

文章编号: 1001-0920(2005)10-1157-04

## 基于DNA进化算法的车间作业调度问题研究

牛群, 顾幸生

(华东理工大学自动化研究所, 上海 200237)

**摘要:** 针对遗传算法解决车间作业调度问题时存在早熟收敛的缺点, 采用一种新型进化算法——DNA进化算法解决车间作业调度问题, 将算法从连续优化问题拓展用于解决离散优化问题, 并将其成功地应用于Job shop生产调度。采用了著名的Muth和Thompson标准问题FT10进行了验证。仿真结果表明, 与遗传算法相比, 该算法简单有效, 不仅具有很好的求解性能, 而且具有更快的收敛速度和全局搜索能力。

**关键词:** DNA进化算法; Job shop; 生产调度; 优化

**中图分类号:** TP278      **文献标识码:** A

## Job Shop Scheduling Problems with DNA Evolutionary Algorithm

NIU Qun, GU Xing-sheng

(Research Institute of Automation, East China University of Science and Technology, Shanghai 200237, China  
Correspondent: GU Xing-sheng, E-mail: xsgu@ecust.edu.cn)

**Abstract:** Aiming at the limitations of genetic algorithm such as converging at local optimum, an original evolutionary algorithm named DNA evolutionary algorithm, is used to solve the Job shop scheduling problems. The DNA evolutionary algorithm is extended to solve discrete optimization problems. And the presented algorithm is successfully applied to Job shop scheduling problems. The simulation results for the famous muth and thompson problem FT10×10 show that the algorithm is quite easy and effective, and not only has rapid convergence ability but also global searching ability compared with genetic algorithm.

**Key words:** DNA evolutionary algorithm; Job shop; Scheduling; Optimization

### 1 引言

车间作业调度问题, 亦称Job shop生产调度, 是一类非常经典的机器调度问题<sup>[1]</sup>, 同时也是最困难的组合优化问题之一。近几年发展起来的以遗传算法为代表的进化算法, 对于求解Job shop调度问题具有较为显著的优势。遗传算法是由Holland<sup>[2]</sup>于1975年提出的, 现有遗传算法虽然在解决车间作业调度问题时体现出较大优越性, 但仍存在两大局限——收敛速度慢和早熟。因此, 为了扩大其应用范围, 人们对它进行了多方面的改进。这些改进工作有力地推动了遗传算法的发展, 使其不断走向成熟, 但这些方法依赖于对优化问题的先验了解, 难以推广应用。

余文等基于单亲生物的分裂繁殖机理, 提出了一种新的进化学习算法——DNA进化算法, 并将其应用于连续优化问题, 取得了良好的效果<sup>[3]</sup>, 但在调度问题上却几乎没有应用。文献[4]用DNA进化算法实现了Flow shop生产调度, 因为Job shop调度问题的模型比Flow shop生产调度复杂得多, 而且在编码和操作算子上也有所区别, 所以用DNA算法实现Job shop调度更为复杂。本文对这种优化算法的变异算子作了适当的改进, 并将其成功地应用于Job shop生产调度。最后的实例仿真和比较表明了DNA进化算法的有效性和优越性, 以及它在调度领域的应用可行性。

收稿日期: 2004-10-27; 修回日期: 2005-03-04

基金项目: 国家自然科学基金项目(60274043); 上海市科委重大科技攻关项目(04dz11008)。

作者简介: 牛群(1980-), 女, 山东泰安人, 博士, 从事流程工业中的生产计划与调度的研究; 顾幸生(1960-), 男, 江苏海门人, 教授, 博士生导师, 从事控制理论与应用、生产计划与调度等研究。

## 2 车间作业调度问题

Job shop 调度问题可描述为: 在  $m$  台机器上安排  $n$  个加工作业。由于工件加工工艺的要求, 给定每个工件使用机器的顺序以及每道工序所花费的时间, 而且: 1) 每个机器在同一时刻只能加工一个工件; 2) 不同工件的工序之间没有顺序约束; 3) 操作一旦开始就不能中途停止。调度的目的是安排每台机器上的工件加工顺序, 并满足约束条件, 使制造系统某些性能指标得到最大限度地优化。

Job shop 调度问题常用的目标函数之一是最小化完工时间 (Makespan), 即

$$\min f(x) = \max_{(k=1, 2, \dots, m)} (C_k),$$

其中  $C_k$  为第  $k$  台机器的加工完成时间。

## 3 DNA 进化算法

### 3.1 DNA 进化算法的基本介绍

遗传算法是对有性生物自然进化的模拟, DNA 进化算法是对单亲生物自然进化过程和行为的一种机器模拟, 引入一个模拟单亲细胞繁殖和分化的算子——分裂算子。在 DNA 进化算法中, 分裂能繁衍生成具有等价关系意义下的“同类”个体, 不“同类”个体间的演化只能由变异操作来实现。分裂和变异操作后, 分别经过水平选择和垂直选择使群体得到双向进化。

### 3.2 基本定义和操作算子<sup>[3]</sup>

**分裂子群:** 个体编码 (DNA 单链) 与其互补链以及它们的逆转 (反演) 所构成的集合, 称为该个体的分裂子群。

**变异域:** 单链经过不多于一次变异所能迁移的局部范围 (构成的点集), 称为该链的变异域。

#### 3.2.1 分裂算子

**定义 1** 如果  $F$  是一个  $D \rightarrow D^*$  上的映射, 且对  $\forall v \in D$ , 有

$$F(v) = \{v, \bar{v}, v^-, \bar{v}^-\} \cap D^*, \quad (1)$$

则称  $F$  是  $D \rightarrow D^*$  上的分裂算子。这里:  $\{v, \bar{v}, v^-, \bar{v}^-\}$  是  $v$  的分裂 (集);  $\bar{v}$  和  $v^-$  分别表示串  $v$  的倒置、同互补和异互补。

#### 3.2.2 变异算子

**定义 2** 如果  $M$  是一个  $D \rightarrow D^*$  上的映射, 且对  $v \in D$ , 有

$$M(v) = \{v_1, v_2, \dots, v_L, v\} \cap D^*, \quad (2)$$

则称  $M$  是  $D \rightarrow D^*$  上的变异算子。这里:  $\{v_1, v_2, \dots, v_L, v\}$  是  $v$  的变异域;  $m_i$  是  $D \rightarrow D$  上的位变映射,  $i = 1, 2, \dots, L$ , 它使得串  $v$  的第  $i$  位发生变异。

### 3.2.3 选择算子

选择算子有 2 个: 一个使从分裂集中选取 1 个 DNA “表形”; 另一个使从变异域中选取 1 个 DNA “变型”。前者称作水平 (分裂) 选择, 后者称作垂直 (变异) 选择。

**定义 3** 如果  $D^* \rightarrow D$  是由  $D^*$  中的元素分裂构成的集类 ( $D^*$  的真子集),  $S_x$  是  $D^* \rightarrow D$  的映射, 且对  $\forall V = [v \circ \bar{v} \circ v^- \circ \bar{v}^-] \in D^*$ , 有

$$S_x(V) = Vx, \quad Vx = [v \circ \bar{v} \circ v^- \circ \bar{v}^-] \in D; \quad (3)$$

$$f(Vx) = \max\{f(v) : v \in V\}. \quad (4)$$

则称  $S_x$  是  $D^* \rightarrow D$  上的水平选择算子。这里:  $f(v)$  代表个体  $v$  的适应值; 算子  $S_x$  使得一个分裂 (集) 返回到它的最优表形; 分裂和水平选择算子结合在一起, 合成了水平算子。

$$E_x = S_x \cdot F. \quad (5)$$

其中:  $F$  是式 (1) 中的分裂算子, “ $\cdot$ ” 是算子乘积,  $E_x$  是  $D \rightarrow D$  上的一对一映射。

**定义 4** 如果  $D^* \rightarrow D$  是由  $D^*$  上的变异域构成的集类 ( $D^*$  的真子集),  $S_y$  是  $D^* \rightarrow D$  上的映射, 且对于  $\forall V_M = [v_1 v_2 v_L \dots v] \in D^*$ , 有

$$S_y(V_M) = v_M, v_M = [v_1, v_2, \dots, v_L, v], \quad (6)$$

则称  $S_y$  是  $D^* \rightarrow D$  上的垂直选择算子。

一般地, 垂直选择算子  $S_y$  使得一个变异域 (矩阵) 返回到其中某一个变性 (包括它自己)。变异和垂直选择结合在一起便构成了垂直演化算子。

$$E_y = S_y \cdot M. \quad (7)$$

这里:  $M$  是式 (2) 的变异算子;  $E_y$  是  $D \rightarrow D$  上的一对一映射, 它使个体迁移到它的某个变性。

## 4 基于 DNA 进化算法的 Job shop 生产调度的实现

### 4.1 编码方法

编码是 DNA 进化算法实施优化非常关键的一个问题。考虑到 Job shop 调度问题的约束性, 编码技术必须考虑其合法性和可行性。如果直接用工序编号进行编码, 在后面的算法中会产生非法解的个体。所以, 结合 DNA 进化算法的特点, 本文采用对工序进行间接编码, 其可行解是所有待加工工件各个工序组成的序列。每个 DNA 单链中的基因代表一个工件, 并给予这个工件新的含义, 即根据他们在 DNA 单链中出现的顺序确定是该工件的第几道工序。例如, 在 2 台机器 3 个工件的 Job shop 调度问题中, 随机产生了一个可行解 [2 1 1 3 2 3], 第 1 个有下划线的数字“1”代表第 1 个工件的第 1 道工序; 第 2 个有下划线的数字“2”表示第 2 个工件的第 2 道

工序。这样定义编码后,可以保证任意的随机排序都是可行解

#### 4.2 DNA 序列及初始群体的产生

首先要构造一个DNA 单链,这里类似于遗传算法中的染色体,针对Job shop 调度问题,将DNA 分子中每个基因用工件数表示,每个DNA 单链长度应该是总的工序数, DNA 单链中某一工件出现的个数等于该工件的工序数。每次产生一个  $1 \sim n$  (工件数) 的随机数,直到每个工件出现的次数与这个工件的工序数相等时停止。这样,所有的工序就随机产生了,一个个体也就形成了。重复上述过程,多个个体便组成初始群体

#### 4.3 个体适应值的确定

适应值是衡量种群中个体优劣的标志,它是执行DNA 进化算法、优胜劣汰的依据。一般情况下,将所研究问题的目标函数作为适应值函数,但在Job shop 调度问题中,目标函数是所有工件的最短完工时间,优化的目的是使得目标函数尽量小。个体的适应值采用其目标函数,即将待进化种群中的个体按其对应的调度最大完工时间从小到大排列,选择最大完工时间最小的作为这组种群中最优值

#### 4.4 DNA 进化算法中的操作算子

##### 4.4.1 分裂算子

分裂算子是DNA 进化算法中产生新个体的方法,不同于遗传算法中的交叉操作,它根据自身进行繁殖,得到后代。每个个体经过分裂后得到 4 个个体,这 4 个个体分别是经过恒等变换、反演变换、同互补和异互补操作后形成的个体。恒等变换就是保持不变,反演变换是将个体的基因倒置,同互补是将  $n$  (工件数) 加 1 减去恒等变换后得到的个体,异互补是同互补后的个体反演变换后得到的个体。例如在 2 台机器 3 个工件的Job shop 调度问题中,随机产生一个可行解 [3 1 2 1 3 2], 得到 4 个新的个体

[3 1 2 1 3 2], [2 3 1 2 1 3],  
[1 3 2 3 1 2], [2 1 3 2 3 1]

##### 4.4.2 水平选择算子

水平选择的目的是从分裂集中选取 1 个DNA “表形”,水平选择可以使得个体迁移到它的最优表形。根据适应值函数求解出每个个体的适应值,取最优的那个保存下来,进入下一个操作

##### 4.4.3 变异算子

经水平选择后的个体,进行变异操作。DNA 进化算法中的变异算子与遗传中的变异算子有些不同,采用的是解空间域变异,因为人们发现Job shop 调度问题中的解空间域很大,变异操作可以产生更

多的解,对种群个体改良有着非常重要的作用。这里对DNA 进化算法进行了一些改进,提出了并行变异操作,即同时产生两个解空间域,将个体中的第 1 个基因和后面的每 1 个基因进行依次交换,得到一个解的空间域;随机产生  $1 \sim k$  (所有工序数) 的 2 个数,将这 2 个数中间的基因段进行交换,这样又得到 1 个新的解的空间域。这样作的目的是,第 1 个解空间域与原来的个体相似性较大,较好地保存了原个体的特征,第 2 个解空间域使得个体变化较大,容易跳出局部最优解。仿真表明,这种改进方法是有效的

##### 4.4.4 垂直选择算子

垂直选择的目的是从变异域中选择出某一“变型”,使得个体迁移到它的最优变型。将变异后得到的个体,通过适应值函数的求解便可得到最优解

##### 4.4.5 停止准则

以预先所设定的最大进化代数作为停止条件,迭代次数选择在 50 ~ 500 之间

##### 4.4.6 具体实现流程

Step 1: 对Job shop 调度问题进行建模和编码;

Step 2: 试验获取群体的大小,建立随机的初始种群;

Step 3: 对上述群体进行分裂、水平选择、并行变异及垂直选择;

Step 4: 判断是否满足停止条件,若不满足,则将变异得到的可行解代替初始种群,转 Step 3,直到满足条件,找到最优解

## 5 仿真实例

为了判断不同算法的优劣,需采用相同的问题进行测试,从而产生了标准问题。Fisher 等于 1963 年提出了 FT 问题<sup>[5]</sup>,包括 FT06 (6 × 6), FT10 (10 × 10) 和 FT20 (20 × 5)。其中 FT10 是最著名的标准问题。本文的仿真实例,选用问题 FT10,优化目标为整个作业完工时间最少。给出 4 个不同的运算规模,每个规模运行 10 次,取出 10 次运算结果中最好的一次作为最好解,将 10 次运算的最好结果平均作为平均解。采用标准遗传算法进行求解,单点交叉概率为 0.85; 换位变异概率为 0.08。得到的对比结果如表 1 所示

表 1 DNA 进化算法和遗传算法的比较结果

代数 × 规模	遗传算法		DNZ 进化算法	
	最好解	平均解	最好解	平均解
50 × 20	1.063	1.074	998	1.009
100 × 50	1.039	1.048	993	998
200 × 50	1.028	1.034	975	983
500 × 50	1.010	1.021	959	967

从表 1 可以看出,改进的DNA 进化算法的性能

几乎在所有规模中都遥遥领先于普通的遗传算法,收敛能力非常强。从图1可以看出,DNA进化算法不存在遗传算法容易早熟的缺点,因此,本文算法无论在求解有效性还是在计算效率方面均具有较大的优势。

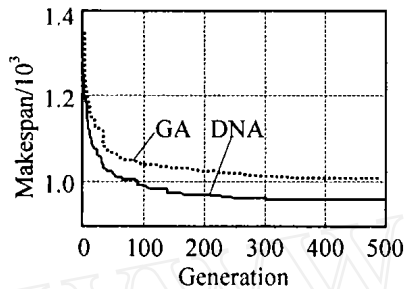


图1 GA和DNA进化算法的收敛曲线图

为了更好地验证算法的有效性,将DNA进化算法与文献[6,7]的7种算法进行了比较,比较结果如表2所示。表中:sGA是局部搜索遗传算法,SA是模拟退火算法,TS是禁忌搜索算法,DNA种群规模为50,DNA进化代数数为500,sGA种群规模为200,sGA进化代数数为500,交叉概率为0.95,变异概率为0.05,SA和TS的迭代次数为1500。

表2 DNA进化算法与其他算法的比较结果

算法	Makespan
DNA	959
Nakano (GA)	965
TS	976
Federico (GA)	946
SA	988
sGA	994
Barker (BAB)	960
Carlier (BAB)	930

从表2的比较结果看,DNA进化算法的结果令人满意。因为sGA种群规模是200,进化代数数是500,SA和TS的迭代次数是1500,而DNA进化算法的种群规模只有50,进化代数数是500,能够在这样

的规模下取得这么好的结果,证明了DNA进化算法求解Job shop生产调度问题确实非常有效。

## 6 结语

本文采用DNA进化算法,对算法中的变异算子加以改进,并将其应用于Job shop生产调度与遗传算法相比,克服了容易早熟的缺陷,提高了优化算法的质量和搜索效率。仿真实例证明,DNA进化算法是解决Job shop生产调度的一个新途径。

## 参考文献 (References)

- [1] Garey E L, Johnson D S, Sethi R. The Complexity of Flow Shop and Job-shop Scheduling [J]. *Mathematics Operations Research*, 1976, 1(2): 117-129.
- [2] Holland J H. *A Adaptation in Natural and Artificial System* [M]. Michigan: The University of Michigan Press, 1975: 1-10.
- [3] 余文, 李人厚. 一种有效的双向进化算法[J]. *小型微型计算机系统*, 2003, 24(3): 527-530.  
(Yu W, Li R H. A Effective Bidirectional Evolutionary Algorithm [J]. *Mini-Micro Systems*, 2003, 24(3): 527-530.)
- [4] 牛群, 顾幸生. 基于DNA进化算法的Flow shop生产调度问题[J]. *上海大学学报*, 2004, 10(S): 88-92.  
(Niu Q, Gu X S. Flow Shop Scheduling Problems Based on DNA Evolutionary Algorithms [J]. *J of Shanghai University*, 2004, 10(S): 88-92.)
- [5] Fisher H, Thompson G L. *Industrial Scheduling* [M]. Englewood Cliffs: Prentice-Hall, 1963: 225-251.
- [6] Ventresca M, Ombukin B M. *Meta-heuristics for the Job Shop Scheduling Problem* [R]. Canada: Brock University, 2003.
- [7] Cai L W, Wu Q H, Yong Z Z. A Genetic Algorithm with Local Search for Solving Job Problems [A]. *Real-world Applications of Evolutionary Computing: Proc of Evo Workshops 2000* [C]. Berlin: Springer-Verlag, 2000: 107-116.

(上接第1146页)

- [3] 王宏飞, 杨成梧. 双被动雷达交会跟踪的精度分析与跟踪算法[J]. *电子学报*, 2003, 31(3): 471-474.  
(Wang H F, Yang C W. Tracking Algorithm and Precision Analysis for Passive Radars from Two Platforms [J]. *Acta Electronica Sinica*, 2003, 31(3): 471-474.)
- [4] 王宏飞, 王永成. 多被动传感器融合跟踪算法[J]. *电子与信息学报*, 2004, 26(8): 1196-1202.  
(Wang H F, Wang Y C. A Tracking Fusion Algorithm for Angle Only Measurements from Multi-sensors [J]. *J of Electronics & Information Technology*, 2004, 26(8): 1196-1202.)
- [5] Farina A. Target Tracking with Bearings Only

Measurements [J]. *Signal Processing*, 1999, 78(1): 61-78.

- [6] Song T L, Speyer J L. A Stochastic Analysis of a Modified Gain Extended Kalman Filter with Applications to Estimation with Bearing-only Measurements [J]. *IEEE Trans on AC*, 1985, 30(3): 940-949.
- [7] Branko Ristic, Sanjeev Arulampalam, Christian Musso. The Influence of Communication Bandwidth on Target Tracking with Angle Only Measurements from Two Platforms [J]. *Signal Processing*, 2001, 81: 1801-1811.