*Type of the Paper (Article, Review, Communication, etc.)*

# Multi-Agent Task Planning Based on Distributed Resource Scheduling under Command and Control Architecture

**Jie ZHANG[1*], Gang WANG[2], Song YaFei [2]**

[1] Graduate School, Air Force Engineering University, Xi'an 710054, China.

[2] College of Air Missile Defense, Air Force Engineering University, Xi'an 710054, China.

* Correspondence: afeu_zhangjie@163.com; Tel.:+86-1333-538-4381

Received: date; Accepted: date; Published: date

**Abstract:** For the task planning problem of the command and control architecture, the existing algorithms have problems such as low efficiency and poor re-planning quality under abnormal conditions. Based on the requirements of the current accusation architecture, this paper constructs a distributed command and control architecture model based on multi-agents, which makes use of the superiority of multi-agents in dealing with complex tasks. The concept of MultiAgents-HTN is proposed under the framework. The original hierarchical task network planning algorithm is optimized, the multi-agent collaboration framework is redefined, and the coordination mechanism of local conflict is designed. Taking the classical resource scheduling problem as the experimental background, the comparison between the proposed algorithm and the classical HTN algorithm is carried out. The experimental results show that the proposed algorithm has higher quality and higher efficiency than the existing algorithm, and the space anomaly is heavy during processing. The planning is more efficient, and the time is more complicated and superior in dealing with the same problem, with good convergence and adaptability. The conclusion proves that the distributed command and control architecture proposed in this paper has high practicability in related fields and can solve the problem of distributed command and control architecture in multi-agent environment.

**Keywords:** Multi-agent, HTN, Distributed Architecture, Command and Control Model, Algorithm Performance Comparison.

---

## 1. Introduction

In the field of artificial intelligence, traditional research objects mainly focus on the intelligence problem of individual agents. However, in practical engineering applications, most of agents have social interactions with other agents like information interaction and cooperation. Therefore, the concept of Multi-agent System (MAS) has been proposed and has become one of the two research branches in distributed artificial intelligence [1]. The research of MAS is mainly divided into the following parts: Agent design, interactive cooperation and utility allocation between agents, and alliance training algorithm design.

In recent years, with the international situation changes, peace and development are the current themes of the times, but disputes still exist in the Middle East and other parts of the region. In order to cope with possible emergencies, it is reasonable to organize relevant departments, mobilize relevant reasonable resources and formulate and implement decision-making plans within the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



shortest time after the occurrence of emergencies, so as to minimize people's property losses and maintain the image of a great country. A quick and effective command and control structure is indispensable in the face of emergencies. Efficient and feasible accusation architecture model can generate the highest quality decision scheme with the fastest speed in the effective time, and timely handle the temporal exception in the execution process of the scheme to ensure the smooth implementation of the decision scheme.

The particularity of the emergency response determines that the accusation architecture is characterized by large scale, hierarchical decision-making, complex cooperative relationship, strong spatio-temporal constraint and complex external situation [4].

1) Large scale. There are a large number of participating units involved in emergency response, with a large number of tasks to be completed [5], a large number of types, and a large number of resources to rely on. The relationship and element connotation of the above issues need to be considered when making a decision plan, resulting in a huge scale of command and control.

2) Hierarchical decision-making. In the face of complex requirements for solution, namely input complex tasks. The task is decomposed layer by layer, several simple tasks are decomposed, and the task is processed cooperatively with the concept of multi-agent system. Based on this principle, the decision-making process requires certain empirical knowledge and rules [6].

3) Complex cooperative relationship. During the execution of the decision scheme, there are complex dependencies and time-space constraints among agents, and resource and time constraints also exist between agents and tasks [7].

4) Strong spatio-temporal constraint [8]. The particularity of this field is that the longer the time consumption, the more serious the loss of people's property. Therefore, the constraint on the generation time of task processing scheme is extremely strong, which requires the generation of high-quality decision-making tasks in a short time.

5) Complex external situation. In the process of implementing the decision scheme, the change trend of the external situation environment cannot be predicted. Environmental factors have a certain impact on the implementation result of the decision scheme, which may easily lead to the unenforceability of the scheme. Therefore, the generation of decision plans is required to conform to the change of external situation.

To sum up, the generation of decision-making schemes for complex task planning has brought great difficulty. Traditional mathematical modeling is not suitable for solving such problems due to its slow solution speed, complex cooperative relationship and spatio-temporal state constraints. In addition, traditional mathematical modeling schemes are usually used for solving static problems, and in the face of dynamic problems, it is necessary to re-model once the problems change [9]. Therefore, the traditional modeling method has some limitations and cannot satisfy the solution of such problems [10].

HTN (Hierarchical Task Network Planning) is one of the most widely used intelligent planning methods in the field of multi-agent. The basic process is to decompose complex tasks until they can be decomposed into directly executable atomic-level tasks. After the decomposed tasks are completed, task allocation is carried out and finally a decision scheme is generated [11]. HTN algorithm has the following advantages in solving such problems. First, the scheme is generated quickly and can deal with complex problems with strong constraints. Second, the existing empirical

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



knowledge can be fully utilized. HTN can efficiently use the existing method set and empirical knowledge to express various constraints, so as to improve the decision-making efficiency and the quality of the decision-making scheme. HTN is currently widely used in robotics, feature modeling, emergency response decision-making and other fields. The above discussion shows that HTN has a good adaptability to the solution of the decision-making scheme of emergency problems [12]. Moreover, in this paper, we conduct targeted optimization based on HTN to solve the decision-making generation problem of distributed accusation architecture.

In summary, the main contributions of this paper are as follows:

• We propose a new expression form, MultiAgents-HTN, which expands the traditional HTN methods to express the time preference of HTN. Preference information is expressed by preference function, which can express continuous preference.

• We use heuristic search algorithm to guide the planning direction in MultiAgents-HTN, and propose three definitions of horizontal consistency to estimate the quality of MultiAgents-HTN, which is used as heuristic information to evaluate the quality of operators or methods. Heuristic searches select appropriate operators or methods based on their rank.

## 2. Related Research

In the field of emergency problem solving network analysis, based on improving the efficiency of decision-making schemes, researchers analyzed them from different perspectives. B. a. Jackson et al. constructed FMECAN (Failure Mode Effects and Critically Analysis Network) as the program analysis model, to achieve the input and output ratios of different failure models, which greatly improved the reliability of the decision-making program generation system and the resource allocation rate [13]. E. pratyszek et al. modeled decision-making scheme for local conflicts by building failure nodes into failure trees [14]. Furthermore, Gracaliz Pereira Dimuro et al. constructed vague meanings for grouping functions. To build the rule of conditioned, research on distributed decision was of great significance [15]. For multi-agent systems with actuator faults, Liu Xiu-hua et al. put forward a new design method of distributed intermediate observer, which can simultaneously estimate the state and fault of the system and is applied to systems with strictly positive and real conditions and unsatisfied observer matching conditions [15]. In view of the Cyber Physical Systems (CPS) with nonlinear coupling characteristics in attack, Ao Wei proposed a distributed safety measurement preselector to ensure that the state of the system was accurately estimated within a preset limited time, and established a distributed finite time safety control algorithm to ensure that the system can track a given signal in a limited time [16]. Based on the environment and situational awareness of the motion control system and summarizing the existing research results at home and abroad, Sun Jian pointed out the existing challenges and future research directions [17]. In another work, Liu Fan et al. proposed a compound distributed inclusive control algorithm based on nonlinear integral sliding mode control and a compound distributed integral sliding mode control protocol based on disturbance observer [18]. Dong Tao et al. designed the event-triggering control mechanism based on the consistency problem of the third-order discrete multi-agent system of event-triggering control, and provided the determination conditions for the event-triggering controller to exclude zeno-like behavior [19]. For the linear heterogeneous multi-agent system, a state and output feedback collaborative controller was designed, which can effectively reduce the network communication load and the number of controller updates, and realize the multiagent system's asymptotic tracking and interference suppression to the external

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



system [20]. Under the condition that the directed graph was strongly connected, Yang Dongyue and Mei Jie designed a distributed algorithm based on disturbance observer to achieve the consistency of the linear multiagent system with unknown disturbance [21]. Furthermore, Chen Shiming et al. proposed a new event-triggered consistency control protocol with state predictor, and demonstrated that the proposed event-triggered control strategy can effectively achieve the average consistency under the combined connected topology by using Lyapunov stability theory and algebraic graph [22]. The method of combining the average dwell time and the joint switching signal was used 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 proposed [23]. Moreover, Zhao Jun and Liu Guoping put forward the position time-varying consistency protocol of planar non-integrity multiagent based on the universal consistency protocol of multi-agent, which can effectively solve the general time-varying formation problem of planar non-integrity multiagent system [24]. H. N. Phuong et al. [25] proposed two methods to solve the problem using the d-relaxed priority rule. In this paper, the authors improved the formulation construction exact solution proposed in the mathematical literature. The meta-heuristic approach based on the iterative local search framework also found approximate solutions the problem operator introduction. However, the major drawback of this method was that it failed to find the best solution and provide a solution that was stable and runs reasonably well on a large instance.

In the field of command and control architecture modeling, Ji Haoran, a domestic scholar, studied and designed the command information system and proposed a command information system architecture based on the mobile cloud mode, which effectively improved the utilization of computing resources of the command information system and further improved the realization efficiency of command and control [26]. Zang Tianxiang completed the design of SOAP security model based on the architecture of command and control system, combined with the protocol of SOA architecture and WS-security related security specifications, and realized multiple security mechanisms for SOAP messages [27]. Furthermore, Wang Jin applied the service-oriented thinking to the architecture design of the command and control information system, which enhanced the information processing capacity of the command and control information system [28]. Zhang Hongming et al. put forward the C/S and B/S hybrid architecture, and on this basis, the system was separated from the business logic through the data access layer, so that the database was completely transparent to the user. By running the RunProxy to connect the client and server, the security of the server was greatly improved [29]. Based on the idea of hierarchical hierarchy and OODA control loop, He Hua et al. proposed a four-layer command and control architecture for the unmanned combat system based on cognition, which ensured the unity of command, the flexibility of control and the scalability of the system [30]. Furthermore, Li Minglei proposed a temporal HTN planner TPHTN (Time Preference HTN) to deal with time constraints with preferences [31]. The planner used STNP (simple time network with preference) to express the time preference information, and extended the operators and methods to express the time preference information in the planning domain knowledge. In the planning process, TPHTN propagated STNP from top to bottom and proposed three definitions of horizontal consistency to estimate the quality of STNP. Besides, it designed a new heuristic search to select appropriate operators or methods and corresponding STNP according to the quality. In the end, TPHTN generated a plan that meets the decision maker's preference when the planning process terminates. Besides such benefits of this method, it neither expressed continuous preference nor used heuristic search algorithm to guide the planning direction. Combined with the above information, it can be seen that the traditional modeling method is not suitable for the current command and control architecture modeling requirements.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



In summary, this study proposes a new method MultiAgents-HTN based on the architecture, which can solve these problems well. Based on the current command and control domain requirements, we build a multiagent-based distributed command and control model architecture, optimize the HTN algorithm and take the resource scheduling problem as the experimental background. The algorithm in this paper will be compared with the original HTN algorithm, TPHTN algorithm, F - HTN algorithm and so on. The comparative experiments show that the proposed algorithm has certain superiority and high convergence in solving problems for distributed cooperative task planning.

## 3. Design of Distributed Command and Control Architecture Model Based on Multi-Agent

According to the modeling requirement of command and control decision-making function, the agent structure block diagram of command and control module can be obtained, as shown in the following figure:

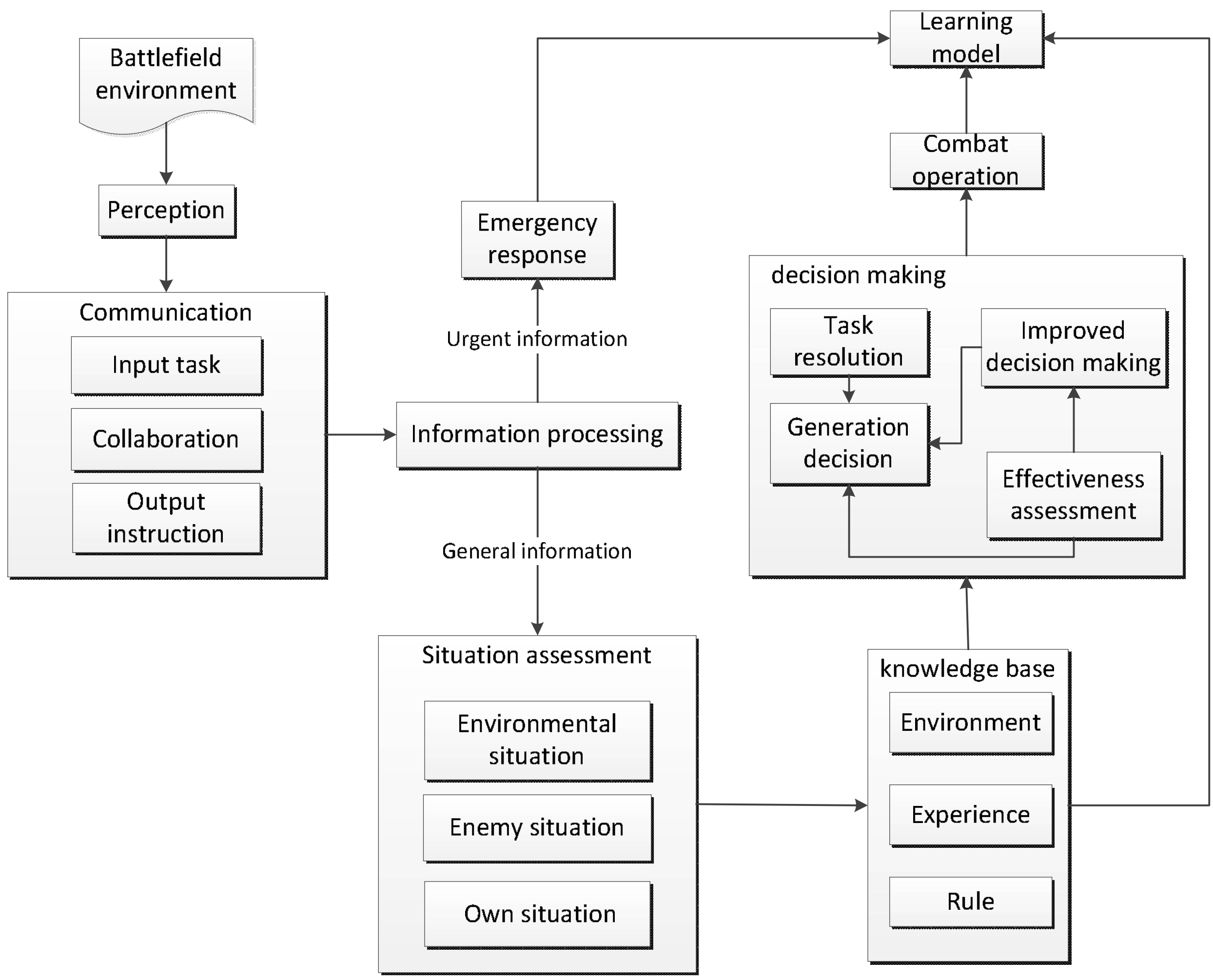


**Figure 1.** Block diagram of the command and control agent

The purpose of the distributed command and control model [32] is to improve the utilization of alleged resources, reduce local contradictions in the command and control process, and reduce the complexity of the model structure while the mission requirements are fulfilled. The following variables are defined based on the input task's timing constraints, resource requirements and other constraints. $d_{im}$ is the allocation variable of the accused resource, $t_{ijm}$ is the resource transfer variable, $o_{ij}$ is the timing variable, assuming as the objective function, and $Y$ is the time required for the completion of the input task. The problem is described as follows:

$$
s.t.\begin{cases}
\sum_{i=0}^{n} t_{ijm} - d_{im} = 0; \; i=1,\cdots,N; m=1,\cdots,K \\
\sum_{i=0}^{n} t_{jim} - d_{im} = 0; \; i=1,\cdots,N; m=1,\cdots,K \\
\sum_{i=0}^{n} t_{i0m} = \sum_{j=0}^{n} t_{ojm} = 1 \\
s_i - s_j + t_{ijm} * \left( \frac{p_{ij}}{q_m} + o_{ij} * T \right) \leq o_{ij} * T - time_i; \; i、j=1,\cdots,N; m=1,\cdots,K \\
\sum_{m=1}^{K} r_{ml} * d_{im} \geq R_{il}; \; i=1,\cdots,N; l=1,\cdots,N \\
s_i - Y \leq -t_i; \; i=1,\cdots,N \\
0 \leq Y \leq T \\
s_i \geq 0 \\
t_{ijk} \in \{0,1\} \\
d_{ik} \in \{0,1\}
\end{cases}
$$

At this time, the relevant mathematical description of command and control model is the mixed element linear programming problem, namely the search problem of state space. Perform clustering analysis on the input tasks.

Different tasks are assigned to different agents according to the clustering results, and then the workload of agents is minimized. The following variables are defined: $W^I$ is the cooperation threshold within agents, $W^K$ is the cooperation threshold outside agents, and the minimum load is set to $\min C_w$.

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

Complex task processing determines the external collaboration threshold of command and control center. To solve the problem of collaboration among agents' internal decision-making, it is necessary to determine the maximum external collaboration threshold between agents, and to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



minimize the indirect collaboration threshold in command and control structure, which is defined as $minW_{\max}$. Define $e_{ij}$ as direct connection variable, $z_{ijk}$ as indirect connection variable, internal collaboration function as $I(n)$, and external collaboration function as $E(n)$, as follows:

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

## 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 need to make a plan to maximize their own interests according to the local situation they have mastered. Based on the above content, in order to avoid the interaction between agents, eliminate local conflicts or low resource utilization, it is necessary to plan the distributed cooperative task [33] to ensure the completion of the global task and the maximization of overall interests. To this end, this chapter mainly describes the cooperative relationship between distributed multi-agents in the decision-making process of command and control. Taking the task planning problem as the background, the multi-agent cooperative mode and relationship network of command and control decision-making link are constructed.

### 4.1. Description of Basic HTN Algorithms

HTN decision system is widely used in the planning system of AI field. HTN represents the change of state by the concept of atomic operators, and a group of atomic operators represents a state. The basic principle of HTN is to generate a series of instruction actions based on a target or task, and input the given problem domain or task, including the initial test state of the process, and the state is the sequence of ordered unfinished task set.

#### 4.1.1. Parameter Definition

Definition 1 HTN is described as a six-tuple $<U,I,C,A,T,N>$ with extended pre-sequence language. In the six-tuple, $U$ represents variable set, $I$ represents constant set, $C$ represents logical consistent set, $A$ represents set the of agent's task action, $T$ represents task set, and $N$ represents task identifier set. These sets are independent of each other.

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

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

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

4.1.2. Constraints

1. Variable constraint

$$(v_1 = c)(v_1 = v_2)$$

Where $v_1, v_2 \in V$ is a variable identifier, $c \in C$, $c$ is a constant identifier

2. Sequential constraint

$$n < n'$$

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

3. State constraint

$$(n,l)(l,n)(n,l,n')$$

$(n,l)=1$ means that $l$ is executed after $n$. $(l,n)=1$ means that $n$ is executed after $l$, and $(n,l,n')=1$ means that $l$ is executed between $n$ and $n'$.

4.1.3. Analysis of Algorithm Flow

The HTN planning is instantiated and analyzed. The specific process is as follows:

**Algorithm 1** HTN Algorithm Pseudo Code

1.**HTN**$(s,U,C,O,M)$

2. if $(U,\mathrm{C})$ can be shown to have no solution

3. Then return failure

4. else if $U$ is primitive then

5. if $(U,\mathrm{C})$ has a solution then

6. non-deterministically let $\pi$ be any such solution

7. return $\pi$

8. else return failure

9. else

10. choose a non-primitive task node $u \in U$

11. $active \leftarrow \{ m \in M \mid \text{task}(m) \text{ is unifiable with } t_u \}$

12. if $active \neq \varnothing$ then

13. non-deterministically choose any $m \in active$

14. $\sigma \leftarrow$ an mgu for $m$ and $t_u$ that renames all variables of $m$

15. $(U', C') \leftarrow \delta(\sigma(U, \mathrm{C}), \sigma(u), \sigma(m))$

16. $(U', C') \leftarrow$ apply-critic $(U', C')$

17. return HTN $(s, U', C', O, M)$

18. else return failure

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

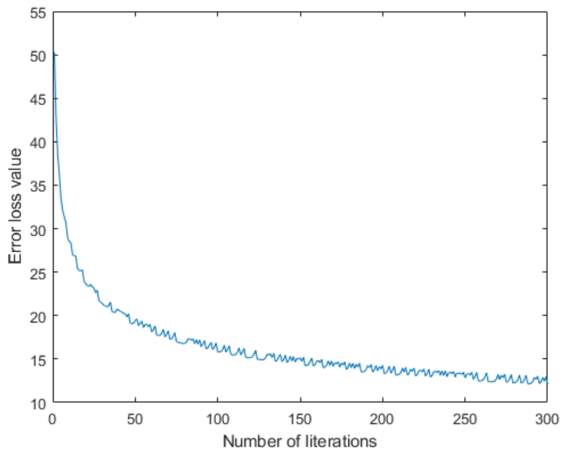


**Figure 2.** Function convergence curve

The simulation takes the number of iterations as the reference object and the error loss value as the criterion of convergence. The experimental result shows that the proposed algorithm has good convergence.

4.2. Distributed Collaboration Framework of MultiAgents Based on HTN

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



Usually, in a centralized environment, the collaboration framework of MultiAgents-HTN (MultiAgents-Hierarchical Task Network Planning , MA-HTN) is to establish a decision-making center, control the overall situation, and deduce and solve the scheme.

However, in the actual command and control process, it is difficult to achieve the overall situation sharing, and cannot guarantee the security of a single decision-making center. Once the center is destroyed, it loses the initiative of war. Therefore, distributed MA-HTN has higher superiority. Each agent deduces the scheme internally and does not share the relevant information with each other in the process of the scheme deduction. Moreover, each agent needs to cooperate with the scheme that has dependency or mutually exclusive relationship to solve local conflicts, so as to construct a task planning collaboration network and maximize the overall interests. The cooperative task planning process framework of distributed MA-HTN is shown in the figure.

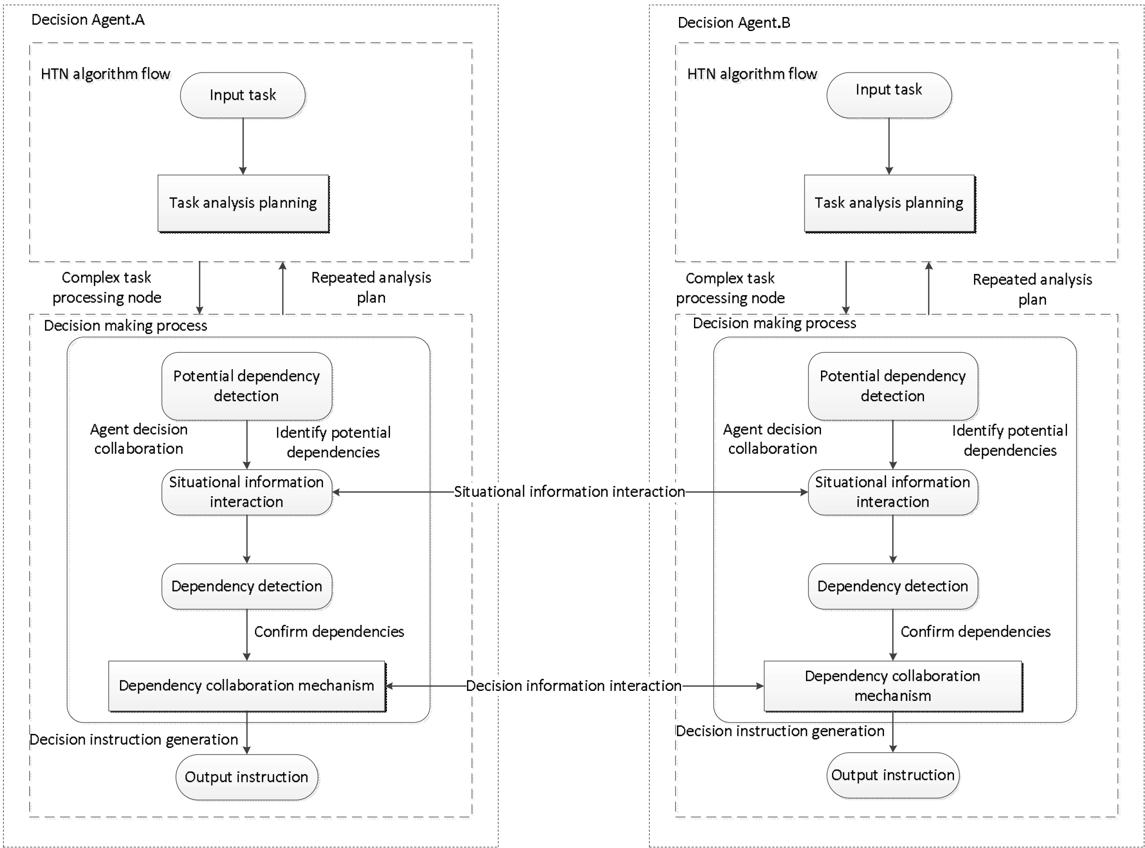


**Figure 3.** Distributed mission planning process for distributed MA-HTN

The cooperative task planning of distributed MA-HTN is based on the modification of the decision-making scheme. Cooperative relationship adjustment mechanism is added to integrate the process of decision-making adjustment and modification. Thus, the cooperative relationship control of decision-making actions is realized in the planning process. The algorithm flow is based on the original HTN planning framework, and adds intermediate steps such as potential cooperative relationship detection and situational information sharing. The potential cooperative relationship detection is used to detect whether there is a potential dependency or contradiction between the

 doi:10.20944/preprints201904.0252.v1



decision instruction of the agent and the decision instruction generated by other agents. After confirming the potential cooperative relationship, the situation interaction module releases its own situation information and cooperative relationship information to other agents, and receives information from other agent nodes. The collaboration detection determines the specific collaboration relationship according to the results of situation information interaction, and processes the specific collaboration relationship through the coordination relationship adjustment mechanism, so as to maximize the overall interests of each agent. The algorithm flow is not only applicable to the planning of cooperative relationships between any two independent agents, but also applies to cooperative planning among multiple agents.

On the one hand, the distributed MA-HTN cooperative task planning process integrates the collaboration mechanism into HTN, making the decision-making instructions formulated and collaborated simultaneously. Therefore, a decision-making scheme compatible with other agents can be generated at one time. On the other hand, we can discover and deal with the confirmed cooperative relationship 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. To sum up, compared with algorithm flow framework for traditional multiagent collaboration, this algorithm has a higher advantage in reducing the global decision generation time and improving the collaboration efficiency. The specific algorithm flow is as follows:

**Algorithm 2** MultiAgents-HTN Algorithm Flow in Distributed Environment

---

1. Procedure MA-HTN $(s,T,D)$

2. set $s$,initial state

3. set $T$,initial task set

4. set $D$,domain

5. initial operator $ActionID = 0$

6. set $P =$ the empty plan

7. $T_0 \leftarrow \{t$ belongs to $T$ : no other task in $T$ is constrained to precede $t\}$

8. loop

9. if $T$ is empty then return $P$

---

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



---

non-deterministically 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

$ActionID = ActionID + 1$

$opmessage(a) = (ID, Resourceused(a), st(a), et(a), cost(a))$

$sendlist \leftarrow opmessage(a)$

send $opmessage(a)$ to the other planners

if $receivelist \neq \varnothing$ :

$ResourceID = Resourceused(a)$

conflict actions $a' \leftarrow \{a'$ belongs to the $extopmessages$ of $receivelist$ :

has $Resource(a') = \text{Re}\,sourceusedID,$ 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 $P$

---

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



else if $st > st'$<br>
delete $opmessage(a)$ in $sendlist$<br>
$Resourcestate(r) = i', Resourcest(r) = st', Resourceet(r) = et'$<br>
Backtrack<br>
else delete $T$ in $T_0$ and add $a$ into $P$<br>
if $t$ is a non-primitive task then<br>
choose a method to decompose $t$ into subtasks $\{t_1, t_2, \cdots t_n\}$<br>
delete $T$ in $T_0$ and add $\{t_1, t_2, \cdots t_n\}$ into $T_0$<br>
if receiving a new $opmessage(a')$ in $receivelist$ of $a'$ and $sendlist$<br>
$\mathrm{Re}\,sourceID = Resourceused(a')$<br>
conflict actions $a'$ belongs to the $extopmessages$ of $receivelist$ :<br>
has $Resourceused(a) = ResourceID$, and has the smallest $st(a)$<br>
and duration $(st, et)$ is overlapped with duration $(st', et')\}$, then<br>
if $st' < st$<br>
delete the rest $opmessage$ in $sendlist$<br>
$Resourcestate(r) = i', Resourcest(r) = st', Resourceet(r) = et'$

42. Backtrack to task node at action $a$

43. else if $st' > st$

44. delete $extopmessage(a')$ in $receivelist$

45. $Resourcestate(r) = i, Resourcest(r) = st, Resourceet(r) = et$

46. delete $T$ in $T_0$ and add $a$ into $P$

47. repeat

48.end MA-HTN

4.3. Coordination Mechanism of Local Conflicts

For the contradiction of local conflicts between the various agents, it is necessary to adopt a corresponding cooperation mechanism for processing. Decker uses GPGP to deal with the coordination of resource conflicts. However, the coordination mechanism used in this paper is that one party keeps the original decision-making plan unchanged, and the other party modifies the original decision-making plan. In the figure, there is a local conflict between Action.A1 and Action.B1 between Agent.A and Agent.B. Action.A1 remains unchanged. Agent.B modifies Action.B1 and abandons Task.B1. Task.B is decomposed into existing sub-tasks Task.B1 and Task.B2, and Action.B3 is selected to avoid the conflict. Considering that local conflicts belong to different agents, their generation time may be different. In the process of agent's cooperative planning, potential conflict contradictions are detected earlier, and the decision-making modification of which party should be made is judged according to the collaborative mechanism. The party that invalid adjusts the decision-making scheme can continue to generate the decision-making instructions. The party that needs to be adjusted gets the conflicting task nodes and makes modifications. Finally, the sent or received conflict information is updated accordingly. The coordination chart of agent decision-making action conflict relationship is as follows.

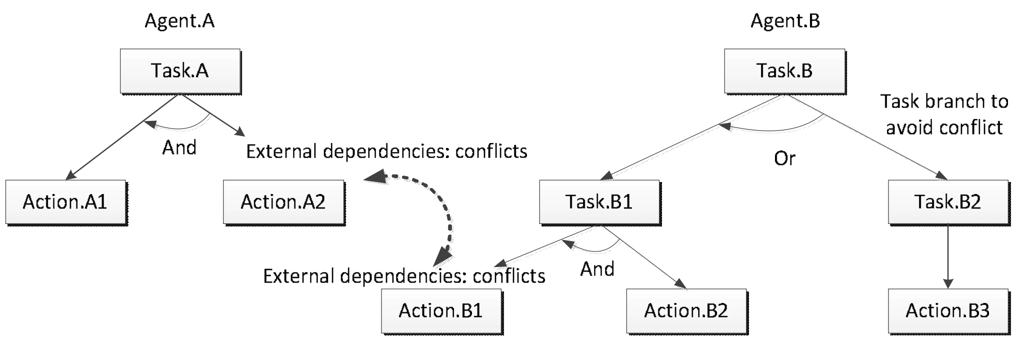

**Figure 4.** Agent conflict coordination mechanism

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 sub-tasks Task.B1 and Task.B2, and the relationship between them is "or" in logical relationship. The sub-task 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, that is, the execution process of A2 will lead to B1 cannot be implemented smoothly.

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

The reusable resource conflict coordination mechanism can be used by multiple decision-making agents at the same time to generate repeated resource conflicts. For these resource conflicts, the tasks with higher priority are reserved and the tasks with lower priorities are coordinated on the basis of determining the priority of the input tasks. The task priority criteria are as follows. First, it is the minimum task that use resources earlier have priority to ensure that the total global task time. Second, the tasks with higher overall interests have higher priority. Third, in the case of similar overall interests, the lower resource consumption has higher priority.

The resource consuming conflict coordination mechanism, that is, part of the consumed resources can be used by multiple agents at the same time, but the number of resources is insufficient. In this case, the resource consuming conflict coordination mechanism is adopted for conflict processing. On the basis of judging the priority of the input task, the agent with high priority is guaranteed to complete the input task in priority, so as 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 architecture 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, a large number of resources need to be transported from resource storage to material demand in limited time. This problem is a complex decision-making problem. It involves a large number of resource scheduling sites and resources. It is necessary to consider both spatiotemporal constraints and numerical logic reasoning in the solution process. In addition, on the premise of meeting the requirements, it is necessary to generate high-quality and fast solutions.

Suppose $P_1, P_2, P_3$ are resource demand points. $Q_1, Q_2, Q_3$ are resource storage points, which store Q $q_1, q_2, q_3, q_4, q_5$ resource types respectively. Locations $O_1, O_2, O_3, O_4, O_5$ are the transfer centers for resource scheduling. Assume that there are seven transport teams that can be used to solve the problem, and that the initial location of all transport teams is in the resource storage point and the transfer center 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 shown in the table.

 doi:10.20944/preprints201904.0252.v1



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

| Number | Transport Team | Transportation Resource Category | Transport Limit | $S\_max$ | $S\_min$ |
|---|---|---|---|---|---|
| 1 | $\text{Team}_1$ | $O_1, O_2$ | 500 | 50 | 75 |
| 2 | $\text{Team}_2$ | $O_1, O_2$ | 400 | 45 | 65 |
| 3 | $\text{Team}_3$ | $O_3, O_4$ | 500 | 45 | 70 |
| 4 | $\text{Team}_4$ | $O_3, O_4$ | 350 | 35 | 50 |
| 5 | $\text{Team}_5$ | $O_4$ | 400 | 30 | 60 |
| 6 | $\text{Team}_6$ | $O_2, O_4$ | 300 | 30 | 55 |
| 7 | $\text{Team}_7$ | $O_1, O_2, O_3, O_4$ | 300 | 45 | 65 |

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

$$T_{\min} \le End - Start \le T_{\max}; \begin{pmatrix} (T_{\min}, 0.5), \\ ((T_{\min} + T_{\max})/2, 1), \\ (0.2 * T_{\min} + 0.8 * T_{\max}, 1), \\ (T_{\max}, 0.5) \end{pmatrix}$$

$$T_{\min} < 0.5, T_{\max} > 0.5$$

$$\frac{(T_{\min} + T_{\max})}{2} < 1; (0.2 * T_{\min} + 0.8 * T_{\max}) > 1$$

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

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



In order to test the performance of Multi-Agents 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 shown in the table:

**Table 2.** Detailed plan of resource scheduling

| Number | Initial state | Temporal Constraint |
|---|---|---|
| Problem_1 | $(resource\_needA_1 p_1 300)$<br>$(resource\_needA_3 p_2 200)$ | $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ |
| Problem_2 | $(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 500)$ | $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ |
| Problem_3 | $(resource\_needA_3 p_2 200)$<br>$(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 500)$ | $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$ |
| Problem_4 | $(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 400)$<br>$(resource\_needA_3 p_2 700)$ | $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(2,2),(3,2),(4,3))$ |
| Problem_5 | $(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 700)$<br>$(resource\_needA_3 p_2 100)$ | $end.1 \le 4((0,0),(1,1),(3,1),(5,1))$<br>$end.1 \le 4((0,0),(4,3),(5,2),(7,1))$<br>$end.1 \le 4((0,0),(3,2),(4,1),(5,4))$<br>$end.1 \le 4((0,0),(1,4),(5,2),(6,3))$ |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



| | | |
|---|---|---|
| | | $end.1 \le 4((0,0),(2,1),(4,2),(6,3))$ |
| Problem_ 6 | $(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 500)$<br>$(resource\_needA_3 p_2 500)$<br>$(resource\_needA_3 p_2 700)$ | $end.1 \le 4((0,0),(2,1),(3,1),(4,1))$<br>$end.1 \le 4((0,0),(1,3),(3,1),(5,7))$<br>$end.1 \le 4((0,0),(3,7),(7,8),(2,5))$<br>$end.1 \le 4((0,0),(3,2),(5,3),(6,5))$<br>$end.1 \le 4((0,0),(4,4),(5,3),(5,1))$ |
| Problem_ 7 | $(resource\_needA_3 p_2 100)$<br>$(resource\_needA_3 p_2 300)$<br>$(resource\_needA_3 p_2 700)$<br>$(resource\_needA_3 p_2 900)$ | $end.1 \le 4((0,0),(4,3),(6,3),(4,7))$<br>$end.1 \le 4((0,0),(6,4),(4,6),(7,3))$<br>$end.1 \le 4((0,0),(4,5),(3,4),(4,1))$<br>$end.1 \le 4((0,0),(5,3),(3,1),(6,4))$<br>$end.1 \le 4((0,0),(6,2),(4,7),(4,1))$<br>$end.1 \le 4((0,0),(7,1),(3,6),(4,4))$ |
| Problem_ 8 | $(resource\_needA_3 p_2 200)$ | $end.1 \le 4((0,0),(3,1),(2,2),(5,5))$<br>$end.1 \le 4((0,0),(3,3),(3,2),(4,1))$ |

### 5.2. Analysis of Experimental Results

In this paper, MultiAgents-HTN, TPHTN [34] and F-HTN algorithms are compared and analyzed. TPHTN algorithm with better temporal preference has been widely used in practical fields. F-HTN is a dynamic task hierarchical planner with good dynamic re-planning capability. Therefore, this paper chooses the above algorithm as the comparison method.

#### 5.2.1. Quality Comparison of Generating Schemes

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



This paper compares the quality of the generating schemes of MultiAgents-HTN and TPHTN algorithm, and solves the above algorithms separately. The quality comparison diagram of the generating schemes is as follows:

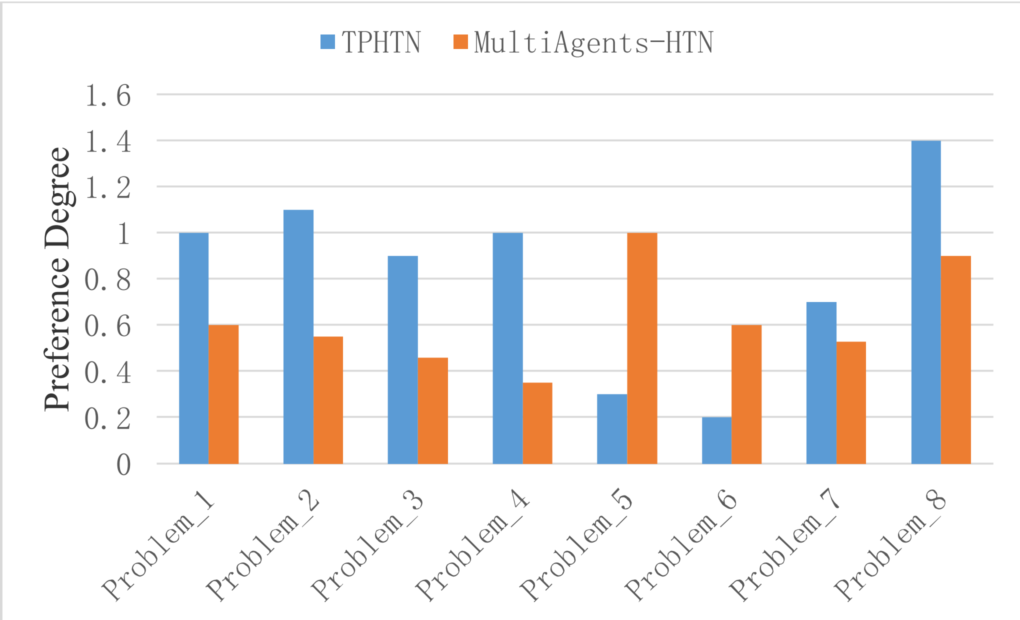


**Figure 5.** Initial state collection of resource scheduling planning

It can be found from the figure that both algorithms can solve the problem with relatively high quality, but MultiAgents-HTN has a better generation scheme than TPHTN. For some complex problems, such as Problem_6 and Roblem_7, the quality of TPHTN generated solutions is obviously not as good as MultiAgents-HTN. The reason is that TPHTN adopts a rule-based planning method to quickly generate the solution. However, when the problem is complex, the exponential growth of the rules that need to be dealt with leads to the insufficient quality of the generating scheme. Multi-Agents-HTN adopts a heuristic search strategy based on depth-first, so it is more prominent in dealing with complex problems.

### 5.2.2. Efficiency Comparison of Generation Schemes

The generation solution requires not only a high-quality solution but also an efficient enough. Excessive decision-making time often has a great impact on the outcome of the decision. The efficiency of the scheme generation of MultiAgents-HTN and TPHTN is compared. The efficiency comparison of generating schemes is shown below.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



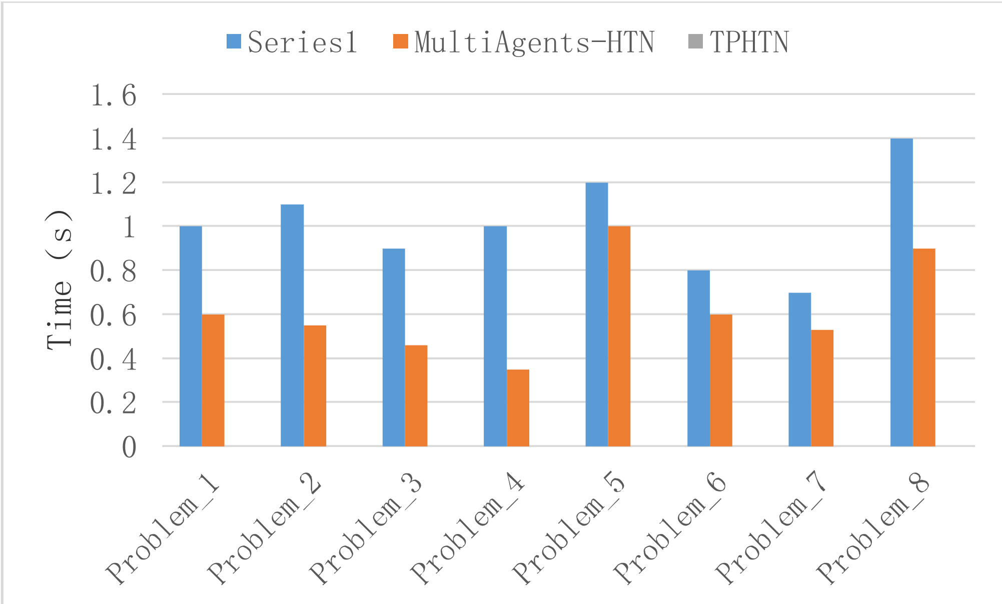


**Figure 6**. Comparison of the generation efficiency of MultiAgents-HTN and TPHTN schemes

By comparing the results of the above figure, it can be found that in general, the generation time of MultiAgents-HTN and TPHTN is approximately the same, but the generation scheme of MultiAgents-HTN has high quality. In dealing with complex problems, MultiAgents-HTN requires more time to generate decision-making solutions. Because of the small time difference, it is almost negligible. In conclusion, MultiAgents-HTN has higher scheme generation efficiency than TPHTN.

5.2.3. Comparisons of Re-Planning Numbers

After generating a decision plan, when a scenario is encountered during the execution of the scenario, there are usually two solutions for that case. Firstly, the decision plan should be revised. When the decision plan encounters an abnormality in the execution process, the decision plan is adjusted according to the previously determined decision modification rule. When the adjustment still encounters a tense anomaly, the re-planning of the decision-making scheme is carried out. Secondly, the temporal exception handling is performed according to the temporal constraints. The re-planning rate is obtained by dividing the average number of re-planning times by the number of occurrences of the temporal anomaly. Select 4 out of the 8 questions above as the subject. Among them, Problem_8 is a simple problem. Problem_2 and Problem_5 are more complex issues. Problem_7 is a complex issue. The specific experimental results are as follows:

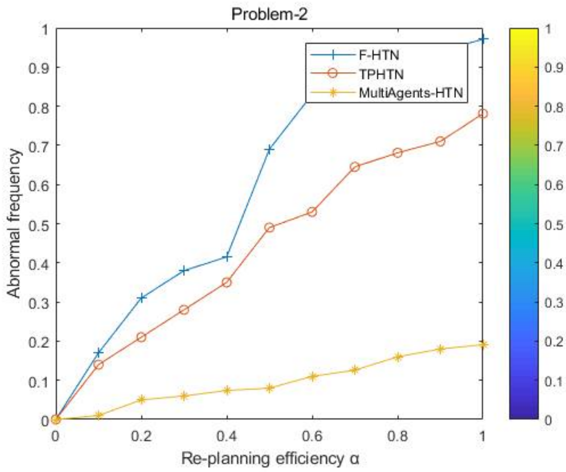


**Figure 7**. Problem_2 comparison of resource re-planning times

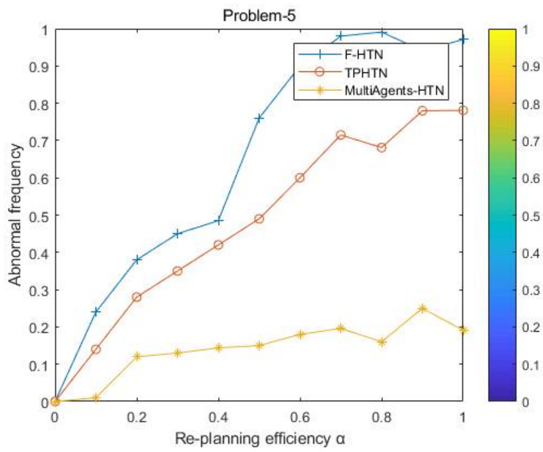


**Figure 8**. Problem_5 resource re-planning comparison

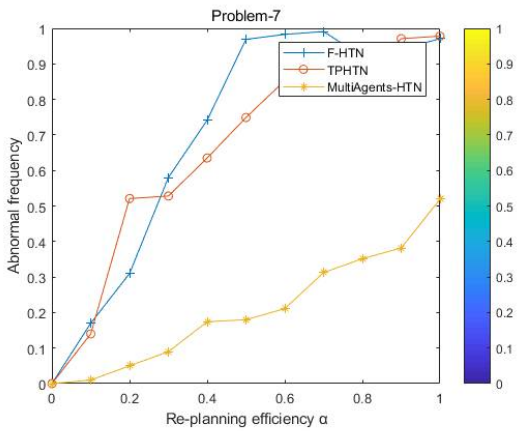


**Figure 9**. Problem_7 resource re-planning comparison

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2019 doi:10.20944/preprints201904.0252.v1



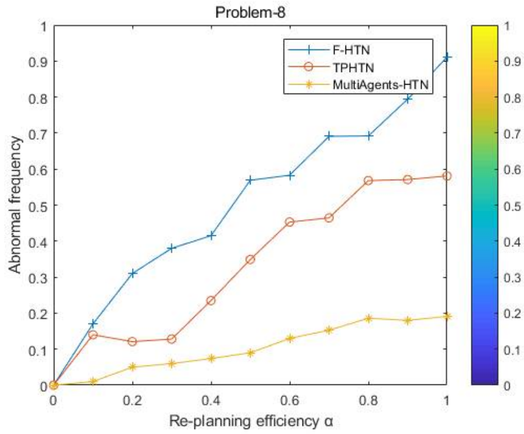


**Figure 10**. Problem_8 comparison of resource re-planning times

From the results of figure 7 to figure 10, it can be found that with the increase of abnormal frequency in the execution of decision-making schemes. The re-planning rate basically shows an increasing trend. But there are also unexpected situations, which occur because the quality of the decision-making plan is contingent. In the same situation environment, the rescheduling rate of MultiAgents-HTN is the lowest and F-HTN is the highest for the rescheduling of four problems. When the problem is too complex or the solution space is too large, F-HTN and TPHTN may not be able to solve and fall into deadlock. MultiAgents-HTN uses repair local decision-making scheme when the decision-making scheme needs to be re-planned. Therefore, the efficiency of reprogramming is much higher than that of the other two algorithms.

### 5.2.4. Comparisons of Algorithmic Complexity

In the same situation, the framework of this algorithm is compared with F-HTN and TPHTN algorithms. The results are shown in the table 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.** Algorithm time space complexity 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 |

It can be seen from the experimental data that the algorithm in this paper analyzes the function value and its time-space complexity under the same number of iterations. It finds that the algorithm can achieve lower objective function value and has better convergence and adaptability.

## 6. Conclusion

With the frequent occurrence of local disputes and emergencies, more and more scholars at home and abroad pay attention to the generation of decision-making schemes for emergency response to real events. The basic purpose of the decision-making scheme is to achieve the tactical objectives of the complete scheme that is to minimize the casualties and property losses as much as possible. Moreover, it is necessary to shorten the generation time of the scheme to a great extent and coordinate the local disputes after the implementation of the scheme quickly and efficiently. As one of the efficient intelligent planning algorithms, HTN has good adaptability and high efficiency in dealing with such problems, and it is the basis of the algorithm framework proposed in this paper.

In this paper, a distributed command and control architecture model based on multi-agent is constructed according to the demand of current command and control structure. The problem of

 doi:10.20944/preprints201904.0252.v1



large amount of data and complex situation environment of emergency events is solved by using the high collaboration efficiency of multiagent and the superiority of dealing with complex problems. On the basis of HTN algorithm, an algorithm framework of MultiAgents-HTN is proposed, and the algorithm is simulated and compared comprehensively. Experiments show that the algorithm has high practicability and research value in solving complex problems with large amount of data and complex situation. In addition, it also has high adaptability and superior self-cooperation ability. Furthermore, the future research direction of this paper involves improving the exact method by adding effective inequalities to solve more instances in the branching and cutting framework. Community based on larger meta-heuristic large neighborhood search (LNS) may be considered as a good candidate approximation method. Moreover, the capacity version of the problem is also an interesting topic for future research.

**Author Contributions:** Conceptualization, G.W. and J.Z. Methodology, J.Z. and Y.S. Validation, Y.S. and G.W. Writing—Original Draft Preparation, J.Z. Writing—Review and Editing, J.Z. Funding Acquisition, Y.S.

**Funding:** The National Natural Science Foundation of China, grant numbers 61703426, 61876189, 61806219, 61273275, and 61503407; the China Postdoctoral Science Foundation, grant number 2018M633680, funded this research.

## References

1. F. Muyun, Z. Chanchan, H. Xin, L. Xiao, 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.
2. W. Zhengxin, F. Jingbo, J. Guoping, C. Jinde, X. Min, 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.
3. Chao Q, Dan W, Muñoz-Avila H, et al. Hierarchical task network planning with resources and temporal constraints. Knowledge-Based Systems, 2017: S0950705117303167.
4. Martin H, Daniel V, Lisa W. Dynamics in matching and coalition formation games with structural constraints. Artificial Intelligence, vol. 262, pp. 222-247, 2018.
5. Torreño, Alejandro, Onaindia E, Komenda, Antonín, et al. Cooperative Multi-Agent Planning: A Survey. Acm Computing Surveys, vol. 50, no. 6, pp. 1-32, 2017.
6. Otte M, Kuhlman M, Sofge D. Competitive target search with multi-agent teams: symmetric and asymmetric communication constraints. Autonomous Robots, 2017.
7. Mi X, Zou Y, Li S, et al. Self-triggered DMPC design for cooperative multi-agent systems. IEEE Transactions on Industrial Electronics, no. 99, pp. 1-1, 2019.
8. Liang Q, She Z. Constraint consensus of heterogeneous multi-agent systems. International Journal of Modern Physics C, 2018:S0129183118400053.
9. Tazi K, Abbou F M, Abdi F. Multi-agent system for microgrids: design, optimization and performance. Artificial Intelligence Review, no. 28, 2019.

10. Li Q. Cooperative control of multi-agent systems. Dissertations & Theses - Gradworks, vol. 7, no. 8, pp. 822 - 831, 2009.
11. Omolbanin Yazdanbakhsh, Scott Dick, FANCFIS: Fast adaptive neuro-complex fuzzy inference system. International Journal of Approximate Reasoning, vol. 105, 2019.
12. Iqbal M, Leth J, Ngo T D. Generalized hierarchical block circulant structure of multi-agent systems. International Journal of Control, Automation and Systems, vol. 15, no. 3, pp. 1275-1286, 2017.
13. B. A. Jackson, K.S. Faith, H.H. Willis. Are we prepared? Using reliability analysis to evaluate emergency response system. Journal of Contingencies and Crisis Management, vol. 9, no. 3, pp. 147–157, 2011.
14. E. Piatyszek, 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.
15. L. Xiuhua, H. Jian, W. Xinjiang. Distributed fault estimation for multi-agent systems based on intermediate observer. Acta Automatica Sinica, [Online], available: 1-11[2019-03-03].https://doi.org/10.16383/j.aas.c180179.
16. Ao, Wei, S. Yongduan, 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.
17. S. Jian, D. Fang, C. Jie. Development status and trend of land motion body control system. Acta Automatica Sinica, vol. 44, no. 11, pp. 1985-1999, 2018.
18. L. Fan, Y. Hongyong, Y. Yize, Li Lingyu, Liu Yunshan. Finite-time integral sliding mode control for multi-agent systems with mismatched disturbances. Acta Automatica Sinica, [Online], available: 1-10[2019-03-03].https://doi.org/10.16383/j.aas.c180315.
19. Dong Tao, Li Xiao-Li, Zhao Da-Duan. Consistency analysis of third-order discrete multi-agent system based on event trigger. Acta Automatica Sinica, [Online], available: 1-7[2019-03-03].https://doi.org/10.16383/j.aas.2017.c170406.
20. L. Juan, Z. Hao, W. Zhuping. Cooperative output regulation of heterogeneous multi-agents based on self-triggering. Acta Automatica Sinica, [Online], available: 1-10[2019-03-03].https://doi.org/10.16383/j.aas.2018.c170660.
21. Y. Dongyue, 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.
22. Ch. Shiming, G. Junjie, G. Yanli, P. Huiqin, Q. Yun. Fast consistency algorithms for multi-agents based on event triggering in composite connected topology. Acta Automatica Sinica, vol. 44, no. 12, pp. 2269-2277, 2018.
23. W. Miaomiao, Z. Hao, Y. Huaicheng, Ch. Shiming. Cooperative output regulation for asynchronously switched multi-agent systems. Acta Automatica Sinica, vol. 43, no. 5, pp. 735-742, 2017.
24. Z. Jun, 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.

25. H. N. Phuong, H. T. N. Nhat, M. Hoàng Hà, L. André, M. Trépanier. Solving the clustered traveling salesman problem with d-relaxed priority rule. Artificial Intelligence, 2018.
26. J. Haoran. Research on the Structure of Command Information System Based on Mobile Cloud Model. Ph.D. dissertation, Dept. National University of Defense Technology, China, 2016.
27. Z. Tianxiang, "Research on security mechanism of SOAP in command and control system of SOA." M.S. thesis, Dept. Shenyang Ligong Univ., China, 2014.
28. W. Jin, D. Wenxue, G. Hong. Design of a service-oriented command and control information system. Shanghai Aerospace Electronic Technology Research Institute, vol. 3, 2018.
29. Z. Hongming, L. Yongzhi, W. Yang. Research and implementation of command and control system based on hybrid architecture, vol. 4, 2008.
30. He Hua, Li Zhi-Fei, Wang Wei-Ping, Zhu Yi-Fan, Li Xiao-Bo. Multi-layer command and control architecture of unmanned combat system based on cognition. In: Proceedings of the 5th China Command and Control Conference. Beijing, China, 2007.
31. 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.
32. Jin Q, Lin R, Zou H, 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), 2018.
33. Feng Z, Wen G, Hu G. Distributed Secure Coordinated Control for Multiagent Systems under Strategic Attacks. IEEE Transactions on Cybernetics, vol. 47, no. 5, pp. 1273-1284, 2017.
34. Wang Z, Wang H W, Qi C, et al. A resource enhanced HTN planning approach for emergency decision-making. Applied Intelligence, vol. 38, no. 2, pp. 226–238, 2013.