第 15 卷 第 6 期　　控 制 与 决 策　　2000 年 11 月
Vol. 15 No. 6　　CONTROL AND DECISION　　Nov. 2000

基于 GA 和 SA 的制造单元成组方法*

李　岩　吴智铭
(上海交通大学自动化研究所　200030)

摘　要　介绍一种综合 GA 和 SA 优点的制造单元成组方法。建立了可变工艺路径的单元成组问题的优化模型, 对 GA 和 SA 操作中的步骤及其相应于可变路径情况下的特殊性做出说明。阐述了基于并行 GA 搜索的 SA 算法, 并对制造单元的实例进行成组, 取得了令人满意的效果。

关键词　遗传算法, 模拟退火法, 制造单元, 成组
分类号　TP 273

GA/ SA-based Grouping Method for Manufacturing Cells

Li Yan, Wu Zhiming
(Shanghai Jiaotong University)

Abstract　A manufacturing cells grouping method combining the advantages of GA and SA is described. A solution scheme of cell formation is discussed and a model of this problem is set up. The procedure of this GA/ SA-based method and its features in application to multi-route environment is explained. A SA algorithm embedded in parallel searching of GA is expatiated. Some practical examples of grouping manufacturing cells are presented and the results are satisfactory.

Key words　genetic algorithms, simulated annealing, manufacturing cells, cell formation(CF)

1　引　言

成组技术(GT) 是一种利用零件加工过程中的相似性进行机器和零件的分组, 从而减少加工量, 节约劳动力, 缩短装夹时间和交货期, 缩小问题解空间的有效方法。GT 定义为: 在机器类型、容量和零件类型、产量和路径已知的条件下, 识别具有相似加工要求的零件簇和能够处理相应零件簇的机器簇。Burbidge 定义了 CF 问题, 利用基于 PFA 的启发式方法(如相似系数法[1]) 确实能够获得某些近优的解决方案, 但解的质量与初始解的设定等与人为因素密切相关。整数规划方法也被用于求解 CF 问题, Ballakur 和 Steudel[2]指出, 在极严格的限制下, CF 问题是一个 NP 难题, 当问题规模增加时, 用整数规划方法求解将相当困难。聚类分析法[3]的效果则因选用方法而异, 算法的鲁棒性不强。

制造单元成组过程中所要考虑的标准很多, 如单元划分过程中的异常元素(这里是指那些需要在零件所在单元外进行加工的工序) 最少, 单元间运送成本最小, 各单元间的负荷差异最小, 机器利用率最大等。而这样的目标通常难以兼顾, 因此需要在几个目标中做出适当的权衡。本文利用混合 GA 和 SA 的方法处理制造单元成组过程中的机器/ 零件分簇问题, 在考虑多种加工路径的前提下解决 CF 问题。该方法具有很强的实用性。

2　问题形成

表 1 给出一些符号及其定义。

假定零件 i 的加工工序形成一个有向图 G_i, 其中节点 $S = g(0)$ 和 $E = g(d+1)$, 由代表机器类型的节点和代表加工进程的有向弧组成。则工件 i 的有向图为

$$G_i = \{S, \{(p_i, g(p_i))\}, E, \{(\alpha, \beta_l)\}\}$$

* 国家自然科学基金项目(59889505、70071017)
1999-03-31 收稿, 1999-06-17 修回

表 1　符　号　定　义

符号	定　义	符号	定　义
K	期望设计的单元数(索引为 k)	G_i	工件 i 的可选路径有向图
n	零件类型数(索引为 i)	R_{iz}	通过 G_i 的第 z 条路径
N_i	零件 i 的数量	d_i	G_i 中的节点数
H	机器类型数(构成集合 $\boldsymbol{H}$, 索引为 h)	p_i	G_i 中标识节点的索引
O_{iz}	零件 i 在路径 z 时的加工工序数	f_{iz}	R_{iz} 的节点数(除去始点 S 和终点 E)
M	机器数, 所有机器构成集合 $\boldsymbol{M}$	l_{iz}	R_{iz} 的节点索引(除去始点 S 和终点 E)
C_M_k	单元 k 中最多的机器数	$\{g(\bullet)\}$	沿序列节点 $\{\bullet\}$ 的机器类型集
U	$=\sum_{i=1}^{n} u_i * N_i$, 总异常元素个数	u_i	零件类型 i 的单元外加工工序数, 或称异常元素个数
A	$=[a_{miz}]$, 机器 - 零件关联矩阵, $a_{miz}=1$ 表示第 i 种零件类型在选用第 z 条路径时需要经过机器 m, 否则 $a_{miz}=0$	AV_{ik}	$=\sum_{k=1}^{K}\sum_{m=1}^{M} T_{im} * x_{mk} \Big\backslash \sum_{m=1}^{M} x_{mk}$, 零件类型 i 在单元 k 内每台机器上的平均加工时间
w_i	指标 i 的权值	$f_{kk'}$	单元 k 与单元 k' 之间的运送成本
T_{imz}	零件类型 i 采用路径 z 时在机器 m 上的处理时间	θ_{il}	$=1$, 表示零件类型 i 的第 l 道和第 $l+1$ 道工序在同一单元内处理, 否则为 0
y_{ik}	$=1/0$, 表示零件类型 i 是否属于单元 k	x_{mk}	$=1$ 表示机器 m 属于单元 k, 否则 $=0$

$$p_i = 1, 2, \cdots, d_i$$

$$\alpha, \beta_l \in \{S, 1, 2, \cdots, d_i, E\}, l = 1, 2, \cdots\}$$

其中节点 S, $\{(p_i, g(p_i))\}$ 和 E 由 G_i 中的有向弧 $\{(\alpha, \beta_l)\}$ 相连。R_{iz} 定义为

$$\{S, g(l_{iz}) \in \boldsymbol{H}, E \big| l_{iz} = 1, 2, \cdots, f_{iz}\}$$

下面以一类零件的加工路径为例(见图 1) 来说明上述符号的定义。

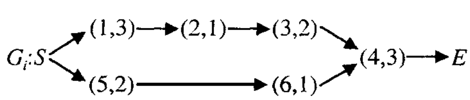

图 1　具有多重路径的零件加工有向图

图 1 中, $H=3$, $O_{i1}=4$, $O_{i2}=3$, $d_i=6$。对于 G_i, 有 $g(0)=S$, $g(7)=E$。对于 R_{i1}, 有以下节点: ① $p_i=1$, $g(1)=3$; ② $p_i=2$, $g(2)=1$; ③ $p_i=3$, $g(3)=2$; ④ $p_i=4$, $g(4)=3$。对于 R_{i2}, 有以下节点: ① $p_i=5$, $g(5)=2$; ② $p_i=6$, $g(6)=1$; ③ $p_i=4$, $g(4)=3$。

由以上的符号定义, 可以计算异常元素惩罚 U, 单元负荷不平衡度 V 可用下式计算

$$V = \sum_{i=1}^{n}\left(\sum_{m=1}^{M}\sum_{k=1}^{K}(T_{imk} - AV_{ik}) * y_{ik} * N_i\right)$$

假定零件类型 i 的第 l 道工序和第 $l+1$ 道工序分别表示在单元 k 和 k' 中进行单元间的运送成本, 则总的运送成本

$$E = \sum_{i=1}^{n}\left(\sum_{l=1}^{O_{iz}}(f_{kk'}(\alpha\theta_l + \beta(1-\theta_l)) * N_i)\right)$$

令 $\alpha+\beta=1$, $\alpha, \beta \in \{0, 1\}$, 如果 $\alpha=1$, 则计算结果为单元内的运送成本; 否则, 计算结果为单元间的运送成本。本文选择了不同的指标, 可由用户根据需要选取不同的目标函数。

3　算法描述

遗传算法(GA) 具有较强的并行搜索能力, 且能在一定程度上保留历史信息, 但交叉与变异只有有限的进化能力, 常出现过早收敛现象。模拟退火方法(SA) 模仿晶体的冷却过程, 虽然收敛速度较慢, 但 SA 的概率突跳性使其具有避免局部极小的能力。本文采用基于 GA 和 SA 的混合方法, 同时利用 GA 的并行搜索特征和 SA 的概率突跳能力帮助算法摆脱局部最优。下面给出算法的主要步骤:

Step 1: 确定 GA 和 SA 参数, 包括任务数据(机器类型数、零件类型数、每类机器零件数、单元数、每组最多机器数限制、零件加工路径、每道工序加工时间)、遗传代数 G、种群大小、交叉概率 P_c、变异概率 P_m、SA 中的初始温度 T_0 等;

Step 2: 形成染色体的初始解 S_0, 对每种零件类型都随机选择一条加工路径, 并随机将机器与零件分配给每个单元, 满足每种机器类型的容量约束;

Step 3: 评估适应度和选择, 计算每个个体的目标函数值及其适应度, 统计最优解与最差解;

Step 4: 如果迭代次数大于 δ, 则对每个个体执行 SA 操作, 否则只进行交叉和变异, 算法如下

```
SA{while( Tk ≠ 0)
    {S = S0
        Generate_ new_ solution(S')
        // 交叉、变异以产生新解
        Δ = fitness(S) - fitness(S')
```

$$\text{Sa-Prob} = \min(1, e^{-\Delta/T_k})$$
$$\text{If}(\text{Sa-Prob} > \text{random}(0,1))\, S = S\ \}$$
$$T_k = T_{k-1} * RF\}$$

Step 5: 判断是否到达指定的迭代次数, 未到达则返回 Step 3 循环执行, 否则结束。

在 GA 搜索前期, 由于群体的较优解尚未得到充分复制, 染色体呈现出多样性, 有利于遗传信息的保留; 而在 GA 搜索后期, 由于较优个体的染色体得到广泛复制, 丧失了群体的多样性, 从而产生过早收敛现象。SA 在退火后期(低温) 接受较差解的概率很小, 而在搜索前期(高温) 接受较差解的概率较大。如果在搜索前期就引入 SA 操作, 则不能充分利用 SA 的优点。因此, 本算法在搜索过程中(δ次迭代后) 引入了 SA。

实数编码方法把解表示为一个实数向量: $\{q_1 \cdots q_m \cdots q_M, s_1 \cdots s_i \cdots s_n\}$, 大小为 $M + n$。其中 $q_m(m = 1,2,\cdots,m)$ 和 $s_i(i = 1,2,\cdots,n)$ 分别表示机器 m 和零件类型 i 属于第几个单元。

以 $I_pC_k = \{1/D(1),\cdots,h/D(h),\cdots,H/D(H)\}$ 表示个体 p 中所有属于单元 k 的机器集合, 其中 $h/D(h)$ 表示 I_pC_k 中共有 $D(h)$ 台 h 类型的机器; $I_{pq}C_{kk'} = \{g_1/D(g_1), g_2/D(g_2),\cdots\}$ 表示所有属于个体p 中单元k 和个体q 中单元k'的机器集合, 比如 $g_1/D(g_1)$ 表示集合 $I_{pq}C_{kk'}$ 中共有 $D(g_1)$ 台 g_1 类型的机器。称$I_{pq}C_{kk'}$为交叉块, 称$[I_{pq}C_{kk}]_{K\times K}$ 为交叉矩阵[3]。满足一台机器只属于一个单元, 即

$$I_{pq}C_{kk'} \cap I_{pq}C_{ll'} = \varnothing$$
$$|k - l| + |k' - l'| \neq 0$$
$$\bigcup_{k'}^{K} I_{pq}C_{kk'} = I_pC_k, \quad k = 1,2,\cdots,K$$
$$\bigcup_{k}^{K} I_{pq}C_{kk'} = I_qC_{k'}, \quad k' = 1,2,\cdots,K$$

每个交叉块为一个机器的集合, 交叉矩阵中的每个元素都是一个交叉块。尽管不同交叉块所含机器数可能不同, 但同一交叉块中的所有机器在两个个体中都在一起。位于交叉矩阵中两个在同一行或同一列的交叉块中的机器之间的关系是: 在一个个体中它们在同一单元, 而在另外一个个体中它们不在一起。位于交叉矩阵中两个既不在同一行也不在同一列的交叉块中的机器关系是: 在两个个体中它们都不在一起。一致交叉算子的操作原则是: 1) 两个父本中都分在一个单元的机器在子代中仍在一起; 2) 以 0.5 的概率把机器分到一个父本中所属的一个单元中; 3) 两个父本中都不分在一个单元的机器在子代中仍在一起的概率为 0。

变异算子的操作包括: 1) 随机从某单元中选择一台机器到另外一个可接受它的单元; 2) 随机交换两个单元中的机器。在评估每个个体的目标函数 $J(p)$ 之后, 求出每代最劣个体(J 最大) 的适值 ($\max\_J$), 然后进行适应度的规范化与定标, 并按下式求出每个个体的被选择概率

$$f(p) = \frac{[(\max\_J - J(p)) * A + B]^2}{\sum_{p=1}[(\max\_J - J(p)) * A + B]^2}$$

其中, A, B 为线性拉伸系数, P 为种群大小。适应度的拉伸减小了寻优过程中的早熟与停滞。

4 计算实例

本例是 Boctor[4] 列举的 10 个规模为 16* 30(机器数* 零件数) 的 CF 问题。每个问题都有不同的单元数和机器容量的限制, 采用异常元素数最少作为目标函数。Boctor 给出了每个问题的最优解及采用单纯 SA 的解。Boctor 取得的最优解为 64%, 单纯采用 GA[3] 取得的最优解为78%, 而本文获得的解全

表 2 10 个问题的最优解和[4, 3] 中的解

问题	K = 2					K = 3			
	C_ M_k = 8	C_ M_k = 9	C_ M_k = 10	C_ M_k = 11	C_ M_k = 12	C_ M_k = 6	C_ M_k = 7	C_ M_k = 8	C_ M_k = 9
1	11, 11, 11	11, 11, 11	11, 11, 11	11, 11, 11	11, 11, 11	27, 28, 28	18, 18, 18	11, 11, 11	11, 11, 11
2	7, 7, 7	6, 6, 6	4, 10, 4	3, 4, 4	3, 3, 3	7, 7, 7	6, 6, 6	6, 7, 6	6, 6, 6
3	4, 5, 4	4, 4, 4	4, 4, 4	3, 4, 4	1, 4, 4	9, 12, 9	4, 8, 8	4, 8, 8	4, 4, 4
4	14, 14, 14	13, 13, 13	13, 13, 13	13, 13, 13	13, 13, 13	27, 27, 27	18, 18, 18	14, 14, 14	13, 13, 13
5	9, 9, 9	6, 6, 9	6, 6, 6	5, 7, 5	4, 4, 4	11, 11, 11	8, 9, 8	8, 9, 9	6, 8, 6
6	5, 5, 5	3, 3, 3	3, 5, 3	3, 3, 3	2, 3, 3	6, 8, 6	4, 5, 4	4, 5, 4	3, 4, 4
7	7, 7, 7	4, 4, 4	4, 4, 4	4, 4, 4	4, 4, 4	11, 11, 11	5, 5, 5	5, 5, 5	4, 5, 4
8	13, 13, 13	10, 20, 10	8, 15, 15	5, 11, 11	5, 7, 7	12, 14, 14	11, 11, 11	8, 11, 11	8, 8, 8
9	8, 13, 8	8, 8, 8	8, 8, 8	5, 8, 8	5, 8, 8	12, 12, 12	12, 12, 12	8, 13, 13	8, 8, 8
10	8, 8, 8	5, 5, 5	5, 5, 5	5, 5, 5	5, 5, 5	10, 12, 10	8, 14, 14	8, 8, 8	5, 8, 8

表 3　文献[4] 中的第 10 个问题

	1	2	3	3A	4	5	6	7	8	9	10	11	12	12A	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1		1		1			1		1		1					1	1											1			
2				1							1					1									1	1				1	1	
3			1													1																
4												1			1					1	1			1			1		1			1
5				1					1		1														1	1					1	
6	1		1		1	1	1			1		1					1	1														1
7				1			1		1		1					1																
8	1	1	1		1	1	1	1		1		1																1		1		1
9	1	1	1		1		1	1				1					1	1														
10	1	1	1			1	1			1		1					1	1														
11									1		1														1					1	1	1
12	1							1						1			1	1														
13			1								1										1				1	1						1
14													1		1				1	1	1	1		1			1		1			
15									1	1						1									1					1	1	
16													1									1						1				

表 4　本文对[4] 中第 10 个问题在可变路径情况下的解

	1	2	4	5	6	7	9	11	15	16	26	12	3	8	10	14	23	24	28	29	30	18	27	17	18	19	20	22	25	13
1	1		1			1	1	1	1	1													1							
6	1		1	1	1		1	1	1	1																				
8	1	1	1	1	1	1	1	1			1																			
9	1	1	1		1	1		1	1	1																				
10	1	1		1	1		1	1	1	1																				
12	1					1			1	1		1																		
16											1																1			
2													1		1	1	1	1	1	1										
3																1					1									
5													1	1	1		1	1			1									
7					1								1	1	1	1			1	1	1									
11														1	1		1		1	1	1									
13															1		1	1		1	1					1				
15							1							1		1	1		1											
4								1														1	1		1	1		1	1	1
14																						1	1	1	1	1	1	1	1	1

部为最优解。表 2 给出了本例 10 个问题的最优解和文献[4, 3] 中的解。其中第 1 项为本文的最优解, 第 2 项为[4] 的解, 第 3 项为[3] 的解。Celeron 300 上的平均CPU 时间为 9s。

考虑问题 10 中 $K = 3, C - M_k = 7$ 时的情况, 标有“A ”者为笔者添加的零件新增路径(见表 3)。对于未添加路径之前的问题, 本文方法结果(最优解) 中有异常工序数为 8(最优解); 添加路径后, 用本文方法进行可变路径的搜索, 异常工序数减少为 6(见表 4) 。可见, 在引入多重路径后, 成组的效果得到了改善。优化后零件 3, 12 采用了新增的路径。

5　结　　论

本文对容量受限、多种优化目标和多加工路径的 CF 问题进行研究, 形成了基于 GA 和 SA 的 CF 问题模型, 并详细描述了问题的解决策略。通过对广泛采用的几个典型实例进行计算, 结果表明该方法具有较高的计算效率和满意的计算结果。

参 考 文 献

1　Gupta T, Seifoddini H. Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system. Int J of Production Research, 1990, 28(7): 1247～1269

2　Ballakur J, Steudel H J. A within-cell utilization based heuristic for designing cellular manufacturing systems. Int J of Production Research, 1987, 25(5): 639～655

3　赵春伟, 吴智铭. 一种新的制造单元设计方法. 上海交通大学学报, 1997, 32(8): 73～76

4　Boctor F F. A linear formulation of the machine-part cell formation problem. Int J of Production Research, 1991, 29(2): 343～356

作 者 简 介

李　岩　男, 1972 年生。1997 年在上海铁道大学获硕士学位, 现为上海交通大学自动化研究所博士研究生。主要研究方向为离散生产调度, 智能控制。

吴智铭　男, 1936 年生。1959 年毕业于上海交通大学电机系, 现为上海交通大学自动化研究所教授, 博士生导师。主要研究方向为离散事件系统理论, 智能控制。