文章编号:1671-4598(2023)09-0207-06 DOI:10.16526/j.cnki.11—4762/tp.2023.09.030 中图分类号:TP15 文献标识码:A

基于聚类与时间耦合执行序列的任务分解方法

龚 雪[1],彭鹏菲[1],姜 俊[2]
(1.海军工程大学 电子工程学院,武汉 430033;
2.海军工程大学 作战运筹与规划系,武汉 430033)

摘要:针对复杂任务分析中的高度耦合任务不易分解且需重构排序的问题,提出了一种基于聚类分析与改进时间一耦合执行序列的自适应任务分解方法;在矩阵最值遴选模型和任务序列转移策略相结合的基础上,设计了基于中间任务序列的任务矩阵分割算法,基于中间任务序列的任务矩阵分割算法很大程度上能够灵活处理离散任务序列并入的问题;并进一步采用粒度自主循环调整机制,此机制能够有效的解决任务粒度分解不均问题,并不断的进行粒度调整重置,从而使得任务粒度自主分解,最终实现了复杂任务的自适应解耦分析;仿真验证结果表明,该方法能够有效的在实现任务粒度自主设计的基础上,进一步实现复杂任务的解耦及序列重构,在作战任务分析领域具有很好的推广应用前景。

关键词:任务分解;粒度自主循环;矩阵最值遴选模型;序列重构;聚类分析

Task Decomposition Method Based on Clustering and Time Coupling Execution Sequence

GONG Xue[1], PENG Pengfei[1], JIANG Jun[2]
(1. Naval University Of Engineering, Wuhan 430033, China;
2. Department of Operational Operations and Planning, Naval Engineering University, Wuhan 430033, China)

Abstract: Aimed at the problem that highly coupled tasks in complex task analysis are not easily decomposed and need to be reordered, an adaptive task decomposition method based on the cluster analysis and improved time-coupled execution sequences is proposed. Based on the combination of the matrix optimal selection model and task sequence transfer strategy, a task matrix partitioning algorithm based on intermediate task sequences is designed to largely handle the problem of merging discrete task sequences flexibly; The autonomous circular adjustment mechanism of granularity is further adopted to effectively solve the problem of uneven task granularity decomposition and continuously reset the granularity adjustment, thus making the task granularity decomposition autonomously, and finally realizing the adaptive decoupling analysis of complex tasks. The simulation results show that on the basis of the autonomous task granularity design, the method can effectively achieve the decoupling and sequence reconstruction of complex tasks , and it has a very good prospect of application in the field of combat mission analysis.

Keywords: task decomposition; granularity autonomous cycle; matrix minimum selection model; sequence reconstruction; cluster analysis

0 引言

作战任务分析[1]是作战任务规划[2-3]的基础和前提,多军种联合复杂任务规划已成为军事指挥决策领域的重点研究方向。因此,众多学者对复杂任务分析方法进行了更加深入的研究,以进一步推动作战任务规划应用技术的发展。例如,传统任务分析方法一空间搜索法,周凌超[4]通过改进模拟退火算法对导弹目标进行分析规划,得到一个较好的导弹目标的分配方案。仿生物学方法一遗传算法,闫玉铎[5]着力分析武器目标的内在联系,应用改进遗传算法,得到武器目标较优的任务分配方案。智能规划方法一基于分层任务网络规划,胡晓峰[6]等人针对决策问题智能化的分析,得出游戏博弈和作战指挥的密切的内在联系。

虽然,以上方法能够解决任务分析领域中的特定问题,但是目前复杂任务分析仍然面临的许多难点问题,这些问题主要体现在:一是任务分析过程中影响因素过多构造模型困难[7]。二是建立的任务分析模型过于繁琐,时效性较差。三是任务分析模型完备性较差无法实时对情况进行分析。

收稿日期:2022-11-01; **修回日期**:2023-01-16。
基金项目:国家重点研发计划项目(2017YFC1405205);海军工程大学科研发展基金自主立项项目(425317S107)。
作者简介:龚 雪(1998-),女,硕士研究生。
通讯作者:彭鹏菲(1977-),男,博士,副教授,硕士生导师。
引用格式:龚 雪,彭鹏菲,姜 俊.基于聚类与时间耦合执行序列的任务分解方法[J].计算机测量与控制,2023,31(9):207-212,227.

本文针对作战任务分析中的高度耦合[8-9]任务不易分解且需重构排序的问题[10]，提出了一种基于聚类分析与改进时间－耦合执行序列的自适应任务分解方法[11]。该方法的主要思想是：对任务矩阵中的每一个任务进行定量分析，再利用聚类后的改进时间—耦合度的执行序列优选方（TATC）耦合度分解算法对最小粒度的任务集解耦运算进行最小粒度的耦合任务集解耦运算。最终，得到一个耦合度较低之后的任务序列执行层次结构图。最后，通过仿真实验，验证了该方法对解决复杂任务的解耦及序列重构难点问题具有较好的运用效果，在复杂任务分析及规划领域应用前景广阔。

1 任务分析的问题描述

1.1 作战任务分析

任务的多样性、任务间的复杂性与不确定性等不确定因素限制了作战任务的详细规划，因此在任务规划前需进行任务分析，将强耦合任务分解，优先级高的任务优先执行。

本文针对作战任务分析中高耦合性任务分解问题[12]，实现高耦合性任务粒度自动分解至最佳粒度的解耦合任务集的解耦[13-15]操作思路，提出了基于自动粒度控制的循环解耦的方法。创新改进了基于任务集解耦算法，本文运用的解耦算法是改进的时间—耦合度的执行序列优选方法。得到了清晰的基于粒度分解的解耦任务集，任务集内子任务执行序列[16]以及解耦任务集中子任务可执行系数。

1.2 任务协同关系类型

因为任务与任务之间存在着繁琐而复杂[17]的关联关系，每个作战任务之间相互影响并彼此相互作用，即对外表现出协同关系。则这种对外的协同模式表现出三种模式[18]（假设有两个作战任务 T_s 和 T_h）：

1）依赖型任务：任务 T_s 对任务 T_h 具有单一方向的传递关系并且 T_s 依赖 T_h 的输出才能执行如图 1 所示。

2）独立型任务：T_s 和 T_h 没有彼此相互作用的信息交互，T_s 和 T_h 各自独立的在不同轨道单独运行如图 2 所示。

3）耦合型任务：T_s 和 T_h 具有双向信息依赖关系也即 T_s 需要 T_h 提供的信息流来作为输入且 T_h 需要 T_s 提供的信息流作为输入，任务 T_s 和任务 T_h 需要通过多次的彼此相互作用才能完成的耦合任务如图 3 所示。

其中图 1～3 中的“＊”应在［0，1］取值，表示为任务间耦合系数。

任务间越是信息交互频繁，则耦合性越高，复杂程度越高，则进行任务分析就越发困难；在紧急的情况下，人脑计算能力有限，且经验性决断过多，因而在很多情况下无法做出决定性的选择，有可能会决断失误。因而任务分析显得尤为重要，所以本文从多种复杂信息交互的角度，设计自适应性任务粒度调整机制，最终达到解耦合的目的。

1.3 任务协同关系表达

本文是基于任务矩阵来进行分类耦合关联任务块，则任务间复杂关系表示如图 1，图 2，图 3 所示。

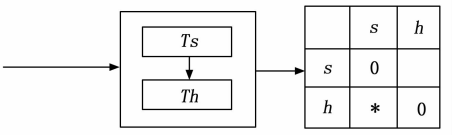

图 1 依赖型任务

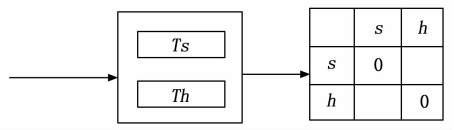

图 2 独立型任务

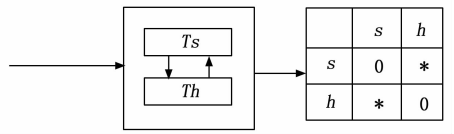

图 3 耦合型任务

1.4 耦合任务问题描述

信息耦合指的是不同优化问题的求解模块或算法的内在机理上存在信息的的传递、调用、影响的作用。有多个算法组成的系统，在算法之间的关系上，可以对系统的信息耦合程度进行量化。针对任意两个问题进行求解时，若是这两者的运行过程相互独立，则两个算法之间不存在信息耦合；若是两者之间存在共享输入，两者之间为松弛耦合状态；如果两者之间存在影响算法运行过程的信息，则两者为紧密耦合状态。

则针对以上叙述，假设在一个作战任务集中，有多个强耦合任务存在。定义任务集合为 $T=\{T_1,T_2,\cdots,T_s\}$，其中 T_s 表示的是任务集合中的第 s 个任务。

若任务分解粒度过小，任务执行难度变大，所需代价太高。若任务分解粒度大，任务整体的执行效果差。因此，对于解耦问题，最关键的一步是设置合适的任务分解粒度。

若一个任务集合中强耦合任务过多，执行强耦合任务时难度系数较高，因此需要进行任务分析时，需要对耦合任务集进行解耦。则耦合任务模型可描述如图 4 所示。

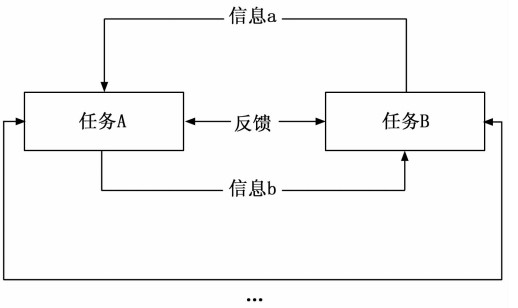

图 4 耦合任务信息交互图

2　任务分析模型构建

在设计任务协同的过程中，设计时不会考虑任务之间的耦合关系，但是在实际情况中多个任务之间存在大量的信息交互且紧密耦合，解耦合就是将任务间的关系斩断，并且研究任务之间存在的耦合关系。本文基于这种特性，构建了任务分析模型。主要分为两个方面：任务分析基本框架的设计和任务分析建模。任务分析基本框架主要是在任务分析模型的基础进行粒度分解和循环解耦的过程，最重要的一环是循环解耦的算法设计。

2.1　任务分析基本框架

本文基于任务间的关联性构建任务关联矩阵，在已知任务间关联关系的基础上设计任务连通效应矩阵。通过连通效应矩阵进行聚类［19－20］分析，设置初始粒度为 3（也即将其作为聚类分析的初始分类）。依据初始粒度进行耦合数据集解耦。在解耦之后，计算任务集间的 jaccard 系数 J，若 $J>0.3$，初始粒度增加一个单位，并进行以上循环。若 $J<0.3$，跳出循环，得到解耦后的任务集并计算任务集的子任务可执行性。

循环解耦大致步骤为：

1）初始化参数，令 $J=1$，初始分解粒度 $G=3$。

2）若 $J>0.3$，则进入循环：聚类分析得到任务集合 TCE，将标记任务分类的标签存入 TCO 矩阵。

（1）循环取 TCO 矩阵中标记系数：

①依据标记类别将分类集分类转存。

②引入了基于中间任务序列的任务矩阵分割算法，通过任务执行周期对连通效应矩阵唯任务分割（也即若该任务属于该耦合任务集则效应连通矩阵对其保留反之删除）。

③TATC 解耦合操作，得解耦之后任务集。

④计算每个任务集的平均 jaccard 系数。

（2）对基于粒度分解的每个任务集进行维度查询，若维度小于 3，则将解耦后删除的任务集与之合并，生成一个新的任务集。并对新合并的任务集进行子任务可行性系数的计算，若可行，则保留；否则对该任务集进行再次解耦。

（3）计算基于粒度分解的任务集平均 jaccard 系数，若大于 0.3 循环继续，粒度增加 1，若小于 0.3，退出循环。

3）计算解耦之后任务集可执行性矩阵，若不符合要求，粒度加一；对不符合要求的任务集单独解耦。循环解耦的简易流程图如图 5 所示。

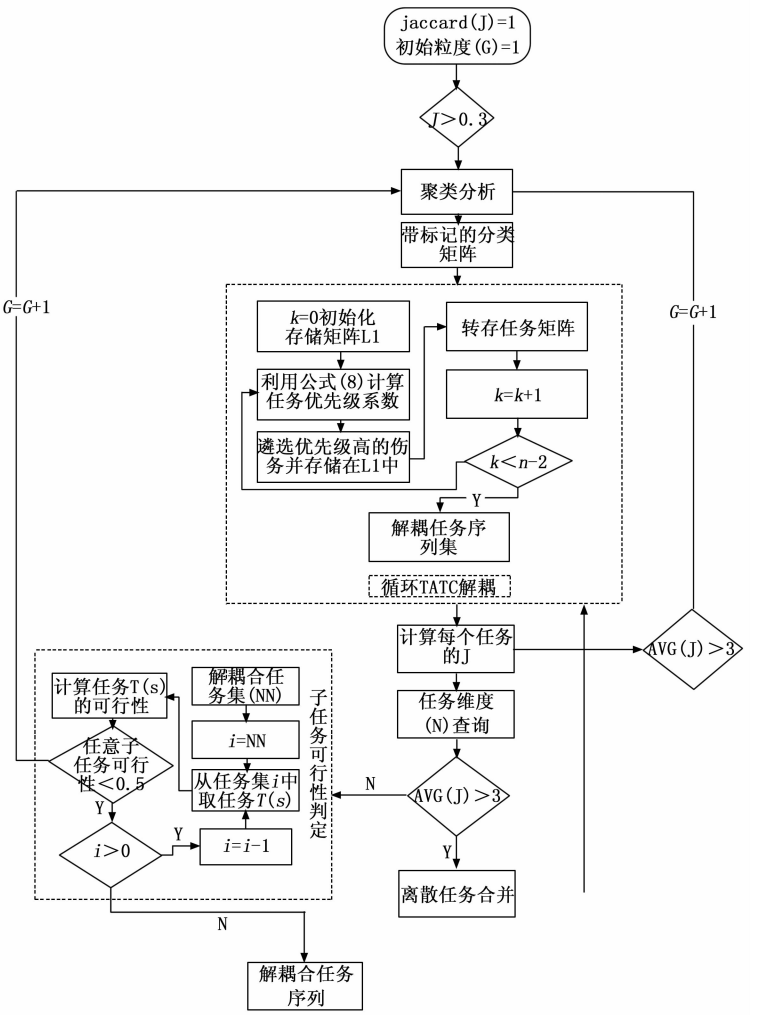

图 5　循环解耦的简易流程

2.2　任务分析建模

2.2.1　任务关联矩阵

本文首先设计任务结构矩阵，计算任务 T_s 与其他任务 T_h 的关联系数，并将生成的关联系数存入矩阵得到相关系数矩阵；将上一步得到的系数矩阵通过公式（1）进行任务间相关性进行比对，最后得到任务 T_s 和 T_h 的相关的显著程度的对比，从而得到强耦合任务关联矩阵。

$$t_s=\frac{\sqrt{R_s(n-2)}}{\sqrt{(1-R_s)^2}} \tag{1}$$

其中：t_s 为任务 T_s 服从自由分布 t 的概率，R_s 则为任务 T_s 的相关系数。最后计算出每一个任务与其他任务的相关联的相关系数矩阵。

2.2.2　任务连通效应矩阵

在任务关联矩阵的基础之上建立基于设计结构矩阵（DSM，design structure matrix）模型[21]得到任务连通效应矩阵。为精确反映任务之间相互作用的强弱程度，本文采用上文任务间相关系数矩阵（RF）来定量描述任务之间相互作用的强弱程度。在现实意义中，该矩阵反应了在不同程度下迭代导致的任务之间相互影响概率。确定 M 个指标 $RF_1(i,j)$，$RF_2(i,j)$，$\cdots$，$RF_m(i,j)$，采用公式（2）进行相乘效用函数法计算，并对任务相关系数矩阵取 λ（$0<\lambda$）截集，得到布尔型 BRF_{ij}。

$$RF_{ij}=\sqrt[m]{RF_1(i,j)\times RF_2(i,j)\times\cdots\times RF_m(i,j)} \tag{2}$$

由于影响因素不同，则 λ 取值也应该不同，因而可得到不同 λ 强度下的 BF_{ij} 矩阵。主对角线上的元素代表了任务

本身产生的影响，其相关程度是最强的，得 $BF_{ii}=1$。若令 $\lambda=0.5$，得到 $\lambda=0.5$ 相关效应矩阵，如当 $RF_{ij}<0.5$ 时，$BF_{ii}=0$ 表示 T_i 和 T_j 之间无相互作用效果的影响；当 $RF_{ij}=0.5$ 时且 $RF_{ij}>0.5$，$BF_{ii}=1$ 表示 T_i 和 T_j 之间有相互作用。其中下标 i 和 j 表示 [1，N]，N 表示 RF 矩阵的维度。

2.2.3 计算可达矩阵

利用关联任务分离算法对连通效应矩阵进行运算，将任务连接通效应矩阵中包含的多个独立的耦合任务块进行分解，并用粗粒度关联任务对连接的大任务模块进行分解。粗粒度的分解实质是一个模块化的聚类过程，它把多个强耦合的耦合任务作为一个整体，通过识别任务模式和重组耦合任务，把各个耦合任务与外界联系转化为耦合的任务集合。

在图论的观点上，为了确定耦合任务块中包含各种任务间数据流相互作用而形成的图的数据流环路，本文采用图论中关键路径的相关算法设计算法模块. 由于任务连通效应矩阵的特殊性可被看作图的邻接矩阵，利用任务连通效应矩阵的幂运算来搜索所有的数据流环，从而初步确定耦合任务块，为后续初始粒度的设置做铺垫。

定义 1：和算子 $\forall$

矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的逻辑和为：

$$\boldsymbol{P}\forall\boldsymbol{Q}=\boldsymbol{M},m_{ij}=\max(p_{ij},q_{ij}) \tag{3}$$

定义 2：乘算子 $*$

矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的逻辑乘为：

$$\boldsymbol{P}*\boldsymbol{Q}=\boldsymbol{M},m_{ji}=\min(p_{ij},q_{ij}) \tag{4}$$

定义 3：算子 $\otimes$

$$\boldsymbol{M}=\boldsymbol{P}\otimes\boldsymbol{Q}=\vee_{k=1}^{n}(p_{ij}\wedge q_{ij})_{n\times n} \tag{5}$$

则根据路径搜索算法得强连通图，并通过矩阵信息的遍历，建立可达矩阵。

则可达矩阵 $\boldsymbol{V}$：

$$\boldsymbol{V}=\vee_{i=1}^{n}P^{(i)}=P^{(1)}\vee P^{(2)}\vee\cdots\vee P^{(N)} \tag{6}$$

通过公式（6）计算可达矩阵 P，根据上文所述根据任务连通效应矩阵来定量反映任务间的联系强弱程度，用 [0，1] 数值来诠释它们之间的依赖关系的强弱。通过聚类分析得到初始耦合任务集并将其作为迭代的初始耦合任务集。首先设置的初始粒度为 3 也即将其作为聚类分析的初始分类，然后，对其进行耦合数据集的解耦操作。解耦之后，计算任务集间的 jaccard 系数 J，若 $J>0.3$，初始粒度增加一个单位，并进行以上循环。若 $J<0.3$，跳出循环，得到解耦后的任务集并计算任务集的子任务可执行性。

3 耦合任务集分解

当 T_s 任务在 T_h 之前执行时，假设 T_h 之后没有任务来执行，那么 T_h 将促使 T_s 返工，和公式（6）的返工概率一样，通常促使 T_s 不止一次返工，因此会不断地产生返工概率。假设每个任务的执行周期为 $S_i=\{S_1,S_2,\cdots,S_m\}$，促使 T_s 返工的概率 $D_i=\{D_1,D_2,\cdots,D_m\}$，则 T_s 返工的期望总和为：

$$R_s=\sum_{i=1}^{n-k}S_iD_i \tag{7}$$

因为 T_s 对其他任务的返工则即其他任务对 T_s 返工的返工概率假设为 $DD_i=\{DD_1,DD_2,\cdots,DD_m\}$，$DD_i$ 的计算方式是：假设的 T_s 对其他任务的返工概率和其他任务对 T_s 的返工概率是相斥的，则其他任务对 T_s 返工的返工期望为：

$$R_{s1}=\sum_{i=1}^{n-k}S_iDD_i \tag{8}$$

综合考虑时间周期对耦合任务块的影响，为此引入基于时间－耦合度的解耦算法，在时间－耦合度的解耦算法中设置计数器，进行任务间优先级计算并对耦合任务块内的子任务进行排序。

则改进的时间－耦合度的解耦算法如下。

Step 1：令 $k=0$，T 为任务集合，L_1 为空矩阵其作用是储存每次得到的最大任务的优先级的任务序号。

Step 2：对于每个耦合任务集合中的某一任务 T_s，计算其优先级的系数。

$$Sort_i=\frac{\sum_{i=1}^{n-k}S_iD_i}{\sum_{i=1}^{n-k}S_iDD_i} \tag{9}$$

Step 3：将矩阵最值遴选模型和任务序列转移策略相结合，即令 $Sort_{max}=max\ [Sort_i]$，$Sort_{max}$ 对应的任务 T_i 存到矩阵 L_1 之中。

Step 4：将解除耦合的任务序列存入 L_2 矩阵，并将剩下的任务的序列存入 L_3 矩阵。

Step 5：$k=k+1$，若 $k<n=2$，转步 Step 2，否则，算法结束。

最后得到按优先级排序的该耦合模块或者该任务集合中子任务的执行顺序。

4 子任务可行性分析

对解耦合后得到的子任务集进行可执行性分析的判定，只有任务在解耦合之后，重新得到的任务集能够执行，则判断解耦合成功，下面是子任务可执行性判定算法。

首先，专家对各个解耦之后任务集中单个任务进行描述。用 $RG(k)_i$ 来描述第 K 个任务集，第 i 个任务。$RG(k)_i$ 的矩阵元素是由任务的功能和任务属性序列这两个重要的影响因素构成，通过两个因素初步确定子任务的可行性。引入用户对该任务的侧重水平，定义模糊性语言变量集合 $I_x=$ {不重要，无要求，基本重要，相较重要，及其重要} 来进行表示，并以数值集合 {0，0.3，0.5，0.7，1.0} 的数值分别量化。设置子任务可行度可用公式（9）表示为：

$$fesibleD(k)_i=\frac{\sum_{i=1}^{n}RSQ(k)_iI(k)_i}{\sum_{j=1}^{n}I(k)_j} \tag{10}$$

式中，$RSQ(k)_i$ 表示任务执行者对第 K 个任务集中的第 i

个任务的属性能否符合要求，若数值为 0 则表示不符合，否则数值为 1 就表示符合。

设定一个阈值 λ，如果存在执行者使 $D>\lambda$，则说明分解合适；反之，则说明分解粒度不适合，需进一步对任务集进行分解。阈值的设置需综合分析子任务重要性、创新程度及任务执行者状况等内容，取值范围为 (0，1.0]，本文取值为 0.5。

5　案例仿真

在作战过程中，任务分析往往是及其最基础的一环。并针对基于矩阵的作战任务建模及重组问题，进行应用拓展。先根据任务在时间、空间和资源上的相互制约和依赖关系确定 15 个任务指标设计原始任务结构矩阵 F，如式 (11) 所示，可以明显看到任务在量化过程中，任务重要程度的指标以及任务间的信息交互程度的表现：

$$\boldsymbol{F}=\begin{bmatrix}
1 & 0.2 & 0 & 0.4 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0.8 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0.8 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0.8 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1\\
0 & 0.1 & 0.7 & 0.2 & 0 & 0.3 & 0.1 & 0 & 0.5 & 0 & 0 & 0.3 & 0 & 0 & 0.1
\end{bmatrix} \tag{11}$$

再次根据 (1) 式和 (2) 式确定任务关联矩阵 F_q 如式 (12) 所示，式中各个任务的通路已然可以通过数据可以表现出，例如任务一合任务二连通且任务一与任务五不连通：

$$\boldsymbol{F}_q=\begin{bmatrix}
1 & 1 & 0.999 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.534\,52 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.534\,52 & 0 & 0 & 0 & 0\\
0.999 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.534\,52 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.534\,52 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.534\,52 & 0.534\,52 & 0.534\,52 & 0.534\,52 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix} \tag{12}$$

将 λ 定义为 0.5 得到 λ-任务连通效应矩阵 B 如式 (13) 所示。

$$\boldsymbol{B}=\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix} \tag{13}$$

根据关键路径幂乘算法调用公式（6）进行可达矩阵的计算，得到矩阵 $\boldsymbol{P}$ 如式（14）所示：

$$\boldsymbol{P}=\begin{bmatrix}1&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\1&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\1&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\1&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&1&1&0&0&0&0&1&0&0&1\\0&0&0&0&0&1&1&0&0&0&0&1&0&0&1\\0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\1&1&1&1&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&1&0&0&0&0&1&0&0&1\\0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&1&1&0&0&0&0&1&0&0&1\end{bmatrix}\tag{14}$$

设置最小的分解粒度为 3，也即进行聚类分析分成生成 3 类将要被解耦合的耦合集，其次将这三类分别用数组转存并分别通过引入任务时间周期 $D=[0.1\ 0.2\ 0.8\ 0.3\ 0.1\ 0.4\ 0.2\ 0.1\ 0.6\ 0.1\ 0.1\ 0.4\ 0.1\ 0.1\ 0.2]$，将引入改进的时间－耦合度的解耦算法进行计算操作，将解耦合后的任务集进行转存，同时进行计算子任务间的 jaccard 系数。将 jaccard 系数进行平均算法操作得到各个解耦合后的任务集的 jaccard 系数，并计算子任务可行性系数，最终实验结果的可行性系数如表 1～3。

表 1 初始解耦合任务集

初始解耦合任务集	任务序列
led_1	10
led_2	[5;8;9;13;14]
led_3	[6;7;12;15]
led_4	[1;2;3;4;11]

其中图中的 led_i 其中 $i\in\{1,2,3,4\}$，表示初始解耦合任务集合，由表一可以看出初始进行解耦时，任务的粒度划分已然完成并且如表 2 可以看出任务粒度的划分在不断进行优化。

表 2 最终解耦任务集

最终解耦合任务集	任务序列
RG_1	[1,4,10,13,14]
RG_2	[9,8,5]
RG_3	[6;7;12;15]
RG_4	[11;3;2]

其中图中的 RG_i 其中 $i\in\{1,2,3,4\}$，表示最终解耦之后的任务集合。

表 3 解耦任务集子任务可行性系数

任务序列	子任务可行性系数
$fesibleD_1$	[*nan*,1,1,1,0.666 7]
$fesibleD_2$	[1,0.7,0.7]
$fesibleD_3$	[1,1,1,1]
$fesibleD_4$	[0,0.769 2,0.625 0]

其中图中的 $fesibleD_i$ 其中 $i\in\{1,2,3,4\}$，表示最终解耦之后的每个任务集合的子任务可行性系数。

对于解耦任务集子任务可行性系数，本文设置的子任务可执行的阈值为 0.5，大于 0.5 则该子任务可执行否则不可执行。从表 3 中可以看出第一个解耦合之后的任务集中‘NaN’是其任务本身，对其任务本身则不可做评价因为任务本身的可执行距离为 0。则得到最终的任务层次结构图如图 6 示，任务执行次序是从第一层到第四层。其中，第一层中任务执行顺序为：任务 1→任务 4→任务 10→任务 13→任务 14；第二层次的任务执行顺序为：任务 9→任务 8→任务 5；第三层的任务执行顺序为：任务 6→任务 7→任务 12→任务 15；第四层的任务执行顺序为：任务 11→任务 3→任务 2。每个层次间互不干扰，层次间的数据传递不具时效性但是层级顺序不可变。因此，任务链（任务 1→任务 4→任务 10→任务 13→任务 14→任务 9→任务 8→任务 5→任务 6→任务 7→任务 12→任务 15→任务 11→任务 3→任务 2。）依据实际情况，在同一层次的任务可进行次序调整，本文中基于聚类分析与改进时间－耦合执行序列的自适应任务分解方法运用耦合任务分析的思想对任务矩阵中的每一个任务进行定量分析，然后采用聚类方法寻找耦合最紧密的任务，再利用粒度自主循环调整机制运行改进 TATC 算法对最小粒度的任务集进行解耦。最终，得到一个耦合度较低之后的任务序列执行层次结构图，从而解决了复杂任务分析中的高度耦合任务不易分解且需重构排序的问题。

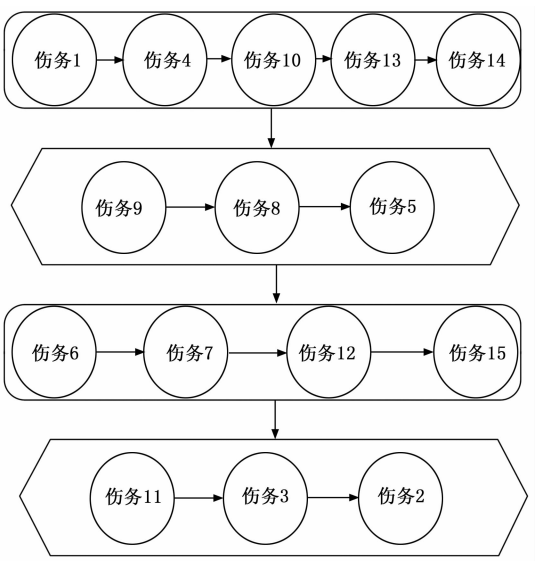

图 6 任务层次结构

（下转第 227 页）