

可动态生成具有优先级工序集的 动态 Job-Shop 调度算法

谢志强¹⁾ 杨 静²⁾ 杨 光¹⁾ 谭光宇³⁾

¹⁾(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

²⁾(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

³⁾(哈尔滨理工大学机械动力工程学院 哈尔滨 150080)

摘 要 为了在满足约束条件的前提下使不同时刻加入到作业集中的全部作业所用的加工总时间尽可能地少,提出算法:根据动态的作业集构造虚拟加工树,虚拟加工树上的叶结点代表最先加工的工序,虚拟根节点代表最后一道工序,边代表偏序关系;以层优先为虚拟加工树上的工序设置优先级,工序的优先级可作为调度过程中考虑的一个因素;除了在工序需要动态调整的情况下,在调度过程中始终遵循着机器忙原则.在调度过程中,首先根据虚拟加工树动态地生成备选工序集合,然后根据工序的优先级并且结合其它的调度策略从备选集合中调度工序,直到备选集合为空,即所有作业加工完毕.这里提到的调度策略包括短用时策略、长路径策略和动态调整策略.通过实例验证,该算法对于动态 Job-Shop 问题可以获得比较好的解.

关键词 车间调度;虚拟加工树;优先级;备选工序集;动态调整

中图法分类号 TP278

Dynamic Job-Shop Scheduling Algorithm with Dynamic Set of Operation Having Priority

XIE Zhi-Qiang¹⁾ YANG Jing²⁾ YANG Guang¹⁾ TAN Guang-Yu³⁾

¹⁾(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

²⁾(School of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

³⁾(School of Mechanical Power Engineering, Harbin University of Science and Technology, Harbin 150080)

Abstract To make the total processing time of job added into job set at different time as short as possible, subject to the constraint, a new algorithm is proposed: A virtual manufacturing tree is constructed according to dynamic job set, where leaf nodes denote the operations to process firstly, root nodes denote the operations to process last and edges denote the partial relation, that is constraint between operations; the priority of operations is set according to level of operation in the virtual manufacturing tree and is regarded as a factor considered to schedule operations; A strategy of keeping machine busy is adopted to make machine try its best to process operations continually except that operations need be modified dynamically. Standby operation set is generated dynamically according to the virtual manufacturing tree, and operations in the standby operation set, subject to the priority, are scheduled with Short-time strategy, Long-path strategy and Dynamic-adjustment strategy until the set is empty, that is all operations unscheduled are finished. The example shows that the algorithm proposed can obtain favorable result for dynamic Job-Shop scheduling problem.

收稿日期:2006-10-25;最终修改稿收到日期:2007-12-03. 本课题得到国家自然科学基金(50575062)、黑龙江省自然科学基金(F0309, F200608)、黑龙江省教育厅重大科学研究项目(10551z0008)和哈尔滨市科技攻关项目(2005AA1CG061-11)资助. 谢志强,男,1962年生,博士研究生,教授,研究领域为CIMS和调度优化. E-mail: xzq011@tom.com. 杨 静,女,1962年生,博士,教授,博士生导师,研究领域为数据挖掘和调度优化. 杨 光,男,1982年生,硕士研究生,研究方向为调度优化. 谭光宇,男,1957年生,博士,教授,博士生导师,研究领域为先进制造技术.

Keywords Job-Shop; virtual manufacturing tree; priority; standby operation set; dynamic adjustment

1 引言

理论上已经证明了作业车间调度问题是 NP-Hard 问题. 本文所研究的问题是在满足 3 个约束条件的前提下使分批加工的作业所用的总时间尽可能的少,这 3 个约束条件是:(1)对于每道工序来说,必须等待其所有的前继工序加工完毕后才能加工此道工序;(2)对于每台机器来说,任意时刻只能加工一道工序,且必须满足加工完一道工序后才能加工另一道工序^[1];(3)不存在相同设备. 本文首先对单个作业的加工环节进行分析,在遵循机器忙原则基础上提出了四种调度策略. 然后提出了一种构造方法,把动态作业集中的作业构造成为一个虚拟的加工树. 在调度过程中根据虚拟加工树的加工流程动态地构造一个可供调度的工序集,同时应用所提出的 4 种调度策略调度工序集中的工序. 最后给出了完整的算法,实例比较表明此算法可以得到较好的解.

2 动态 Job-Shop 问题描述

初始时,作业集合 $P^0 = \{P_1^0, P_2^0, \dots, P_k^0\}$, t_i 时刻,作业集合 $P^i = \{P_{i1}^0, P_{i2}^0, \dots, P_{ij}^0\}$, 且 $P^i \subseteq P^0$, 同时又有新的作业 $P_1^i, P_2^i, \dots, P_l^i$ 加入到作业集合 P^i 中,使 $P^i = \{P_{i1}^0, P_{i2}^0, \dots, P_{ij}^0, P_1^i, P_2^i, \dots, P_l^i\}$. 对于任意时刻作业集中的某个作业 P_i ,若 P_i 中有工序 $p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{il}$ ($l=1, 2, \dots$),且满足 $p_{ij} < p_{ik}$ ($j, k=1, 2, \dots, l$), $<$ 为一个加工顺序的偏序关系^[2],即问题中的第一个约束条件;当加工机器集合 $M = \{M_1, M_2, \dots, M_i, \dots, M_k\}$ 时,结合第二个约束条件 M_i 为空闲机器可加工工序;由于在 M_i ($i=1, 2, \dots, k$) 上加工的工序是已知的,设 T_i 为在 M_i 上加工完最后工序的时间,则问题的解可表示为

$$T = \min\{\max\{T_1, T_2, \dots, T_i, \dots, T_k\}\}.$$

3 问题分析

定义 1(加工树). 单个作业的加工流程图为树形结构,这里称之为加工树.

规定:加工树上的节点代表工序,边代表偏序关

系. 显然,叶节点工序为初始时可被加工工序,根节点工序为最后被加工的工序. 当根节点工序加工完毕,表明此作业加工完毕.

初始时,作业集中包含 k 个作业. t_i 时刻,有的作业可能已经加工完毕,还有的作业部分加工,此时作业集中剩余 j 个作业,由文献[3]知作业部分加工剩余工序仍为树状结构,说明作业集中有 j 个加工树. 若此时又有 l 个作业加入到作业集中,则作业集中包含 $j+l$ 个作业. 尽管每个作业的加工流程各不相同,但是它们的加工流程都属于同一种类型的加工结构,即树形结构;如果把每个作业都看作是一个更大的虚拟加工树的根子树,则可以把问题简化为加工单个作业. 因此,应该首先分析单个作业的加工流程.

由于在此问题中相同的机器只有一台,所以很有可能出现多个工序争夺机器的情况. 假设在某一时刻有 m 个工序争夺机器,则将这些工序调度到机器上的排列一共有 $m!$ 种,从而使总的加工时间有 $m!$ 种. 选择哪个排序使加工单个作业所用的时间较少是本文首先要研究的内容. 下面对加工树进行了具体的分析,并提出具体的调度策略.

3.1 工序类型分类

由加工树和问题的约束条件可知,初始时叶节点工序是被调度的对象,而非叶节点工序肯定不会被调度. 所以,初始时只能调度某个叶节点工序. 当某个工序的所有子节点工序全部加工完毕,则此工序就转换为叶节点工序.

定义 2(可调度工序). 设 $p_{l1}, p_{l2}, \dots, p_{lm}$ 为叶节点工序,而且都需要在同一台机器 M_i 上加工. 当 M_i 空闲时,称 $p_{l1}, p_{l2}, \dots, p_{lm}$ 为可调度工序.

定义 3(准可调度工序). 设 $p_{l1}, p_{l2}, \dots, p_{lm}$ 为叶节点工序,而且都需要在同一台机器 M_i 上加工. 当 M_i 忙时,称 $p_{l1}, p_{l2}, \dots, p_{lm}$ 为准可调度工序.

定义 4(不可调度工序). 称所有非叶节点工序为不可调度工序.

定义 5(备选工序集). 所有可调度工序和准可调工序的工序集.

定义 6(直接后继工序). 若满足 $p_{ij} < p_{ik}$,则称 p_{ik} 为 p_{ij} 的直接后继工序.

3.2 调度策略设计

在加工过程中尽可能地使不同的机器同时工作

可以缩短总的加工时间,但是还应该遵循一个原则——机器忙原则,即只要存在可调度工序便根据调度策略调度某个工序,尽量使机器不停地工作.由于工序是从备选集中调度的,所以很可能出现一种情况,就是有多多个可调度工序同时争夺一台机器.此时,若只遵循机器忙原则并不能保证使总的加工时间较少.下面提出了4种调度策略,通过分析和实例验证可以看出,综合这几种调度策略可以较好地解决竞争问题,而且减少了总的加工时间.

3.2.1 优先级策略

首先,为工序设置优先级.设加工树为 m 层,则将根节点工序的优先级设置为1,其所有子节点工序的优先级设置为2,依此类推.规定根节点工序的优先级最低, m 层上的工序的优先级最高^[3].若存在可调度工序 $p_{i1}, p_{i2}, \dots, p_{im}$,优先级分别为 L_1, L_2, \dots, L_m ($L_i \neq L_j$, 当 $i \neq j$),且它们都需要在机器 M_i 上加工.设 p_{ij} 为优先级最小的工序,其优先级为 L_j ,从 p_{ij} 到根节点的路径上的工序序列为 $p_{ij}, p_{ij}^1, p_{ij}^2, \dots, p_{ij}^r$; p_{ik} 为优先级次小的工序,其优先级为 L_k ,从 p_{ik} 到根节点的路径上的工序序列为 $p_{ik}, p_{ik}^1, p_{ik}^2, \dots, p_{ik}^r$.由于 $L_j < L_k$,如果优先调度 p_{ij} ,则很可能使 p_{ij} 路径上剩余的工序与 p_{ik} 路径上的工序可并行加工的工序数减少,从而使并行加工的时间减少,使总的加工时间增加^[4].最极端的情况会是这样:当调度完 p_{ij} 后, p_{ij} 这条路径上没有工序可以与 $p_{ik}, p_{ik}^1, p_{ik}^2, \dots, p_{ik}^r$ 上的工序并行加工,用于加工这两条路径上的工序的时间增加了 p_{ij} 加工时间 t_{ij} .如果优先调度 p_{ik} ,由于 p_{ik} 的优先级较高,当调度完 p_{ik} 后,一定不会使路径 $p_{ij}, p_{ij}^1, p_{ij}^2, \dots, p_{ij}^r$ 和路径 $p_{ik}, p_{ik}^1, p_{ik}^2, \dots, p_{ik}^r$ 上可并行加工的工序数减少,这样可以保证不会使总的加工时间增加.对于优先级更高的工序 $p_{ix} \in \{p_{i1}, p_{i2}, \dots, p_{im}\}$ 和 p_{ik} ,由以上分析可以看出,应该优先调度 p_{ix} .由此可知,对于 $p_{i1}, p_{i2}, \dots, p_{im}$,应该优先调度优先级最高的工序.

3.2.2 短用时策略

若存在可调度工序 $p_{i1}, p_{i2}, \dots, p_{im}$,它们的优先级相同,都为 L_i ,而且都需要在同一台机器 M_i 上加工,用时分别为 $t_{i1}, t_{i2}, \dots, t_{im}$.在这种情况下, $p_{i1}, p_{i2}, \dots, p_{im}$ 在 M_i 上一定是串行加工的,又由机器忙原则和优先级策略可知加工是紧密的,即加工完一个工序后立刻加工另一个工序.此时可以理解为在 M_i 上加工一个用时为 $\sum_{j=1}^m t_{ij}$ 的工序.所以,无论先加工哪个工序都不会影响与正在其它机器上加工的工

序的并行加工时间.又设 $p_{ij} \in \{p_{i1}, p_{i2}, \dots, p_{im}\}$, $t_{ij} = \min\{t_{i1}, t_{i2}, \dots, t_{im}\}$, p_{ij}^1 为 p_{ij} 的直接后继工序.由于 p_{ij} 的用时最少,优先调度它有两个明显的优点.首先,调度完 p_{ij} ,可以使其后继工序 p_{ij}^1 获得比较早的可被加工的时间^[5-6].其次,对于集合 $\{p_{i1}, p_{i2}, \dots, p_{im}\} - \{p_{ij}\}$ 中的工序来说,也可以比较早获得机器 M_i ,从而使其后继的工序也能够较早被加工,这样就可以缩短总的加工时间.综合以上分析,当工序具有相同的优先级时应该调度用时少的工序.

3.2.3 长路径策略

定义7(路径). 设 n_1, n_2, \dots, n_r 为加工树上的节点, n_1 为叶节点, n_r 为根节点, n_{i+1} 是 n_i 的父节点,称此节点序列为路径.

定义8(路径长度). 设路径 n_1, n_2, \dots, n_r ,加工 r 个节点上的工序需要时间为 t_1, t_2, \dots, t_r ,称 $T = \sum_{i=1}^r t_i$ 为 n_1 的路径长度.

若存在可调度工序 $p_{i1}, p_{i2}, \dots, p_{im}$ 满足以下条件:它们的优先级相同,都为 L_i ;都需要在同一台机器 M_i 上加工;用时分别为 $t_{i1}, t_{i2}, \dots, t_{im}$,而且 $t_{i1} = t_{i2} = \dots = t_{im}$,设从 p_{is} ($s=1, 2, \dots, m$)到根节点的路径上的工序序列为 $p_{is}, p_{is}^1, p_{is}^2, \dots, p_{is}^r$,用时分别为 $t_{is}, t_{is}^1, t_{is}^2, \dots, t_{is}^r$, p_{is} 的路径长度为 $T_{is} = \sum_{k=0}^r t_{is}^k$ (其中 $t_{is}^0 = t_{is}$), p_{is}^1 的路径长度为 $T_{is}^1 = \sum_{k=1}^r t_{is}^k$,由于 $t_{ij} = t_{ik}$ ($j, k=1, 2, \dots, m$),所以无论优先调度哪个工序都不会使后续工序的调度时间提前,但是可以影响总的加工时间,因为总的加工时间取决于并行加工的时间,并行加工的时间越长,总的加工时间就会缩短,反之,就会增加^[2,7-8].又因为并行加工的时间取决于短路径,所以当优先调度 p_{ij} ($T_{ij} = \min\{T_{i1}, T_{i2}, \dots, T_{im}\}$),后面工序的并行时间便取决于 T_{ij}^1 .而 $T_{ij}^1 = \min\{T_{i1}^1, T_{i2}^1, \dots, T_{im}^1\}$,也就是说后续工序并行加工的时间会最短.这样会使整个加工的时间增加.所以,在这种情况下应该优先调度路径较长的工序.

3.2.4 动态调整策略

若存在不可调度工序 p_{ij} ,优先级为 L_j ;可调度工序 p_{ik} ,优先级为 L_k , p_{ij} 和 p_{ik} 需要在 M_i 上加工,且满足 $L_j \geq L_k + 1$, p_{ik} 是可调度工序,所以 $p_{ik} \in$ 备选工序集,而且根据机器忙原则,此时应该调度 p_{ik} .但是由于存在 p_{ij} ,如果调度 p_{ik} 后会出现两种情况:第一,当 p_{ik} 加工完毕时, p_{ij} 仍为不可调度工序,也

就是说 p_{ij} 不在备选工序集中; 第二, 在 p_{ik} 加工完毕之前, p_{ij} 成为准可调度工序, 此时 p_{ij} 被放置到备选工序集中. 对于第一种情况, 在 p_{ik} 加工完毕之前没有优先级更高的可调度工序, 而且根据机器忙的原则, 应该调度 p_{ik} ; 对于第二种情况, 若仍然先调度 p_{ik} , 则在 p_{ik} 加工完毕之前的某一时刻, p_{ij} 成为了准可调度工序, 此时违背了优先级策略. 所以, 对于第二种情况应该使 p_{ik} 等待, 不予调度.

3.3 构造虚拟加工树

定义 9(虚拟加工树). 采用某种构造方法将所有作业构造成为一个加工树, 称这个构造的加工树为虚拟加工树.

定义 10(虚拟根节点). 构造虚拟加工树的过程中添加的根节点.

对于作业集合 $P^0 = \{P_1^0, P_2^0, \dots, P_k^0\}$, 本文采取根对齐的方式构造虚拟加工树, 即将每个作业的加工树作为虚拟根节点的根子树. 虚拟根节点的优先级设置为 0, 它的优先级最低, 每个子树的根节点的优先级设置为 1, 依此类推. 这样就将多个作业的问题简化成为单个作业的问题. 显然, 虚拟加工树包含所有待加工的作业, 而且这个虚拟加工树满足单个作业的性质, 所以调度工序时完全可以应用以上提出的四种调度策略调度虚拟加工树上的工序. 对于 t_i 时刻, 当作业 $P_1^i, P_2^i, \dots, P_l^i$ 加入到作业集合以后, 可以按照上述的构造方法重新构造虚拟加工树, 然后调度这个新的虚拟加工树.

4 算法设计

算法设计思想如下.

1. 根据作业集合构造虚拟加工树.
2. 根据虚拟加工树的结构为工序设置优先级.
3. 根据虚拟加工树生成备选工序集. 若工序集为空, 则转到步 6.
4. 从备选工序集中选取优先级最高的可调度工序. 若工序唯一, 则选取此工序, 若工序不唯一, 则选取用时最少的工序; 若用时最少的工序不唯一, 则选取路径最长的工序.
5. 若选取的工序满足动态调整条件或者在所有需要在同机器上加工的工序中优先级最高, 则调度此工序, 然后从加工树上删除此节点, 从备选工序集中删除此工序. 如果有新的作业加入到集合中, 则转到步 1; 否则, 转到步 3.
6. 结束.

注. 选取不是调度, 步 4 是选取的过程, 步 5 是决定是否调度的过程.

所设计的算法流程图见附录.

5 算法复杂度分析

设所有作业的工序总数为 m , 则虚拟加工树上有 $m+1$ 个节点. 初始时, 备选工序集中最少有 1 个工序, 最多有 $m-1$ 个工序. 这里只分析备选集中包含工序最多的情况. 因为初始时备选集中有 $m-1$ 个工序, 则最极端的情况是: 调度第一个工序需要比较 $4(m-2)$ 次, 调度第二个工序需要比较 $4(m-3)$ 次, 当备选集中只包含最后两个工序时需要比较 1 次. 所以算法共需要比较次数为: $4[(m-2) + (m-3) + \dots + 2 + 1] = 2m^2 - 6m + 4$, 所以算法的复杂度为 $O(m^2)$.

6 实例对比

初始时, 作业 A 对应的加工树如图 1 所示. 当调度进行到 55 时, 作业 B 加入到作业集合中, 其对应的加工树如图 2 所示. 图 3 为 B 加入到作业集合

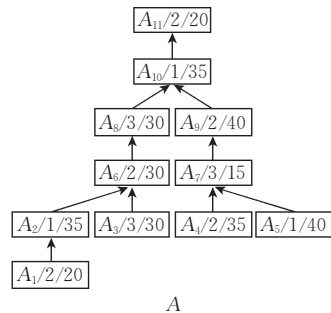


图 1 作业 A 对应的加工树

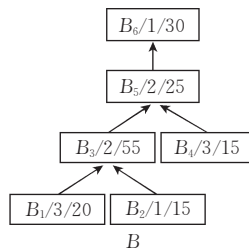


图 2 作业 B 对应的加工树

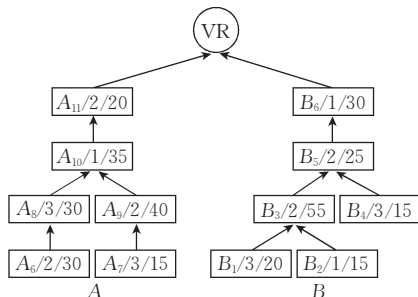


图 3 当作业 B 加入到作业集合以后重新构造的虚拟加工树

以后重新构造的虚拟加工树. 分别采用: ① 基于 ACPM 和 BFSM 的动态 Job-Shop 调度算法^[8]; ② 本文提出的算法调度作业 A 和作业 B. 图 4 和图 5 分别为两种算法生成的 Gantt 图. 结果显示, 算法②

比算法①缩短了 5 个单位时间. 由于算法①对于求解动态 Job-Shop 问题已经较优了, 能够在此基础上得到更优的解, 说明本算法调度优化更进一步.

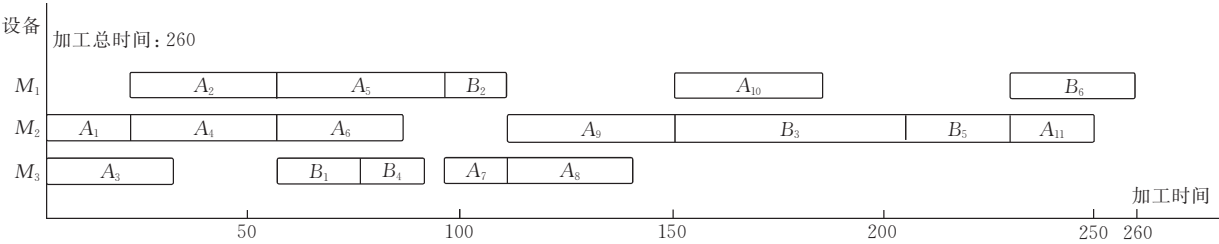


图 4 应用算法①加工完所有作业的时间(260)

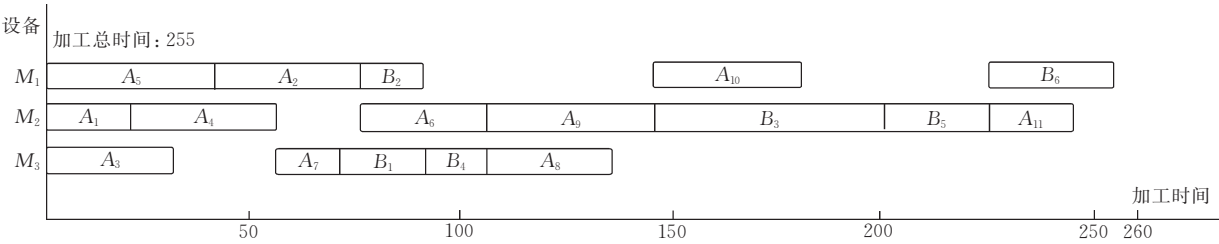


图 5 应用本文算法加工完所有作业的时间(255)

7 结 论

通过上面的分析以及实例的验证, 可以看出本文所提出的算法对于动态 Job-Shop 问题能够得到较优的解. 由于本文提出的算法在调度工序的过程中动态地生成备选工序集, 除非工序被调度到机器上, 否则工序不会被调出备选工序集, 所以本文提出的算法能够避免操作系统调度中的抖动问题, 且具有一定的理论和实用价值.

参 考 文 献

[1] Wang Lei, Huang Wen-Qi. A new local search algorithm for Job Shop scheduling problem. Chinese Journal of Computers, 2005, 28(5): 60-67(in Chinese)
(王磊, 黄文奇. 求解工件车间调度问题的一种新的邻域搜索算法. 计算机学报, 2005, 28(5): 60-67)

[2] Qiao Pei-Li, Xie Zhi-Qiang. Classified and grouped Job-Shop scheduling algorithm based on key equipments' compact procedures. Chinese Journal of Mechanical Engineering, 2004, 40(8): 17-21(in Chinese)
(乔佩利, 谢志强. 基于关键设备工序紧凑的工序分类、分批的 Job-Shop 调度算法. 机械工程学报, 2004, 40(8): 17-21)

[3] Xie Zhi-Qiang, Yang Guang, Tan Guang-Yu. An algorithm of JSSP with dynamic collection of job with priority//Proceedings of the International Technology and Innovation Conference 2006-Advanced Manufacturing Technologies. Hangzhou, China, 2006: 106-111

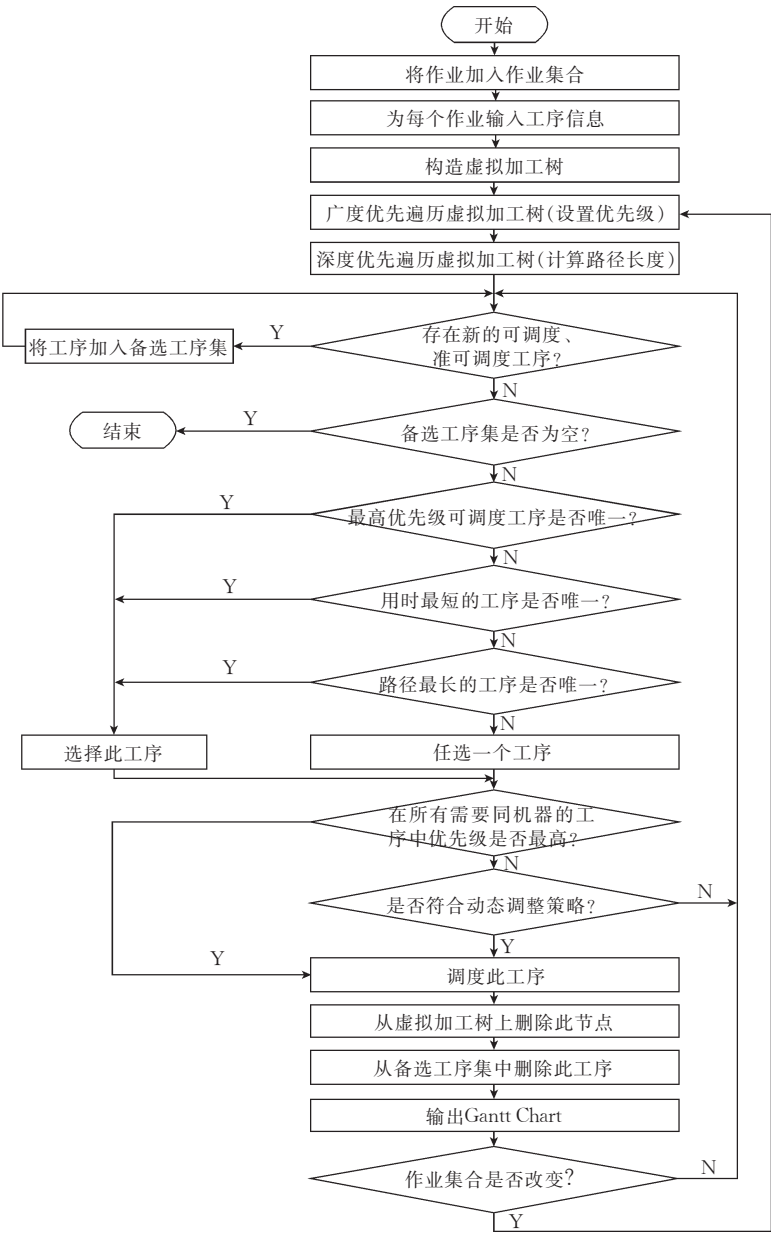
[4] Dominic P D D, Kaliyamoorthy S, Murugan R. A Conflict-Based Priority Dispatching Rule and Operation-Based Approaches to Job Shops. London: Springer, 2004: 76-80

[5] Brucker P, Heitmann S, Hurink J, Nieberg T. Job-Shop scheduling with limited capacity buffers. OR Spectrum, 2006, 28(2): 151-176

[6] Zhang De-Fu, Li Xin. Fast heuristic algorithm for Job Shop scheduling problem. Computer Integrated Manufacturing Systems-CIMS, 2005(2): 89-93(in Chinese)
(张德富, 李新. 求解作业车间调度问题的快速启发式算法. 计算机集成制造系统-CIMS, 2005(2): 89-93)

[7] Xie Zhi-Qiang, Cong Jing, Yang Jing. Scheduling algorithm of special assembly problems based on classifying operations//Proceeding of the International Conference on Advanced Design and Manufacture (ADM2006). Harbin, China, 2006: 337-340

[8] Xie Zhi-Qiang, Liu Sheng-Hui, Qiao Pei-Li. Dynamic Job-Shop scheduling algorithm based on ACPM and BFSM. Journal of Computer Research and Development, 2003, 40(7): 79-85(in Chinese)
(谢志强, 刘胜辉, 乔佩利. 基于 ACPM 和 BFSM 的动态 Job-Shop 调度算法. 计算机研究与发展, 2003, 40(7): 79-85)



附图 算法流程图



XIE Zhi-Qiang, born in 1962, Ph. D. candidate, professor. His research interests include CIMS and scheduling optimization etc.

supervisor. Her research interests include data mining and scheduling optimization etc.

YANG Guang, born in 1982, M. S. candidate. His research interests include scheduling optimization etc.

TAN Guang-Yu, born in 1957, Ph. D. , professor, Ph. D. supervisor. His research interests include advanced manufacturing technology etc.

YANG Jing, born in 1962, Ph. D. , professor, Ph. D.

Background

According to statistic, the cost about processing procedure of product, comparing with the total cost of product, is more than 40 percent in the manufacturing. So it is necessary

to optimize the processing procedure of product, it can not only improve production efficiency and reduce cost, but also make corporation arrange production effectively and quickly

and avoid risk for corporation. And it is also according with the fact that country advocates to develop the manufacturing greatly. Now the optimization of job set with non-constraint is studied mainly in abroad, but our focus is the optimization of job set with constraint. This kind of problem is based on the idea to decompose manufacturing tree of job proposed by myself, and to realize dynamic optimization procedure of job set with constraint by constructing virtual manufacturing tree and confirm the priority of operations. This study is supported by the National Natural Science Foundation of China (No. 50575062), the Natural Science Foundation of Heilongjiang Province (No. F0309, F200608), Scientific Research

Fund of Heilongjiang Provincial Education Department (No. 10551z0008), the Key Project of Science and Technology of Harbin (No. 2005AA1CG061-11) and the Key Project of Scientific Research Subsidy of Abroad Scholars of Heilongjiang Provincial Education Department (No. 1152hq08). This study was started in 2003, the corresponding content about 15 papers are published by Journal of Computer Research and Development, Computer Integrated Manufacturing Systems, Chinese Journal of Mechanical Engineering and so on and international conference organized by IEEE and IET, and 4 papers of them have been accepted by EI.