

# 全局资源视图生成战术资源视图的方法研究<sup>\*</sup>

刘宏芳<sup>1</sup> 阳东升<sup>2</sup> 刘忠<sup>1</sup> 张维明<sup>1</sup>

(国防科技大学信息系统与管理学院 长沙 410073)<sup>1</sup> (海军兵种指挥学院 广州 510431)<sup>2</sup>

**摘要** 战术视图生成是从战役全局资源视图产生基于任务的作战视图,本文通过解决作战资源实体在作战任务上的聚类问题来生成战场战术态势图的基本要素之一——作战实体。基于功能能力、作战平台资源和作战任务的定义,提出了基于任务的作战资源实体生成方法。该方法包括两个部分:资源实体裁剪过程的数学描述以及作战资源实体分配到任务的匹配算法,并以联合战役为例,分析并设计了从联合战役作战资源的全局视图生成战术资源视图。

**关键词** 全局资源视图,基于任务的战术资源视图,使命分解

## Research on Methodology of Generating Task-Based Tactical Situation Picture from Global Resource Picture

LIU Hong-Fang<sup>1</sup> YANG Dong-Sheng<sup>2</sup> LIU Zhong<sup>1</sup> ZHANG Wei-Ming<sup>1</sup>

(College of Information System and Management, National University of Defense Technology, Changsha 410073)<sup>1</sup>

(College of Naval Arms Command, Guangzhou 510431)<sup>2</sup>

**Abstract** Generating Tactical Situation Pictures(TSP)is to construct TSP based on tactical task from global situation picture. In this paper,we solve the problems of combat entity clustering on combat task,i.e.,combat entities,one of essential factors of tactical situation pictures,are clustered on task. By definitions about function capability,combat platform and task,we advance an approach of Task-based tailoring(TBA)to build TSP. And the approach is composed of two parts: one is mathematics description about TBA process,another is arithmetic about combat entity resource clustering on task. With a case of joint campaign,we analyze and design TSP by the new approach.

**Keywords** Global resource picture, Task-based tactical situation pictures, Mission decomposition

在未来一体化联合作战中,由于分布式环境中并行决策(或者同步决策)是实施快速指挥的必要条件<sup>[1]</sup>,态势的共享与感知是并行决策的基础和前提。一体化联合作战在很大程度上模糊了战争的层次(如战役、战术和战斗)<sup>[2]</sup>,置身于战场环境中的指挥员通过公共作战视图(Common Operational Picture,COP)来共享态势与感知。

公共操作视图 COP 是一种全局视图,具有接收、关联、显示战场态势图像及战略战术命令的集成能力,能利用公共作战环境提供的服务为战斗人员生成通用作战视图。同时,COP 指相对稳定的保持时间长度的全局性的战略级别上的针对高层作战指挥人员的战场态势,它综合来自各个公共战术视图(Common Tactical Picture,CTP)的信息<sup>[3,4]</sup>。由于美军的传感器数据非常丰富,网络基础设施完备,各种数据标准相对完善,因此 COP 的生成采用了自底向上的体系结构方式,由来自各个 CTP 的信息进行融合处理产生 COP。

由于 COP 是全局视图,丰富的数据和信息在保证战场指挥员感知整个战场环境的同时也增加了指挥员快速、正确感知与决策的难度。而单独的 CTP 又是战场态势的被动反应,既有与任务密切相关的信息和数据,也有不相关或者相关的信息和数据,同时也并不具备战术执行的所有条件(包括作战资源和环境),也就是说 CTP 不具备任务执行的完备性。

一体化联合作战中的 CTP 既需要对环境被动的反应,也需要主动获取相关任务执行的条件,如执行任务的作战资源、作战编成以及环境状态等等。这种灵敏的 CTP 是各决策结点决策与行动同步的保证。本文通过作战资源实体基于任务

的裁剪实现了 CTP 中战术任务执行资源的主动聚类。

### 1 资源实体基于任务的裁剪流程

在全局资源视图生成战术资源视图过程中,资源实体的裁剪包括任务产生、资源的描述和资源实体的裁剪三个部分。

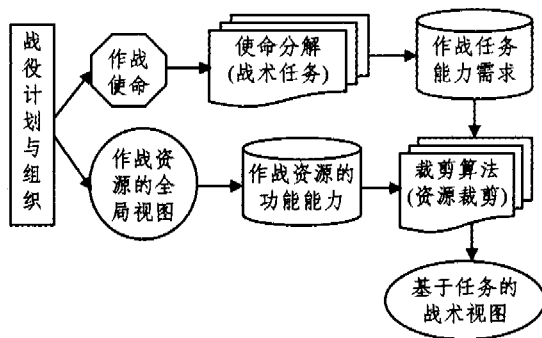


图1 资源实体基于任务的裁剪流程

第一部分:作战任务的产生。作战任务源于作战使命的分解,作战使命到任务分解的关键是分解依据与分解粒度。分解依据是指作战使命到可执行的任务所遵循的原则,可遵循的原则或依据包括以目标为依据,以平台资源的功能能力为依据,以战场区域划分为依据,或者以混合方式为依据,等等。分解粒度决定了问题的复杂性,也提高了资源裁剪结果的精确性,分解粒度越细则裁剪结果越精确,反之则粗略<sup>[5,6]</sup>。通常,分解粒度以作战资源的功能能力为依据,以子

<sup>\*</sup>国家自然科学基金(60504036,70401003)资助。刘宏芳 博士,主要研究方向为战场环境建模,作战态势评估和作战计划的生成技术等。

任务能够关联到具体作战平台资源为原则,也可以说是把作战使命分解到具备可操作性(被具体平台执行)的子任务为止,这一分解粒度依据为作战资源基于任务的裁剪奠定了基础。

第二部分:作战资源的描述。全局资源的描述是建立战场作战实体资源模型,是通过定量手段测度作战实体资源的能力,其描述工作包括作战实体资源的功能能力矢量、行进速度和探测范围等,其中作战实体资源的功能能力矢量是描述工作的关键。资源的描述与作战使命的分解产生任务共同奠定作战资源裁剪的基础工作。

第三部分:作战资源实体基于任务的裁剪。基于使命分解工作得到可执行的任务以及作战资源功能能力矢量的建立,作战资源的裁剪即作战资源在任务上的聚集,也可以说是任务需求与作战资源功能能力的匹配,以实现最佳资源组合去执行子任务,从而达到最佳执行效果。

## 2 基本概念

功能为作战资源实体所拥有,是执行使命任务的直接凭借,能力是作战资源所具备的功能大小的定量描述。作战资源是功能的载体,是战场作战平台的集合,这里的平台不仅包括舰艇、飞机和坦克等作战单元,还包括能够执行某种功能的任何作战实体,如可以实施地面进攻的步兵作战单元,指挥与控制指挥中心,可以实施跟踪与监视的传感器,等等。

根据所具备功能的多少,平台可以分单一功能平台和多功能平台。为建立平台资源能力度量的统一形式,我们以矢量形式描述平台资源能力。在某一次战役筹划中,设定其基本功能的资源矢量为 $[f_1, f_2, \dots, f_n]$ , $n$ 为战场作战平台资源的基本功能数量, $f_i$ ( $1 \leq i \leq n$ )表示某一作战平台资源具备的第 $i$ 项功能的能力大小。在本文案例中,我们设定资源能力划分为防空类 $f_1$ 、反舰类 $f_2$ 、反潜类 $f_3$ 、地面攻击类 $f_4$ 、火力(班)类 $f_5$ 、装甲类 $f_6$ 、扫雷类 $f_7$ 和识别探测类 $f_8$ 。定义平台资源的能力矢量 $C=[c_1, c_2, \dots, c_8]$ , $c_i$ 为 $f_i$ 类资源(在案例中 $1 \leq i \leq 8$ )的能力度量,则作战平台具备的资源以及战役任务对资源的需求都可以以矢量 $C$ 进行描述。

任务是作战平台资源根据行动的需要,在某一时间和空间执行其相关的功能。某一具体的任务包括以下几方面的信息<sup>[7,8]</sup>:一是由哪一个或哪些平台执行,即执行任务的平台集合 $GP(t)=\{p_1, p_2, \dots, p_k\}$ ;二是怎样执行任务,即平台使用发挥其什么功能去完成。通常以矢量 $[f_1, f_2, \dots, f_n]$ 表示某一平台资源在执行某一具体任务时所采用的功能能力;三是在什么时间什么区域执行,包括任务执行的开始时间 $St$ 、地点 $L$ 和持续时间 $D$ ;四是任务间的逻辑关系,或者说任务执行的前提条件与执行后的输出结果,包括任务执行的信息需求、任务的执行的准备和任务执行后的信息发布等。任务的逻辑关系通常以任务图 $T_G$ 的形式进行描述。

任务通常根据更高层次的作战使命分解而得到,依据不同的分解方法而分解不同的任务集合 $T=\{t_1, t_2, \dots, t_N\}$ ( $N$ 为任务的数量)。

## 3 作战任务及任务的数据属性表示

任务源于作战使命的分解。作战使命不同的分解方法(如目标分解、功能分解、域分解和混合分解,等等)确定不同的任务起点和不同任务类型,同样,不同的分解粒度也确定不同的子任务集合。通常,任务规划设计者对具体分解技术和模型粒度(在任务分解里的任务量)的选择应视设计过程的计算效率和其支持算法而定。在确定模型粒度的条件下,使命

分解得到的结果是类似的,尽管伴随有不同的分解方法。

分解的逆向操作为还原,而还原的最终目标为战役使命。因此,作战使命的分解过程以战役使命为起点,使命不同的分解方法以设定的任务粒度和辨识的战场重心为基础,通过关联战场重心、对作战平台资源的需求以及任务的其它基本属性来设置子任务集合。

在分解过程中以作战平台资源基本功能的划分为依据,即分解到树的叶节点任务的执行或完成与作战平台的功能能力相关联,以此为作战平台基于任务的裁剪奠定基础。同时,树节点与战场重心关联,明确任务的优先等级与重要性。使命分解得到的结构树为任务分析奠定了基础,任务分析首先需要从使命结构树中提取可执行的任务集合,集合中的任务通常被称之为子任务,子任务的数据属性以及子任务执行的逻辑顺序关系是任务分析的核心工作。

### 3.1 任务逻辑关系

使命分解产生完成使命所需要执行的任务集合,集合中的任务被称之为子任务。集合中的子任务依据其战场环境的复杂性可划分确定性任务和随机任务,所谓确定性任务是指明确指定执行的任务,这类任务建立在可靠的信息情报基础之上;所谓随机任务是指可能遭遇的任务,这类任务的产生源于信息情报的不确定。

任务逻辑关系分析是建立集合中所有任务执行时的顺序关系。任务的执行可能存在一个任务的完成是另一任务执行开始的前提条件,或者一个任务执行的开始需要某些任务均被成功完成,这便构成了集合中任务间的顺序关系。任务间的逻辑关系通常以图 $T_G$ 的形式描述。图2描述了所有任务间的逻辑关系,并对所有任务依据其战场环境的复杂性进行了分类。

### 3.2 作战任务的描述——任务的数据模型

任务的数据模型是确立任务的属性数据,其属性数据包括任务成功执行的功能能力需求 $R$ 、任务处理的估计时间(持续时间) $D$ 、任务的地理位置 $la$ 等。任务逻辑关系分析是对任务间关系的定性研究,而任务的数据模型是建立任务的定量描述,进一步为作战平台资源的裁剪奠定基础,使得从战役全局视图生成基于任务的战术视图成为可能。

对任务的基本属性以下列符号表示:

- $D(t_i)$ 任务 $t_i$ 的处理时间( $i=1, 2, \dots, N$ ,  $N$ 是任务的数目),即任务的持续时间;
- 任务执行的地理约束矢量 $la(t_i)=[x_i, y_i]$ ;
- 任务成功执行的功能能力需求矢量 $R_i=[r_{i1}, r_{i2}, \dots, r_{iL}]$ , $r_{il}$ 是指完成任务 $t_i$ 所需资源 $l$ ( $l=1, 2, \dots, L$ ,  $L$ 为资源的类型数量)的数量。

表1建立了本文战役案中分解所得到的所有子任务的属性数据。

## 4 资源实体的描述

在本文中,全局资源视图是对战役资源的管理,与战役环境密切相关。资源实体描述是建立作战平台资源的数据属性。作战平台资源的数据属性包括平台资源集合 $P_i$ 和平台自身属性 $P_A$ , $P_i=\{p_1, p_2, \dots, p_h\}$ , $h$ 是可获取作战平台的数量, $P_A$ 定义为平台的类型、资源能力、信息获取范围、地理位置以及其他属性。平台 $m$ 具备的资源能力矢量为 $S_m=[s_{m1}, s_{m2}, \dots, s_{mL}]$ ,其中 $s_{ml}$ 中表示在平台 $P_m$ 上可获取 $l$ 类型资源的数量(案例中 $1 \leq l \leq 8$ )。战役所有平台的数据属性参数如表2,表2描述了各参战平台属性,包括参数:资源能力、平台类型、运行速度。

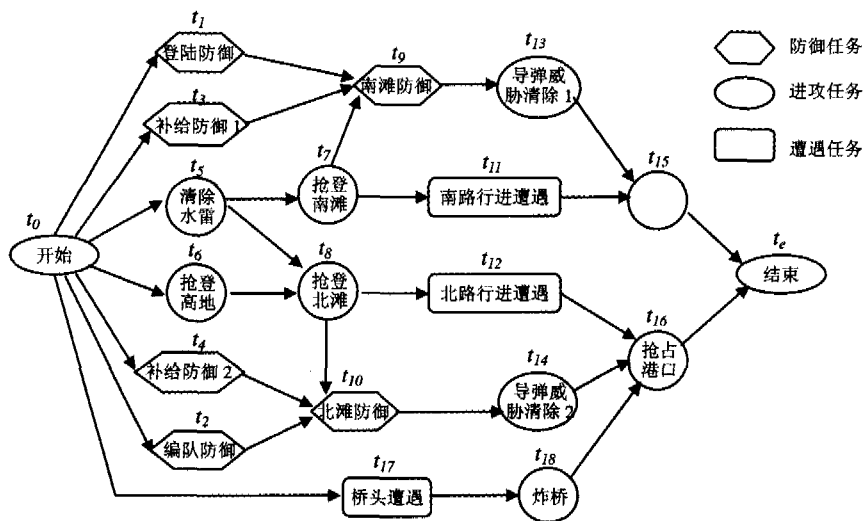


图2 任务逻辑关系图(TG)

表1 任务属性数据

任务	功能资源需求(R <sub>i</sub> )								处理时间 D(t <sub>i</sub> )	la(x, y)
	r <sub>1</sub>	r <sub>2</sub>	r <sub>3</sub>	r <sub>4</sub>	r <sub>5</sub>	r <sub>6</sub>	r <sub>7</sub>	r <sub>8</sub>		
t <sub>1</sub>	5	3	10	0	0	8	0	6	30	[70, 15]
t <sub>2</sub>	5	3	10	0	0	8	0	6	30	[64, 75]
t <sub>3</sub>	0	3	0	0	0	0	0	0	10	[15, 40]
t <sub>4</sub>	0	3	0	0	0	0	0	0	10	[30, 95]
t <sub>5</sub>	0	3	0	0	0	0	10	0	10	[28, 73]
---	---	---	---	---	---	---	---	---	---	---
t <sub>17</sub>	0	0	0	0	0	8	0	4	10	[5, 60]
t <sub>18</sub>	0	0	0	8	6	0	4	10	20	[5, 60]

表2 作战平台数据属性表

标识	名称	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	f <sub>6</sub>	f <sub>7</sub>	f <sub>8</sub>	速度
1	驱逐舰	10	10	1	0	9	5	0	0	2
2	护卫舰	1	4	10	0	4	3	0	0	2
3	巡洋舰	10	10	1	0	9	5	0	0	2
4	航空母舰	10	10	1	0	9	5	0	0	2
---	---	---	---	---	---	---	---	---	---	---

## 5 资源裁剪算法

### 5.1 基于任务裁剪的假设条件

基于任务的裁剪是分派作战平台资源到具体的作战任务,即解决某一具体任务的执行需要聚集哪些作战平台资源。

通常,某一作战任务的执行需要具备处理这一任务的所有条件,这些条件包括足够的平台资源的聚集、平台资源在任务区域集结完毕以及该任务准备就绪。由此,假设某一具体任务由平台资源进行处理需要具备如下条件和约束:

- (1) 在这一任务之前的所有任务都已处理完毕;
- (2) 分配到这任务的所有作战平台资源已到达指定地点;
- (3) 聚集的平台资源功能的执行不存在冲突;
- (4) 聚集的平台资源的功能能力与具体的子任务的需要匹配,即该平台资源的组合能在任务的执行过程中达到最佳的执行效果;
- (5) 聚集的平台资源的功能能力不小于该任务对平台资源功能能力的需求;
- (6) 一个作战平台每次只能处理一个任务。

通常,资源的裁剪或者说平台资源在任务上的聚类,其目

标是缩短完成总任务的时间,提高完成战役使命的有效性,其有效性的提高需要分配合适的平台或平台组到正确的区域去执行合理的任务,具体地说,就是在满足任务的资源需求的情况下提高平台资源的利用率,缩短完成任务过程的时间。同时,减少作战平台在任务执行上不必要的协作或冲突。

因此,对平台资源的聚类过程可以进行如下步骤的描述:

(1) 定义平台资源聚类过程变量

平台-任务分配变量  $w_{im}$ : 平台  $p_m$  分配给任务  $t_i$  时  $w_{im} = 1$ , 否则  $w_{im} = 0$ ;

平台在任务间的转移变量  $x_{ijm}$ : 平台  $p_m$  处理任务  $t_i$  后分配给任务  $t_j$ , 则  $x_{ijm} = 1$ , 不分配给任务则  $x_{ijm} = 0$ ;

任务顺序变量  $a_{ji}$ : 如果任务  $t_j$  的处理必须在任务  $t_i$  处理完后才能开始则为 1, 否则  $a_{ji} = 0$ ;

时间变量  $s_i$ :  $s_i$  任务  $t_i$  的处理开始时间。

(2) 平台资源聚类过程的约束分析

对任意平台  $pm(1 \leq m \leq K)$  和任务  $t_i(1 \leq i \leq N)$  之间如果存在聚集或分配关系 ( $w_{im} = 1$ ), 则平台  $p_m$  有且仅有两种情况被分配去执行任务  $t_i$ : 一种情况是平台  $p_m$  在处理完任务  $t_j$  后被分配处理任务  $t_i$ , 即转移变量  $x_{ijm} = 1$ ; 另一种是平台  $p_m$  被首次使用, 直接被分配处理任务  $t_i$ , 在这种情况下不存在转移变量, 即转移变量  $x_{ijm} = 1$ 。由此, 我们假设存在虚拟任务  $t_0$ ,  $t_0$  是所有任务的起点, 在分配初始, 所有平台资源都在虚拟任务  $t_0$  上, 记  $x_{im} = x_{ijm} = 0$ , 则平台-任务分配变量  $w_{im}$  和平台在任务间的转移变量  $x_{ijm}$  存在如下约束关系:

$$\sum_{j=0}^N x_{jim} - w_{im} = 0 \quad i=1, 2, \dots, N; m=1, 2, \dots, K \quad (1)$$

同时, 被分配处理任务  $t_i$  的平台  $p_m$ , 由于每次只能处理一个任务, 在处理完任务  $t_i$  后只能被分配到某一个任务, 而不能同时被分配处理多个任务, 即

$$\sum_{j=0}^N x_{ijm} \leq w_{im} \quad i=1, 2, \dots, N; m=1, 2, \dots, K \quad (2)$$

由于任务间的顺序关系, 任务  $t_i$  的处理开始必须在其所有前导 ( $pr(t_i)$ ) 任务处理完毕之后。因此, 存在顺序关系的任务的处理时间存在如下约束:

$$\begin{cases} s_j - s_i \geq D(t_i) \\ a_{ij} = 1 \\ i, j = 1, 2, \dots, N \end{cases} \quad (3)$$

(3) 式中  $D(t_i)$  为任务  $t_i$  的处理时间。

平台  $p_m (1 \leq m \leq K)$  在处理完任务  $t_i$  后被分配处理任务  $t_j$  时, 由于任务的处理需要处理这一任务的所有平台都达到任务区域。显然执行这一任务的所有平台不可能同时到达, 先到达的平台需要等待。因此, 平台  $p_m$  开始处理任务  $t_j$  的开始时间  $s_j$  不小于平台  $p_m$  到任务  $t_j$  区域的时间, 即

$$s_j \geq s_i + x_{ijm} \cdot \frac{d_{ij}}{v_m} + D(t_i) \quad (4)$$

$i, j=1, 2, \dots, N; m=1, 2, \dots, K$

式中  $D(t_i)$  为任务  $t_i$  的处理时间,  $d_{ij}$  为任务  $t_i$  与任务  $t_j$  间的空间距离:

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad i, j=1, 2, \dots, N$$

$(x_i, y_i), (x_j, y_j)$  分别为任务  $t_i$  与任务  $t_j$  的地理位置。

综合(3)和(4)式, 记  $Y'$  为所有任务完成时间的上界(一般设置为较大的值), 则作战平台在任务的分配以及任务间的顺序关系约束可如下式描述:

$$s_j \geq s_i + D(t_i) + x_{ijm} \cdot \left( \frac{d_{ij}}{v_m} + a_{ij} \cdot Y' \right) - a_{ij} \cdot Y' \quad (5)$$

$i, j=1, 2, \dots, N; m=1, 2, \dots, K$

当任务  $t_i$  与任务  $t_j$  之间存在顺序关系  $a_{ij}=1$ , 且  $x_{ijm}=1$  时, (5)式就描述了平台在分配过程中的等待行为, 即(4)式; 当  $a_{ij}=1$ , 且  $x_{ijm}=0$  时, (5)式可进一步表示如下式:

$$s_j \geq s_i + D(t_i) - a_{ij} \cdot Y', \quad i, j=1, 2, \dots, N; m=1, 2, \dots, K$$

如果设置任务完成的初始值  $Y'$  较大, 上式显然是成立的。

任务  $t_i$  的成功处理的条件是被分配处理这一任务的所有平台资源的能力不小于任务  $t_i$  的需求  $R_i$ , 即

$$\sum_{m=1}^K p_{Cm} \cdot \omega_m \geq r_i \quad i=1, 2, \dots, N; l=1, 2, \dots, n \quad (6)$$

式中  $n$  为功能资源类型(案例中  $n=8$ )。令所有任务全部完成的时间为  $Y$ , 则对任意任务的处理的时间约束下式总能成立:

$$Y \geq s_i + D(t_i) \quad (7)$$

综上所述, 平台资源在任务上的聚集过程可以描述如下式:

$$\min Y \quad (8)$$

$$\begin{cases} \sum_{j=0}^N x_{ijm} - \omega_m = 0; \sum_{j=0}^N x_{ijm} \leq \omega_m \\ s_j \geq s_i + D(t_i) + x_{ijm} \cdot \left( \frac{d_{ij}}{v_m} + a_{ij} \cdot Y' \right) - a_{ij} \cdot Y' \\ \sum_{m=1}^K p_{Cm} \cdot \omega_m \geq r_i \\ Y \geq s_i + D(t_i) \end{cases}$$

(8)式中  $N, K, l$  分别为子任务总数、可获取的平台资源数量(全局资源)和功能能力类型(案例中  $1 \leq l \leq 8$ )。

### 5.2 状态空间的定义与资源实体聚类过程的求解

(8)式的数学描述是混元线性规划问题, 式中包含了多个连续的二元变量。这一问题的求解已被证明为 NP 难问题, 涉及到一个众所周知的问题集: 当可获取的平台资源仅有一个时, (8)式的求解是一个“货郎担”问题(TSP)和“货郎担”问题的扩展(如时序依赖 TSP, 优先关系 TSP, 等等); 当所有的平台能处理所有的任务时, 这一问题的求解又简化为一个伴随优先关系的多 TSP 问题; 如果任务的处理可在不同平台之间适时分解, 这一问题就是车辆路径规划问题和这一问题的扩展; 如果平台在不同任务地点之间的行进时间远小于对该任务的处理时间(可以忽略不计的话), 此问题就是一个伴随

优先次序约束的多处理器调度问题。

对(8)式的求解可以转化为状态空间的搜索求解。

定义状态空间  $\Phi$ :

$$\Phi = (M, Lt_1, Lt_2, \dots, Lt_K, f_1, f_2, \dots, f_K)$$

其中  $M$  为所选任务集,  $M \subset \{t_1, t_2, \dots, t_N\}$ ,  $Lt_j$  是平台  $p_j$  最后处理的任务,  $Lt_j \in \{0\} \cup M$ ,  $f_j$  是平台  $p_j$  处理完最后的任务到达的时间。

则每个  $\Phi$  状态包含有如下三个方面的含义:

一是任务集  $M$ ;

二是任务集  $M$  中被各平台最后处理的任务集  $Lt_1, Lt_2, \dots, Lt_K$ ;

三是与  $Lt_1, Lt_2, \dots, Lt_K$  对应的平台处理最后任务所到达的时间  $f_1, f_2, \dots, f_K$ 。

基于状态空间  $\Phi$  的定义, 一次完整的任务计划的状态空间  $\Phi$  可表示为( $N$  为任务数量):

$$\Phi = \Phi_1 \cup \Phi_2 \cup \dots \cup \Phi_N, \Phi_m = \{(M, Lt_1, Lt_2, \dots, Lt_K, f_1, f_2, \dots, f_K) \in \Phi, |M|=m\}$$

由此(8)式对平台资源聚集过程的数学描述可简化为

$$Y = \min_{(M, Lt_1, Lt_2, \dots, Lt_K, f_1, f_2, \dots, f_K)} \max\{f_1, f_2, \dots, f_K\} \quad (9)$$

状态  $\Phi_m$  到  $\Phi_{m+1} = \{(M', Lt'_1, Lt'_2, \dots, Lt'_K, f'_1, f'_2, \dots, f'_K) \in \Phi, |M'|=m+1\}$  的转换通过在任务集中选择  $M$  中没有的任务分配到  $M'$ , 该任务的所有前导任务都必须在  $M$  中, 否则不能被选入分配。如果选入分配的任务为  $t_j$ , 且被分配平台组  $G_P(t_j)$  来处理, 记  $t_j$  的前导任务集为  $Pr(t_j)$ , 则新的状态可用(10)式表示:

$$\begin{aligned} M' &= M \cup \{t_j\} \\ Lt'_i &= \begin{cases} Lt_i, & \text{if } p_i \notin G_P(t_j) \\ t_j, & \text{if } p_i \in G_P(t_j) \end{cases} \\ f'_i &= \begin{cases} f_i, & \text{if } p_i \notin G_P(t_j) \\ D(t_i) + \max_{r \in Pr(t_j)} \{a(r)\} \cdot \max_{p_z \in G_P(t_j)} (f_i + d_{iz}/v_z) & \text{if } p_i \in G_P(t_j) \end{cases} \end{aligned} \quad (10)$$

(10)式中  $v_z$  为处理任务  $t_j$  的平台  $p_z$  从任务  $Lt_i$  转到任务  $t_j$  的速度,  $a(r)$  表示任务  $r$  处理完到达的时间,  $D(t)$  为任务  $t$  的处理持续时间。

$\max_{p_z \in G_P(t_j)} (f_i + d_{iz}/v_z)$  表示任务  $t_i$  的处理必须等到所有处理该任务的平台到达任务  $t_i$  的处理地点时才能开始进行任务的处理。

由此, 求解(8)式的问题就转换为对(10)式状态空间的搜索, 缩小状态空间, 寻找最优解。

### 5.3 基于任务的资源裁剪算法

由于案例中作战平台资源与任务需求的匹配需要多维变量的测度, 同时, 任务需要不同平台资源的协同处理, 对(10)式状态空间的搜索求解是复杂的调度搜索问题, 对这一问题解决的解决本文采用了 MDLS 算法<sup>[8,9]</sup>。

MDLS 算法可以分为两个关键步骤:

(1) 从可选任务集中选择可被处理的任务。可选任务集是所有前导任务已被处理的任务集合。在可选任务集中选择准备处理的任务需要确定任务选择的优先权函数, 对优先权的确定 MDLS 算法分析比较了关键路径算法(CP 算法)、层次分配算法(LA)和长度加权算法(WL)。三种算法虽然在可选任务集中确定任务优先权的方法不同, 但其计算复杂性一样, 都是任务图边的函数  $O(L)$ ,  $L$  为任务图边的数量。

(2) 选择最佳平台组执行需要处理的任务。如何选择最

佳平台或平台组去执行需要处理的任任务是搜索最佳平台资源-任务调度方案的关键。在执行任务最佳平台组的选择过程中,MDLS算法依据了最小化平台资源消耗和平台转移路径的原则,依据这一原则确定了平台组选择的优先权函数,从而根据可获取的平台优先权来选择平台组成执行某一任务的平台集合。

不同的优先权函数导致不同的求解结果,MDLS算法关于平台组选择的优先权函数有如下几种选择:

$$V_m(m) = \frac{B(m,i)}{BR(m) - B(m,i)} \quad (11)$$

$$V_2(m) = \left( s_{l(m)} + t_{l(m)} + \frac{d_{l(m),i}}{v_m} \right) \frac{B(m,i)}{BR(m) - B(m,i)} \quad (12)$$

$$V_3(m) = s_{l(m)} + t_{l(m)} + \frac{d_{l(m),i}}{v_m} + \frac{B(m,i)}{BR(m) - B(m,i)} \quad (13)$$

式中  $l(m)$  为平台  $m$  在当前时间最后处理的任任务,  $s_{l(m)}$  是平台  $m$  处理的最后任任务的开始时间,  $t_{l(m)}$  是平台  $m$  处理最后一个任任务的持续时间,  $d_{l(m),i}$  是任任务  $l(m)$  到当前选出的待处理任任务  $i$  的距离,  $B(m,i)$  是处理当前任任务  $i$  时平台  $m$  所提供的能满足任任务  $i$  的各类资源的总和,  $BR(m)$  就是平台  $m$  对就绪任任务集(可进行资源分配的所有任任务)中任任务的资源满足程度,即

$$B(m,i) = \sum_{k=1}^L \min(s_{ik}, r_{ik}), BR(m) = \sum_{i \in \text{READY tasks}} B(m,i) \quad (14)$$

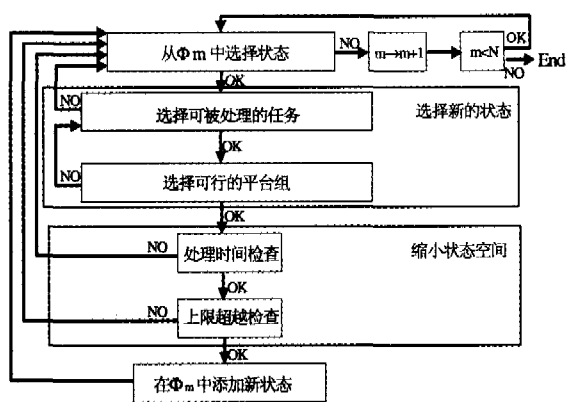


图3 平台资源裁剪算法流程

MDLS算法流程如图3所示。图示过程包括了任任务选择、平台组选择以及任任务处理的时间检查和更新,直到所有任任务被处理完毕算法流程结束。算法的输出包括:平台-任任务分配方案,即分配变量  $w_m$ ; 平台行进的路径,即转移变量  $x_{ijm}$ ; 所有任任务的开始时间和结束时间;以及所有任任务完成所需要的总的任时间  $Y$ 。

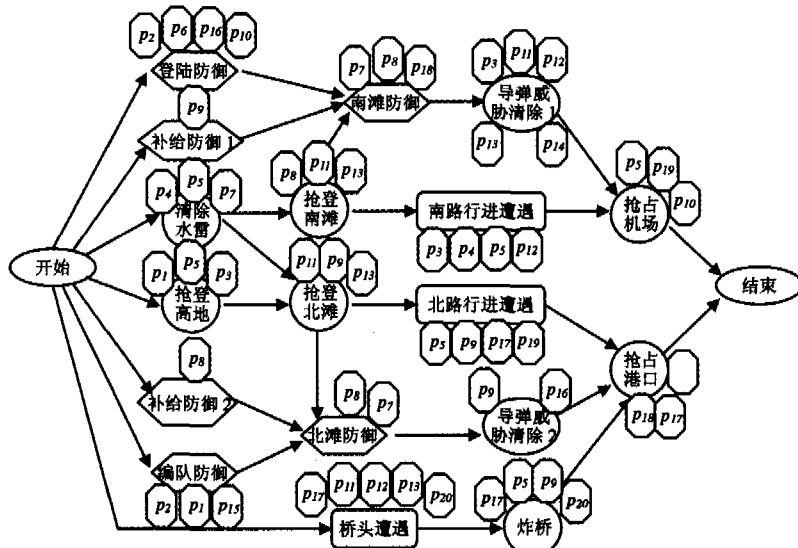


图4 作战平台资源基于任任务的裁剪结果

**结论与讨论** 采用(13)式所确定的平台组选择优先权函数,对本文联合战役全局资源基于任任务的裁剪结果如图4所示。

本文建立了战役全局资源产生基于任任务的战术资源视图的框架模型,在该框架模型中,通过对作战平台资源和作战任任务的定量描述,建立了基于任任务的战术视图生成过程的数学描述,在数学描述的基础上,对任问题的求解构建了作战平台资源动态分配执行任任务的状态空间概念以简化任问题的求解。

对本文任问题的进一步深入研究一方面是采用进化算法优化求解过程,另一方面是在本文结论(即基于任任务战术视图裁剪)的基础上进一步研究基于任任务的作战编成裁剪。

**参考文献**

- 1 Alberts D S, Hayes R E. Power to the Edge: Command and Control in the Information Age. CCRP(Jun 03); 14~15
- 2 Alberts D S, Garstka J J, Stein F P. Network-Centric Warfare: Developing and Leveraging Information Superiority. CCRP, 2nd ed. (Jul 02)

- 3 Mittu R, Segaria F. Common Operational Picture (COP) and Common Tactical Picture (CTP) Management via a Consistent Networked Information Stream. Naval Research Laboratory
- 4 Selinger M. DoD needs joint task force for C2 programs, CSIS says. Aerospace Daily. Retrieved March 23, 2004, from Lexis-Nexis NDU Library
- 5 Carolan T, MacMillan J, Entin E B, et al. Integrated Performance Measurement and Assessment in Distributed Mission Operations Environments: Relating Measures to Competencies. In: Proc. of the 25th Interservice/Industry Training, Simulation & Education Conference, Orlando, FL, Dec. 2003
- 6 Carroll E B. The formulation and testing of schemes to allow three-dimensionally consistent manual alteration of NWP fields. Forecasting Systems Technical Note No. 3, UK Meteorological Office, March, 1996
- 7 Decker K S, Lesser V R. Generalizing the partial global planning algorithm. Int. Journal of Intelligent and Cooperative Information Systems, 1992, 1(2)
- 8 Fikes R E, Nilsson N J. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 1971, 2(3-4); 189~208
- 9 Levchuk G M, Levchuk Y N, Luo L, et al. Normative Design of Organizations - Part II: Organizational Structure. in IEEE Transactions on SMC, 2002, 32(3); 360~375