

DOI:10.13196/j.cims.2015.04.018

求解混合流水车间调度问题的变量相关 EDA 算法

刘 昶¹, 李 冬^{1,2}, 彭 慧¹, 史海波¹

(1. 中国科学院 沈阳自动化研究所, 辽宁 沈阳 110016; 2. 中国科学院大学, 北京 100049)

摘 要:针对混合流水车间调度问题和分布估计算法的特点,提出将变量按工序分组,通过组内概率约束、组间概率耦合的方式建立混合流水车间调度问题变量间概率关系的新方法。对分布估计算法中的紧致遗传算法的种群产生和概率更新机制进行了改进,以解决流水车间调度问题等复杂问题。通过仿真实验、与其他算法比较以及在大规模生产实际问题中的应用,验证了该算法的有效性和鲁棒性。

关键词:分布估计算法;紧致遗传算法;混合流水车间;概率模型

中图分类号:TP315 文献标识码:A

EDA algorithm with correlated variables for solving hybrid flow-shop scheduling problem

LIU Chang¹, LI Dong^{1,2}, PENG Hui¹, SHI Hai-bo¹

(1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: According to the characteristic of Hybrid Flow-shop Scheduling Problem (HFSP) and Estimation of Distribution Algorithm (EDA), a new method of building probabilistic relationships between HFSP variables was proposed. In this new method, the variable was grouped by process, and the variable probabilistic relationship in a high level was built through the probabilistic constraints in the same group and the probabilistic coupling in different groups. The mechanism of population generation and probability updating was improved in the standard compact Genetic Algorithm(cGA) which accelerated the convergence rate of algorithm. The effectiveness and robustness of the algorithm was proved by simulation and the comparison with other algorithms and the verification of large-scale practical problem.

Key words: estimation of distribution algorithm; compact genetic algorithm; hybrid flow-shop; probability model

0 引言

混合流水车间调度问题(Hybrid Flow-shop Scheduling Problem, HFSP)是车间调度中的一类经典问题,具有广泛的应用背景,很多行业的调度问题均可归结为 HFSP,如机械制造、装配、半导体封装和冶金等。HFSP 综合了流水车间调度与并行机调度的特点,其求解难度大,即使是仅包含两工序的 HFSP 也是 NP 问题^[1]。因此,对 HFSP 的研究具有很强的应用价值和重要的学术意义^[2]。

HFSP 的传统求解方法主要有精确算法^[3]和启发式方法^[4]。精确算法理论上能得到最优解,但计算时间难以接受,通常适用于小规模问题;启发式方法可以在较短时间复杂度内求出问题的解,但无法保证解的质量。由于传统方法的缺陷,求解 HFSP 的群智能方法成为当前研究的热点,并取得了良好的效果^[5-11]。

分布估计算法(Estimation of Distribution Algorithm, EDA)是一种概率分析进化算法,该算法采用一种全新的进化模式,没有遗传算法中的交叉、

收稿日期:2014-01-17;修订日期:2014-04-25。Received 17 Jan. 2014; accepted 25 Apr. 2014.

基金项目:国家科技重大专项资助项目(2011ZX02601-005)。**Foundation item:** Project supported by the National Science and Technology Major Project, China(No. 2011ZX02601-005).

变异操作,而是采用统计学习的手段从宏观角度建立一个描述解分布的概率模型,通过对概率模型反复抽样产生新群体来实现种群的进化^[12]。分布估计算法通过建立概率模型描述变量之间的相互关系,能有效地实现构造块的混合与重组,因此可以解决已有群智能算法难以解决的问题,从而更有效地处理非线性、高维复杂问题^[13]。目前分布估计算法已广泛用于工程优化、机器学习、模式识别和运筹学等众多领域的优化问题^[14-27]。

根据概率模型如何表示变量间的相互关系,分布估计算法可以分为变量无关、双变量相关、多变量相关三种。但在实际问题中,变量之间存在复杂的依赖关系,变量无关和双变量相关算法由于没有考虑变量相关关系或仅考虑了简单的相关关系,在求解复杂问题时表现并不理想。因此,近年来研究更多的是多变量相关的分布估计算法,这类算法通过建立多变量间的复杂关系反映问题变量间的关联关系,在采样过程中,多变量间的关联关系使得个体可以根据已采样结果自适应地调整,提高了算法的连锁学习能力,能够有效地解决非线性、复杂问题。

混合流水车间调度问题是一个集分配和排序为一体的问题,由于 HFSP 比一般的 flow shop 调度问题要复杂得多且变量间的概率关系十分复杂,目前尚无针对 HFSP 变量相关的概率分布估计算法,使该算法的应用受到限制。文献[2]针对 HFSP 的特点提出一种分布估计方法,取得了较好的效果,但由于算法中的概率模型未能反映不同工序间的变量相关关系,仅对第一道工序进行了优化,其余工序通过启发式规则完成,降低了算法的灵活性,并未充分发挥分布估计算法的优势。本文根据 HFSP 的特点,通过将变量按工序分组,组内变量通过概率约束,组间变量根据目标函数建立组间概率耦合关系,提出一种新的建立高层次间变量概率关系的方法,并对估计分布算法中的紧致遗传算法(compact Genetic Algorithm, cGA)的算法流程进行了改进,针对 HFSP 中最小化最大加工时间问题的求解取得了比较好的效果。

1 混合流水车间调度问题

如图 1 所示,在 HFSP 中,任一道工序有一台或多台机器并且至少存在一道工序有两台或多台机器,工件的加工需要满足相同的工艺顺序。记 J_i 为工件序号, $i=1,2,\dots,n$ 。其中: n 为工件总数; m_j 为第一道工序的机器数, $j=1,2,\dots,S$, S 为工序总数; $t_{i,j,k}$ 为工件 J_i 在第 j 道工序第 k 台机器上的加工时

间; $s_{i,j,k}$ 为工件 J_i 在第 j 道工序第 k 台机器上的开始加工时间; $e_{i,j,k}$ 为工件 J_i 在第 j 道工序第 k 台机器上的完工时间; C_i 为工件 C_i 的完工时间; $C_{\max} = \max\{C_1, C_2, \dots, C_n\}$ 为最大完工时间。

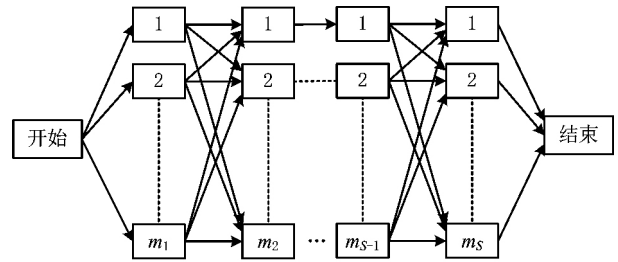


图1 HFSP图例

本文假设:工件在加工过程中不可中断;在某一道工序,工件可在该工序的任意一台机器上加工。参考文献[2],以最小化最大加工时间为目标函数,给出 HFSP 的数学模型:

$$\min C_{\max} \quad (1)$$

$$\text{s. t.} \quad \sum_{l=1}^n x_{i,l} = 1, l = 1, 2, \dots, n; \quad (2)$$

$$\sum_{l=1}^n x_{i,l} = 1, i = 1, 2, \dots, n; \quad (3)$$

$$\sum_{k=1}^{m_j} y_{i,j,k} = 1, i = 1, 2, \dots, n, j = 1, 2, \dots, S; \quad (4)$$

$$e_{i,j,k} = s_{i,j,k} + t_{i,j,k}, i = 1, 2, \dots, n, j = 1, 2, \dots, S, k = 1, 2, \dots, m_j; \quad (5)$$

$$e_{i,j,k} \leq s_{i,j,k'}, i = 1, 2, \dots, n, j = 1, 2, \dots, S-1, k = 1, 2, \dots, m_j; k' = 1, 2, \dots, m_{j+1}; \quad (6)$$

$$\sum_{l=1}^n x_{i,l} s_{i,1,k} \leq \sum_{i=1}^n x_{i,l+1} s_{i,1,k}, i = 1, 2, \dots, n, l = 1, 2, \dots, n-1, k = 1, 2, \dots, m_1; \quad (7)$$

$$\sum_{i=1}^n x_{i,l_1} y_{i,j,k} e_{i,j,k} \leq \sum_{i=1}^n x_{i,l_2} y_{i,j,k} s_{i,j,k'} + (1 - \sum_{i=1}^n x_{i,l_2} y_{i,j,k}) L, l_1, l_2 = 1, 2, \dots, n, l_1 \leq l_2, j = 1, 2, \dots, S, k, k' = 1, 2, \dots, m_j; \quad (8)$$

$$x_{i,l} = \begin{cases} 1 & \text{工件 } i \text{ 被安排在第 } l \text{ 个位置;} \\ 0 & \text{否则。} \end{cases} \quad (9)$$

$$y_{i,j,k} = \begin{cases} 1 & \text{工件 } i \text{ 的工序 } j \text{ 在机器 } k \text{ 上;} \\ 0 & \text{否则。} \end{cases} \quad (10)$$

其中:式(1)表示最小化最大加工时间为目标函数;式(2)确保任一工序上工件加工顺序的某个位置只能安排一个工件;式(3)确保任一工序上的一个工件仅能被安排在一个位置上加工;式(4)表示在某道工序上一个工件仅能在一台机器上加工;式(5)表示

某一道工序上工件的加工完成时间等于工件的开始加工时间加上工件的加工时间;式(6)表示工件需要满足工艺顺序约束;式(7)表示工件在任一道工序的优先级越高则工件在工序开始处理的时间就越早;式(8)表示如果同一道工序的两个工件被分配到同一台机器上加工,则优先级低的工件必须等待优先级高的工件加工完毕以后才能加工。

2 紧致遗传算法

cGA 是 EDA 算法中的一种,相对于其他分布估计算法,该算法不但具有不同的概率模型的更新算法,而且种群的规模更小^[28]。在 cGA 算法中,每次仅由概率模型随机产生两个个体,然后对两个个体进行比较,按照一定的策略对概率模型进行更新^[29]。cGA 的操作过程非常简单清晰,只需保存个体不同基因位置上的概率分布,大大降低了存储空间;通过每一代产生两个个体进行比较和更新,取代了标准遗传算法大规模群体之间的交叉变异,提高了运算速度,通过概率指导优化方向,提高了收敛的稳定性^[30]。然而 cGA 算法在解决复杂问题时,由于每一代的种群规模只有两个,会导致概率模型早熟,易陷入局部最小值。另外,cGA 算法的每一代都会放弃原有的个体而根据概率模型产生新的种群,因此不能保存系统的优势个体,即优势个体对概率模型的贡献可能仅有一次,而算法给出的最优解未必会是算法搜索中遇到的最好个体。因此,针对 cGA 算法在解决复杂问题时存在的缺陷,在保持算法简单清晰的基础上对 cGA 算法做如下改进:

(1)增加种群规模,每次仍然由概率矩阵产生两个个体,并比较个体的适应度值,保留优势个体,舍弃劣势个体。然后由概率模型再次产生一个新的个体并与原有优势个体进行比较,保留优势个体,舍弃劣势个体。依此类推,直到产生的个体满足设定的种群规模,取出优势个体并对概率模型进行更新。这样处理能够保证每一代用来更新概率模型的都是一个整体比较优秀的个体(而不是两个中的优秀个体),有利于概率模型沿着正确的方向收敛。

(2)保留每一代中的优良个体,参与下一代的比较。标准 cGA 算法每次都会产生两个新的个体,这样处理使得优秀个体与普通个体对概率模型收敛的贡献度相同,会引起算法的震荡,同时无法保证算法最后给出的个体是搜索过程中遇到的最好个体。因此结合传统遗传算法的优点,保留每一代的获胜个体参与下一代竞争。

改进后的 cGA 算法的流程图如图 2 所示。

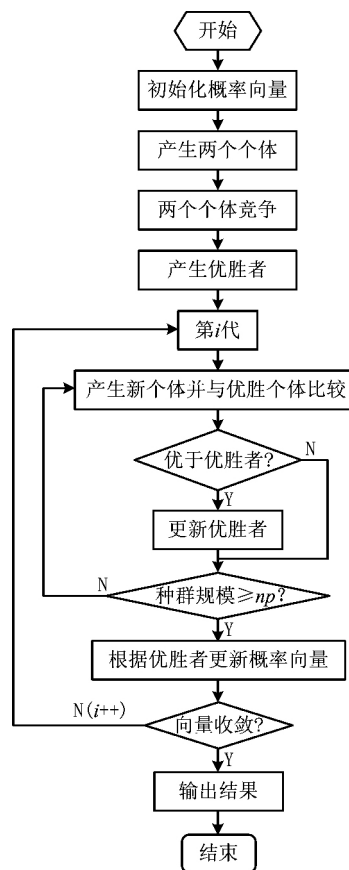


图2 改进的cGA算法流程

3 求解 HFSP 的变量相关的 cGA 算法

3.1 编码和解码

设 HFSP 的问题规模为 $N \times M$, N 表示工件数, M 表示工序数, 个体采用分组排列编码。将长度为 $N \times M$ 的编码按工序分为 M 组, 在每一组中, 整数部分表示工件在该工序的加工顺序, 小数部分表示加工机器。例如对于 3 工件 2 工序的问题, 个体 $\{[2.1, 1.3, 3.2], [1.1, 2.2, 3.1]\}$ 表示在第 1 道工序时, 工件 2 在工序 1 的第 1 台机器上加工, 工件 1 在工序 1 的第 3 台机器上加工, 工件 3 在工序 1 的第 2 台机器上加工, 由于第 1 道工序 3 个工件没有发生机器冲突, 3 个工件可同时并行加工, 如果发生机器冲突, 则按整数部分排列的先后顺序依次加工; 对于工序 2, 工件 1 在工序 2 的第 1 台机器上加工, 工件 2 在工序 2 的第 2 台机器上加工, 工件 3 在工序 2 的第 1 台机器上加工, 由于工件 1 和工件 3 有机器冲突, 按整数部分排序由工件 1 先加工, 工件 3 后加工。图 3 所示为该个体的甘特图。

3.2 概率模型

cGA 通过概率模型来体现个体的分布, 并通过

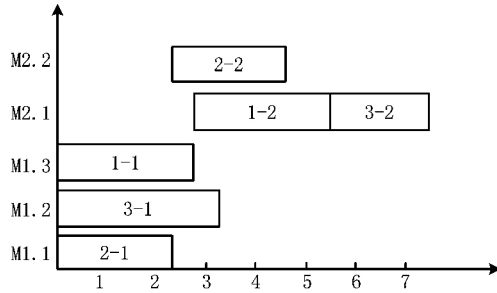


图3 个体甘特图

对概率模型采样产生新的个体。因此,概率模型是能否正确反映变量间的概率关系,对 cGA 的性能起决定性作用。结合变量分组的思想,对概率模型也按工序进行分组处理,用 M 个 $N \times N$ 维矩阵 (P_1, P_2, \dots, P_M) 表示解空间分布的概率模型,其中第 m 个矩阵 $P_m (m=0, 1, 2, \dots, M)$ 表示第 m 道工序工件的排列顺序概率模型;用元素 f_{mij} 表示第 $m (m=0, 1, 2, \dots, M)$ 道工序中工件 $j (j=1, 2, \dots, N)$ 在第 $i (i=0, 1, 2, \dots, N)$ 位加工的概率, p_{mij} 越大,表示工件 j 在第 i 位上或之前出现的概率就越大。因此,第 m 道工序不同工件的加工优先顺序通过矩阵 P_m 得以体现。图 4 为 3 工件 2 工序的概率模型, P_1 表示工序 1 各工件的排列概率模型, P_2 表示工序 2 各工件的排列概率模型。

P_m 的每一个元素 p_{mij} 表示事件发生的概率,因此 $p_{mij} \in [0, 1]$ 。对于给定位置 i ,一定会有一个工件得到加工,因此存在约束

$$\forall i, \sum_{j=1}^n p_{ij} = 1, \quad (11)$$

即 P_m 的每行元素之和均为 1,这便是组内变量概率约束,即对于某道工序,各工件在该工序第 i 位得到加工的概率和始终为 1。该约束在个体采样概率模型的更新中得到体现,保证了组内变量概率模型的收敛性。理论上经过多次迭代,概率矩阵 P_m 某一位置上仅有一个工件得到加工的概率趋于 1,而其他工件在该位置得到加工的概率趋于 0。

P_1			P_2		
0.3	0.4	0.3	0.7	0.2	0.1
0.2	0.7	0.1	0.3	0.2	0.5
0.2	0.6	0.2	0.1	0.4	0.5

图4 概率模型示例

为保证算法初始阶段解空间被均匀采样,概率矩阵 $P_m (m=1, 2, \dots, M)$ 按如下方式初始化:

$$P_{mij} = \frac{1}{n}. \quad (12)$$

图 5 所示为 3 工件 2 工序初始化后的概率模型。

P_1			P_2		
1/3	1/3	1/3	1/3	1/3	1/3
1/3	1/3	1/3	1/3	1/3	1/3
1/3	1/3	1/3	1/3	1/3	1/3

图5 初始化概率模型

3.3 个体采样

本文在个体采样过程中,根据 HFSP 的特点将变量以工序为单位分成组,按工序分组串行采样,通过目标函数建立起组间概率联系,从而在较高层次上建立起变量之间的概率关系。本文的目标函数为最小化最大加工时间,为了缩短加工时间,结合正序排产的思想建立工序之间的概率联系。具体实现步骤如下:

步骤 1 确定第 1 道工序工件的排列顺序。对于第 1 道工序,从第 1 个加工位置到第 n 个加工位置依次采样,找到每个位置的待加工工件。通过轮盘赌方式确定第 i 个位置上的待加工工件—工件 j 被选中的概率为 p_{1ij} ,由于已选择的工件不能在本工序再次出现,工件 j 一经选中,需将矩阵 P_1 的第 j 列元素全部置 0,表示该工件在本工序已完成,为了满足组内变量概率约束,同时将矩阵 P_1 的其余各行归一化。

步骤 2 在得到第 1 道工序各工件的加工顺序后,根据原有空闲机器 (First Available Machine, FAM)^[33] 规则,为各工件在第 1 道工序的加工分配机器。具体做法如下:

(1) 根据各工件在本工序上的加工顺序,判断每个工件 j 在第 k 台机器上的最早允许加工时间,即机器 k 的释放时间 r_k 与 $C_{i,j-1}$ 之间的最大值, $\max(r_k, C_{i,j-1})$, 其中 $C_{i,0} = 0$ 。

(2) 对于每个工件 j ,根据 $\max(r_k + C_{i,j-1}) + t_{mjk}$ (t_{mjk} 为在第 m 道工序的工件 j 在第 k 台机器上的加工时间) 选择值最小的机器作为工件 j 的加工机器,更新工件 j 在第 1 道工序的加工完成时间和机器 k 的释放时间。

步骤 3 在得到第 1 道工序的采样个体后,便可以计算出各工件在第 1 道工序的加工完成时间。第 1 道工序的加工完成时间对第 2 道工序的加工序列有直接影响。结合最小化最大完工时间的目标函数,以第 1 道工序各工件的加工完成时间为纽带,建立两道工序间的概率关系。具体做法如下:

(1) 计算各工件在第 1 道工序的完成时间,并由小到大进行排列,得到第 1 道工序工件加工时间的

先后序列 S_1 。

(2) 根据 S_1 对 P_2 进行关联耦合:

$$p_{2ij} = (1 - \partial)p_{2ij} + \frac{\partial}{i} I_{2ij} \quad (13)$$

式中: $\partial \in (0, 1)$ 为关联因子,

$$I_{2ij} = \begin{cases} 1, & \text{工件 } j \text{ 在 } S_1 \text{ 中排在第 } i \text{ 位或之前;} \\ 0, & \text{否则。} \end{cases} \quad (14)$$

可以证明经过式(13)修正的概率模型 P_2 的每一行之和仍为 1, 保证了式(11)的组内变量概率约束。经过概率耦合后的 P_2 仅在局部起作用, 即只在本次采样过程中起作用。修正过的概率模型使得在第 1 道工序先完工的工件在第 2 道工序优先加工的概率增强, 从而建立起两个工序之间的概率联系, 提高了个体的质量, 加快了算法的收敛性。通过调整修正幅度的大小, 可以对两道工序之间的联系进行控制。依次类推, 直到完成所有工序的个体采样。

3.4 概率模型的更新

选择每一代最终的优秀个体 x 作为优势个体, 采用如下增量学习的方法更新概率模型:

$$p_{mij}(l+1) = (1 - \beta)p_{mij}(l) + \frac{\beta}{i} L_{mij} \quad (15)$$

式中: $\beta \in (0, 1)$ 为学习速率; l 为迭代代数;

$$L_{mij}(l) = \begin{cases} 1, & \text{工件在第 } m \text{ 道工序位置 } i \text{ 或} \\ & \text{之前出现;} \\ 0, & \text{否则。} \end{cases} \quad (16)$$

通过不断的迭代, 使得概率矩阵不断趋于成熟, 经过迭代后的概率矩阵 P_m 中每一行的和仍为 1, 满足组内变量概率约束, 保证了组内概率模型 P_m 的收敛性。在本算法中设置最大迭代次数 ($MaxGen$) 作为算法的终止条件。

4 实验结果与比较

4.1 关联因子设置

为了验证关联因子对算法的影响, 分别取 ∂ 为 $(0, 0.3, 0.6, 0.9, 1)$ 四个水平, 对 5 种不同规模 ($8 \times 6, 10 \times 10, 15 \times 15, 20 \times 15, 20 \times 20$) 的随机实例进行测试, 在每种规模下不同水平的 ∂ 分别计算 10 次, 取 10 次运行的平均值 AGV 作为评价指标。除关联因子不同外, 其余各参数均保持一致, 取种群规模 np 为 20, 最大迭代次数为 50 次, 学习速率 $\beta=1$ 。表 1 所示为不同问题规模下不同 ∂ 设置下的测试结果。

通过对表 1 中的数据进行分析可以发现, 随着 ∂ 的增大, 仿真结果逐渐变好。前 4 种规模下, $\partial=1$ 时

表 1 结果对比表

规模	∂				
	0	0.3	0.6	0.9	1.0
8×6	369	334	290	283	291
10×10	587	535	516	505	513
15×15	923	712	657	657	672
20×15	1 257	1 034	957	924	931
20×20	1 363	1 259	1 223	1 148	1 035

的收敛结果比 $\partial=0.9$ 时要差一些, 这是因为当 $\partial=1$ 时, 紧邻两道工序属于强耦合, 丧失了算法的收敛能力与灵活性。对于第 5 种规模, $\partial=1$ 时的收敛结果最好, 这是因为对于 20×20 规模的问题, 当前的参数设置 (种群规模 np 为 20, 最大迭代次数 50 次, $\beta=1$) 无法使算法充分收敛。综上, 关联因子 ∂ 对提高算法的搜索能力和效率的作用非常明显, 因此要根据不同的问题规模和参数, 选择合适的关联因子。

4.2 算法比较

仿真的硬件环境为: Core 双核 2.93 GHz CPU, 2 GB 内存, 仿真程序开发环境为 Visual Studio 2010, 编程语言为 C#。为验证算法的有效性, 选择 5 种不同规模 ($8 \times 6, 10 \times 10, 15 \times 15, 20 \times 15, 20 \times 20$) 的实例进行仿真测试。在迭代次数与种群规模相同的情况下与标准 cGA 算法、差分进化 (Differential Evolution, DE) 算法和改进的 PSO 算法做对比。

实例 1 采用 8 工件、6 工序的小规模实例, 每道工序都有两台或多台机器可以加工。表 2 所示为各工件在不同机器上的加工时间表。

设置种群规模 $np=20$, 最大进化代数 300 代, 学习速率 $\beta=0.1$, 修正幅度 $\partial=0.9$, 每种算法均运行 10 次。表 3 所示为四种算法仿真结果比较以及四种算法的平均计算时间对比。结果表明, 在种群规模和迭代次数相同的情况下, 改进的 cGA 得到的结果远优于 DE 算法和 PSO 算法, 而且收敛的结果的稳定性非常好, 有很好的鲁棒性, 较标准的 cGA 的搜索能力也有一定提高。

实例 2~实例 5 分别采用 $10 \times 10, 15 \times 15, 20 \times 15, 20 \times 20$ 规模的实例, 每道工序均有多台机器可以选择。由于数据量较大, 本文只给出改进的 cGA (设置种群规模 $np=20$, 最大进化代数 100 代, 学习速率 $\beta=0.1$, 修正幅度 $\partial=0.9$)、标准 cGA (种群规模为 2, 迭代代数 1 000 代)、DE 算法、PSO 算法四种算法运行 10 次的结果对比, 如表 4~表 7 所示。

表 2 8×6 规模问题排产参数

工件	工序 1		工序 2		工序 3			工序 4			工序 5		工序 6		
	M1	M2	M1	M2	M1	M2	M3	M1	M2	M3	M1	M2	M1	M2	M3
1	43	41	43	34	47	39	46	40	43	35	42	45	35	45	46
2	46	32	32	42	37	34	49	42	31	39	36	48	38	43	34
3	36	45	45	49	46	33	33	41	36	42	33	41	46	42	49
4	44	31	31	37	42	38	35	30	34	33	36	41	46	35	47
5	47	49	49	33	32	43	34	39	40	45	46	49	49	49	34
6	49	39	39	47	43	34	41	40	37	44	30	47	40	33	34
7	44	36	36	38	40	42	43	49	45	44	44	45	36	44	45
8	43	49	49	45	49	47	34	42	43	38	30	34	34	33	48

表 3 8×6 规模问题计算结果对比(cGA,DE,PSO)

算法	次数										时间/s
	1	2	3	4	5	6	7	8	9	10	
改进的 cGA	313	317	313	317	317	317	313	317	317	313	9.2
标准的 cGA	321	324	317	319	315	322	325	317	316	325	9.5
DE	369	364	381	371	365	374	382	356	376	356	10.9
PSO	379	373	385	385	383	372	381	370	375	365	9.1

表 4 10×10 规模问题计算结果对比(cGA,DE,PSO)

算法	次数										时间/s
	1	2	3	4	5	6	7	8	9	10	
改进的 cGA	499	502	503	499	503	513	498	508	506	499	20.3
标准的 cGA	535	527	518	524	533	536	528	514	526	519	22.7
DE	587	599	603	596	607	614	605	613	606	593	25.1
PSO	623	615	618	591	600	597	604	628	606	612	18.4

表 5 15×15 规模问题计算结果对比(cGA,DE,PSO)

算法	次数										时间/s
	1	2	3	4	5	6	7	8	9	10	
改进的 cGA	730	729	729	726	728	730	741	738	730	729	83.2
标准的 cGA	751	739	748	740	756	738	736	758	736	741	93.2
DE	917	908	922	913	925	923	922	895	892	918	98.5
PSO	906	901	913	893	913	896	928	932	899	919	65.2

表 6 20×15 规模问题计算结果对比(cGA,DE,PSO)

算法	次数										时间/s
	1	2	3	4	5	6	7	8	9	10	
改进的 cGA	730	729	729	726	728	730	741	738	730	729	132.1
标准的 cGA	776	830	857	821	790	798	794	802	832	793	152.6
DE	1 040	1 047	1 026	1 056	1 043	1 056	1 026	1 053	1 026	1 048	166.3
PSO	1 077	1 051	1 086	1 083	1 076	1 089	1 093	1 078	1 083	1 087	109.4

表 7 20×20 规模问题计算结果对比(cGA,DE,PSO)

算法	次数										时间/s
	1	2	3	4	5	6	7	8	9	10	
改进的 cGA	1 158	1 145	1 136	1 156	1 153	1 148	1 134	1 148	1 158	1 136	223.4
标准的 cGA	1 194	1 207	1 233	1 204	1 224	1 256	1 197	1 159	1 209	1 230	240.5
DE	1 363	1 406	1 347	1 356	1 343	1 356	1 326	1 353	1 326	1 348	275.7
PSO	1 383	1 372	1 340	1 359	1 362	1 326	1 369	1 357	1 372	1 358	186.0

通过 5 种不同规模实例的比较可以发现,在种群规模和迭代次数相同的情况下,改进的 cGA 算法的时间开销仅比 PSO 算法高,但其优化结果远远优于 DE 算法与 PSO 算法,而且得到的结果十分稳定,相对于标准 cGA 的搜索能力也有较大的提高。

4.3 大规模问题验证

半导体封装测试工艺流程主要包括划片、装片、键合、塑封、去飞边、电镀、打印、切筋和成型、外观检查、成品测试以及包装出货等工序,半导体封装测试企业的加工种类众多,同一道工序存在多种型号设备,是一个典型的 HFSP。半导体封装测试的加工单位是 Lot,每个 Lot 是一个不可分的工件集合,本文采用 100 个 Lot,8 工序的实例进行测试。测试环境为普通 PC 机,设置种群规模 $np=20$,最大进化代数 100 代,学习速率 $\beta=0.1$,修正幅度 $\delta=0.9$ 。表 8 所示为分别运行三次所用时间和所得结果。

表 8 实际问题排产结果

运行次数	最大加工周期/min	时间/min
1	2 331	12
2	2 325	12
3	2 329	12

基于正序规则排产得到的结果为 2 433 min,通过实验数据对比可以发现:对于 100 个 Lot 排产,改进的 cGA 算法的最长加工时间比基于正序规则的算法节省约 110 min,对于大规模问题,该算法的时间开销可以接受,同时通过调整学习速率和迭代次数,能够在更短的时间内获得一个近优解,以解决实际的混合流水生产线排产问题。

5 结束语

本文提出了解决 HFSP 的变量相关 EDA 算法。在经过变量分组后,通过组内概率约束保证了组内变量概率模型的收敛性,通过采样过程中的组间概率耦合,建立了紧邻工序之间的联系,提升了个体的质量,降低了冗余计算,提高了算法的效率。通过对 cGA 算法进行改进,使其更适应于解决大规模的复杂问题,提高其搜索能力。通过不同规模实例的数值仿真、与其他算法的比较、实际应用对比验证了算法的有效性与可靠性。本文以最小化最大加工时间作为目标函数建立组间概率关系,对于其他的目标函数也可以通过相应的映射建立紧邻工序间或各工序与瓶颈工序之间的关系。此外,对于其他复杂的组合优化问题,也可以采样类似的思想,将原本

复杂的无法建立概率相关模型的变量通过分组的思想建立更高层次的概率联系。这将是未来的研究方向。

参考文献:

- [1] HOOGEVEEN J A, LENSTRA J K, VELTMAN B. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard[J]. European Journal of Operational Research, 1996, 89(1):172-175.
- [2] WANG Shengyao, WANG Ling, XU Ye, et. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problems[J]. Acta Automatica Sinica, 2012, 38(3):437-443(in Chinese). [王圣尧,王凌,许烨,等.求解混合流水车间调度问题的分布估计算法[J].自动化学报,2012,38(3):437-443.]
- [3] XUAN Hua, TANG Lixin. Lagrangian relaxation algorithm for real-time hybrid flow-shop scheduling with no-wait in process[J]. Control and Decision, 2006, 21(4):376-380(in Chinese). [轩华,唐立新.实时无等待 HFS 调度的一种拉格朗日松弛算法[J].控制与决策,2006,21(4):376-380.]
- [4] RIANE F, ARTIBA A, ELMAGHRABY S E. Sequencing a hybrid two-stage flowshop with dedicated machines[J]. International Journal of Production Research, 2002, 40(17):4353-4380.
- [5] ZHOU Huiren, TANG Wansheng, WEI Yinghui. Optimize flexible flow-shop scheduling using genetic algorithm[J]. Computer Engineering and Applications, 2009, 45(30):224-226(in Chinese). [周辉仁,唐万生,魏颖辉.柔性 Flow-Shop 调度的遗传算法优化[J].计算机工程与应用,2009,45(30):224-226.]
- [6] LOW C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines[J]. Computers and Operations Research, 2005, 32(8):2013-2025.
- [7] WANG X, TANG L. A tabu search heuristic for the hybrid flow-shop scheduling with finite intermediate buffers [J]. Computers and Operations Research, 2009, 36(3):907-918.
- [8] ALAYKYRAN K, ENGIN O, DOYEN A. Using ant colony optimization to solve hybrid flow shop scheduling problems [J]. International Journal of Advanced Manufacturing Technology, 2007, 35(5/6):541-550.
- [9] TSENG C T, LIAO C J. A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks[J]. International Journal of Production Research, 2008, 46(17):4655-4670.
- [10] LIU F, ZHANG X P, ZOU F X, et al. Immune clonal selection algorithm for hybrid flow-shop scheduling problem[C]//Proceedings of Conference on the Chinese Control and Decision. Washington, D. C., USA: IEEE, 2009:2605-2609.
- [11] XU Y, WANG L, ZHOU G, et al. An effective shuffled frog leaping algorithm for solving hybrid flow shop scheduling problem[C]//Proceedings of the International Conference on Intelligent Computing. Berlin, Germany: Springer-Verlag, 2011:560-567.
- [12] PELIKAN M, GOLDBERG D E, LOBO F. A survey of opti-

- mization by building and using probabilistic models[J]. Computational Optimization and Application, 2002, 21(1): 5-20.
- [13] HE Xiaojuan. Estimation of distribution algorithm and its application in scheduling problem solving [D]. Lanzhou: Lanzhou University of Technology, 2011 (in Chinese). [何小娟. 分布估计算法在生产调度问题中的应用研究[D]. 兰州: 兰州理工大学, 2011.]
- [14] HANDA H. EDA-RL: estimation of distribution algorithms for reinforcement learning problems[C]//Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. New York, N. Y., USA: ACM, 2009: 405-412.
- [15] ROMERO T, LARRANAGA P. Triangulation of Bayesian networks with recursive estimation of distribution algorithms [J]. International Journal of Approximate Reasoning, 2009, 50(3): 472-484.
- [16] SIMONESCU P, BEALE D, DOZIER G. Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm[J]. Journal of Mechanical Design, 2006, 128(1): 108-119.
- [17] INZA I, LARRANAGA P, ETXEBERRIA R, et al. Feature subset selection by Bayesian network-based optimization[J]. Artificial Intelligence, 2000, 123(1/2): 157-184.
- [18] CESAR R. Inexact graph matching for model-based recognition: evaluation and comparison of optimization algorithms [J]. Pattern Recognition, 2005, 38(11): 2099-2113.
- [19] SAGARNA R, LOZANO J. On the performance of estimation of distribution algorithms applied to software testing[J]. Applied Artificial Intelligence, 2005, 19(5): 457-489.
- [20] BLANCO R, LARRANAGA P, INZA I, et al. Gene selection for cancer classification using wrapper approaches[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18(8): 1373-1390.
- [21] SAEYS Y, DEGROEVE S, AEYELS D, et al. Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction[J]. Bioinformatics, 2003, 19(S2): 179-188.
- [22] GOLDINHO P, MEIGUINS A, OLIVEIRA R, et al. An estimation of distribution algorithms applied to sequence pattern mining[M]//Innovations in Computing Sciences and Software Engineering. Berlin, Germany: Springer-Verlag, 2010: 589-593.
- [23] SANTARELLI S, YU T, GOLDBERG D, et al. Military antenna design using simple and competent genetic algorithms[J]. Mathematical and Computer Modelling, 2006, 43(9/10): 990-1022.
- [24] NAEEM M, LEE D. Estimation of distribution algorithm for sensor selection problems [C]//Proceedings of Radio and Wireless Symposium. Washington, D. C., USA: IEEE, 2010: 388-391.
- [25] LIU Xinliang, ZHANG Tao, GUO Bo. Model of spare parts optimization based on estimation of distribution algorithms [J]. Systems Engineering—Theory & Practice, 2009, 29(2): 146-152 (in Chinese). [刘新亮, 张涛, 郭波. 基于分布估计算法的备件优化配置[J]. 系统工程理论与实践, 2009, 29(2): 146-152.]
- [26] DOU Lihua, WANG Gaopeng, CHEN Jie, et al. A hybrid algorithm of computing cannonball dispersion evenness [J]. Contron Theory & Application, 2009, 26(6): 624-628 (in Chinese). [杜丽华, 王高鹏, 陈杰, 等. 解弹头散布均匀度的分布估计算法[J]. 控制理论与应用, 2009, 26(6): 624-628.]
- [27] WU Hong, WANG Weiping, WANG Lei, et al. Application of EDA in path planning of cruise missile [J]. Electronics Optics & Control, 2010, 17(7): 6-10 (in Chinese). [武红, 王维平, 王磊, 等. 分布估计算法在巡航导弹航迹规划中的应用[J]. 光电与控制, 2010, 17(7): 6-10.]
- [28] RASTEGAR R, HARIRI A. A step forward in studying the compact genetic algorithm [J]. Evolutionary Computation, 2006, 14(3): 277-289.
- [29] YE Miao, CHENG Xiaohui. An improvement of the compact genetic algorithm for solving clustering traveling salesman problem [J]. Microelectronics & Computer, 2013, 30(8): 7-10 (in Chinese). [叶苗, 程小辉. 改进的紧致遗传算法求解簇状旅行商问题[J]. 微电子学与计算机, 2013, 30(8): 7-10.]
- [30] ZHOU Shude, SUN Zengqi. A survey on estimation of distribution algorithms [J]. Acta Automatica Sinica, 2007, 33(2): 113-124.
- [31] WANG Shenyao, WANG Ling, XU Ye. Estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machine [J]. Computer Integrated Manufacturing Systems, 2013, 19(6): 1304-1311 (in Chinese). [王圣尧, 王凌, 许焱. 求解相同并行机混合流水线车间调度问题的分布估计算法[J]. 计算机集成制造系统, 2013, 19(6): 1304-1311.]
- [32] WANG Ling. Shop scheduling with genetic algorithms [M]. Beijing: Tsinghua University Press, 2003: 138-145 (in Chinese). [王凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003: 138-145.]
- [33] WANG Ling, ZHOU Gang, XU Ye, et al. Advances in the study on hybrid flow-shop scheduling [J]. Control and Instrument in Chemical Industry, 2011, 38(1): 1-8 (in Chinese). [王凌, 周刚, 许焱, 等. 混合流水线调度研究进展[J]. 化工自动化及仪表, 2011, 38(1): 1-8.]

作者简介:

刘昶(1973—),女,辽宁辽阳人,副研究员,硕士生导师,研究方向:制造系统建模与性能评价、计划调度、数据分析与智能决策, E-mail: changl@sia.cn;

李冬(1990—),男,山西长治人,硕士研究生,研究方向:制造执行系统与群智能算法;

彭慧(1963—),男,辽宁沈阳人,研究员,硕士生导师,研究方向:制造执行系统软件体系结构与应用开发、网络化制造;

史海波(1966—),男,浙江宁波人,研究员,博士生导师,研究方向:生产与运作管理理论、制造系统建模与性能评价、网络化制造。