

# 基于约束满足的 Job-Shop 调度问题求解方法研究\*

陈恩红 薛瀚宏

(中国科学技术大学计算机科学与技术系 合肥 230027)

**摘要** 针对一类典型的约束满足问题——Job-Shop 调度问题,在深度优先搜索算法的基础上,给出了形式化的约束一致性实施、操作选择策略和开始时间选择策略以及启发式的约束一致性实施和不完全回跳策略,从而提高了求解 Job-Shop 调度问题的效率。

**关键词** 一致性实施,操作选择,开始时间选择,不完全回跳,困难操作集。

**中图法分类号** TP18

## 1 Job-Shop 调度的约束满足问题简介

Job-Shop 调度是调度问题研究中的一类重要问题,它要求在一定的时间范围内和一定的资源集上完成一组作业的调度。<sup>[1~3]</sup>给定资源集  $RES = \{1, \dots, v\}$  和作业集  $JOB = \{J_1, \dots, J_w\}$ , 其中作业  $J_i$  的提交时间和截止时间分别是  $B_i$  和  $E_i$ , 并且  $J_i$  是由  $q_i$  个不重叠的操作组成,即  $J_i = \{o_{i1}, \dots, o_{iq_i}\}$ , 而每个操作  $o_{im}$  又有一定的执行时间  $d_{im}^i$  和资源需求  $R_{im}^i \in RES$ . 求解操作  $o_{im}$  的开始时间  $s_{im}^i$ , 要求满足: (1) 作业的时间约束: 任何操作  $o_{im}^i$  有  $s_{im}^i \geq B_i \wedge s_{im}^i + d_{im}^i \leq E_i$ ; (2) 操作的顺序约束: 作业  $J_i$  的两个操作  $o_{im}^i$  和  $o_{in}^i$ , 若  $m < n$ , 则  $s_{im}^i + d_{im}^i \leq s_{in}^i$ ; (3) 资源的相容约束: 任两个操作  $o_{im}^i$  和  $o_{jn}^j$ ,  $R_{im}^i \neq R_{jn}^j \wedge s_{im}^i + d_{im}^i \leq s_{jn}^j \vee s_{jn}^j + d_{jn}^j \leq s_{im}^i$ .

## 2 求解 Job-Shop 调度约束满足问题的搜索算法

求解约束满足问题的一般方法是深度优先搜索,通过尝试对变量的不同赋值,不满足约束时则通过回溯的方式来得到问题的解或证明问题无解。<sup>[4,5]</sup>

为每个操作  $o_{im}^i$  定义可能开始时间集  $S_{im}^i$ , 初始化时只要求满足作业的时间约束,并定义操作的最早开始时间  $es_{im}^i = \min(S_{im}^i)$  和最迟开始时间  $ls_{im}^i = \max(S_{im}^i)$ . 定义困难操作集  $D$  为等待调度操作集的一个子集,该子集中的操作曾经产生冲突(即失败的局部调度). 困难操作集在搜索过程中对一致性实施和变量选择都有启发作用. 算法 1 是本文所介绍的算法的轮廓,其中所采用的策略细节将在后面阐述.

算法 1. 带启发式的深度优先搜索求解 Job-Shop 问题

/\*  $U$ : 等待调度操作集;  $S[o]$ : 操作  $o$  的可能开始时间集;  $D$ : 困难操作集;  $C$ : 冲突操作集 \*/

```

1 function HDFS()
2   if  $U = \emptyset$  then return SUCCESS
3    $o \leftarrow \text{select\_operation}(U, D)$  /* 选择进行调度的操作 */
4    $D \leftarrow D \setminus \{o\}$  /* 把  $o$  从困难操作集中删除 */
5   while enforce_consistency( $U, C$ )  $\neq$  ONFLICT do /* 一致性实施发现冲突的操作集  $C$  */
6      $s \leftarrow \text{select\_start\_time}(S[o])$  /* 选择操作的开始时间 */
7      $U \leftarrow U \setminus \{o\}$  /*  $o$  已被调度 */
8     if HDFS() = SUCCESS then return SUCCESS
9      $U \leftarrow U \cup \{o\}$  /* 回溯,取消对  $o$  的调度 */
10     $D \leftarrow D \cup C$  /* 把曾经冲突的操作加入困难操作集 */
11     $S[o] \leftarrow S[o] \setminus \{s\}$  /* 删除该可能开始时间 */

```

\* 本文研究得到国家自然科学基金和中国科学技术大学青年基金资助. 作者陈恩红, 1968年生, 博士, 主要研究领域为约束满足问题, 遗传算法, 机器学习. 薛瀚宏, 1972年生, 硕士生, 主要研究领域为约束满足问题, 数据库中的知识发现.

本文通讯联系人: 陈恩红, 合肥 230027, 中国科学技术大学计算机科学与技术系

本文 1997-03-19 收到原稿, 1997-11-06 收到修改稿

```

12     end while
13     return FAILED
14 end function

```

### 2.1 约束的一致性实施方法

约束的一致性实施分为约束传播和约束检查.约束传播包括顺序约束传播和资源相容约束传播,它根据操作使得二元约束直接修剪可能开始时间集;约束检查则发现操作间的冲突,但不修剪可能开始时间集.

顺序约束传播的一致性实施方法:从作业的时间约束和操作的顺序约束知,对于作业  $J_i$  有  $es_i^j \geq B_i$  和  $ls_i^j \leq E_i - d_i^j$ ,而且  $es_{m+1}^i \geq es_m^i + d_m^i, ls_m^i \leq ls_{m+1}^i - d_m^i, m=1, \dots, q_i - 1$ .

资源相容约束传播的一致性实施方法,若操作  $o_m^i$  和  $o_n^k$  共享同一资源.从资源的相容约束条件可以得出

$$S_m^i \cap (S_n^k - d_n^k, ls_n^k + d_n^k) = \emptyset.$$

资源容量约束检查,设  $G$  是一组共享同一资源的操作集,则  $G$  中的操作可被一致调度的必要条件是

$$\max(\{ls_m^i + d_m^i | o_m^i \in G\}) - \min(\{es_n^k | o_n^k \in G\}) \geq \sum_{o_m^i \in G} d_m^i.$$

资源容量约束检查只对  $U$  的子集进行,并要求该子集和困难操作集的交不为空.

### 2.2 操作(变量)选择策略

本文采用困难操作优先的调度策略.首先,曾经产生冲突的操作被认为是困难的,其次,和已被调度的操作需要相同资源的操作也被认为是困难的.这两类操作被优先处理.定义操作的困难度函数

$$Dif(o_m^i, U) = \sum_{o_n^k \in U \wedge R_{o_n^k} = R_{o_m^i}^k} |S_m^i \cap S_n^k|.$$

$Dif$  值越大,则该操作和其他操作在执行时间上重叠的可能性越大,所以应该优先调度.

### 2.3 开始时间(赋值)选择策略

本文采用相容性好的赋值优先策略.定义相容度函数

$$Com(s_m^i, U) = \sum_{o_n^k \in U \wedge R_{o_n^k} = R_{o_m^i}^k} |(s_m^i - d_m^i, s_m^i - d_m^i) \cap S_n^k|.$$

选择了一个操作的开始时间,进行一致性实施之后,其他共享同一资源的操作的开始时间集就被修剪.相容度越大,则修剪中所做的删除越少.

### 2.4 不完全回跳法

当在搜索树的某个分枝上的回跳次数超过给定的阈值时,则认为对该分枝的搜索是低效的,立即回溯到搜索树的根结点.阈值越小,搜索剪枝越多.由于回跳法可能删除存在解的分枝,因而搜索是不完全的,但可以加速求解过程.<sup>[6,7]</sup>

## 3 实验结果和比较

本文对算法 1 用文献[1,2]所提供的 60 个 Job-Shop 调度问题进行了测试.每个问题包含 10 个作业,每个作业由 5 个操作组成,分别需要 5 种不同资源.以问题求解的困难程度分为 6 个等级,每个等级包括 10 个问题.表 1 列出了 3 种算法的比较结果,其中算法 1 的回跳阈值定为 30 次回跳.搜索效率定义为有用赋值次数(即操作个数)和总赋值次数(即搜索结点数)的比值,效率为 1 时表明搜索是无回溯的.算法 1 对于 60 个问题中的 50 个,求解时是无回溯的,并且在绝大多数的问题中的搜索效率都有显著提高.

表 1 3 种深度优先搜索算法的实验结果比较(搜索规模限制在 500 个结点以内)

问题困难度		1	2	3	4	5	6	合计/平均
文献[1]算法	解决问题数	10	10	8	9	7	8	52
	搜索效率	0.96	0.99	0.78	0.87	0.73	0.82	0.86
文献[2]算法	解决问题数	10	10	10	10	10	10	60
	搜索效率	0.96	0.99	0.91	0.93	0.88	0.84	0.92
本文算法 1	解决问题数	10	10	10	10	10	10	60
	搜索效率	0.99	1.00	0.99	0.92	0.98	0.91	0.96

## 参考文献

- 1 Bitner J R, Reingold E M. Backtrack programming techniques. *Communications of the ACM*, 1975,18(11),651~655
- 2 Sadeh N, Sycara K, Xiong U. Backtracking techniques for the Job Shop scheduling constraint satisfaction problem. *Artificial Intelligence*, 1995,76(1),455~480
- 3 Uekun S, Bagchi S, Kawamura K. Managing genetic search in Job Shop scheduling. *IEEE Expert*, 1993,15~24
- 4 Lhomme O. Consistency techniques for numerical CSPs. In: Bajcsy Ruzens ed. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. San Mateo, California, Morgan Kaufmann Publishers, Inc., 1993. 232~238
- 5 Schiex T, Fargier H, Verfaillie G. Valued constraint satisfaction problems, hard and easy problems. In: Mellish Chris S ed. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. San Mateo, California, Morgan Kaufmann Publishers, Inc., 1995. 631~637
- 6 Dechter R. Enhancement schemes for constraint processing, backjumping, learning and cutset decomposition. *Artificial Intelligence*. 1989,41(2),273~312
- 7 Williams C P, Hogg T. Using deep structure to locate hard problems. In: *Proceedings of the 10th National Conference on Artificial Intelligence*. AAAI Press/the MIT Press, 1992. 472~477

## On Solving Constraint Satisfaction Based Job-Shop Scheduling Problems

CHEN En-hong XUE Han-hong

*(Department of Computer Science and Technology University of Science and Technology of China Hefei 230027)*

**Abstract** In this paper, the authors discuss how to solve a set of typical constraint satisfaction problems, Job-Shop scheduling problems. Based upon the depth-first search, formal strategies of enforcing consistency, selecting operation and selecting start time are given, heuristic strategies of enforcing consistency and incomplete back jumping are introduced to further enhance the efficiency in solving Job-Shop scheduling problems.

**Key words** Enforcing consistency, selecting operation, selecting start time, incomplete back jumping, difficult operation set.