

文章编号:1001-9081(2007)01-0202-03

## 一种新的项目调度解表示方法

黄志宇

(中国科学院 自动化研究所,北京 100080)

(huangzhiyu2004@163.com)

**摘要:**基于解集合的准启发式方法是解决资源约束下项目调度问题的有效方法,解的表示形式一直是这种方法的一个重要研究问题。只有充分利用解的形式和目标函数之间的联系,才可能达到在少数枚举下得到尽可能好的解。详细分析了解空间性质,提出了用额外关系表示一个可行解的方法,给出了这种表示方法的理论依据。并介绍了用该方法产生邻域的方法。

**关键词:**准启发式方法;解空间;邻域;分散搜索算法

**中图分类号:** TP306 **文献标识码:** A

## A new representation for the solution of resource constrained project scheduling

HUANG Zhi-yu

(Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** How to represent a solution of a resource constrained project scheduling is an import problem for the metaheuristic method. Only after understanding the relation between solution representation and objective function adequately, a good solution can be reached within fewer search steps. After analyzing the solution space in detail, a new solution representation based on the extra relation was provided as well as the theoretical basis of this representation. How to make use of the method to create a neighborhood was also introduced.

**Key words:** metaheuristic method; solution space; neighborhood; scatter search

具有资源约束的项目调度可以描述为:给定项目的资源约束,组成项目活动的执行时间、执行时的资源消耗,以及活动间的先后关系约束、开始时间约束,求解项目目标函数的最优解。除特殊说明外,下文中“资源”即指可再生资源,“项目目标”是最小工期。

基于解空间的准启发式方法,因其有效性一直是研究的重点,其中分散搜索算法<sup>[4]</sup>方法由于它直接涉及到解空间的性质,正在和其他算法相结合,成为近来的研究热点。以往的研究在如何表示解空间、解空间距离定义以及解邻域定义上过于依赖解的产生方法。本文以文献[2]中提出的额外关系概念为基础来阐述解的性质以及邻域的构造。

### 1 Active 解的额外关系表示

#### 1.1 关于 Active 解的定义和相关理论

**定义 1** 最早开始时间调度。在不考虑资源冲突的情况下,直接按照问题的其他约束进行调度,而得到的解,称为最早开始时间调度。

**定义 2** Active 解。一个可行解,如果任何一个活动的开始时间在不延迟其他活动的开始时间和满足资源约束的条件下都不能左移,那么这个解是 Active 的。

**定理 1** 最早开始时间调度所对应的解,在第一次资源冲突时间点之前结束的活动的开始时间,在所有的 Active 解中都相同。

**证明** 文献[2]中提到的分支定界方法,相当于隐性枚举了所有的 Active 解。而按照文中的分支方法,在某个时间点只需考虑正在执行的活动,在这个时间点以前完成的活动

对于这个分支的所有子分支都不改变开始时间。对于最早开始时间对应解的第一次资源冲突,它产生的子分支将是产生所有 Active 解分支。而在这个时间点前结束的活动开始时间在所有子分支中不变,所以结论成立。

**引理 1** 一个 Active 解中每个活动都在这个活动的最早开始时间或者其他某个活动的结束时间开始。其中最早开始时间是最早时间调度决定的。

**证明** 假设一个 Active 解,存在某个活动不在上面两种情况下开始,那么它将开始于大于其最早时间并且该时间不是任何一个活动的结束时间,这样可以在保持其他活动时间不变和它本身的开始时间约束条件下,将其左移至少一个时间单位。因为一个活动在开始后消耗资源,而在结束后就释放资源,所以判定一个调度的活动序列是否在资源约束下是可行的,只要计算各个活动的开始时间点处的正在执行的活动是否满足资源约束就行了。设活动原来的开始时间是  $t$ ,活动左移一个时间单位到  $t'$  后,因为没有任何活动在  $t$  处结束,所以  $t'$  处的正在执行活动集合被  $t$  处的正在执行活动集合包含。因为在未移动之前,调度是可行的,所以在原来的活动开始时间  $t$  处,没有资源冲突,就是说在  $t$  处正在执行的活动集合所需的资源小于总资源。这样在  $t'$  处也不存在资源冲突。这与原来的解是 Active 的矛盾。证毕。

**额外关系的添加方法:**如果某个活动  $j$  的开始时间  $s_j$  等于某个活  $i$  的结束时间  $f_i$ ,而且活动  $i$  和  $j$  之间没有先后关系,那么就添加从  $i$  到  $j$  的额外关系。

**定理 2** 如果对一个 Active 解按照上面的方法添加额外关系,当把这些添加关系看成是网络的先后关系后,直接按照

最早开始时间调度,得到的解与原来的 Active 解相同。

证明 首先这个 Active 解是添加额外关系后问题的一个可行解。这是因为添加额外关系后的问题只是比原问题在先后关系上多出了一些约束,其他约束不变。这个 Active 解,在原问题下可行,而根据添加方法,添加的额外关系并没有改变这个 Active 解本身,所以这个 Active 解是添加关系后问题的一个可行解。其次这个解是添加后网络的 Active 解。因为这个解对于原问题是 Active 的,所以对于比原问题具有更严格约束的添加额外关系后的问题,必然是 Active 的。最后在添加额外关系后考虑问题,根据引理 1,这个 Active 解所决定的活动开始时间,不是最早开始时间就是某活动的结束时间。注意到额外关系的添加方法,这里的开始时间都是最早开始时间。所以这个 Active 解对于添加额外关系后问题是最早开始时间调度。根据最早开始时间调度的唯一性,结论成立。

### 1.2 Active 解的额外关系表示

文献[5]指出对于一个调度解的表示可以分成两类: Random-Key(RK)和 Activity-List(AL),它们都对应于一个  $n$  维向量,为了方便用  $X$  表示这个向量,并写成:  $X = \{x_i\}, i = 1, 2, \dots, n, n$  表示网络的节点数目。

在 RK 中,  $x_i$  表示活动  $i$  在调度中被选择的权值大小,权值大的活动如果它可以调度,那么首先调度;而在 AL 中,  $x_i$  表示活动  $x_i$  应该在调度序列的第  $i$  个位置,也就是所有活动的一个偏序排列。

启发式产生一个调度解有两种框架<sup>[5]</sup>: 串行产生框架 SSGS(Serial Schedule Generation Scheme) 和并行产生框架 PSGS(Parallel Schedule Generation Scheme)。按照不同的框架即使是同一个 RK 和 AL 表示,也可能产生不同的调度结果。

RK 和 AL 都能完整地表示解空间,但由于它们与采用何种调度框架相绑定,与其说它们到解空间的映射,不如说是在某种框架下的映射。所以不可能用它们来描述问题本质特征。

从定理 2 可以看出,给定一个 Active 解按照给定的额外关系添加方法,存在着一个额外关系序列和这个 Active 解一一对应。但并不是所有的额外添加关系都能决定一个 Active 解。因为在不知道问题的解的情况下,添加什么样的额外关系是盲目的。如果把所有可能添加的额外关系组合看成一个空间,并称之为额外关系空间,那么额外关系空间将包括解空间。

可以用一个连接矩阵来表示任何两个活动的连接关系,设  $L = \{l_{ij}\}$ , 其中  $l_{ij} = 1$  表示在原来的活动网络有向图中存在活动  $i$  到  $j$  的有向路,  $l_{ij} = 0$  表示不存在这样的路。

从上面的额外关系添加方法可以看出,额外关系只能添加在不存在先后关系的活动对中,也就是矩阵  $L$  中元素值等于零并且这个元素的对角对称位置的元素值也等于零的位置。用数学表达式表示为:

$$l_{ij} = 0 \wedge l_{ji} = 0, i \neq j \quad (1)$$

连接矩阵  $L'$  中满足上述关系的全体位置,决定了所有可能添加的额外关系全体。对于每个可以添加的位置,如果对应的元素值置为 1,  $L'$  其他元素值置为 0, 这样决定了一个矩阵组。而任何一个添加关系,都可以用这个矩阵组来线性表示。这个矩阵组就是额外关系所决定线性空间的基。定义了额外关系空间的基就可以决定任何一个额外关系所对应的坐标,如果两个额外关系坐标分别为  $X_1 = \{x_{1k}\}, X_2 = \{x_{2k}\}$ , 它们的距离定义为:

$$|X_1 - X_2| = \sum |x_{1k} - x_{2k}| \quad (2)$$

这种距离定义方法,衡量了两个解具有不同额外关系的个数。可以近似反映两个解基于额外关系表示的临近程度。具有小距离的两个解可能在相同冲突时间点上选择相近的额外关系,得到的目标函数值也相近。

## 2 基于额外关系空间构造邻域的方法

用准启发式方法一般都涉及到构造一个解邻域,并在这个邻域内进行局部搜索的问题。用 RK 表示的解进行局部搜索时一般采用变更某个 RK 中的值,来实现解的小扰动,进而构造邻域;AL 表示的解,一般采用把某两个活动对换实现解的邻域。第二部分已经指出 RK 和 AL 表示都没有体现出真正解空间的性质,所以在它们基础上构造邻域也很难反映问题的本质。对于额外关系表示的解,可以认为具有相似额外关系的解构成了这个解的邻域。

因为最优解一定是 Active 解,而凡是有效的分支定界方法,都要隐性枚举所有的 Active 解,所以可以把一个 Active 解看成是分支定界枚举树中的叶节点。这里所谓的隐性枚举是指真正的分支定界算法必然要有剪枝算法,剪掉的分支仍然可以得到 Active 解,把这些被剪掉的 Active 解与有效的 Active 解合在一起,一定是整个 Active 解集合。其实是否能够枚举整个 Active 解集合是判断一个分支定界算法是否有效的最基本准则。

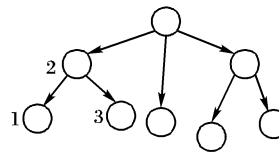


图 1 分支定界树示意图

按照文献[2]产生的分支定界算法的枚举树可以想象成图 1 的样子。图中每个圆圈都代表分支定界时的一个部分调度,每个分支代表选择了不同的额外关系,也就是采用了延迟不同的活动,每分支一次,就相当于在原来的基础上添加了一些额外关系,其中最下层的代表最后得到的 Active 解。对于一个 Active 解,如果能够得到它的上一层分支点(对于图中节点 1 它的上一层就是 2),就可以通过枚举上一层节点处的所有可能分支,而得到这个解邻域内所有的解(对应图中节点 3 和 1 属于同一邻域)。

为了以下叙述不致混乱,称添加的额外关系中作为父节点的活动为额外父节点,子节点的活动称为额外子节点。

构造一个 Active 解的邻域,需要得到在分支定界枚举树中这个解的上层部分调度,正是在这个部分调度的基础上添加了一些额外关系才形成了这个解。根据一个解一般无法精确得到枚举过程,但由于开始时间大的额外子节点比开始时间小的更晚添加到网络,所以可以在当前 Active 解所确定的额外关系中,去掉一些具有较大开始时间额外子节点的额外关系,用剩下的额外关系近似这个解的上层部分调度。设去掉的额外子节点中最小的开始时间为  $t_{mins}$ , 计算剩下的额外关系中子节点开始时间离这个时间最近的值,设为  $t_{near}$ , 把它作为当前 Active 解的上一层部分调度的决策时间。由于去掉的额外子节点可能在当前决策时间之前就是被延迟的,为此采用的判断方法是:如果在被去掉的额外子节点集合中,存在某个节点满足当前决策时间大于它的所有父节点的结束时间,那么这个子节点在当前决策时间以前就被延迟。对于这样的额外子节点,要添加在当前决策时间处结束的活动节点到这个节点的额外关系。这样得到的额外关系也属于上层部分调度。把得到的部分调度的额外关系添加到原来的网络中形成一个新的网络,以这个网络的解空间作为当前 Active 解的邻域。一般情况下,对同一解去掉越多的额外关系新形成网络的解空间越大,这个解的邻域也就越大。

### 3 仿真结果

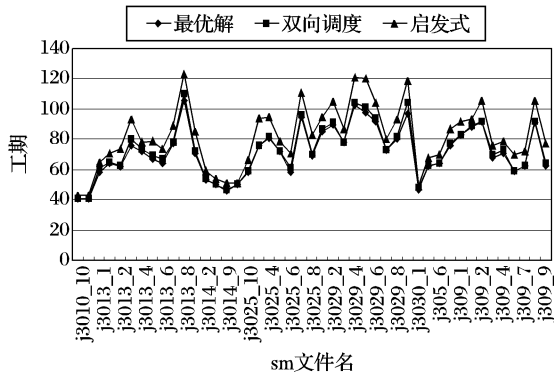


图2 j30 仿真结果

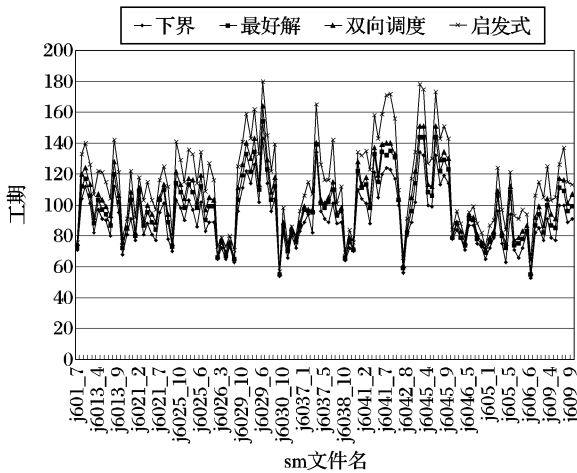


图3 j60 仿真结果

(上接第 201 页)

法和本文提出的发现方法的平均查全率为 20% ,68% 和 87% ,平均查准率为 15% ,59% 和 82% 。这里本文假设各个阶段的相似度阈值均为 50% ,返回的是与查询条件相似度在 50% 以上的 Web 服务。具体测试结果如图 2 和图 3 所示。

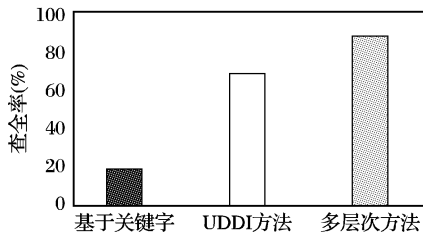


图2 查全率

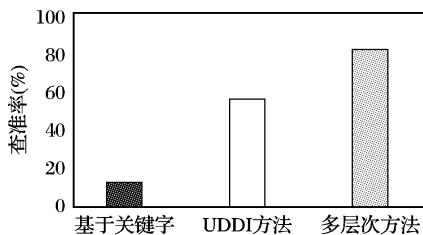


图3 查准率

测试结果表明基于关键字的服务发现方法查准率和查全率都较低,文献[1]的服务发现方法的性能比基于关键字的发现方法有明显的改善。而本文提出的多层次服务发现方法的查准率和查全率都达到 80% 以上,较前面两种方法,性能均有较大的提高。

结合反向调度笔者设计了一个双向搜索算法,对于文献 [6]中提出的标准问题进行了仿真,由于有些问题用一般的启发式方法就可以得到比较好的解,本文对于 j30 库中用分支定界计算超过 10s 的 46 个问题和 j60 库中至今未找到最优解的 122 个问题进行了仿真,仿真结果见图 2 和图 3。图中横轴表示文献 [6]中给出的 sm 文件对应的文件名,纵轴表示工期,单位是单位时间。

本文提出了用额外关系表示一个可行解的方法,清晰而简洁地给出了解的邻域构造方法以及解之间距离的定义,为设计分散搜索算法提供了坚实有力的基础。因为这种表示方法与产生一个解的启发式框架无关,这给分析问题和计算带来了许多方便。

最后的程序仿真结果说明用额外关系表示可行解是可行的,本文提供的算法也是比较有效的。

#### 参考文献:

- [1] WEGLARZ J. Project scheduling: Recent Models, algorithms and applications[M]. Boston: Kluwer Academic Publishers, 1999.
- [2] DEMEULEMEESTER E, HERROELEN W. A branch - and - bound procedure for the multiple resource-constrained project scheduling problem[J]. Management Science, 1992, 38(12): 1803 - 1881.
- [3] ALVAREZ-VALDEZ R, TAMARIT JM. Heuristic algorithms for resource-constrained project scheduling: a review and an empirical analysis[A]. Advances in Project Scheduling[C]. 1989. 113 - 134.
- [4] GLOVER F. Heuristics for integer programming using surrogate constraints[J]. Decision Sciences, 1977, (8): 156 - 166.
- [5] KOLISCH R, HARTMANN S. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis[M]. Kluwer, 1998. 147 - 178.
- [6] Http://129. 187. 106. 231/psplib/data. html[ EB/OL], 2005 - 12 - 10.

### 4 结语

本文提出了“五层次三阶段”的语义 Web 服务多层次服务发现方法,在匹配过程中充分考虑了功能属性和非功能性属性,采用语义相似度,提高了查找的精确性;另外基本信息的匹配起到了过滤作用,很大程度上缩短了查找时间,同时在计算服务相似度时不再使用匹配级别,而是使用更精确的通过相似度计算得到的介于 [0,1] 之间的数值来表示,从细粒度角度表示了服务匹配的结果。将来的工作是将该方法推广到多本体环境中,将用户的反馈信息添加到 QoS 模型中,使其不断更新,从而更接近用户的需求。

#### 参考文献:

- [1] PAOLUCCI M, KAWAMURA T, PAYNE T, et al. Semantic matching of Web services capabilities[A]. Proceedings of the First International Semantic Web Conference (ISWC)[C]. 2002. 333 - 347.
- [2] MARTIN D. OWL-S: semantic markup for Web services[ EB/OL]. Http://www. w3. org/Submission/OWL-S, 2005 - 12 - 10.
- [3] GILLELAND M. Levenshtein Distance, in Three Flavors[ EB/OL]. Http://www. merriampark. com/ld. htm, 2005 - 12 - 10.
- [4] COURTNEY C, RADA M. Measuring the Semantic Similarity of Texts[A]. Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment[C]. 2005. 13 - 18.
- [5] WordNet- introduction [ EB/OL]. Http://ccl. pku. edu. cn/doubtfire/semantics/WordNet/C-wordnet/wordnet-c-index. htm, 2005 - 12 - 10.
- [6] 徐德智, 郑春卉, Passi K. 基于 SUMO 的概念语义相似度研究 [J]. 计算机应用, 2006, 26(1).
- [7] 朱礼军, 陶兰. 领域本体中的概念相似度研究 [J]. 华南理工大学学报(自然科学版), 2004, (S1).