Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4259649, 14 pages
https://doi.org/10.1155/2019/4259649

Research Article

Multiagent Task Planning Based on Distributed Resource Scheduling under Command and Control Structure

Jie Zhang,[1] **Gang Wang**,[2] **Yafei Song**,[2] **Fangzheng Zhao**,[1] **and Siyuan Wang**[1]

[1]*Graduate School, Air Force Engineering University, Xi'an 710054, China*
[2]*Department College of Air Missile Defense, Air Force Engineering University, Xi'an 710054, China*

Correspondence should be addressed to Jie Zhang; 824519928@qq.com

Received 7 May 2019; Revised 15 August 2019; Accepted 6 September 2019; Published 6 November 2019

Academic Editor: Emilio Insfran Pelozo

For task planning of the command and control structure, the existing algorithms exhibit low efficiency and poor replanning quality under abnormal conditions. Given the requirements of the current accusation architecture, a distributed command and control structure model is built in this paper based on multiagents, which exploits the superiority of multiagents in achieving complex tasks. The concept of MultiAgent-HTN is proposed based on the framework. The original hierarchical task network planning algorithm is optimized, the multiagent collaboration framework is redefined, and the coordination mechanism of local conflict is developed. With the classical resource scheduling problem as the experimental background, the proposed algorithm compared with the classical HTN algorithm is drawn. According to the experimental results, the proposed algorithm exhibits higher quality and higher efficiency than the existing algorithm and the space anomaly is significant in the course of processing. The planning is more efficient and the time is more complicated and superior in solving the same problem, and the algorithm exhibits good convergence and adaptability. In the conclusion, it is proved that the distributed command and control structure proposed in this paper exhibits high practicability in relevant fields and can solve the problem of distributed command and control structure in a multiagent scenario.

1. Introduction

Nowadays, there are many methods for optimization of distributed programming, which are mainly divided into three types: numerical analysis method, heuristic algorithm, and hybrid algorithm. At present, the common numerical analysis methods include linear programming method, nonlinear programming method, branch exchange method, branch demarcation method, enumeration method, dynamic programming method, stochastic programming method, and fuzzy programming method [1]. The principle of numerical analysis method is relatively simple, and it has a high operability [2]. However, for the complicated distributed resource planning problem, it is impossible to obtain the satisfactory optimal solution by using only the numerical analysis method. At present, the common heuristic algorithms that scholars at home and abroad use to study the optimal solution of distributed resource planning include genetic algorithm, particle swarm algorithm, tabu search algorithm, ant colony algorithm, and immune clone algorithm. The heuristic algorithm can get the optimal solution to some extent, but in many cases, it cannot get an optimal solution, for example, when the genetic algorithm is calculated, all the alternatives are listed as a set, and then each scheme in the set is analysed, and the optimal solution is obtained through many operations, but the algorithm is prone to fall into prematurity and long-time operation of CPU. The particle swarm optimization (PSO) algorithm and genetic algorithm have similar characteristics to some extent, which is to find the set of all the options and then analyze and calculate each scheme to find the optimal solution. However, compared with the genetic algorithm, it is simpler, accurate, faster, and easier to implement to converge. In order to overcome the drawbacks of heuristic algorithm, to deal with the problem of complex distributed programming constraint model and to find the best solution, a hybrid algorithm is proposed. In the face of emergencies, a quick and effective command and control structure is indispensable. Feasible command and control structure is capable of generating the highest quality decision scheme at the fastest speed in the effective time and timely handle the temporal exception in the execution process of the scheme to ensure the smooth execution of the decision scheme. The multiagent technology [3], originated in the 1980s, has been widely used and become a hot topic in the field of distributed artificial intelligence. Distributed multiagent system shows its superiority and efficiency in dealing with complex tasks. The formation of alliance is the foundation of all the activities for alliance. As a hot research object, the construction of alliance [4] is an important research content. Therefore, how to form a balanced alliance and to make it develop in an expected direction has become one of the main issues in the artificial intelligence research.

For the particularity of the emergency response, the accusation architecture is characterized by large scale, hierarchical decision-making, complex cooperative relationship, and robust spatio-temporal constraint as well as complex external situation [5].

1.1. Large Scale. There are considerable participating units involved in emergency response, with considerable tasks to be completed [6], numerous types, and plenty of resources to rely on. The relationship and element connotation of the mentioned problems should be considered when making a decision plan, thus leading to a large scale of command and control.

1.2. Hierarchical Decision-Making. Under complex requirements for solution, i.e., input complex tasks, the task is decomposed layer by layer, several simple tasks are decomposed, and the task is processed according to the concept of multiagent system. Following this principle, the decision-making process requires certain empirical knowledge and rules [7].

1.3. Complex Cooperative Relationship. In the execution of the decision scheme, there are complex dependencies and time-space constraints among agents, and there also exist resource and time constraints between agents and tasks [8].

1.4. Robust Spatio-Temporal Constraint [9]. The longer the time consumption, the more serious the loss of people's property will be. Accordingly, the constraint on the generation time of task processing scheme is extremely robust, requiring the generation of high-quality decision-making tasks in a short time.

1.5. Complex External Situation. In the execution of the decision scheme, the change trend of the external situation environment cannot be predicted. Environmental factors affect the implementation result of the decision scheme, thus easily leading to the unenforceability of the scheme. Thus, the generation of decision plans is required to be consistent with the change of external situation.

In brief, the generation of decision-making schemes for complex task planning has led to great difficulty. Conventional mathematical modelling is not suitable for solving such problems for its slow solution speed, complex cooperative relationship, and spatio-temporal state constraints. Besides, conventional mathematical modelling schemes are usually used for solving static problems, and in the case of dynamic problems, it is necessary to remodel once the problems change [10]. Accordingly, the conventional modelling method has some limitations and cannot fit the solution of such problems [11].

HTN (Hierarchical Task Network) planning is one of the most commonly used intelligent planning methods in the field of multiagent [12]. The basic process is to decompose complex tasks until they can be split into directly executable atomic-level tasks. After the decomposed tasks are achieved, task allocation is performed, and finally a decision scheme is yielded [13]. The HTN algorithm has the following advantages in solving such problems. First, the scheme is yielded quickly and can solve complex problems with robust constraints. Second, the existing empirical knowledge can be fully exploited. The HTN can efficiently use the existing method set and empirical knowledge to express various constraints to enhance the decision-making efficiency and improve the quality of the decision-making scheme. The HTN has been extensively used in robotics, feature modelling, emergency response decision-making, etc. The above discussion suggests that the HTN exhibits a good adaptability to the solution of the decision-making scheme of emergency problems [13]. Moreover, targeted optimization is conducted in this paper based on the HTN to solve the decision-making generation problem of distributed accusation architecture.

In brief, the major contributions of this paper are as follows:

(i) A novel expression form, MultiAgent-HTN, is proposed, which expands the conventional HTN methods to express the time preference of the HTN. Preference information is expressed by preference function, which can express continuous preference.

(ii) The heuristic search algorithm is used to guide the planning direction in the MultiAgent-HTN, and three definitions of horizontal consistency are proposed to estimate the quality of the MultiAgent-HTN, which acts as heuristic information to assess the quality of operators or methods. Heuristic searches select appropriate operators or methods based on their ranks.

2. Related Research

In the field of emergency problem solving network analysis, based on enhancing the efficiency of decision-making schemes, researchers analysed them from different perspectives. Jackson et al. built FMECAN (Failure Mode Effects and Criticality Analysis Network) as the program analysis model to achieve the input and output ratios of different failure models. This built network significantly enhanced the reliability of the decision-making program generation system and up-regulated the resource allocation rate [14]. Pratyszek and Karagiannis modelled the decision-making scheme for local conflicts by building failure nodes into failure trees [15]. Moreover, Xiuhua et al. established vague meanings for grouping functions. To formulate the rule of conditioned, research on distributed decision is of great significance [16]. For multiagent systems with actuator faults, Xiuhua et al. proposed a novel design method of distributed intermediate observer, capable of simultaneously estimating the state and fault of the system. This proposed method was applied in systems with strictly positive and real conditions and unsatisfied observer matching conditions [16]. Given the Cyber Physical Systems (CPSs) with nonlinear coupling characteristics in attack, Ao presented a distributed safety measurement preselector to ensure that the state of the system was accurately estimated in a preset limited time. This researcher also developed a distributed finite time safety control algorithm to ensure that the system can track a given signal in a limited time [17]. Based on the environmental and situational awareness of the motion control system and the existing research results worldwide, Jian et al. highlighted the existing challenges and future research directions [18]. In another work, Fan et al. proposed a compound distributed inclusive control algorithm and a compound distributed integral sliding mode control protocol based on nonlinear integral sliding mode control and disturbance observer, respectively [19]. Dong et al. designed the event-triggering control mechanism based on the consistency of the third-order discrete multiagent system of event-triggering control. They provided the determination conditions for the event-triggering controller to exclude zeno-like behaviour [20]. For the linear heterogeneous multiagent system, a state and output feedback collaborative controller was designed, capable of effectively reducing the network communication load and the number of controller updates and achieving the multiagent system's asymptotic tracking and interference suppression to the external system [21]. If the directed graph was strongly connected, Dongyue and Jie designed a distributed algorithm based on the disturbance observer to reach the consistency of the linear multiagent system with unknown disturbance [22]. Furthermore, Shiming et al. presented a novel event-triggered consistency control protocol with the state predictor. They demonstrated that the proposed event-triggered control strategy is capable of effectively achieving the average consistency under the combined connected topology using Lyapunov stability theory and algebraic graph [23]. The method to combine the average dwell time and the joint switching signal was employed to deal with the system instability caused by the delay of switching between the controller and the system mode. A switching control strategy based on output feedback was formulated [24]. Moreover, based on the universal consistency protocol of multiagent, Jun and Guoping developed the position time-varying consistency protocol of planar nonintegrity multiagent, which can effectively solve the general time-varying formation of the planar nonintegrity multiagent system [25]. Phuong et al. [26] proposed two methods to solve the problem following the d-relaxed priority rule. In this paper, the formulation construction exact solution proposed in the mathematical literature is improved. The metaheuristic approach based on the iterative local search framework also found approximate solutions the problem operator introduction. However, the major drawback of this method is that it cannot find the best solution and provide a solution that is stable and runs reasonably on a large instance.

In the field of command and control structure modelling, Haoran, a domestic scholar, studied and designed the command information system. He proposed a command information system architecture based on the mobile cloud mode, which significantly optimized the utilization of computing resources of the command information system and further enhanced the realization efficiency of command and control [27]. Liping et al. proposed a preference vector guided coevolutionary algorithm for Many-Objective Optimization and compared ASF-PICEA-g with g-NSGA-II and r-NSGA-II on WFG and DTLZ benchmark test functions based on 3 to 20 dimension. The experimental results showed that ASF-PICEA-g exhibited good performance and better stability, and the obtained solution set had better convergence and distribution than other algorithms [28]. Furthermore, Jin introduced service-oriented thinking into the architecture design of the command and control information system, thus enhancing the information processing capacity of the command and control information system [29]. Hongming et al. proposed the C/S and B/S hybrid architecture, and then the system was separated from the business logic through the data access layer, so the database was completely transparent to the user. By running the RunProxy to connect the client and server, the security of the server was significantly improved [30]. Based on the hierarchical hierarchy and OODA control loop, He et al. proposed a four-layer command and control structure for the unmanned combat system based on cognition, ensuring the unity of command, the flexibility of control, and the scalability of the system [31]. Furthermore, Minglei proposed a temporal HTN planner TPHTN (time preference HTN) to process time constraints with preferences [32]. The planner expressed the time preference information using STNP (simple time network with preference) and expressed the time preference information in the planning domain knowledge by extending the operators and methods. In the planning process, TPHTN propagated STNP from top to bottom and three definitions of horizontal consistency were proposed to estimate the quality of STNP. Besides, a new heuristic search was designed to select appropriate operators or methods and corresponding STNP according to the quality. Lastly, TPHTN generated a plan that meets the decision maker's preference when the planning terminates. Besides such benefits of this method, it neither expressed continuous preference nor used the heuristic search algorithm to guide the planning direction. Given the above information, it is speculated that the conventional modelling method is not suitable for the current command and control structure modelling requirements.

In brief, this study proposes a new method MultiAgent-HTN based on the architecture, which can solve these problems well. Based on the current command and control domain requirements, a multiagent-based distributed command and control model architecture is built, the HTN algorithm is optimized, and the resource scheduling problem is taken as the experimental background. The algorithm in this paper is to be compared with the original HTN algorithm, TPHTN algorithm, FHTN algorithm, etc. The results of comparative experiments reveal that the proposed algorithm exhibits certain superiority and high convergence in solving problems for distributed cooperative task planning.

3. Design of Distributed Command and Control Structure Model Based on MultiAgent

According to the modelling requirement of command and control decision-making function, the agent structure block diagram of command and control model can be obtained as shown in the following Figure 1:

The distributed command and control model [33] aims to optimize the utilization of alleged resources, deal with local contradictions in the command and control process, and reduce the complexity of the model structure while the mission requirements are satisfied. The following variables are defined based on the input task's timing constraints, resource requirements, and other constraints. d_{im} denotes the allocation variable of the command and control resource, t_{ijm} the resource transfer variable, o_{ij} the timing variable, K and N are the numbers of the resource planning of each party, respectively, assumed internal collaboration function $I(n)$ and external collaboration function $E(n)$ as the objective function, and Y is the time required for the completion of the input task. The purpose is to ensure that the objective function values of $I(n)$ and $E(n)$ are the minimum, thus making internal collaboration and external collaboration the least complex. The problem is expressed as follows:

$$\text{s.t.}\begin{cases} \sum_{i=0}^{n} t_{ijm} - d_{im} = 0; \quad i = 1,\dots,N; m = 1,\dots,K \\ \sum_{i=0}^{n} t_{jim} - d_{im} = 0; \quad i = 1,\dots,N; m = 1,\dots,K \\ \sum_{i=0}^{n} t_{i0m} = \sum_{j=0}^{n} t_{ojm} = 1 \\ s_i - s_j + t_{ijm} * \left(\dfrac{p_{ij}}{q_m} + o_{ij} * T\right) \le o_{ij} * T - \text{time}_i; \\ \qquad\qquad i, j = 1,\dots,N; m = 1,\dots,K \\ \sum_{m=1}^{K} r_{ml} * d_{im} \ge R_{il}; \quad i = 1,\dots,N; l = 1,\dots,N \\ s_i - Y \le -t_i, \quad i = 1,\dots,N \\ \qquad\qquad 0 \le Y \le T \\ \qquad\qquad s_i \ge 0 \\ \qquad\qquad t_{ijk} \in \{0,1\} \\ \qquad\qquad d_{ik} \in \{0,1\}. \end{cases} \tag{1}$$

Then, the relevant mathematical description of the command and control model is the mixed element linear programming problem, i.e., the search problem of state space. Clustering analysis is conducted on the input tasks. Different tasks are assigned to different agents in accordance with the clustering results, and then the workload of agents is minimized. The following variables are defined: W^I denotes the cooperation threshold within agents, W^K is the cooperation threshold outside agents, the minimum load is set to $\min C_w$, and M and N are the value parameters of search space and the range of their values are $[1, D]$:

$$\text{s.t.}\begin{cases} \sum_{m=1}^{D} t_{mj} = 1 \\ y_{nmi} \ge d_{ji} * x_{mj}; \quad m,n = 1,\dots,D; i = 1,\dots,N; j = 1,\dots,K \\ y_{nmi} \ge d_{ji} * x_{nj}; \quad m,n = 1,\dots,D; i = 1,\dots,N; j = 1,\dots,K \\ C_w \ge W^I * \sum_{m=1}^{D} t_{nj} + W^K * \sum_{m=1,m\neq n}^{D} \sum_{i=1}^{N} y_{nmi}; \quad n = 1,\dots,D \\ \qquad\qquad x_{nj}, x_{mj} \in \{0,1\}. \end{cases} \tag{2}$$

Complex task processing determines the external collaboration threshold of the command and control centre. To deal with the collaboration among agents' internal decision-making, the maximum external collaboration threshold between agents should be determined and the indirect collaboration threshold in command and control structure should be minimized, which is defined as $\min W_{\max}$. Define e_{ij} as direct connection variable, z_{ijk} as indirect connection variable, the internal collaboration function $I(n)$ represents the quantitative expression of the cooperative ability of each member within the unit, and external collaboration function $E(n)$ represents the quantitative expression of the cooperative ability and the cooperative efficiency among the units as follows:

$$\text{s.t.}\begin{cases} \sum_{i,j=1}^{D} e_{ij} = D - 1; \quad j = 1,\dots,K \\ \sum_{j=0}^{D} e_{j0} = 0, \sum_{j=0}^{D} e_{ji} = 1; \quad i = 1,\dots,D \\ l_j \ge l_j + 1 + \left(e_{ij} - 1\right)(D+1); \quad i, j = 1,\dots,D \\ e_{ij} + e_{ji} + \sum_{i=1}^{n} z_{ijk} \ge d_{ij}; \quad i, j = 1,\dots,D \\ e_{ik} + e_{ki} + e_{jk} + e_{kj} \ge 2 * z_{ijk}; \quad i, j, k = 1,\dots,D \\ W_{\max} \ge W^I + I(n) + W^E * \left(E(n) + \sum_{i<j} z_{ijn} c_{ij}\right); \\ \qquad\qquad n = 1,\dots,D \\ \qquad\qquad e_{ij} \in \{0,1\} \\ \qquad\qquad z_{ijk} \in \{0,1\}. \end{cases} \tag{3}$$

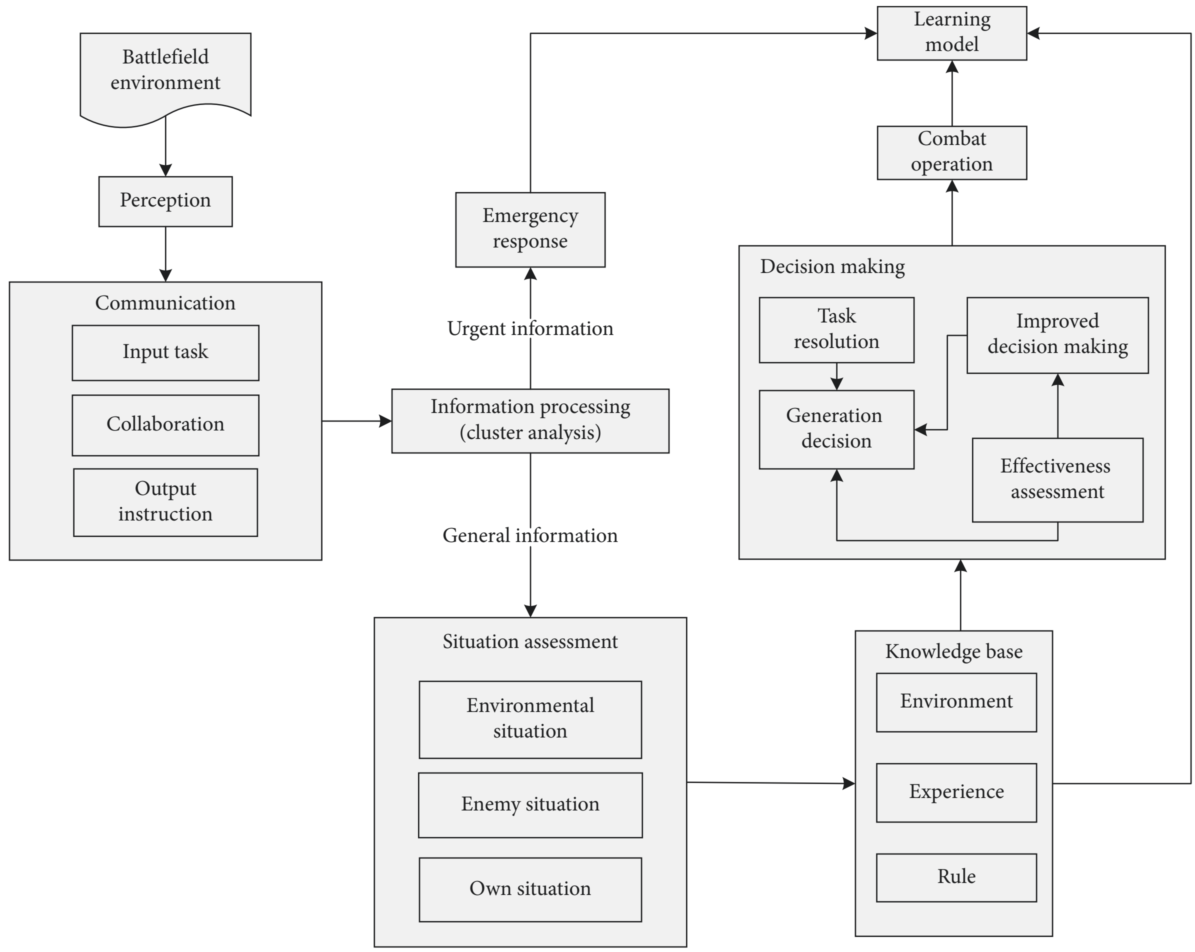

FIGURE 1: Block diagram of the command and control agent.

4. Optimization of Distributed Cooperative Task Planning Algorithms Based on HTN

In the process of command and control decision-making, the situation information of each agent cannot be fully shared, so agents should plan to maximize their own interests according to the local situation they have mastered. Given the above content, to avoid the interaction between agents, eliminate local conflicts or enhance resource utilization, the distributed cooperative task should be planned [34] to ensure the completion of the global task and the maximization of overall interests. For this end, this chapter primarily describes the cooperative relationship between distributed multiagents in the decision-making of command and control. Taking the task planning as the background, the multiagent cooperative mode and relationship network of command and control decision-making link are built.

4.1. Description of Basic HTN Algorithms. The HTN decision system has been extensively used in the planning system of AI field. The HTN represents the change of state following the concept of atomic operators, and a group of atomic operators represents a state. The basic principle of the HTN aims to generate a range of instruction actions based on a target or task and input the given problem domain or task (e.g., the initial test state of the process), and the state is the sequence of the ordered unfinished task set.

4.2. Parameter Definition.

Definition 1. HTN is expressed as a six-tuple $\langle U, I, C, A, T, N \rangle$ with extended presequence language. In the six-tuple, U denotes variable set, I constant set, C logical consistent set, A agent's task action set, T task set, and N task identifier set. These sets are independent of each other.

Definition 2. State refers to the basic state list of agent, i.e., the atomic state list. If the value is not 0, the state will be true; otherwise, it will be 0.

Definition 3. Agent's task action is Agent.$f(x_1, x_2, \ldots, x_k)$, where $f \in F, x_1, x_2, \ldots, x_k$ is the action item. The prerequisites and prediction results should be declared.

Definition 4. The complex task is $f_t(x_1, x_2, \ldots, x_k)$.

4.3. Constraints

4.3.1. Variable Constraint.

$$(v_1 = c)(v_1 = v_2), \tag{4}$$

where $v_1, v_2 \in V$ denotes a variable identifier and $c \in C$ c a constant identifier.

4.3.2. Sequential Constraint.

$$n < n', \tag{5}$$

where $n, n' \in N$. "$<$" denotes the logical sequential operator, i.e., the task content of n must be completed before the decision instruction of n' task is generated.

4.3.3. State Constraint.

$$(n, l)(l, n)(n, l, n'), \tag{6}$$

n and l are, respectively, the numbers of two agents performing tasks. $(n, l) = 1$ indicates that l is executed after n; $(l, n) = 1$ indicates that n is executed after l; and $(n, l, n') = 1$ indicates that l is executed between n and n'.

4.4. Analysis of Algorithm Flow. The HTN planning is instantiated and analysed, which is detailed as follows Algorithm 1:

The algorithm exhibits good convergence. As the iteration proceeds, the quality of the decision scheme generated by the algorithm becomes increasingly higher. The specific convergence function is presented as follows:

The aim of the experiment is to prove that the algorithm has good convergence. In Figure 2, the horizontal axis is the number of iterations and the vertical axis is the value of error loss. According to the simulation, the number of iterations is taken as the reference object and the error loss value as the criterion of convergence. The experimental results show that the error loss value decreases with the increase of iteration times, so the algorithm has good convergence.

```
HTN (s, U, C, O, M)
if (U, C) can be shown to have no solution
   Then return failure
else if U is primitive then
   if (U, C) has a solution then
      nondeterministically let π be any such solution
      return π
   else return failure
else
   choose a nonprimitive task node u ∈ U
   active ⟵ {m ∈ M|task(m) is unifiable with t_u}
   if active ≠ ∅ then
      nondeterministically choose any m ∈ active
      σ ⟵ an mgu for m and t_u that renames all
      variables of m
      (U', C') ⟵ δ(σ(U, C), σ(u), σ(m))
      (U', C') ⟵ apply-critic (U', C')
      return HTN (s, U', C', O, M)
   else return failure
```

Algorithm 1: HTN algorithm Pseudo Code.

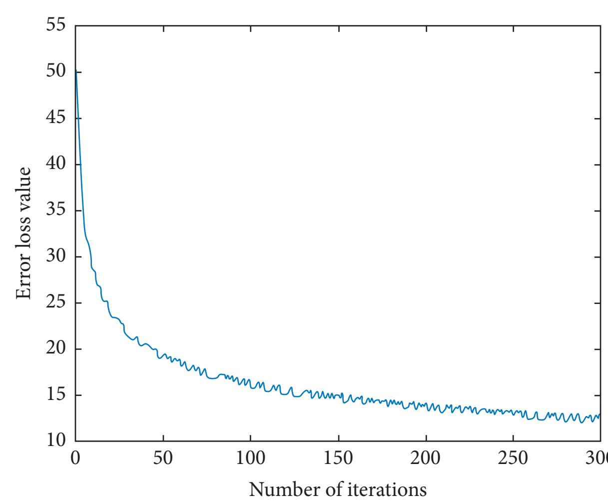

Figure 2: Function convergence curve.

4.5. Multiagent Distributed Collaboration Framework Based on HTN. Usually, in a centralized environment, the collaboration framework of MultiAgent-HTN (MultiAgent-Hierarchical Task Network Planning, MA-HTN) is to establish a decision-making centre, control the overall situation, and deduce and solve the scheme.

However, in the actual command and control process, the overall situation sharing is difficult to achieve, and the security of a single decision-making centre cannot be ensured. Once the centre is destroyed, it loses the initiative of war. Thus, distributed MA-HTN exhibits higher superiority. Each agent deduces the scheme internally and does not share the relevant information with each other in the process of scheme deduction. Moreover, each agent should cooperate with the scheme that has dependency or mutually exclusive relationship to solve local conflicts to build a task planning collaboration network and maximize the overall interests. The cooperative task planning process framework of distributed MA-HTN is illustrated in Figure 3.

The cooperative task planning of distributed MA-HTN is based on the modification of the decision-making scheme. Cooperative relationship adjustment mechanism is introduced to integrate the process of decision-making adjustment and modification. Thus, the cooperative relationship control of decision-making actions is achieved in the planning process. The algorithm flow follows the original HTN planning framework and intermediate steps (e.g., potential cooperative relationship detection and situational information sharing) are added. The potential cooperative relationship detection is employed to detect whether there is a potential dependency or contradiction between the decision instruction of the agent and the decision instruction generated by other agents. After the potential cooperative relationship is confirmed, the situation interaction module releases its own situation information and cooperative relationship

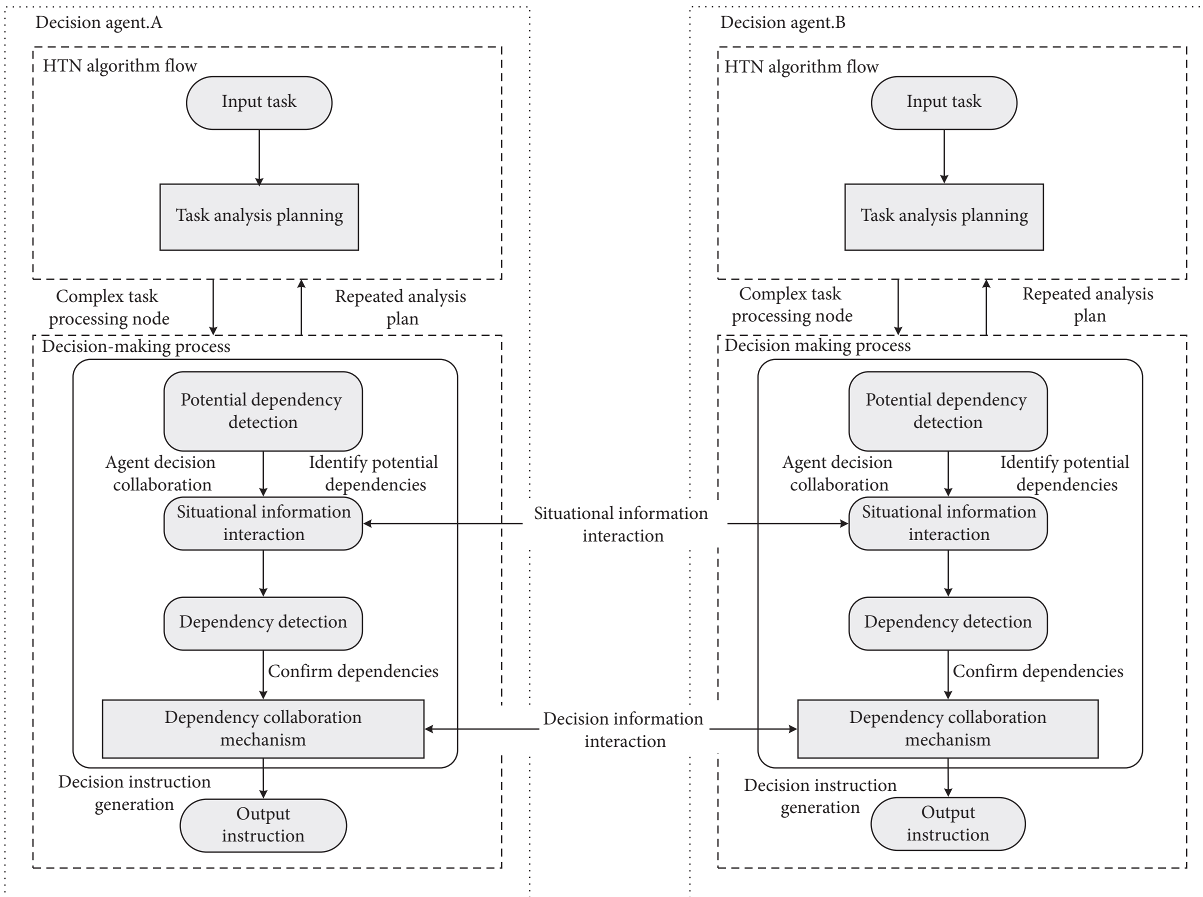

Figure 3: Distributed mission planning process for distributed MA-HTN.

information to other agents and then receives information from other agent nodes. According to the collaboration detection, the specific collaboration relationship is determined according to the results of situation information interaction, and the specific collaboration relationship is processed through the coordination relationship adjustment mechanism to maximize the overall interests of each agent. The algorithm flow not only applies to the planning of cooperative relationships between any two independent agents, but also to cooperative planning among multiple agents.

On the one hand, the distributed MA-HTN cooperative task planning process integrates the collaboration mechanism into the HTN, thus formulating and collaborating the decision-making instructions simultaneously. Accordingly, a decision-making scheme compatible with other agents can be yielded in one time. On the other hand, the confirmed cooperative relationship can be identified and processed as soon as possible. The original decision-making scheme is modified, which reduces the invalid decision-making actions of the adjusted decision-making instructions, and even may not need to make corresponding decisions at all. In brief, compared with the algorithm flow framework for conventional multiagent collaboration, this algorithm shows a higher advantage in reducing the global decision generation time and enhancing the collaboration efficiency. The specific algorithm flow is illustrated as follows Algorithm 2.

4.6. Coordination Mechanism of Local Conflicts. For the contradiction of local conflicts between various agents, a corresponding cooperation mechanism should be adopted for processing. Decker uses GPGP to deal with the coordination of resource conflicts. However, the coordination mechanism employed in this paper consists of one party that keeps the original decision-making plan unchanged and the other party that modifies the original decision-making plan. In Figure 4, there is a local conflict between Action.A1 and Action.B1 and between Agent.A and Agent.B. Action.A1 remains unchanged. Agent.B modifies Action.B1 and abandons Task.B1. Task.B is split into subtasks Task.B1 and Task.B2, and Action.B3 is selected to avoid the conflict. Since local conflicts belong to different agents, they may have different generation time. In the process of the agent's cooperative planning, potential conflict contradictions are detected earlier, and according to the collaborative mechanism, the decision-making modification of which party should be made is determined. The party of the invalid adjustment decision scheme can continue to yield the decision-making instructions, and the party that needs adjustment receives the conflicting task nodes and makes the modification for the original task, which deletes the task node in the conflict section or replaces the conflicting task content to adjust the plan. Lastly, the sent or received conflict information is updated. The coordination chart of

```
Procedure MA-HTN (s, T, D)
set s, initial state
set T, initial task set
set D, domain
initial operator ActionID = 0
set P = the empty plan
T_0 ⟵ {t belongs to T : no other task in T is constrained to precede t}
loop
  if T is empty then return P
    nondeterministically choose any t in T_0
      if t is a primitive task then
        choose a ground instance a for t with the smallest cost among the available resources
        ActionID = ActionID + 1
        opmessage(a) = (ID, Resourceused(a), st(a), et(a), cost(a))
        sendlist ⟵ opmessage(a)
        send opmessage(a) to the other planners
        if receivelist ≠ ∅ :
          ResourceID = Resourceused(a)
          conflict actions a′ ⟵ {a′ belongs to the extopmessages of receivelist :
          has Resource(a′) = ResourceusedID and has the smallest st(a′)
          and duration (st, et) is overlapped with duration (st′, et′)}, then
        if st < st′
          delete the rest extopmessage in receivelist
          Resourcestate(r) = i, Resourcest(r) = st, Resourceet(r) = et
          delete T in T_0 and add a into P
        else if st > st′
          delete opmessage(a) in sendlist
          Resourcestate(r) = i′, Resourcest(r) = st′, Resourceet(r) = et′
          Backtrack
        else delete T in T_0 and add a into P
      if t is a nonprimitive task then
        choose a method to decompose t into subtasks {t_1, t_2, ..., t_n}
        delete T in T_0 and add {t_1, t_2, ..., t_n} into T_0
      if receiving a new opmessage(a′) in receivelist of a′ and sendlist
        ResourceID = Resourceused(a′)
        conflict actions a′ belongs to the extopmessages of receivelist :
        has Resourceused(a) = ResourceID, and has the smallest st(a)
        and duration (st, et)is overlapped with duration(st′, et′)}, then
      if st′ < st
        delete the rest opmessage in sendlist
        Resourcestate(r) = i′, Resourcest(r) = st′, Resourceet(r) = et′
        Backtrack to task node at action a
      else if st′ > st
        delete extopmessage(a′) in receivelist
        Resourcestate(r) = i, Resourcest(r) = st, Resourceet(r) = et
        delete T in T_0 and add a into P
  repeat
end MA-HTN
```

Algorithm 2: MultiAgent-HTN algorithm flow in distributed environment.

agent decision-making action conflict relationship is presented as follows.

The Task.A of Agent.A can be decomposed into Action.A1 and Action.A2 of decision-making action, and the relationship between them is "and" in logical relation. Task.B of Agent.B can be decomposed into subtasks Task.B1 and Task.B2, and the relationship between them is "or" in logical relationship. The subtask Task.B1 can be further decomposed into decision-making actions Action.B1 and Action.B2. There is a conflict between Action.A2 and Action.B1, i.e., the execution of A2 will lead to unsmooth execution of B1.

Based on this conflict coordination mechanism, a coordination mechanism is designed for reusable resource conflicts and consumptive resource conflicts.

The reusable resource conflict coordination mechanism can be used by multiple decision-making agents simultaneously to generate repeated resource conflicts. For these resource conflicts, the tasks exhibiting higher priority are reserved and the tasks with lower priorities are coordinated

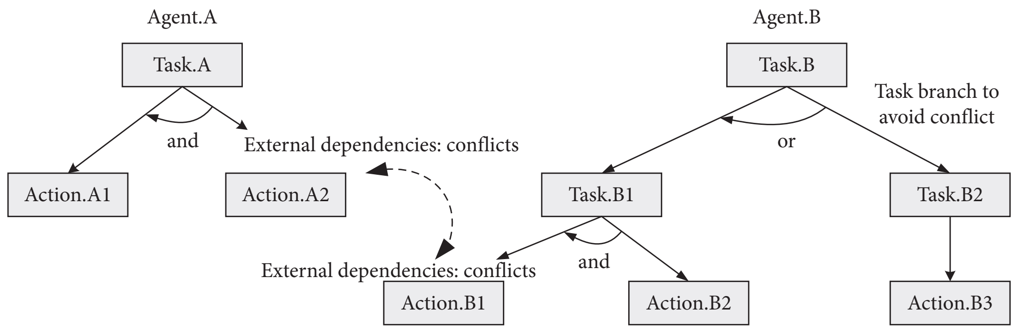

Figure 4: Agent conflict coordination mechanism.

based on determining the priority of the input tasks. The task priority criteria are the following three points. First, tasks that use resources earlier prioritize to ensure that the total global task time is the minimum. Second, tasks with higher overall interests have higher priority. Lastly, in the case of similar overall interests, the lower resource consumption exhibits higher priority.

The resource consuming conflict coordination mechanism, i.e., part of the consumed resources can be used by multiple agents simultaneously, whereas the number of resources is insufficient. In this case, the resource consuming conflict coordination mechanism is employed to process conflicts. Based on judging the priority of the input task, the agent with high priority is guaranteed to complete the input task in priority to avoid multiple conflicts and lead to great loss of overall interests.

5. Experimental Analysis

5.1. Problem Description. Resource scheduling in distributed command and control structure has always been one of the key issues in the field. Resource scheduling is a typical decision-making problem in the process of command and control. In the command process, many resources should be transported from resource storage to material demand in limited time. This problem is a complex decision-making problem. It involves numerous resource scheduling sites and resources. Both spatio-temporal constraints and numerical logic reasoning should be considered in the solution process. Besides, on the premise of meeting the requirements, high-quality and fast solutions should be generated.

Suppose P_1, P_2, and P_3 are resource demand points. Q_1, Q_2, and Q_3 are resource storage points, which store Q q_1, q_2, q_3, q_4, and q_5 resource types, respectively. Locations O_1, O_2, O_3, O_4, and O_5 are the transfer centres for resource scheduling. Assume that seven transport teams can be used to solve the problem and the initial location of all transport teams is in the resource storage point and the transfer centre of resource scheduling. The transport capacity of each transport team should not exceed the transport limit of the team. The maximum speed and minimum speed of the team are defined as $S_{-}\max$ and $S_{-}\min$, respectively. The specific information of the transport team is listed in Table 1.

It is assumed that the temporal constraints for each transport team per transport are as follows:

$$\begin{aligned}
&T_{\min} \leq End - Start \leq T_{\max};\\
&\left(\begin{array}{c}
(T_{\min}, 0.5),\\[2ex]
\left(\dfrac{(T_{\min} + T_{\max})}{2}, 1\right),\\[2ex]
(0.2 * T_{\min} + 0.8 * T_{\max}, 1),\\[2ex]
(T_{\max}, 0.5)
\end{array}\right)\\
&T_{\min} < 0.5,\\
&T_{\max} > 0.5\\
&\frac{(T_{\min} + T_{\max})}{2} < 1;\\
&(0.2 * T_{\min} + 0.8 * T_{\max}) > 1,
\end{aligned} \tag{7}$$

where $T_{\min}$ denotes the time required for the transport team to transport resources at the minimum speed and $T_{\max}$ is the time required for the transport team to transport resources at the maximum speed. Decision-making principle requires the resources to be transported from the resource storage place as soon as possible. At the same time, the required resources can reach the resource demand point as possible without considering the accidents in the transport process.

To test the performance of Multiagents HTN, eight resource scheduling problems were designed. This is the initial state of the problem and the temporal constraints are more different. The specific information is listed in Table 2.

5.2. Analysis of Experimental Results. In this paper, Multi-Agent-HTN, TPHTN [35], and F-HTN algorithms are compared and analysed. The TPHTN algorithm exhibiting better temporal preference has been extensively applied practically. F-HTN is a dynamic task hierarchical planner with good dynamic replanning capability and is a fast HTN algorithm, which is optimized on the original HTN algorithm model to enhance the iteration rate of the algorithm.

Table 1: Basic information of the resource dispatching transport team.

Number	Transport team	Transportation resource category	Transport limit	S_max	S_min
1	$Team_1$	O_1, O_2	500	50	75
2	$Team_2$	O_1, O_2	400	45	65
3	$Team_3$	O_3, O_4	500	45	70
4	$Team_4$	O_3, O_4	350	35	50
5	$Team_5$	O_4	400	30	60
6	$Team_6$	O_2, O_4	300	30	55
7	$Team_7$	O_1, O_2, O_3, O_4	300	45	65

Table 2: Detailed plan of resource scheduling.

Number	Initial state	Temporal constraint
Problem_1	$(resource\_needA_1p_1300)$ $(resource\_needA_3p_2200)$	$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$
Problem_2	$(resource\_needA_3p_2300)$ $(resource\_needA_3p_2500)$	$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$
Problem_3	$(resource\_needA_3p_2200)$ $(resource\_needA_3p_2300)$ $(resource\_needA_3p_2500)$	$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$
Problem_4	$(resource\_needA_3p_2300)$ $(resource\_needA_3p_2400)$ $(resource\_needA_3p_2700)$	$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(2,2),(3,2),(4,3))$
Problem_5	$(resource\_needA_3p_2300)$ $(resource\_needA_3p_2700)$ $(resource\_needA_3p_2100)$	$end.1 \le 4((0,0),(1,1),(3,1),(5,1))$ $end.1 \le 4((0,0),(4,3),(5,2),(7,1))$ $end.1 \le 4((0,0),(3,2),(4,1),(5,4))$ $end.1 \le 4((0,0),(1,4),(5,2),(6,3))$ $end.1 \le 4((0,0),(2,1),(4,2),(6,3))$
Problem_6	$(resource\_needA_3p_2300)$ $(resource\_needA_3p_2500)$ $(resource\_needA_3p_2500)$ $(resource\_needA_3p_2700)$	$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ $end.1 \le 4((0,0),(1,3),(3,1),(5,7))$ $end.1 \le 4((0,0),(3,7),(7,8),(2,5))$ $end.1 \le 4((0,0),(3,2),(5,3),(6,5))$ $end.1 \le 4((0,0),(4,4),(5,3),(5,1))$
Problem_7	$(resource\_needA_3p_2100)$ $(resource\_needA_3p_2300)$ $(resource\_needA_3p_2700)$ $(resource\_needA_3p_2900)$	$end.1 \le 4((0,0),(4,3),(6,3),(4,7))$ $end.1 \le 4((0,0),(6,4),(4,6),(7,3))$ $end.1 \le 4((0,0),(4,5),(3,4),(4,1))$ $end.1 \le 4((0,0),(5,3),(3,1),(6,4))$ $end.1 \le 4((0,0),(6,2),(4,7),(4,1))$ $end.1 \le 4((0,0),(7,1),(3,6),(4,4))$
Problem_8	$(resource\_needA_3p_2200)$	$end.1 \le 4((0,0),(3,1),(2,2),(5,5))$ $end.1 \le 4((0,0),(3,3),(3,2),(4,1))$

The advantage is the output rate of the final scheme is accelerated, but the defect is that the generated scheme may fall into the local optimal, resulting in the loss of global revenue. Thus, this paper takes the above algorithm as the comparison method.

5.2.1. Quality Comparison of Generating Schemes. This paper compares the quality of the generating schemes of MultiAgent-HTN and TPHTN algorithm and solves the above algorithms, respectively. The quality comparison diagram of the generating schemes is given as follows.

The preference degree is one of the important evaluation indexes for the quality of the generated scheme. The higher the preference degree is, the higher the quality of the final generated scheme will be. It can be found from Figure 5 that both algorithms can solve the problem with relatively high quality, but MultiAgent-HTN has a better generation scheme than TPHTN. For some complex problems, e.g., Problem_6 and Problem_7, the quality of TPHTN generated solutions is obviously not as good as that of MultiAgent-HTN. The reason is that TPHTN adopts a rule-based planning method to quickly generate the solution. However, when the problem is complicated, the exponential growth of the rules that should be dealt with leads to the insufficient quality of the generating scheme. MultiAgent-HTN adopts a heuristic search strategy based on depth-first, so it is more prominent in solving complex problems.

5.2.2. Efficiency Comparison of Generation Schemes. The generation solution requires not only a high-quality solution

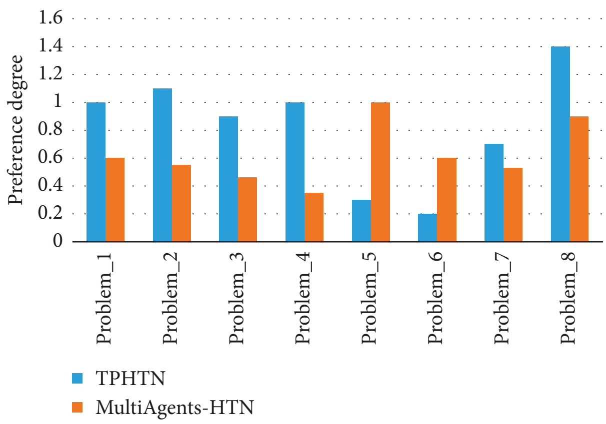

Figure 5: Initial state collection of resource scheduling planning.

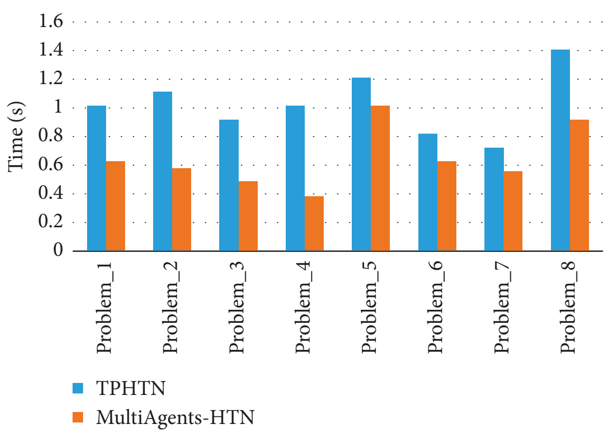

Figure 6: Comparison of the generation efficiency of MultiAgent-HTN and TPHTN schemes.

but also a sufficient efficiency. Excessive decision-making time often significantly affects the outcome of the decision. The efficiency of the scheme generation of MultiAgent-HTN and TPHTN is compared. The efficiency comparison of generating schemes is presented below.

The comparison of the results of the above figure suggests that in general, the generation time of MultiAgent-HTN and TPHTN is approximately the same, but by combining Figures 5 and 6, it can be concluded that the generation scheme of MultiAgent-HTN has relatively high quality. In solving complex problems, MultiAgent-HTN requires more time to generate decision-making solutions. Because of the small time difference, it is almost negligible. In conclusion, MultiAgent-HTN exhibits higher scheme generation efficiency than TPHTN.

5.2.3. Comparisons of Replanning Numbers. After a decision plan is generated, when a scenario is faced during the execution of the scenario, there are usually two solutions for that case. First, the decision plan should be revised. When the decision plan encounters an abnormality in the execution, the decision plan is adjusted following the determined decision modification rule. When the adjustment still encounters a tense anomaly, the replanning of the decision-

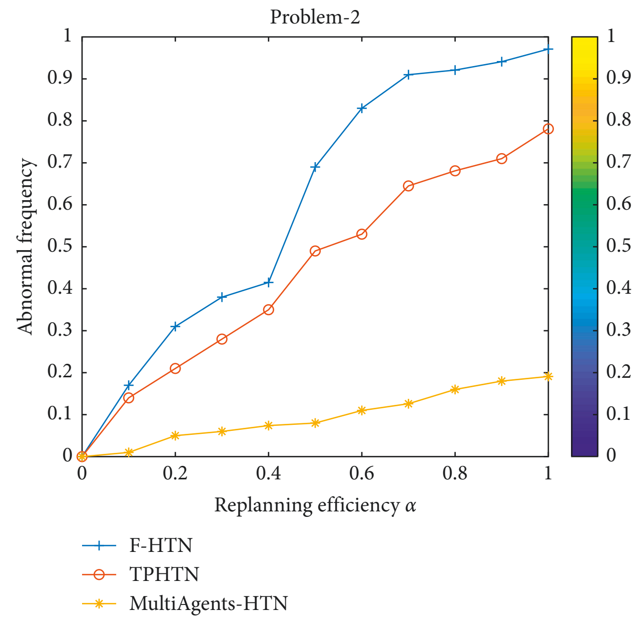

Figure 7: Problem_2 comparison of resource replanning times.

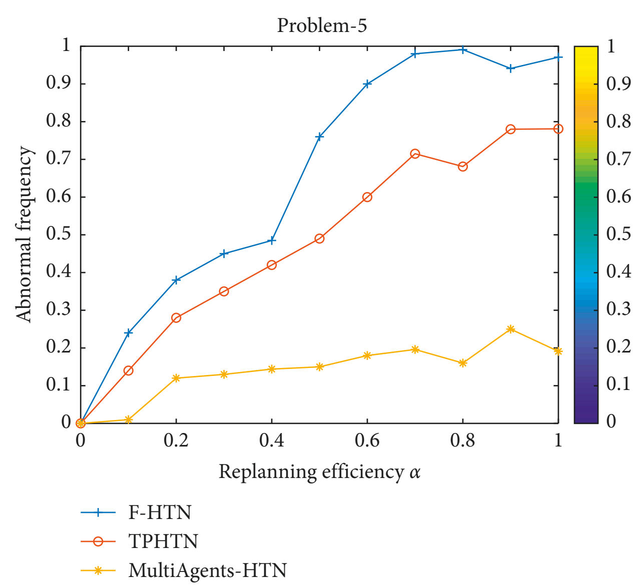

Figure 8: Problem_5 resource replanning comparison.

making scheme is conducted. Second, the temporal exception handling is performed according to the temporal constraints. The replanning rate is obtained by dividing the average number of replanning times by the number of occurrences of the temporal anomaly. 4 out of the 8 questions above are selected as the subject. Among them, Problem_8 is simple. Problem_2 and Problem_5 are more complex issues. Problem_7 is complex. The specific experimental results are presented as follows.

The results of Figures 7–10 suggest that with the rise in abnormal frequency in the execution of decision-making schemes. The replanning rate basically shows an increasing trend. But there are also unexpected situations, which occur because the quality of the decision-making plan is

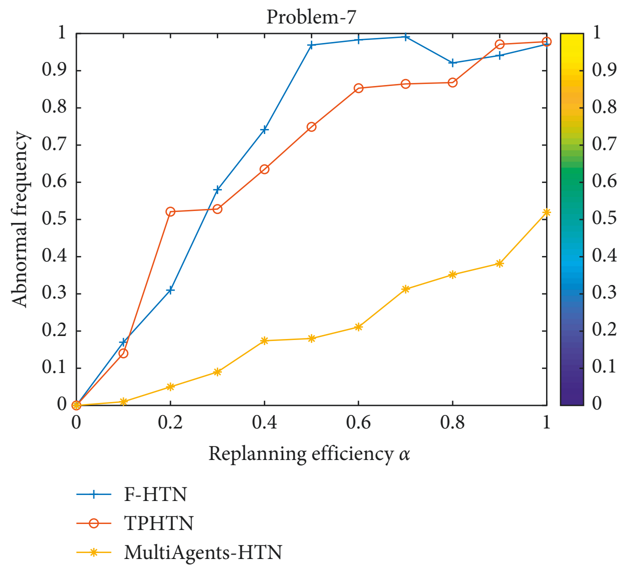

Figure 9: Problem_7 resource replanning comparison.

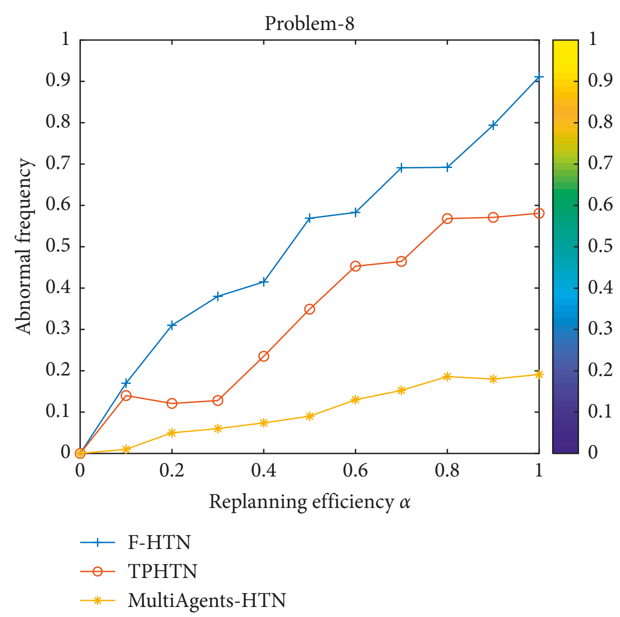

Figure 10: Problem_8 comparison of resource replanning times.

contingent. In the same scenario, the rescheduling rate of MultiAgent-HTN is the lowest and that of F-HTN is the highest for the rescheduling of four problems. When the problem is too complicated or the solution space is too large, F-HTN and TPHTN may not be able to solve and will fall into deadlock. MultiAgent-HTN uses repair local decision-making scheme when the decision-making scheme needs to be replanned. Accordingly, the efficiency of reprogramming is much higher than those of the other two algorithms.

5.2.4. Comparisons of Algorithmic Complexity. In the same scenario, the framework of this algorithm is compared with F-HTN and TPHTN algorithms. The results are listed in Tables 3 and 4 below.

Table 3: Algorithmic iteration efficiency comparison.

Algorithm	Limited time (s)	Number of iterations	Average objective function value
Algorithm in this paper	1	50	7.215
	5	300	6.813
	10	1000	6.126
HTN	1	50	11.215
	5	300	10.813
	10	1000	10.126
F-HTN	1	50	9.165
	5	300	9.133
	10	1000	8.562
TPHTN	1	50	9.935
	5	300	9.423
	10	1000	9.024

Table 4: Algorithmic iteration efficiency comparison.

Algorithm	Time complexity	Space complexity	Sample data volume
Algorithm in this paper	$O(1)$	$O(m)$	5000
	$O(1)$	$O(m)$	50000
	$O(1)$	$O(m)$	1000000
HTN	$O(n^2)$	$O(m^n)$	5000
	$O(1)$	$O(m)$	50000
	$O(1)$	$O(m)$	1000000
F-HTN	$O(n^2)$	$O(m^n)$	5000
	$O(\log^{2n})$	$O(m)$	50000
	$O(n^2)$	$O(m)$	1000000
TPHTN	$O(\log^{2n})$	$O(m^n)$	5000
	$O(n^2)$	$O(m^n)$	50000
	$O(1)$	$O(m)$	1000000

The experimental data suggest that the algorithm in this paper analyses the function value and its time-space complexity under the same number of iterations. It is found that the algorithm is capable of achieving lower objective function value, and it exhibits better convergence and adaptability.

6. Conclusions

With the frequent occurrence of local disputes and emergencies, a growing number of scholars worldwide pay attention to the generation of decision-making schemes for emergency response to real events. The basic aim of the decision-making scheme is to fulfill the tactical objectives of the complete scheme that is to minimize the casualties and property losses as much as possible. Moreover, the generation time of the scheme should be shortened to a great extent and the local disputes should be coordinated quickly and efficiently after the implementation of the scheme. As one of the efficient intelligent planning algorithms, HTN exhibits good adaptability and high efficiency in solving such problems, which is the basis of the algorithm framework proposed in this paper.

In this paper, a distributed command and control structure model based on multiagent is constructed to meet the demand of current command and control structure. The problem of large amount of data and complex situation environment of emergency events is solved using the high collaboration efficiency of multiagent and the superiority of solving complex problems. Based on the HTN algorithm, an algorithm framework of MultiAgent-HTN is proposed and the algorithm is simulated and compared comprehensively. The results of experiments reveal that the algorithm exhibits high practicability and research value in solving complex problems with large amount of data and complex situation. Besides, it also exhibits high adaptability and superior self-cooperation ability. Furthermore, the future research direction of this paper involves optimizing the exact method by adding effective inequalities to solve more instances in the branching and cutting framework. Community based on larger metaheuristic large neighbourhood search (LNS) may be considered a good candidate approximation method. Moreover, the capacity version of the problem is also a noteworthy topic for future study.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The National Natural Science Foundation of China (Grant nos. 61703426, 61876189, 61806219, 61273275, and 61503407).

References

[1] Y. Song, X. Wang, W. Quan, and W. Huang, "A new approach to construct similarity measure for intuitionistic fuzzy sets," *Soft Computing*, vol. 23, no. 6, pp. 1985–1998, 2019.

[2] Y. Song, X. Wang, J. Zhu, and L. Lei, "Sensor Dynamic Reliability Evaluation Based on Evidence Theory and Intuitionistic Fuzzy sets," *Applied Intelligence*, vol. 48, no. 11, 2018.

[3] F. Muyun, Z. Chanchan, H. Xin, L. Xiao, and Z. Jianping, "H couple-group consensus of stochastic multi-agent systems with fixed and Markovian switching communication topologies," *Chinese Physics B*, vol. 28, no. 1, pp. 313–322, 2019.

[4] W. Zhengxin, F. Jingbo, J. Guoping, C. Jinde, X. Min, and A. Alsaedi, "Consensus in nonlinear multi-agent systems with nonidentical nodes and sampled-data control," *Science China (Information Sciences)*, vol. 61, no. 12, pp. 141–154, 2018.

[5] Q. Chao, W. Dan, H. Muñoz-Avila, P. Zhao, and H. Wang, "Hierarchical task network planning with resources and temporal constraints," *Knowledge-Based Systems*, vol. 133, pp. 17–32, 2017.

[6] H. Martin, V. Daniel, and W. Lisa, "Dynamics in matching and coalition formation games with structural constraints," *Artificial Intelligence*, vol. 262, pp. 222–247, 2018.

[7] A. Torreño, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative multi-agent planning," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–32, 2017.

[8] M. Otte, M. Kuhlman, and D. Sofge, "Competitive target search with multi-agent teams: symmetric and asymmetric communication constraints," *Autonomous Robots*, vol. 42, no. 6, pp. 1207–1230, 2018.

[9] X. Mi, Y. Zou, S. Li, and H. R. Karimi, "Self-triggered DMPC design for cooperative multi-agent systems," *IEEE Transactions on Industrial Electronics*, vol. 99, p. 1, 2019.

[10] Q. Liang and Z. She, "Constraint consensus of heterogeneous multi-agent systems," *International Journal of Modern Physics C*, vol. 29, no. 5, Article ID 1840005, 2018.

[11] K. Tazi, F. M. Abbou, and F. Abdi, "Multi-agent system for microgrids: design, optimization and performance," *Artificial Intelligence Review*, no. 28, pp. 1–60, 2019.

[12] O. Yazdanbakhsh and D. Scott, "FANCFIS: fast adaptive neuro-complex fuzzy inference system," *International Journal of Approximate Reasoning*, vol. 105, 2019.

[13] M. Iqbal, J. Leth, and T. D. Ngo, "Generalized hierarchical block circulant structure of multi-agent systems," *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1275–1286, 2017.

[14] B. A. Jackson, K. Sullivan Faith, and H. H. Willis, "Are we prepared? Using reliability analysis to evaluate emergency response systems," *Journal of Contingencies and Crisis Management*, vol. 19, no. 3, pp. 147–157, 2011.

[15] E. Piatyszek and G. M. Karagiannis, "A model-based approach for a systematic risk analysis of local flood emergency operation plans: a first step toward a decision support system," *Natural Hazards*, vol. 61, no. 3, pp. 1443–1462, 2012.

[16] L. Xiuhua, H. Jian, and W. Xinjiang, "Distributed fault estimation for multi-agent systems based on intermediate observer," *Acta Automatica Sinica*, pp. 1–11, 2019.

[17] W. Ao, S. Yongduan, and W. Changyun, "Distributed security state estimation and control for attacked information physics systems: a finite time method," *Acta Automatica Sinica*, vol. 45, no. 1, pp. 174–184, 2019.

[18] S. Jian, D. Fang, and C. Jie, "Development status and trend of land motion body control system," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 1985–1999, 2018.

[19] L. Fan, Y. Hongyong, Y. Yize, L. Li, and Y. Liu, "Finite-time integral sliding mode control for multi-agent systems with mismatched disturbances," *Acta Automatica Sinica*, pp. 1–10, 2019.

[20] T. Dong, L. Xiao-Li, and Da-D. Zhao, "Consistency analysis of third-order discrete multi-agent system based on event trigger," *Acta Automatica Sinica*, pp. 1–7, 2019.

[21] L. Juan, Z. Hao, and W. Zhuping, "Cooperative output regulation of heterogeneous multi-agents based on self-triggering," *Acta Automatica Sinica*, pp. 1–10, 2019.

[22] Y. Dongyue and M. Jie, "Unified consistency of linear multi-agent systems based on disturbance observer in directed graphs," *Acta Automatica Sinica*, vol. 44, no. 6, pp. 1037–1044, 2018.

[23] C. Shiming, G. Junjie, G. Yanli, P. Huiqin, and Q. Yun, "Fast consistency algorithms for multi-agents based on 524 event triggering in composite connected topology," *Acta Automatica Sinica*, vol. 44, no. 12, pp. 2269–2277, 2018.

[24] W. Miaomiao, Z. Hao, Y. Huaicheng, and Ch. Shiming, "Cooperative output regulation for asynchronously switched multi-agent systems," *Acta Automatica Sinica*, vol. 43, no. 5, pp. 735–742, 2017.

[25] Z. Jun and L. Guoping, "Time-varying position consistency control for planar multi-agent with nonholonomic constraints," *Acta Automatica Sinica*, vol. 43, no. 7, pp. 1169–1177, 2017.

[26] H. N. Phuong, H. T. N. Nhat, M. Hoàng Hà, L. André, and M. Trépanier, "Solving the clustered traveling salesman problem with d-relaxed priority rule," 2018, https://arxiv.org/abs/1810.03981.

[27] J. Haoran, *Research on the structure of command information system based on mobile cloud model*, Ph.D. dissertation, Dept. National University of Defense Technology, Changsha, China, 2016.

[28] W. Liping, C. Hong, Du. Jiejie, Y. Wei, and Q. Feiyue, "Preference vector guided co-evolutionary algorithm for Many-Objective Optimization," *Journal of Software*, pp. 1–19, 2019.

[29] W. Jin, D. Wenxue, and G. Hong, "Design of a service-oriented command and control information system," *Shanghai Aerospace Electronic Technology Research Institute*, vol. 3, 2018.

[30] Z. Hongming, L. Yongzhi, and W. Yang, "Research and implementation of command and control system based on hybrid architecture," vol. 4, 2008.

[31] H. He, L. Zhi-Fei, W.-P. Wang, and Yi-Fan Zhu, L. Xiao-Bo, Multi-layer command and control structure of unmanned combat system based on cognition," in *Proceedings of the 5th China Command and Control Conference*, Beijing, China, Apirl 2007.

[32] L. Minglei, W. Hongwei, Q. Chao, and Z. Chao, "Handling temporal constraints with preferences in HTN planning for emergency decision-making," *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 4, pp. 1881–1891, 2016.

[33] Q. Jin, R. Lin, H. Zou et al., "A distributed fog computing architecture supporting multiple migrating mode," in *Proceedings of 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Shanghai, China, June 2018.

[34] Z. Feng, G. Wen, and G. Hu, "Distributed secure coordinated control for multiagent systems under strategic attacks," *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1273–1284, 2017.

[35] Z. Wang, H.-W. Wang, C. Qi, and J. Wang, "A resource enhanced HTN planning approach for emergency decision-making," *Applied Intelligence*, vol. 38, no. 2, pp. 226–238, 2013.