·综述·

(23) 133－138

基于约束分析的 Job Shop 调度算法的综述*

TH165
TH186

段黎明　陈进　刘飞
（重庆大学机械工程一系，重庆，400044，第一作者 33 岁，男，副教授，博士生）

摘　要　综述了基于约束分析求解 Job Shop 调度的算法，并对进一步研究进行了讨论。

关键词　车间调度；约束满足
中国图书资料分类法分类号　TH165；TP18

约束分析

0　引　　言

调度问题，指的是按一定的要求把一组资源分配给一组任务[1]。它对制造系统中的计划和控制是特别重要的。Job Shop 的调度问题又是调度问题中最困难的一种。一般来讲，解决 Job Shop 调度有三种方法：优先规则、组合优化和约束分析。优先规则具有计算有效并能应用到实际的优点，但使用优先规则时，对得到的答案质量没有保证，特别是在某些时间约束应重点考虑时更是如此；组合优化有对求解大规模的调度问题不易处理的缺点，此外，一个数学意义上的最优答案在实际问题中不总是有用的；约束分析方法，寻找一组服从几个时间或技术约束的可能答案，此后把最后答案的选择留给使用者。这种 方法在某种意义上比其它二种方法更具有一般性，这是因为：(1)它是基于约束的系统作用；(2)它能利用启发式知识(如优先规则等)来指导系统在可能的范围内搜索答案；它也很容易和知识基系统相结合，构成知识基调度系统。

1　Job Shop 调度问题的一般描述

Job Shop 调度问题指需要在一组物理资源 $RES = \{R_1, R_2, \cdots, R_m\}$ 上调度一组任务 $J = \{J_1, J_2, \cdots, J_n\}$. 每个任务 J_i 包含一组活动 $O = \{O_1, O_2, \cdots, O_n\}$，这组活动(工序) 按工艺规划指 定了一个先后顺序(即 O_i 在 O_j 之前)。每个任务有一释放期 rd_i 和交货期 dd_i，在时间区间 $[rd_i, dd_i]$ 内，必须完成该任务的所有工序。每个工序 O_i 有一个加工时间 dd_i 和可选择的开始时间 st_i. st_i 受任务的释放期和加工时间约束。每个工序 O_i 需 P_i 个不同的资源 $R_{ij}(1 \leqslant j \leqslant P_i)$，$R_{ij} \in RES$.

一般来讲，问题可定义如下：

变量：

(1) 开始时间：st_i；

* 收文日期　1997-01-23
国家自然科学基金资助(59585005)

(2) 资源需求：$R_{ij}(1 \leqslant j \leqslant P_i)$，$R_{ij} \in RES$；

约束：

(1) 顺序约束：$st_i^k + dd_i^k \leqslant st_j^k$($O_i^k$ 在 O_j^k 之前)；

(2) 能力约束：$(\forall\ p,q \quad R_{ip}^k \neq R_{jq}^h)$

$\vee\ st_i^k + dd_i^k \leqslant st_j^h \vee st_j^h + dd_j^h \leqslant st_i^k$ 换句话说，除非 O_i^k 和 O_j^h 使用不同的资源，否则 O_i^k 和 O_j^h 不能同时进行。此外，还有任务的释放期和交货期约束。

2 基于约束分析求 Job Shop 调度问题简介

由以上分析可看出，Job Shop 调度问题实际上是一特殊的约束满足问题(Constraint Satisfaction Problem，简称 CSP)。解决 CSP 的方法是深度优先回溯法。使用该方法，调度问题能通过不断地选择下一个将要调度的工序(即选择变量)和实验性地为选择的工序安排一个开始时间(变量的值，简称值)。如果在构造调度的过程中，有能力和时间冲突发现，必须取消一个或几个更早工序的开始时间，重新为工序安排一个开始时间。这个过程称为回溯。回溯搜索的最糟情况具有指数复杂性，它恶化了搜索过程，增加了搜索空间和时间。

深度优先回溯搜索求解 Job Shop 调度问题方法描述如下：

step 1：如果所有的工序已被调度，那么停止，否则转到 step 2；

step 2：应用一致性增强方案；

step 3：如果一个冲突被发现，那么回溯(如果已被安排调度的工序有余下的开工时间，那么选择另一个开始时间，否则，或者停止并报告问题不可能，或者转到 step 4)；

step 4：选择下一个工序(根据变量排序启发式选择)；

step 5：为选择的工序安排一个开始时间(根据值排序启发式选择)；

step 6：转到 step 1.

搜索空间的大小很大程度上取决于约束的表达形式和用户指定的选择(如变量排序等)[2]。一般来讲，如约束为更显式的表达式时，对搜索更为有利。

提高回溯搜索的有效性可降低算法扩展的搜索空间。为提高回溯的有效性，研究者已提出了许多方案。归纳起来，这些方案可分为两类：搜索之前使用的方案和搜索过程中使用的方案[3]。

3 搜索之前使用的方案

对普通的 CSP 来讲，该方案主要包括不同的一致性增强算法。这些算法的主要思想是把一个给定的约束网转换为一个等价的，但更显式的网[2,3]。对 Job Shop 调度来讲，该算法是一个混合过程，顺序约束和能力约束的一致性增强方案是不同的[4,5]。

关于顺序约束的一致性增强方案是通过更新未调度工序的一对最早/最迟可能开始时间来实现的。任务的最早开始时间向后传播，最迟开始时间向前传播，在无能力约束情况下，该算法能充分保证回溯自由搜索[4]，即无需回溯，就可发现答案。

关于能力约束更复杂，这是由于这些约束之间的析取(disjunctive)性质引起的。[4]提出当一个活动在一段时间内占有一个资源时，前向检查(forward checking)方案检查需同一资源的其它工序的可能开始时间，并去掉和新的安排相冲突的开始时间。

一致性增强方案修剪了搜索空间，但修剪搜索空间也要花费时间，因此，在整个系统扩

展的搜索空间和花费的时间之间应有一个平衡。

4 搜索过程中动态使用的方案

该方案可被分为向前看(lookahead schemes)和向后看(lookback schemes)方案。

4.1 向前看方案

向前看方案在算法决定下一个应选择什么变量，和决定对所选的变量应赋什么值时使用[2,3,6]。该方案包括变量排序启发式(variable ordering heuristics)和值排序启发式(value ordering heuristics)算法。

4.1.1 变量排序启发式算法

减少回溯搜索的一个有效方法是按照适当的顺序来选择变量。通常，首先选择困难的变量，回溯搜索将避免构造后来不能被完成的部分答案。实际上，通过首先选择困难变量，系统移动到更容易检测的冲突状态，这样减少了系统花在完成一个不能完成的部分答案的时间。

有两类变量排序启发式[5]：

(1) 固定变量排序启发式：先于搜索之前决定选择变量的顺序，并且在每个搜索树的每一个分枝中都作用。

(2) 动态变量排序启发式：为了考虑更早变量的赋值，在每个搜索状态安排变量选择的顺序。搜索树中不同的分枝通常具有不同的变量排序。

很明显，由于固定变量排序启发式只被确定一次，因此它需要较少的计算时间。另一方面，动态变量排序启发式更有效，这是因为该启发式不是在整个搜索树，而是在指定的搜索状态来辨识困难的变量。几个较早期的 CSP[7]研究表明，解决简单 CSP 问题如 N-queens 或中等规模的 CSP 时，使用动态变量排序启发式太费时。[7]中指出对一些更困难的 CSPS，使用动态变量排序启发式为得出一个答案，可大大降低所需的平均搜索状态数。对更困难的问题，一般的文献推荐使用动态搜索再安排启发式(DSR)[7,8,9]，在每个搜索状态，DSR 寻找具有最少剩余值的变量，并选择这个变量为下一个将要赋值的变量。[5]中的试验表明 Job Shop 调度属于更困难的 CSPS，DSR 对 Job Shop 调度来讲，是一个太弱的启发式。这是因为 DSR 仅考虑了每个变量余下值的数量，不能估计这些余下值在后来有多大可能的可用性。[3,16]中提出了一个最小宽度(MW)启发式。MW 启发式的基本思想是这样的：在约束图中，删去已被选择赋值的所有结点(变量)后，选择余下部分有一个最小度(结点的度，联系到那个结点的约束事件的数量)的结点为下一个赋值的变量。MW 有一个修正形式。叫最小最(MD)启发式。MD 根据初始约束图中结点的度的大小来简单排序变量[10]。通常，Job Shop 调度问题十分复杂，使用 MD、MW 启发式的效果都不太理想，这是因为 MW 和 MD 启发式都没考虑约束的紧度(满足一个指定约束的困难性)[11]。

分析 CSP 文献中提出的普通变量排序启发式，可以看出，这些启发式均匀地处理所有问题的约束。在许多实际的 CSPS 中，不同类型的约束具有不同水平的约束检查，这个结果影响到不同的变量排序启发式的有效性。在 Job Shop 调度中，一致性检查技术能有效地保证在有能力冲突时才发生回溯。结果一个工序(变量)的困难性仅仅是发现不发生任何能力冲突的工序开始时间的困难性的函数。[12]中提出了一个广义化的 DSR，其中，给每个工序的开始时间安排一个可生存计算，它反映满足能力约束的概率(即它和分配给相同资源的其它工序竞争时生存下来的机会)。下一个将要调度的工序是具有总体可生存能力最小的工序，

这个启发式比普通的启发式更有效,但使用这个启发式也是十分费时的.因为它要求检查所有的余下的开始时间(所有未调度工序的)。在有几百个工序或更多,每个工序有几百个可能的开始时间和几个可利用的资源时.这个启发式效率太低。[5]中提出给搜索空间引入一个概率模型,根据该模型对每一个未调度工序安排一个重要性指标计量,重要性指标近似了工序将被包含到冲突的可能性。具有最高重要性的工序被首先调度。[5]中试验表明这个变量排序启发式效果较好。

4.1.2 值排序启发式算法

这是一个有效的降低回溯搜索平均复杂性的方法。该方法是为变量的值选择一个适当的顺序,变量根据此顺序试一试每个可能的值,这就是值排序启发式算法。一个好的值排序启发式是最少约束值启发式。所谓最少约束值启发式是对整个问题而言,被期望能参与进许多答案中的约束值,更好地,能参与进适合目前搜索状态的大量答案中。搜索时,通过首先让变量试一试最少约束值,系统一般将使仍未赋值的变量的值的数量最大化,因此,将避免构造不能被完成的部分答案。

试图准确地计算一个值所参与进的全体答案的数量将是无用的,因为它将必须发现这个问题的所有答案。[3]中提出一个值排序启发式,它依赖于问题的树松驰来评估一个值的优缺点。在树型松驰中,一个值能参与进的答案数量以 $O(nk^2)$ 步计算出来。这里 n 是变量数量,k 是一个变量可能值的最大数。该启发式需对搜索树中的 n 层中的每一层计算一个固定的极小化树(MST),这个累计需 n 个 MST 计算,每一个 MST 计算需 $O(n^2)$ 基本计算,因此总共需 $O(n^3)$ 步基本计算。该值排序启发式对解决某些较简单的 CSPS 十分有效,但对 Job Shop 调度这样约束很紧的 CSPS 问题,效果太差。

[12]中提出了一个值排序启发式,它能更有效地处理能力约束。该启发式首先把对每个资源的竞争评估为时间的函数,基于这些评估,工序开始时间按照它们被期望和其它工序的资源需求有多大冲突来排序。然而,该启发式没有对相同任务的其它工序留足够的值,换句话说,该启发式仅对目前的工序考虑了能力约束,但没对相同任务中的其它工序考虑能力约束。

[5]中根据搜索空间的概率模型,提出了两种值排序启发式:最少约束值(LCV)启发式和贪婪值(GV)排序启发式。在 LCV 中,每个工序的每个可能的发生时间,按照它不和另一工序开始冲突的概率来给出一个评价指标,工序开始时间按照评价指标排序。在 GV 中,仅仅根据偏爱来排序工序的开始时间。

4.2 向后看方案

以上介绍的向前看方案能有效地降低系统的回溯,但一般不能完全避免回溯。搜索过程中,冲突发生时,向后看方案被设计用来帮助系统从冲突中恢复[4]。

求解 Job Shop 调度时,当冲突发生时,最简单的策略是回到最近被安排的工序(变量)[4],该工序至少有一个可供选择的开始时间留下。重新安排留下来的另一个值给工序。该策略被称为慢慢回溯(chroological backtracking)。通常,引起目前冲突的根源不是最近的安排,而是更早的一个。这样慢慢回溯策略很可能修正了对目前冲突无影响的安排,降低了回溯的有效性。

在一般的 CSPS 求解中,研究者们提出了几个向后看策略[2]。这些策略可分为两类。第一类是通过分析冲突产生的原因,来直接回到冲突的根源处。直向相关回溯(dependency-di-

rected backtracking)，回跳(backjumping)，基于约束图的回跳(graph-based jackjumping)等就是这样的技术。第二类是以新的约束形式来记录更早的冲突，使系统在后来的搜索过程中免于过去发生的错误，这种方法被称为学习。[2]中提出的 n 阶深和浅学习仅记录包含几个更少变量的冲突就是这样的学习方法。尽管使用向后看方案能大大减少搜索状态空间的数量，但约束记录最糟情况具有指数复杂性，因此使用这些方法时必须考虑一些平衡。

Job Shop 调度中的约束具有很高的相互关联性，且能力约束具有析取性质，使其约束比一般的 CSP 复杂。根据[4]中实验研究，直接利用一般 CSP 的向后看方案不会提高 Job Shop 调度问题搜索的有效性。[4]中提出了三种方案：(1)动态一致性增强方案(DCE)，该方案启发式地识别一个或几个在冲突发生时的工序，在这些工序中增强完全一致性来确定回溯多远，(2)从失败中学习排序(LOEF)方案。LOEF 是基于这样的思想：发生冲突时表明所用的变量排序启发式对处理目前的问题是不合适的，因此首先处理参与进冲突的工序是恰当的，这样，避免了变量排序启发式。(3)不完全回跳(IBH)。IBH 基于这样一个思想：冲突发生时，不是因为它们已被证明为不一致，而是因为似乎有过多的约束。这样，发生冲突时，IBH 首先取消导致更紧搜索状态的安排。IBH 的搜索过程是不完全的，甚至不能找到可能的答案，将 DCE、LOEF、IBH 三种方法结合起来使用，效果更好些。[13]中提出了一种使用基于解释的学习(PEBL)来动态获取调度系统的搜索控制规则。PEBL 给出一个决策函数来指示解释是否正确的概率。如果从前的解释成立，决策函数将增加。决策函数达到一定的程度，将产生一个新的搜索控制规则。[13]中实验表明，使用 PEBL 大大提高了调度过程中搜索的有效性。

实际的调度环境往往是动态的，不可预测的事件经常发生，而且多个优化目标之间相互作用，会产生冲突。[14]中结合约束分析和基于案例的学习开发了一个调度系统，该系统功能有：(1)动态获取用户的调度目标的偏好；(2)能在一个完成的，但次优的初始调度基础上，通过几个启发式修改技术来逐步修正调度系统，使最后的结果满足用户的偏爱。这个系统还只是一个实验性的系统，离实用化还有段距离。

基于约束分析解决 Job Shop 调度问题另一种有前途的方法是：取一个初始的，非一致性的安排，然后利用局部修改技术来逐步修复冲突，直到得到完全一致的的安排。这种方法称为基于修改(repair-based)的约束分析方法。对普通的 CSPS，[10]中提出两种修改策略：深度零(depth-0)策略和可变深度(variable-depth)策略。[15][16]中结合神经网络方法，提出一种极小化冲突启发式修改方法。但还未从已有的文献中发现利用基于修改的约束分析方法来求解 Job Shop 调度问题。

5 结　　语

本文全面分析了基于约束分析求解 Job Shop 调度的主要算法：一致性增强方案、变量排序启发式和值排序启发式算法、向后看方案中的主要方法。开发具体的 Job Shop 调度系统时，应根据要求，综合应用本文中介绍的各种方法，这是因为 Job Shop 调度问题是十分复杂的 CSPS，单一的方法往往效果很差。

从本文的分析可以看出，大多数算法的有效性和复杂度缺乏理论分析，只有根据实验来验证。如向后看中的学习方法，如学习程度太深，系统花在学习上的时间较长，虽降低了系统的搜索空间，但系统得到最后答案所花费的总的时间可能增加。这样，由于缺少必要的理论

分析、应用学习方法时，可能有些盲目性。因此，应加强理论上的分析和研究。

为了能处理更复杂的目标和更大规模的调度系统，必须结合人工智能和机器学习中的最新成果研究更高效的搜索方法。

参 考 文 献

1 刘飞．制造系统工程．北京：国防工业出版社，1995

2 Dechter R. Enhancement schemes for constraint processing: Backjumping, learning, and cuts[...]composition, Artificial Intelligence, 1990, 41: 273～312

3 Dechter R. Network-based heuristics for constraint satisfaction problem. Artificial In[...], 34: 1～38

4 Sadeh N. Backtracking techniques for the job shop scheduling constraint satisfactio[...]telligence, 1995, 76: 455～480

5 Sadeh N. Variable and value ordering heuristic for activety0based job shop scheduling[...]onf. Expert syst. in production and operations management, 1990, 134～144

6 Frender E C. A sufficient condition of backtrack free search. J. ACM, 1982, 29(1): 24

7 Purdom P W. Search rearrangement backtracking and polyomial average time. Artificia[...]983, 21: 117～133

8 Dechter R. Experiment evaluation of preprocessing algorithms for constraint statisfation p[...]tificial Intelligence, 1994, 68: 211～241

9 Ginsberg M L. search lessons learned from crossword puzzle. In: proc. of the Eight National[...]tificial Intellinence, 1990, 210～215

10 Dechter R. Experiment evaluation of preprocessing techniques in constraint satisfaction proble[...]IJCAI-89, 1989, 271～277

11 Fox M S. Constraint heuristic search. In: proc. IJCAI-89, 1989, 309～315

12 Keng N. Aplanning/scheduling methodology for the constrained resource problem. In: proc. IJC[...]89, 998～1003

13 Zweben M. Learning to improve constraint-based scheduling. Artificial Intelligence, 1992, 58: 271～296

14 Migashita K. CABINS: a framework of knowledge acquistion and iterative revision for schedule improvem[...]nd reactive repair. Artificial Intelligence, 1995, 76: 377～426

15 Minton S. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair metho[...]: proc. AAAI-90, 1990, 17～24

16 Minton S. Minimizing conflicts: a heuristic repair method for constraint saticfaction and scheduling problems. tificial Intepligence, 1992, 58: 161～205

The Algorithms of Constraint—Based Job Shop Scheduling

Duan Liming *Chen Jin*

(Department of Mechanical Engineering I, Chongqing University)

ABSTRACT A survey of the constraint-based job shop scheduling was given, and the future research was discussed.

KEYWORDS job shop scheduling; constraint satisfaction; algorithms