that may destroy agents, the ability of the agent during the execution of the mission may change. Therefore, studying the multi-agent dynamic task assignment problem is of great significance for improving multi-agent coordinated combat capability. Before the start of an operation, the command center usually forms complex tasks based on the information on the battlefield that has been mastered. First, the complex tasks must be decoupled, decomposed into a number of different meta-tasks, and then pre-allocated as tasks according to the capabilities of each of their own combat units. Later, as the battlefield environment changes and the agents have an unexpected situation during the execution of the mission, if the task is executed according to the previous scheme, the performance of the task may be reduced. Therefore, an online task assignment method must be used to reallocate these meta-tasks in real time according to the capabilities of the current agent.

How to select the appropriate agent to complete the assignment of tasks has always been a popular and difficult research topic. Contract net protocol based on the market mechanism is an important task allocation strategy [1]. Tasks are usually time-sensitive and have priority, and also must be completed in time to meet the priority order. On the other hand, it is often difficult to find a global optimal solution for the assignment of combat tasks. Therefore, the task allocation goal is to use the least resources possible to make all tasks efficient. In other words, it can be said that they are looking for the best sub-optimal solution. In this study, a solution was found that makes the task completion rate 100%, while using the least amount of traffic and run-time.

Contract net protocol was originally proposed by R.G.Smith, as a classical negotiation strategy for human and resource allocation in distributed sensor systems [2]. It has been widely used in task allocation of distributed systems and production scheduling. Furthermore, it has achieved good results in the field of dynamic task allocation. Researchers at home and abroad have also proposed many expanded and improved forms for traditional contract net protocols (CNP). Wunan et al. [3] reduced traffic by introducing a case-based reasoning contract net through historical information to make decisions. Chen et al. [4] directly sent the bidding notice to agents by introducing interactive trust, acquaintance trust and other strategies to reduce the amount of communication. Sandbolm et al. [5] proposed a variety of methods, including receiver restriction, centralized selection, and even ignoring expired messages. Zhang [6] introduced the concept of trust degree and proposed the dynamic contract net protocol (DCNP), which can be updated continuously. A previous study [7] introduced the buffer pool idea into the bidding stage of the contract net protocol, so as to solve the problems of prominent uncertainty and low efficiency in the bidding stage, while another study [8] introduced task transfer stations and set the bidding threshold to improve the algorithm of the dynamic contract net protocol.

As an important task allocation method in MAS, the basic idea of contract net agreement is "tender-bidding–winning", which enables the whole system to complete task allocation with a lower cost and higher quality. However, there are still some blind spots in the research of contract net protocols when facing specific problems [9,10]. For example, the method of reducing communication traffic in the initial bidding stage depends on historical information, but the lack of historical information in the initial stage means the overall allocation efficiency is not high. Additionally, there are many instances of negotiation communication between the manager agent and the contractor agent, and a lot of redundant information is generated in the process of repeated negotiation, which can easily cause network congestion and deadlock. Based on the research of scholars at home and abroad, this study establishes the dynamic response threshold model and pheromone flow model by combining an ant colony algorithm. We have also designed three experiments to simulate the Repast platform. The results of the three experiments are explained in Sections 5.1–5.3. The improved contract net protocol (ICNP) not only has obvious advantages in operation time and communication traffic, but can also effectively improve the rationality of task allocation and the efficiency of task completion, as well as achieve optimal task allocation results.

2. Improvement of the Contract Net Protocol Model

We were given a MAS with a manager agent and multiple contractor agents connected by a distributed network topology. The manager agent is mainly responsible for the decoupling of tasks, the issuance of tenders, and the assignment of tasks. Each contractor agent can feed back the cost and load of performing this task according to the bidding task. The manager agent and contractor agent are not fixed and the roles can be interchanged as needed. They are given tasks and each task can be decomposed into several different types of subtasks. The same type of subtasks form a task set. Each subtask has its own parameters, including priority, deadline, corresponding remuneration, etc.

2.1. The Agent Model

For a task assignment under the MAS system, the participation of two kinds of agents and the public environment are needed, including a manager agent and contractor agent. The manager agent is the manager who formulates the bidding task and evaluates the winning bidding node. It is mainly responsible for the decomposition of tasks, the formulation and publication of tenders, the collection of

tenders, the judgment of winning bids and the signing of contracts, as well as the supervision of task execution and effect evaluation. The contractor agent is mainly responsible for bidding according to its own working status and ability. In the whole contract negotiation process, the role of the agent does not need to be pre-defined, any agent can become a manager by releasing tasks and can be a contractor by participating in bidding activities. We first needed to model the agent [11–15]. The basic structure of a single agent is shown in Figure 1 below.



Figure 1. Agent basic structure.

When acting as a manager agent, each agent's knowledge base, processing and decision center and communication coordinator play an effective role, while when acting as a contractor agent, the task actuator also runs. The knowledge base stores historical assignment information, task related information, agent status and capability information. The processing and decision-making center can decouple tasks, publishing tenders and assigning tasks when acting as a manager agent. According to the information of the bidding task, the contractor agent can feedback costs, loads and the other information of executing the task. After the contractor agent signs the contract with the manager agent, the task actuator can perform corresponding tasks according to the decision of the processing and decision-making center. The communication coordinator can publish or receive information to communicate between nodes.

2.2. The Dynamic Response Threshold Model

In any population, worker ants can complete different tasks according to different ages, and we introduced a task stimulus accordingly. At the same time, we introduced a dynamic response threshold according to the idea that ants choose a shorter path with pheromone concentration [16,17]. Each task has a task stimulus and each agent has a response threshold related to a class of tasks to be completed (the initial response threshold is determined according to the static history instruction set). The manager agent can make a preliminary bidding judgment, but only when the stimulus of the task exceeds the response threshold of an agent will the tender be sent to it. Also, the lower the threshold, the higher the probability of accepting the task. If no agent is involved in bidding, the stimulus of the task will be increased until it can stimulate multiple agents. On the other hand, we set up a relationship between the threshold value and the trust degree, that is, the response threshold value changes dynamically. If the contractor agent completes the assigned task, the trust degree of the agent for a certain type of task will increase and the corresponding threshold for such a task will decrease. On the contrary, if the task is not completed, the trust degree will decrease and the response threshold will increase [18,19]. The response threshold is non-linearly variable. The more the number of consecutive and successfully completed tasks, the greater the increase of the response threshold. At the same time, the more the

task fails, the greater the response threshold decreases. Assuming that each task assignment process is recorded, *Val* represents the response threshold, which is the lower bidding limit for the different tasks of each agent, *Val* is calculated as follows:

$$\begin{cases} Y = \lambda_1 \frac{C_{acq}}{C} + \lambda_2 \frac{C_{true}}{C_{acq}} \\ \lambda_1 + \lambda_2 = 1 \end{cases}$$
(1)

$$Val = \begin{cases} Sti + Y + \alpha, & task \ accomplished \\ Sti - \beta, & task \ undone \end{cases}$$
(2)

where, *C* represents the number of tenders obtained by this agent, C_{acq} is the total number of tasks completed by the current contractor agent, C_{true} is the number of tasks successfully completed in the end. *Y* represents the magnitude of the increase in the *Val* value. The number of agents that successfully completes multiple tasks in succession is larger than the *Y* value of the agent that successfully completes the task, which is adjusted by using λ_1 and λ_2 (the initial values of λ_1 and λ_2 are both 0.5). α and β are constant and are related to the difficulty of the task and the priority of the task (both α and β have initial values of zero). When the contractor agent completes a lower priority and easier task, the value of α is larger when a higher priority and more difficult task is completed. When the contract agent does not complete an easy task with lower priority, the value of β is larger, and when the harder task with higher priority is not completed, the value is correspondingly smaller.

2.3. The Pheromone Flow Model

The information between the manager agent and the contractor agent is transmitted by a pheromone. According to the different roles of the agent at this time, the pheromone information released is also different [20]. The manager agent can release the task pheromone to the environment and the contractor agent can calculate the bidding pheromone according to the perceived task pheromone and release it to the public environment. Task pheromones and bid pheromones in the environment are constantly changing and remain for a period of time, as shown in Figure 2 below. Suppose there are *m* contractor agents and *n* tasks need to be completed, after four stages of pre bid-tender-bidding–winning, the classical contract net protocol and dynamic contract net protocol need more communication times, while the improved contractor net protocol in this study reduced the number of communication times of the classic contract net protocol, 2mn + 2n to mn + 2n times, reducing the number of communication times by $m \times n$ [21,22].



Figure 2. The flow of pheromones.

3. Task Allocation under the Improved Contract Net Protocol

3.1. Task Description

Once the manager agent receives the task, it firstly decouples the task and determines the stimulus of the task. Then the task pheromone is released to the public environment by a quaternions: *<ID*, *Nam*, *Pri*, *Sti*, *Rew>*, where, *ID* is the task number; *Nam* is the name of the task; *Pri* is the priority of the task, which is divided into five levels in total, with the highest priority level being five; *Sti* is the stimulus degree of each task, ranging from 0 to 1; and *Rew* is used to measure the complexity of the task and the difficulty of completing the task, with a size from 0.1 to 6, where 6 indicates that the complexity is the highest and the most difficult to accomplish.

3.2. Contractor Agent Description

The contractor agent perceives the task pheromone from the public environment according to its own response threshold. If the threshold value is less than the task stimulus, it indicates that it has the ability to accept the task. Then the contractor agent converts the task pheromone into the bidding pheromone by calculation and releases it into the public environment according to its current ability. The bidding pheromone is described by seven-tuples: *<ID*, *Nam*, H, *Cap*, *CosT*, *L*, *Val>*, where, *ID* and *Nam* are inherited from the task pheromone accepted in the public environment, describing the number and name of the task, respectively; H is the ID number of the contractor agent; *Cap* is the current task load of the contractor agent; *and Val* is the response threshold.

3.3. The Task Assignment Process

The task assignment process includes the whole bidding process of the improved contract net protocol, releasing a task pheromone to the public environment by the manager agent, obtaining a bidding pheromone by the contractor agent according to the accepted task pheromone and the manager agent selecting a winning contractor agent to complete the task according to the right bidding pheromone [23]. The whole process of assigned tasks is shown in the following steps 1 through 7.

Step 2: The contractor agent can perceive the task pheromone from the public environment, then the contractor agent will judge its own value of *Val* and the value of *Sti* in this task pheromone and the contractor agent will calculate the value of α , β and *CosT* according to Formulas (3) and (4) if *Val* > *Sti*.

$$\begin{cases} \alpha = \frac{Pri}{100} + \frac{Rew}{10} \\ \beta = \frac{\sin(2\pi pri \times Rew)}{100} + \frac{\exp(-(3Rew)^2)}{10} \end{cases} \quad \alpha, \beta \in [0.001, 0.1)$$
(3)

$$CosT = \sum_{i=1}^{r} T_{0i} \tag{4}$$

where, T_{0i} is the estimated time for the current agent to complete the task *i*, *r* is the number of tasks to be completed by the contractor agent and the lower limit of β is 0.001.

Step 3: The calculated value of *CosT* is compared with the agent's own ability value of *Cap* and it proceeds to the next step only when the value of *Cap* is greater than *CosT*. The bidding pheromone B<*ID*, *Nam*, H, *Cap*, *CosT*, *L*,*Val*> will be calculated and released into the public environment again. Otherwise, the contractor agent will abandon this task.

Step 4: The manager agent perceives the same *ID* and *Nam* of the bidding pheromones and ranks all bidding pheromones by a calculation. In the case of the same task set, the priority of the task is sorted from the value of the largest to the smallest. For two tasks with the same priority, they are sorted by a value from large to small. Furthermore, the comprehensive evaluation sequence table of the tasks is obtained accordingly. The value calculation is shown by Formula (5).

$$Z = \lambda_3 \times Y - \lambda_4 \times \frac{T_{\max} - CosT}{T_{\max} - T_{\min}} - (1 - \lambda_4) \times \frac{L_{\max} - L}{L_{\max} - L_{\min}}$$
(5)

where, $0 < \lambda_3$, $\lambda_4 < 1$ and T_{max} is the task completion time of the agent with the maximum value of *Val*, T_{min} is the task completion time of the agent whose value of *Val* is greater than *Sti* and less than the other *Val*, *L* is the number of tasks waiting to be completed in the task list of the current agent, L_{max} is the maximum number of waiting tasks among all contractor agents, L_{min} is the minimum value.

Step 5: Assign the task to the first agent in the list and inform it of winning the bid. Update the *L* value as shown in Formula (6) and check whether the task is completed with the contractor agent after the time of *CosT*.

$$L = L + 1 \tag{6}$$

Step 6: If the contractor agent that obtains the task successfully completes the task, it will update the value of *Val*; if the task is abandoned in the process of completing the task, it will update the current agent's value of *Val*, then notify the manager agent to check the second contractor agent in the comprehensive evaluation sequence of tasks. If the agent does not get the task after that, it will assign the task to that agent, otherwise it will repeat Steps 1–4.

Step 7: Complete the task.

The specific protocol flow of the improved contract net is shown in Figure 3. In order to make the whole process easy to explain, only one manager agent and one contractor agent are depicted in Figure 3.



Figure 3. Improving the contract net protocol process.

4. Simulation Experiment

In order to verify the performance of the new improved contract net protocol, the Repast platform was used to do the corresponding simulation, mainly through the communication traffic and run-time to compare the performance of the improved contract net protocol, dynamic contract net protocol and classical contract net protocol. As the number of agents is limited in the initial tender release stage and the number of point-to-point communications is reduced in the whole process, our experiment is specifically divided into three steps to verify the superiority of the improved contract net protocol [24].

4.1. Repast Simulation Platform

A Repast simulation toolkit (Repast) is a set of programming elements, which provides a series of nearly 130 class libraries. The core of the Repast simulation toolkit is based on agent modeling service kernel [25]. It has many similarities with Swarm's design, so it is called "Swarm-like architecture", but the difference is that Repast supports many high-level languages. The Repast system is mainly

divided into six modules: Analysis library, Engine library, Games library, Space library, GUI and Util [26]. The class diagram of the manager agent and contractor agent is shown in Figure 4.

Manager Agent	Contractor Agent
<pre>-ID:int +taskName:String +taskParameter[]:double +communacation:int -count:int #taskPerTick:object +ManagerAgent(SimModel, int, double, int, int)() +getID():int +getTaskName():String +getTaskParameter[]:double +getTaskPerTick():int +setTaskPerTick():int +setTaskPerTick(int)() +assigmentTask() +communacation():int +getCount():int -calculate()</pre>	<pre>-ID: int +taskName:String +agentValue:double +agentCost:double +agentCap:double -agentL: int +maxAgentL: int +rCount -taskList:object -temp:object *ContractorAgentModel(double, double, double, int)() +getID(): int +setID(int)() +getTaskName():String +getTaskName():String +getTaskValue:double -resetIndices()() +getAgentL(): int +getTaskList()():object +performTask()() +performTask()() +addTask():bool +getAgentCost()() +changeCost(int, int)() +getrCount(): int -update(): int +draw()()</pre>

Figure 4. Unified Modeling Language (UML) class diagram of manager agent and contractor agent.

4.2. Experimental Settings

A manager agent and a plurality of contractor agents participating in the bidding together constitute the main body of the entire bidding process. There are five different task sets in M1–M5, the number of tasks in each task set is the same and each task group belongs to the same type but the parameters are different. The priority, stimulus and complexity of two tasks in each group are shown in Table 1 below.

Task Set	Task Number	Pri	Sti	Rew
M1	101	1	0.2	3.4
	102	3	0.6	2.7
M2	201	2	0.4	1.6
	202	2	0.6	4.2
M3	301	4	0.2	5.1
	302	3	0.7	2.7
M4	401	3	0.1	3.4
	402	5	0.8	5.1
M5	501	4	0.6	3.0
	502	1	0.3	0.9

Table 1. Part task parameter setting information.

5. Results

5.1. Fixed Agent Quantity

For the convenience of comparison, two variables, the total number of completed tasks and the maximum load number of each agent, were set respectively in Experiment 1 on the basis of the

improved contract net protocol. This study records and compares the changes of the communication traffic and run-time in the whole system. When the number of agents is fixed at 20, the total number of tasks is set at 50, 100, 200, 500, 800 and 1000 for task allocation. For example, when the total number of tasks is 100, 20 tasks are included from each group of 1–5, where, the maximum load is equal to 1, 2, 3, and 4, respectively.

As can be seen from Figure 5, with an increase in the number of completed tasks under the improved contract net algorithm, both the communication volume between agents and the run-time to complete all tasks increase accordingly. On the other hand, all agents set a maximum load not less than one, namely each contractor agent has a buffer pool, when the maximum load is one, the number of buffer pools is zero. The L_{max} value of the contractor agent is greater than one, that is, after setting up the buffer pool for the contractor agent, both the traffic and the run-time decrease, but when the load number increases further, the traffic and the run-time show a downward trend. When $L_{max} = 3$, the number of tasks completed after 500 times, the amount of traffic and run-time were less.



Figure 5. Quantitative analysis with different task numbers. (a) The traffic of the improved contract net protocol with different numbers of tasks; (b) The run-time of the improved contract net protocol with different numbers of tasks.

The overall sample is 24, divided into four groups according to L = 1, 2, 3 and 4, each group is the traffic required to complete the unit task. Using the Kruskal–Wallis test [27] to begin, we established an α level equal to 0.05 and we proposed a null hypothesis. H0 was that the four groups of data are the same and H1 was that the overall distribution of each group is not the same. All data were combined and sorted and given the sequence number value, and the four groups of sequence numbers were summed to obtain R1, R2, R3 and R4, respectively. Then, the H test statistic was calculated again by spss. The p value = 0.038 (<0.05), so we rejected H0, i.e., the traffic is different when the load value is different and the overall traffic is the smallest when L = 3. Using the same method, you can get less run time when L = 3. When $L_{max} = 1$, only the contractor agent not executing a task can accept a new task when the buffer pool is not set. If the contractor agent with a strong ability to complete this task is performing tasks, the manager agent can only send this task to the agent with a weak ability. As the weaker contractor agent takes a long time to complete this task, the actual completion time may be longer than the sum of the completion time of the strong contractor agent and the waiting time of the buffer pool. When the value of L_{max} is large and the number of buffer pools is set too high, the waiting task of a certain contractor agent with strong capability may be too much and the run-time is longer than the time to re-select other contractor agents, thus resulting in an increase of traffic. Therefore, the contractor agent should choose the appropriate L_{max} value, which can reduce the run-time, traffic and improve the overall performance of the system.

5.2. Change of Agent Quantity

Based on the analysis of the results of Experiment 1, we designed a new experiment. On the basis of fixing the total number of tasks and the number of loads, the performance of three algorithms, the classical contract net protocol, dynamic contract net protocol and improved contract net protocol, were compared and analyzed by changing the total number of contract agents in Experiment 2.

When the buffer pools are set for each contractor agent, the L_{max} value is equal to 3 and the number of tasks is 1000, and the number of contractor agents is changed to 10, 20, 30, 40 and 50, respectively. The performance comparison of the three protocols is shown in Figure 6.



Figure 6. Comparative analysis of the three algorithms with different agent numbers. (**a**) The traffic of the improved contract net protocol with a different number of tasks; (**b**) The run-time of the improved contract net protocol with a different number of tasks.

As can be seen from the two figures above, when the number of tasks is fixed, with an increase in the number of agents participating in task allocation, the traffic of the three contract network protocols increases and the run-time increases. Compared with the dynamic contract net and classical contract net protocols, the improved contract net has a lower communication volume and lower run-time, with the advantage of growing with the increase in the number of agents. As there is no dynamic response threshold setting in the classical contract net and dynamic contract net protocols, the initial communication volume is large and the maximum load number is not set in the two algorithms, the manager agent selects the contractor agent, but does not always choose the agent with the strongest ability. Furthermore, the run-time is large in these protocols. In the improved contract net protocol, the agents have different response thresholds for different tasks and the response threshold can change with completion of the task. The stronger the ability to complete the task, the larger the response threshold and the fewer communication times. Therefore, the improved contract net protocol has advantages over other contract net protocols.

Similarly, the three algorithms are divided into three groups and the rank sum test is used to obtain the traffic and run-time distribution position when the number of agents is 30.

5.3. All Variables Fixed

On the basis of Experiment 1 and Experiment 2, Experiment 3 sets the number of agents as 30 and the number of tasks as 1000, L_{max} is equal to 3, and three contract net protocol algorithms are used for task allocation. The statistics of the number of errors and the overall task completion rate are shown in Table 2 and the assignment of 30 agents is shown in Figure 7.

Algorithms	Number of Error Tasks	Number of Successful Re-Forward Tasks	Task Completion Rate
CNP	133	0	86.7%
DCNP	68	0	93.2%
ICNP	11	11	100%

Table 2. Number of errant tasks and task completion rate.

From Table 2, we can see that the classical contract net protocol has the largest number of errors, followed by the dynamic contract net protocol. The improved contract net not only has the least number of errors, but can also re-forward these error tasks, which can be successfully completed by other contractor agents. This is because the traditional contract net protocol does not respond to the threshold and trust degree, and the number of task errors is relatively large. Although the dynamic contract net protocol has a trust degree, it is only linearly increased and reduced, and the contractor agent with a strong ability cannot be selected.



Figure 7. Task allocation of 30 agents under three algorithms.

As can be seen from Figure 7, most of these error tasks are concentrated on a few agents. When the contractor agents fail under the first two contract net protocols, they do not timely notify the manager agent, resulting in the accumulation of tasks and a failure to complete. In the improved contract net protocol, the manager agent can identify whether the contractor agent is at fault after *CosT* time, change its response threshold, and then ensure the re-completion of the task according to the comprehensive evaluation sequence table.

6. Conclusions

Although the improved contract net protocol has greater advantages in communication and run-time than the other two algorithms, the large amount of redundant information in the public environment occupies the storage capacity, which may make the efficiency of the contracting agent lower. In addition to the following summary of the work of the article, the constraints on agent collaborative bidding and task assignment are the focus of future research.

- 1. We set different task stimulus levels for different tasks and established the dynamic response threshold model to solve the problem of less restrictions on initial tender publication in traditional contract net protocols. A combination of history and norms was used to filter some meaningless contractors, and real-time adjust the trust level of contractor agents to reduce the traffic.
- 2. The pheromone flow model was constructed, and the task pheromone released by the manager agent and the bidding pheromone released by the contractor agent were used to coordinate each other in the public environment to complete the task allocation model. Furthermore, a complete process was designed for the whole task allocation process.
- 3. The Repast platform simulated the three experimental settings to get the tasks assignment: the number of tasks and the maximum load are uncertain, the number of agents is uncertain and all parameters are determined. The results show that the improved contract net protocol has the advantages of a small communication volume, short run-time, high task completion rate and multi-party coordination ability. Compared with the classical contract net protocol and dynamic contract net protocol, it has obvious advantages.

Author Contributions: Conceptualization, G.W. and J.Z.; methodology, G.W.; software, J.Z.; validation, G.W., J.Z. and Y.S.; formal analysis, J.Z.; resources, G.W.; data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, G.W.; project administration, Y.S.; funding acquisition, Y.S.

Funding: This research was funded by NATURAL SCIENCE FOUNDATION OF CHINA, grant number: 61703426.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. González-Briones, A.; De La Prieta, F.; Mohamad, M.; Omatu, S.; Corchado, J. Multi-agent systems applications in energy optimization problems: A state-of-the-art review. *Energies* **2018**, *11*, 1928. [CrossRef]
- 2. Smith, R.G. The contract net protocol. High level communication and control in distributed problem solver. *IEEE Trans. Comput.* **1980**, *29*, 1104–1113. [CrossRef]
- Wunan, W.; Xiaojing, G.; Chunyu, S. Contract Net Model Based on Case Based Reasoning. *Mini-Micro Syst.* 2005, 26, 1578–1581.
- 4. Chen, J.; Liao, S.; Deng, F. A collaboration algorithm for computer generated forces based on multi-agent systems. *Comput. Simul.* **2010**, *27*, 113–117.
- Sandbolm, T.W.; Lesser, V.R. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In Proceedings of the First International Conference on Multi-Agent System, San Francisco, CA, USA, 12–14 June 1995; pp. 66–73.
- 6. Zhang, H.; Shi, Z. Dynamic contract net protocol. *Comput. Eng.* **2004**, *30*, 44–57.
- Liu, L.; Gao, F. Research on the Negotiation Strategy of Mufti-agent Based on Extended Contract net. In Proceedings of the International Symposium on Parallel and Distributed Processing with Applications, Chengdu, China, 10–12 August 2009; pp. 11–18.
- 8. Sun, D.; Wu, J. Multi-Agent coordination based on contract net protocol. In Proceedings of the 2009 International Symposium on Intelligent Ubiquitous Computing and Education, Chengdu, China, 15–16 May 2009; pp. 353–357.
- 9. Carbó, J.; Molina, J.M.; Patricio, M.A. Asset Management System through the design of a Jadex Agent System. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **2016**, *5*, 1–14. [CrossRef]
- 10. González-Briones, A.; Prieto, J.; De La Prieta, F.; Herrera-Viedma, E.; Corchado, J.M. Energy optimization using a case-based reasoning strategy. *Sensors* **2018**, *18*, 865. [CrossRef] [PubMed]

- Giuseppe, C.; Marcello, F.; Giacome, L. A network flow based heuristic approach for optimizing AGV movements. J. Intell. Manuf. 2013, 24, 405–419.
- Buehler, J.; Pagnucco, M. A framework for task planning in heterogeneous multirobot systems based on robot capabilities. In Proceedings of the National Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 2527–2533.
- 13. Ming, L.; Wei, L.; Yanduo, Z. Mulit-Agent dynamic task allocation based on improved contract net protocol. *J. Shandong Univ.* **2016**, *46*, 51–56.
- 14. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [CrossRef]
- 15. Lee, W.; Kim, D. History-Based Response Threshold Model for Division of Labor in Multi-Agent Systems. *Sensors* 2017, 17, 1232. [CrossRef] [PubMed]
- 16. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern.* **1996**, *26*, 29–41. [CrossRef] [PubMed]
- 17. He, Z.; Huang, J.; Shuai, Y. Research on benefit allocation of multimodal transport based on ant colony's labor division. *J. China Railw. Soc.* **2018**, *40*, 19–26.
- Yang, Y.; Liu, X. Task assignment based on improved dynamic contract net and ant colony search strategy. In Proceedings of the International Conference on Mechatronic Sciences, Shengyang, China, 20–22 December 2013.
- 19. Xiang, W.; Lee, H.P. Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *Eng. Appl. Artif. Intell.* **2008**, *21*, 73–85. [CrossRef]
- 20. Xu, X. From cloud computing to cloud manufacturing. Robot. Comput. Integr. Manuf. 2012, 28, 75–86. [CrossRef]
- 21. Wang, L. Research on Coordination Mechanism and Key Technique of Bio-Inspired Manufacturing System; Nanjing University of Acronautics and Astronautics: Nanjing, China, 2010.
- 22. De Moraes, R.S.; De Freitas, E.P. Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services. *J. Intell. Robot. Syst.* 2017. [CrossRef]
- 23. Panescu, D.; Pascal, C. Holonic coordination obtained by joining the contract net protocol with constraint satisfaction. *Comput. Ind.* **2015**, *81*, 36–46. [CrossRef]
- 24. Hooshangi, N.; Alesheikh, A.A. Developing an Agent-Based Simulation System for Post-Earthquake Operations in Uncertainty Conditions: A Proposed Method for Collaboration among Agents. *ISPRS Int. J. Geo-Inf.* 2018, 7, 27. [CrossRef]
- Guo, C.; Xiong, W. Parallel Agent-based simulation of complex system based on repast HPC. In Proceedings of the International Symposium on Instrumentation & Measurement, Toronto, ON, Canada, 23–24 December 2013.
- 26. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An overview of recent progress in the study of distributed multi-Agent coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438. [CrossRef]
- 27. González-Briones, A.; Villarrubia, G.; De Paz, J.F.; Corchado, J.M. A multi-agent system for the classification of gender and age from images. *Comput. Vis. Image Understand.* **2018**, 172, 98–106. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).