

作业车间调度问题的一种混合式算法

谢丽芳, 费跃农

(深圳大学机电与控制工程学院, 广东 深圳 518060)

摘要:通过对传统模拟退火算法和移动瓶颈法各自算法流程的分析,提出了一种改进的模拟退火算法.该算法以模拟退火算法为主体流程,在主体流程中融入移动瓶颈技术,及记忆功能和倒转技术.通过对JSP优化问题的仿真试验,验证了改进后的模拟退火算法对大规模组合优化问题可以得到好的优化效果.

关键词:作业车间调度;NP-hard;启发式;模拟退火算法;移动瓶颈法

中图分类号: O224; F406.6; TP312

文献标识码: A

生产调度,即对生产过程进行作业计划,是整个先进制造系统实现管理技术、运筹技术、优化技术、自动化与计算机技术发展的核心.作业车间调度问题(Job-Shop Scheduling Problem, JSP)是一类具有时间约束、次序约束和资源约束的组合优化问题,是生产调度领域的一个热点难点,是许多实际生产调度问题的简化模型,是一个典型的 NP-hard 问题^[1].因此,具有重要的理论意义和工程价值,也是目前研究最广泛的一类典型调度问题.解决这类问题的关键在于寻求有效的优化算法,这本身也是一个 NP 完全问题^[2].当问题的规模较大时,很难找到问题的全局最优解或者满意解,而且随着问题规模的加大,算法的实现时间成指数倍增长,因此,研究的重点就转移到优化各种启发式算法及提高算法优化效率方面.

求解调度优化问题的方法有很多,常用的有分支定界法、移动瓶颈法^[2]、局域搜索算法、基因算法及神经网络算法等.其中移动瓶颈算法与局域搜索算法是当前相互结合,构造混合式算法中应用最多、最有效的方法.模拟退火算法是局域搜索算法中最有效的算法之一,本文基于模拟退火算法(SA)和移

动瓶颈算法(SB)提出了一种改进的混合式启发式算法(SA_SB).

1 作业车间调度问题

作业车间调度问题可以描述为一些工件在一些机器上加工,需满足下面一些约束:(1)每个工件在机器上加工的次序给定;(2)每台机器同时只能处理一个工件,一个工件在一台机器上的处理过程称为一个工序,工序的加工时间是固定的且不可打断,找出使所有工件都完工的时间最短的调度^[3-4].

令 $N = \{0, 1, \dots, *\}$ 表示工序集,其中 0 和 * 是两个虚拟操作,分别表示加工开始和结束; M 为机器集; A 为约束工序对的集合; E_k 为在机器 k 上加工的约束工件对的集合. JSP 问题可以下面形式进行描述:

$$t_j - t_i \geq d_j, (i, j) \in A. \quad (1)$$

式(1)中: t_i, t_j 分别表示工序约束集中存在相互约束关系的先后工序对 (i, j) 的完工时间(如图 1 中各有向实线上先后两个工序对); d_j 为工序 j 的加工时间.

收稿日期:2006-08-21

作者简介:谢丽芳(1980-),女,广东河源人,硕士研究生.

$$t_i \geq 0, i \in N, \quad (2)$$

式(2)中 t_i 为工序 i 的完工时间。

$$t_j - t_i \geq d_j \vee t_i - t_j \geq d_i, (i, j) \in E_k, k \in M. \quad (3)$$

式(3)中, (i, j) 表示机器 k 上存在的加工约束集合 E_k 中的一个工序对(如图1中同一条虚线组成结点集合中相连的一对结点)。

满足约束不等式(1),(2),(3)的调度问题的一个可行解称为一个调度,该问题也可以用析取图 $G=(V, A, E)$ 来表示。图1为一个3工件、3机器的JSP的析取图,其中 V 为所有操作构成的顶点集,0和*为两个虚拟操作,分别表示加工开始和结束; A 为 n 条子边(实线)构成的边集,子边表示某工件按约束条件在所有机器上从开始到结束的加工路径; E 为 m 条子弧(虚线)构成的弧集,子弧表示同一机器上加工的工件的先后加工顺序的选择;各节点上的标示是该节点工序。因此,用析取图表示JSP问题,就是找一个使有向图中最长路径最小的有向无循环图,即找到使有向图中的最长路径长度 $C_{\max}(\pi)$ 最小的加工顺序 π 。

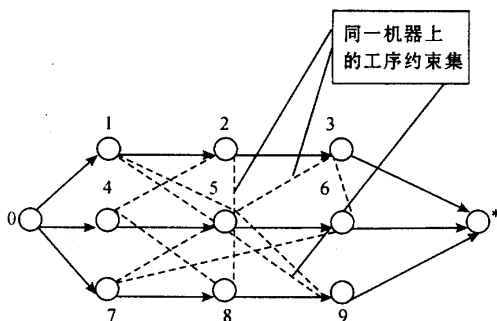


图1 3工件、3机器JSP的析取图

Fig. 1 A disjunctive graph representation of a 3×3 problem

2 算法

2.1 初始解

采用任何启发式算法都可以得到初始解。本文采用 Giffler 和 Thompson 提出的 GT 算法^[3]得到初始解。如果采用更有效的启发式算法来优化初始解,可以相对地减少退火次数。

2.2 模拟退火算法

模拟退火算法(SA)是一种主要解决组合优化问题的启发式算法,其基本思想是由 Metropolis 等

人^[1]于1953年首次提出的,1983年由 Kirk-Patrick^[2]将退火思想引入组合优化领域,解决大规模组合优化问题,对NP完全组合优化问题尤其有效。SA算法具有一种时变且最终趋于零的概率突跳性,可以有效地避免陷入局部最优并最终趋于全局最优,但是对JSP的一些大规模的优化问题,其实现全局收敛的时间性能较差。而 Adams 等人^[5]提出的移动瓶颈法(SB)是解决JSP问题最有效的启发式算法之一。SB算法通过对瓶颈机器进行单机调度,修改该机器上的调度顺序来优化当前调度,从而获得更好的次优解。针对基本模拟退火算法的一些不足,将移动瓶颈技术引入到模拟退火算法中,使改进后的模拟退火算法在保持原有的具有跳出局部最优特性的前提下,进一步优化当前状态下的解,提高传统SA算法的优化效率。同时利用记忆装置,保证在一定终止条件下所得的最终解是搜索过程中曾得到的所有解中的最优解。

2.2.1 领域结构

对于作业车间调度问题产生领域的方式有多种,一般是交换相邻两个工序的位置,或者是将一个工序插在另一个工序的前面或后面。本文采用 Van Laarhoven^[6]和 Taillard^[7]等人提出的加工顺序 π 的移动来产生领域,也就是交换(翻转)工序对 (i, j) ,产生解的集合。其中工序对 (i, j) 需满足:(1) i 和 j 是在相同机器上的两个连续的工序;(2) i 和 j 是在析取图 $G(\pi)$ 中的同一条关键路径上。

2.2.2 温度衰减函数

温度衰减函数的设计包括设置初温 T_0 、终温 T_c 及衰减因子 α 。温度控制问题将会影响算法运行速度。 T_0 大,便于激活各状态的接受概率,从而获得高质量解的几率也增大,但计算时间也将成倍增加。因此,通过实验来调试温度衰减。初温时,保证接受概率 $\exp(-\Delta/T) \approx 1$;终温时,保证接受概率 $\exp(-\Delta/T) \approx 0$,这样可以保证全局的跳跃性跟局部的收敛性。

2.2.3 内外循环终止准则

内循环终止准则采用 Metropolis 抽样稳定准则,也就是当接受新状态或者当前状态的领域解空间已搜索完时,跳出内循环,进行降温回到外循环。外循环终止准则即算法终止准则,在算法设计中采用设定终止温度,当前温度达到终止温度时,终止外循环,算法结束。

2.3 移动瓶颈技术

给定一个活动调度 S , 选择该调度的一个关键路径, 将非关键机器上的排序删除, 进行重新排序, 生成新的调度 S' . 将新调度 S' 作为移动瓶颈法的可行输入, 然后执行移动瓶颈技术进行优化, 最后将由移动瓶颈法得到的新调度进行全局协调, 以保证优化后的调度为活动的可行调度. 具体的移动瓶颈法的运算过程, 详见文献[2].

2.4 倒转技术

为了改善算法的计算结果, 引入倒转技术. 对一个给定的实例 G (如图1所示一个实例的析取图), 将其每个工件加工的顺序倒过来得到实例 G' (如图2), 称为实例 G 的倒转实例^[4]. 为了减少算法的运算时间, 采用运行速度较快的 GT 算法来找到倒转实例 G' 的一个可行解 (加工顺序 π'), 然后将加工顺序 π' 翻转得到加工顺序 π , 而 π 正好是实例 G 的可行解. 将由 G 得到的解与翻转实例 G' 得到的解进行比较, 取较小的解作为下一次退火搜索的起点.

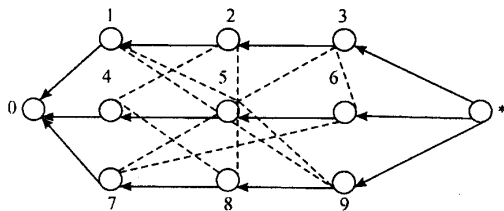


图2 3工件、3机器JSP的倒转析取图

Fig. 2 A reverse disjunctive graph representation of a 3×3 problem

2.5 改进后的 SA_SB 算法

改进后的 SA_SB 算法是一种基于模拟退火算法和移动瓶颈技术的混合式算法, 为了得到更好的计算结果, 在改进后的 SA_SB 算法中引入了记忆功能和倒转技术. 下面为改进后的 SA_SB 算法在 JSP 上的应用.

步骤1 由 GT 算法产生一个活动调度 S_k , 初始温度为 T_k , 令 $k=0$, 且记忆最优解 $S^* = S_k$, 最优值为 $C_{\max}(S^*)$.

步骤2 若满足算法的终止条件, 则搜索结束, 并输出最优解 S^* , 否则继续以下步骤.

步骤3 求解当前调度 S_k 的领域搜索空间 $N(S_k)$

, N 为该领域解空间元素的总数, 且令 $n=0$, n 表示当前解空间元素被计算的个数.

步骤4 判断是否接受新状态 S_i : (1) 随机选取领域空间 $N(S_k)$ 中一个解 S_i , 如果这个解是第一次被选中, 那么计算这个解的适配值, 记为 $C_{\max}(S_i)$, 同时令 $n=n+1$. (2) 采用 Metropolis 接受准则判断是否接受随机解 S_i . 如果 $C_{\max}(S_i) \leq C_{\max}(S_k)$, 那么接受当前调度为新调度, 令 $S_k = S_i$, 否则产生一个随机数 ξ , 有 $\xi \in (0, 1)$. 如果 $\xi \leq \min\{1, \exp(-\Delta/T_k)\}$, 其中 $\Delta = C_{\max}(S_i) - C_{\max}(S_k)$, 那么对新状态 S_i 采用移动瓶颈技术进行优化, 优化后的状态记为 S' , 并且令 $S_k = S'$. 若满足抽样稳定准则 (接受当前状态或者 $n=N$ 时), 则转步骤5, 否则重复步骤4.

步骤5 利用倒转技术再次优化 S_k , 得到 S'' , 如果 $C(S'') \leq C(S_k)$, 令 $S_k = S''$, 否则 S_k 保持不变, 然后进行退火降温, 即令 $T_{k+1} = \alpha T_k$, 并令 $k=k+1$, 同时更新记忆最优解. 如果 $C_{\max}(S_k) < C_{\max}(S^*)$, 那么更新最优解. 令 $S^* = S_k$, 则 $C_{\max}(S^*) = C_{\max}(S_k)$, 否则最优解保持不变.

步骤6 直到满足算法终止条件时, 输出最优解 S^* .

3 结果与讨论

分别采用改进后的 SA_SB 算法、传统的 SA 算法及插入 GT 算法后的 SA_GT 算法, 在 MATLAB6.5 的仿真环境下, 采用 128MB RAM 的 AMD/750 计算机, 对典型的 FT06, FT10 问题仿真 20 次, 仿真数据处理结果列于表1. 其中 FT06 问题的参数为 $T_0=100, T_c=1, \alpha=0.992$; FT10 问题的参数为 $T_0=400, T_c=1, \alpha=0.994$. 表1中 n 和 m 分别为工件数和机器数; C^{**} 为问题的最优值; $\bar{t}$ 为 20 次随机仿真的平均 CPU 时间; C^* 为 20 次随机仿真得到的最优值; $\bar{C}$ 为 20 次随机仿真得到的平均值.

由表1可知: 在相同的模拟退火参数设置条件下, 采用传统的 SA 算法对问题求解, CPU 的运行时间较短, 但得到的结果与问题最优值有较大的偏差; 改进后的 SA_SB 算法, 虽然其 CPU 的运行时间较长, 但得到的结果与问题最优值的偏差较小; SA_GT 算法的求解结果最差. 由此可见, 改进后的 SA_SB 算法能得到更好的优化效果.

表 1 改进后 SA_SB 算法、SA 算法和 SA_GT 算法对典型 Benchmark 问题求解结果的比较
Table 1 Comparisons between SA_SB, SA and SA_GT using MT benchmark problems

问题	$n \times m$	C^{**}	SA 算法			改进后的 SA_SB 算法			SA_GT 算法		
			C^*	$\bar{C}$	$\bar{t}/s$	C^*	$\bar{C}$	$\bar{t}/s$	C^*	$\bar{C}$	$\bar{t}/s$
FT06	6×6	55	55	55.9	54.5	55	55.4	84.1	55	56.4	72.1
FT10	10×10	930	993	1027.4	200	930	954.8	520	987	1013.3	320

4 结 论

将移动瓶颈技术和倒转技术引进到模拟退火算法中求解 JSP 调度问题,改进后的 SA_SB 算法对求解最优问题更为有效,可以求得更好的结果,因而该算法对求解 JSP 这类 NP-hard 问题确实可行,而且随着问题规模的增大,该算法的优化效果将更加明显.

参考文献:

[1] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社,2001.
[2] TAKESHI Y. Studies on metaheuristics for jobshop and

flowshop scheduling problems [D]. Kyoto: Kyoto University, 2003.
[3] 王凌. 车间调度及其遗传算法[M]. 北京:清华大学出版社,2003.
[4] 黄志,黄文奇. 一种基于禁忌搜索的作业车间调度算法[J]. 计算机工程与应用,2006,12(3):12-14.
[5] ADAMS J, BALAS E, ZAWACK D. The shifting bottleneck procedure for job shop scheduling [J]. Management Science, 1988,34(3):391-401.
[6] PETER J M, EMILE H L, LENSTRA J K. Job shop scheduling by simulated annealing[J]. Operations Research,1992,40(1):113-125.
[7] TAILLARD E. Paralled taboo search techniques for the job shop scheduling problem [J]. ORSA J Computing, 1994(6):118-125.

Hybrid algorithm for job shop scheduling problem

XIE Li-fang, FEI Yue-nong
(College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China)

Abstract: Based on the analysis of the process of the Simulated Annealing Algorithm and the Shifting Bottleneck procedure, an improved Simulated Annealing was proposed. The simulated annealing was served as the main flow of improved algorithm which syncretized the mechanism of improved shifting bottleneck procedure. To get some improvement, the memory function and reverse technique are applied in the algorithm. Through the simulation test for JSP optimization problem, the result indicates that the improved algorithm can gain better optimization effect especially combination optimization problem with large size.
Key words: job shop scheduling; NP-hard; heuristic; simulated annealing algorithm; shifting bottleneck