

文章编号: 1002-0446(2002)04-0342-04

## 含机器人的作业车间双资源智能优化调度\*

孙志峻 朱剑英

(南京航空航天大学机电工程学院 南京 210016)

**摘要:** 用遗传算法研究了含机器人的作业车间双资源调度优化问题, 提出了一种将遗传算法和分派规则相结合的调度算法, 将加工机床和机器人合理地分配给加工任务, 使评价指标获得最优. 最后给出了此调度算法的仿真结果, 证明该算法是可行的, 并获得良好的结果.

**关键词:** 车间调度; 遗传算法; 机器人

**中图分类号:** TP24 **文献标识码:** B

## A GENETIC ALGORITHM BASED APPROACH TO INTELLIGENT OPTIMIZATION FOR SCHEDULING OF DUAL-RESOURCES WITH ROBOTS

SUN Zhijun ZHU Jianying

(Mechatronic Engineering Institute Nanjing University of Aeronautics & Astronautics, Nanjing 210016)

**Abstract:** Based on genetic algorithms, dual-resource constrained job-shop-scheduling problem with robots is studied. A new scheduling algorithm is proposed, which combines genetic algorithms and dispatching rules. It reasonably assigns the resources of machines and robots to jobs. Finally, a simulation example is presented to illustrate the feasibility of the algorithm and it is confirmed that the results are good.

**Keywords:** job shop schedule, genetic algorithms, robot

### 1 引言( Introduction)

到目前为止, 大量的关于作业车间调度的文献资料只考虑单资源(机床)的情况. 国际上有少量学者研究了由于工人的技术熟练程度与工作能力的不同, 不同工人操作设备的种类和数量也是不同的情况. 在这种情况下, 一个工人需要操纵数台机床设备. 一般地, 对于车间加工能力可能受到机床和工人同时制约的这种系统, 称其为双资源作业车间(dual-resources constrained job-shop, DRC)<sup>[1]</sup>. 但与生产发达国家相比, 我国的人工费低廉, 不太会出现一个工人管理数台机床的情况; 相反, 由于设备购置和管理费用居高不下, 不可能为每台机床配备一套完整的工装设备.

车间作业调度问题在数学上属于 NP 完备问题, 求其最优解是数学尚未解决的难题. 对于双资源的

调度问题, 就是难上加难的问题. 在最近十年中, 有一些学者用模拟和启发式方法研究双资源作业车间调度问题. 一些决策规则被用在模拟方法中, 以分配工人<sup>[2,3]</sup>. 但是, 采用模拟方法要得到一个最优解, 需要花费较长的时间; 另一方面, 由于采用局部贪婪启发式策略, 很可能陷入局部最优的陷阱<sup>[4]</sup>. 逐渐地, 随着遗传算法在传统的旅行商问题求解的成功运用, 学者们开始把注意力集中到遗传算法, 用它来解决制造系统中的计划和调度问题<sup>[5-9]</sup>.

本文对含机器人的作业车间调度问题进行了研究, 并提出了解决方法. 由于机器人属于贵重设备, 不可能为每台机床配备一台, 这里设定一台机器人为三台机床服务. 机器人的作用是在加工之前和加工之后, 负责工件的上料和下料. 类似这样的机器人代表了一类生产资源, 它们与操作工人这种生产资

\* 基金项目: 国家自然科学基金重大项目(59990470)资助.

收稿日期: 2001-09-10

源的不同之处在于: 在一道工序的加工过程当中, 操作工人作为资源被占用的时间与加工时间一致, 操作工人在这段时间内不可能为其他工序服务; 而机器人作为资源被占用的时间被分成两段, 分别是上料时间和下料时间, 但机器人在该工序加工过程中, 完全可能被另一道工序征用. 如文中所述的机器人所代表的生产资源还有物料运输车等等.

本文基于遗传算法提出了一种调度算法, 来有效地分配机床和机器人以达到某种最佳评价指标, 如生产周期.

## 2 双资源调度问题描述 (Description of dual-scheduling problem)

本文的作业车间调度问题中, 有四个不同类型的工件, 标记为  $J_i (i= 1, 2, 3, 4)$ ; 且待加工工序都是三道, 每道工序均可被不止一台的机床加工. 这与古典作业车间调度问题有着本质的不同: 在古典作业车间调度中, 每个工序使用的机床(唯一)被事先确定. 这样, 实际上是先制定加工计划, 然后进行生产调度. 其结果必然导致加工计划和生产调度相脱节, 而且也不符合实际加工情况. 在实际的加工中, 一道工序应该可以在多台机床上被加工, 不应该硬性规定只能在某一台机床上被加工. 在本调度问题中, 允许工序可以由一个机床集合中的多台机床完成加

工. 由于一道工序可以在多台机床上被加工, 在调度过程中就必须根据当时加工车间的实际情况, 为工序选择合适的机床和机器人, 以满足某种评价指标. 这个过程是本作业车间调度的核心和难点, 也是古典作业车间调度所欠缺的.

车间的生产能力受到机床设备和机器人这两种生产资源的制约. 各种不同类型的机床设备总数是六台, 标记为  $M_k (k= 1, 2, \dots, 6)$ ; 一台机器人需要为三台机床服务, 进行加工前的上料和加工后的下料, 机器人总数为两台, 标记为  $R_r (r= 1, 2)$ . 调度的目标是寻找一个可行的调度方案, 使得在给定的约束条件下, 全部工序加工完毕的生产周期最短.

每台机器人为机床服务的信息由机器人工作表提供, 如表 1 所示; 四个不同类型工件的加工时间以及每道工序的上下料时间信息由加工时间/上下料时间表提供, 如表 2 所示; 从表中可以看到工序在不同机床设备上的加工时间各不相同.

表 1 机器人工作表

**Table 1 Working capability of robots**

R	M1	M2	M3	M4	M5	M6
R1	able	able	able	-	-	-
R2	-	-		able	able	able

表 2 加工时间/上下料时间表

**Table 2 Period of time for machining and loading /unloading for robots**

Job	Oper	M1	M2	M3	M4	M5	M6	L	UnL
J1	1- 1	2	3	4	-	-	-	0.2	0.1
	1- 2	-	3	-	2	4	-	0.2	0.1
	1- 3	1	4	5	-	-	-	0.4	0.3
J2	2- 1	3	-	5	-	2	-	0.3	0.2
	2- 2	4	3	-	-	7	-	0.4	0.3
	2- 3	-	-	4	-	7	11	0.4	0.3
J3	3- 1	5	6	-	-	-	-	0.3	0.2
	3- 2	-	-	-	4	3	5	0.3	0.2
	3- 3	-	-	13	-	9	12	0.3	0.2
J4	4- 1	9	-	7	9	-	-	0.4	0.3
	4- 2	-	6	-	4	-	5	0.4	0.3
	4- 3	1	-	3	-	-	3	0.5	0.4

## 3 算法实现(Implementation of algorithm)

遗传算法是基于群体进化的一种方法, 它的最大优势在于它有能力在各种调度方案之间随机地选择, 可以跳出局部最优的陷阱; 而其它算法则常常会陷入局部最优.

### 3.1 编码

选择一种合适的染色体基因型是应用遗传算法寻优的第一步. 在文献[10]中分析了多种染色体基因型的优劣. 考虑到在调度之前, 无法确定工件具体在哪台机床上被加工的特点, 采用了基于工序的表达法. 给每一个工件一个十进制的编号, 如第二个类型工件的编号就是“2”; 工件的工序编号则是两位十

进制编号,如第二个类型工件的三道工序的编号分别是“21”,“22”和“23”.在染色体编码中,给所有同一工件的工序指定相同的符号,即工件编号,根据它们在给定染色体中出现的先后顺序就可以确认是第几道工序.如表 3 所示.工序“21”,“22”和“23”的编

码都是“2”,由于工件的加工工序是确定不变的,因此首先出现的“2”代表“21”,第二次出现的“2”代表“22”,第三次出现的“2”代表“23”.很容易看出染色体的任意排列总能产生可行调度,而且可以肯定这种编码方式一定含有最优调度.

表 3 调度的染色体基因型

Table 3 Genotype of chromosome for scheduling

工序	21	11	31	41	42	32	22	23	33	12	13	43
编码	2	1	3	4	4	3	2	2	3	1	1	4

在该编码中,不包含机器人信息,这样可以使染色体的交叉和变异容易操作,节省大量的计算机运算时间.在文献[9]的双资源调度中,将工人的加工信息也放入编码中,一方面编码形式复杂;另一方面使得交叉和变异变得非常困难,在出现不合理染色体后,修复过程要考虑工序和工人两方面的因素,操作规则十分繁杂,耗费大量运算时间.这里,机器人信息将在以下的解码算法中体现,它极大地简化了交叉和变异操作.

3.2 解码算法描述

先说明在文中出现的标号:

$i$ ——工件号 ( $i=1, 2, \dots, n$ );

$k$ ——机床号 ( $k=1, 2, \dots, m$ );

$w$ ——工人号 ( $w=1, 2, \dots, d$ );

$M_k$ ——可以使用机床  $k$  的时刻(机床  $k$  在此刻开始空闲);

$P_w$ ——可以使用工人  $w$  的时刻(工人  $w$  在此刻开始空闲);

$R_r$ ——机器人  $r$  工作时间区间集合;

$O_{ij}$ ——工件  $i$  的第  $j$  道工序;

$T_{ij}$ ——工件  $i$  的第  $j$  道工序可被加工的时刻;

$J_i$ ——工件  $i$  的总工序号;

$t_{ijk}$ ——在机床  $k$  上加工工序  $O_{ij}$  的所需时间;

$L_{ij}$ ——工件  $i$  的第  $j$  道工序的安装时间;

$U_{ij}$ ——工件  $i$  的第  $j$  道工序的拆卸时间;

$f_{ij}$ ——在机床  $k$  上完成工件  $i$  的第  $j$  道工序的时刻;

$E_{ij}$ ——工件  $i$  的第  $j$  道工序的最早完工时刻;

$F_i$ ——工件  $i$  的完工时刻;

$S$ ——一种调度方案中所有未被调度的工序集合;

$L(k, 1)$ ——机器人在机床  $k$  上开始安装工件

( $O_{ij}$ ) 的理想时刻;

$L(k, 2)$ ——机器人在机床  $k$  上完成安装工件 ( $O_{ij}$ ) 的理想时刻;

$U(k, 1)$ ——机器人在机床  $k$  上开始拆卸工件 ( $O_{ij}$ ) 的理想时刻;

$U(k, 2)$ ——机器人在机床  $k$  上完成拆卸工件 ( $O_{ij}$ ) 的理想时刻.

解码算法过程:

Step 1: 计算所有工序的理想可被加工起始时刻

$$\text{设 } M_k = 0, P_w = 0, R_r = (0, 0), T_{i1} = 0$$

$$T_{ij} = L_{i(j-1)} + T_{i(j-1)} + t_{i(j-1)k} + U_{i(j-1)}$$

$$(i = 1, 2, \dots, n \text{ and } j = 2, \dots, J_i)$$

Step 2: 如果在集合工序中还有未被调度的工序,从集合中取出此刻排在第一位尚未被调度的工序转 Step 3

否则,设  $\text{makespan} = \max(F_i) (i = 1, 2, \dots, n)$

$$\text{Step 3: } L(k, 1) = \max\{M_k, P_w, T_{ij}\}$$

$$L(k, 2) = L(k, 1) + L_{ij}$$

如果  $(L(k, 1), L(k, 2)) \in R_r$ , 则调整  $L(k, 1)$  和  $L(k, 2)$

$$U(k, 1) = L(k, 2) + t_{ijk}$$

$$U(k, 2) = L(k, 1) + U_{ij}$$

如果  $(U(k, 1), U(k, 2)) \in R_r$ , 则调整  $U(k, 1)$  和  $U(k, 2)$

$$f_{ijk} = U(k, 2)$$

Step 4: 设  $E_{ij} = \min(f_{ijk})$ .

如果  $j = J_i$ , 则  $F_i = E_{ij}$

否则,计算  $T_{i(j+h)} = T_{i(j+h)} + (E_{ij} - T_{i(j+1)})$ , ( $h = 1, 2, \dots, (J_i - j)$ )

Step 5: 设  $M_k = E_{ij}, P_w = E_{ij}$

把  $(L(k, 1), L(k, 2))$  和  $(U(k, 1), U(k, 2))$  并入相应的集合  $R_r$

Step 6: 检查集合  $S$  中是否还有未被调度的工序; 如果还有, 则转 Step 2. 否则,  $makespan(\text{生产周期}) = \max(F_i)$ .

在每代中, 该算法将机床和机器人分配给各工序, 并确保与机器人的工作时间没有干涉.

### 3.3 选择、交叉和变异

选择阶段由两部分组成:

(1) 适值标定. 采用线性标定法,  $f = (a - makespan) * b$ , 由于目标是寻求最小的生产周期, 为了便于概率选择, 所以在公式中用一个较大的正整数  $a$  减去生产周期, 其结果乘以常数  $b$ , 是为了能更好地产生指定的数个后代.

(2) 确定染色体的期望值, 并将期望值转化为后代个数. 采用 Holland 正比选择法生成生存概率, 即  $P_s = f_i / \sum f_i$ . 然后, 用 Baker 提出的随机通用采样法只旋转一次转轮而得到后代数.

交叉操作是遗传算法中关键的一步, 它可以将父代的良好基因通过信息互换, 而产生更好的子代. 然而, 众所周知, 对于调度和旅行商问题, 交叉操作是最棘手的难题. 考虑到染色体基因型的实际意义,

采用随机两点交叉操作, 两个父代产生一个子代. 一般这样操作很容易产生不合理的染色体基因型. 这里, 采用了一种非常好的交叉操作方式<sup>[11]</sup>, 不仅可以完全避免不合理的染色体基因型的产生, 而且在操作时间上也大大缩短. 最后, 将两个父代与一个子代染色体比较, 选择最好的两个染色体放入种群中.

变异操作在本问题中, 相对而言比较简单. 在随机选择的染色体内随机交换两个基因, 形成一个新的染色体即可.

## 4 仿真结果与分析 (Simulation and conclusions)

本问题的评价指标是寻求最短生产周期, 因而采用的分派规则是 SPT (Shortest Processing Time). 使用的一些算法参数是: 种群个数是 50, 交叉率是 0.6, 变异率是 0.01. 适值计算用参数为:  $a = 35, b = 2$ .

由遗传算法获得的最佳调度图显示在图 1 和图 2 中.

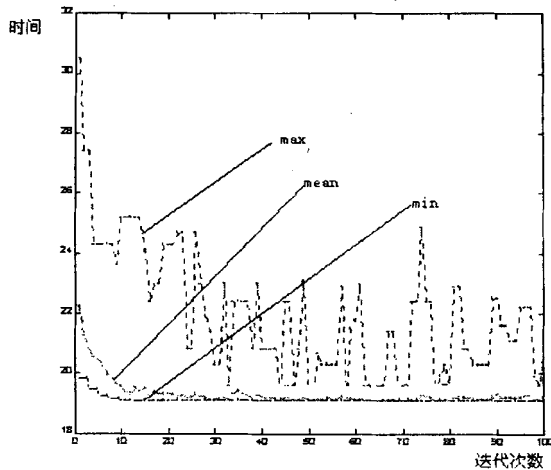


图 1 遗传算法曲线图  
Fig. 1 Curve graph using genetic algorithm

图 1 是遗传算法的收敛过程. 从图中可以看出, 目标函数的最小值 ( $min$ ) 随着群体的进化在减小, 最后收敛于极值的速度较快. 群体中目标函数的平均值 ( $mean$ ) 也随着群体的进化逐渐减小, 并趋向目标函数的最小值.

图 2 显示了工序与机床和机器人的分配关系, 其间还标明了机器人为每道工序服务的时间. 图中横坐标表明了这批工件加工的时间历程, 纵坐标表

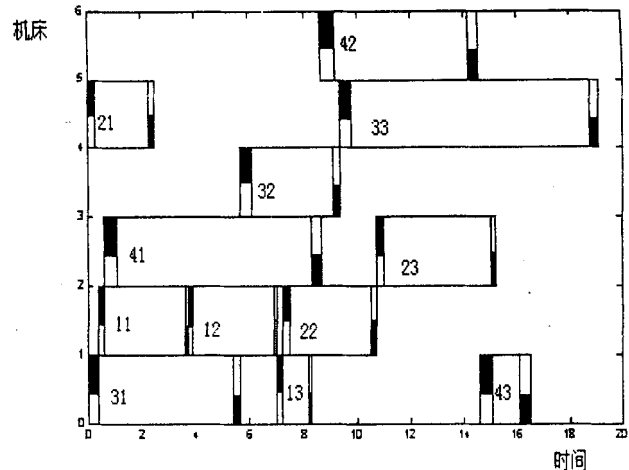


图 2 调度 Gantt 图  
Fig. 2 Gantt graph for schedule

明机床设备, 用两个字符标识的方框代表一道工序. 例如, 第一台机床加工的第一道工序是“31”, “3”表示该工件是第三个类型的工件; “1”表示这是该工件的第一道工序, 从横坐标上同样可以查出加工该工序的起始和完毕时刻. 在该工序的前后有两条被涂黑的细方框, 它表示了相关的机器人为该工序上下料的时间.