

文章编号:1673-0062(2014)02-0084-05

基于混合蜂群算法的柔性作业车间调度优化

伍大清¹, 罗江琴¹, 李 俊¹, 邱 玲²

(1. 南华大学 计算机科学与技术学院, 湖南 衡阳 421001; 2. 四川理工学院 计算机学院, 四川 自贡 643000)

摘 要:针对柔性作业车间调度求解问题,提出一种新型混合蜂群智能优化算法.该算法采用独特的编码方式和位置更新策略来避免不合法解的产生,在蜂群算法的基础上增加了有利于局部搜索的混沌算子提高了算法搜索精度,实现了全局搜索与局部搜索能力的有效平衡.最后将此算法在不同规模的实例上进行了仿真测试,并与最近提出的其他几种具有代表性的算法进行了比较,验证了该算法的有效性和优越性.

关键词:柔性作业车间调度问题;蜂群优化算法;混沌算子

中图分类号:TP301

文献标识码:B

A Hybrid Artificial Bee Colony Optimization Based on Flexible Job-shop Scheduling Problem

WU Da-qing¹, LUO Jiang-qin¹, LI Quan¹, QIU Ling²

(1. School of Computer Science and Technology, University of South China, Hengyang, Hunan 421001, China; 2. School of Computer Science, Sichuan University of Science and Engineering, Zigong, Sichuan 643000, China)

Abstract: A novel hybrid artificial bee colony optimization algorithm (HABC) is presented for flexible Job-shop Scheduling Problem. This algorithm applies the novel coding mode to scheduling problem directly, and uses the chaotic operator in ABC algorithm to improve the search precision. The simulation results of some classical Job-shop scheduling problems and instance demonstrated that the proposed algorithm could effectively overcome the early-maturing and improve global search capability. Compared to other algorithms, the optimal solution or near optimal solution obtained by the proposed algorithm was better.

key words: flexible Job-shop scheduling problem; artificial bee colony optimization algorithm; hybrid algorithm

收稿日期:2013-09-24

基金项目:湖南省教育厅基金资助项目(13C818;13C819);衡阳市科技局基金资助项目(2013KG63);人工智能四川省重点实验室资助项目(2012RYJ03)

作者简介:伍大清(1982-),女,湖南衡阳人,南华大学计算机科学与技术学院讲师,博士研究生.主要研究方向:智能决策与数据挖掘研究.

0 引言

目前,生产调度问题^[1]得到了学术界和工业界广泛和深入的研究,并取得了很大的进展,但尚存在很多问题.譬如,模型假设过多、约束条件过于简单、调度目标相对单一等.这造成了调度问题的研究与实际应用相差较大,研究成果难以得到实际应用.因此,针对实际生产过程,研究合理有效的建模和优化调度理论与方法具有重要的理论意义和实际应用价值.柔性生产调度问题就是在此背景下应运而生的.柔性调度问题(Flexible Job-shop Scheduling Problem, FJSP)广泛存在于冶金、机械、化工、物流、建筑、造纸、自动装配、半导体生产和交通运输等行业,其最主要的特征在于允许一个工序可在多台机器上进行加工,这更加接近以加工中心为基础的实际工作环境.除了整体上具有传统车间问题的特征,柔性调度问题中某些工序上还存在并行机的特点,因此柔性调度问题比传统调度求解难度更大,在数学上已被证明为 NP-hard 问题.

近年来,智能优化算法已成为求解 NP 问题的有效工具,是诸多学科的热点研究课题,如遗传算法、模拟退火、微粒群算法、蚁群算法等.作为一种新颖的群智能优化方法,人工蜂群算法(artificial bee colony, ABC)^[2]是建立在蜜蜂自组织模型和群体智能基础上的一种优化方法,算法因其机制简单、鲁棒性强和参数少的特点被广泛应用于诸多优化问题,在调度问题方面,目前一些学者也开始尝试 ABC 算法的应用. Pan 等^[3]针对批量流水线调度问题提出了一种基于自适应搜索策略的离散人工蜂群算法,算法使用排列编码并采用插入和交换等操作实现离散空间的搜索, Tasgetiren 等^[4]以总流经时间为目标,研究了置换流水车间调度问题.文献[5]以最优化最小完成时间为目标,研究了带阻塞的流水车间调度问题. Gupta 等^[6]提出了一种改进的人工蜂群算法用于解决作业车间调度问题,并设计了有效的交叉和变异算子.桑红燕^[7]等将蜂群算法以及邻域搜索策略应用于流水车间调度研究; Banharnsakun 等^[8]对群算法中跟随蜂和侦察蜂环节进行了改进,并将其应用于作业. Zhang 等^[9]针对带随机处理时间的动态作业车间调度问题,以最优化拖后时间为目标,提出了有效的人工蜂群算法. Zhou 等^[10]使用加权法将多转化为单目标问题 FJSP,并使用人工蜂群算法对其进行解决.

本文结合蜂群算法的优点,提出一种新型混合蜂群算法(Hybrid artificial bee colony, HABC),初始化阶段采用独特的编码方式和位置更新策略来避免不合法解的产生,综合动态协同 ABC 算法和局部混沌算子策略,实现对复杂柔性作业车间调度问题求解,从而规避了单种智能算法早熟、收敛速度及其精度上的不足.

1 柔性车间调度问题描述

柔性作业车间调度问题描述如下^[1]: n 个工件在 m 台设备上加工. n 个工件的编号用 i 来表示, $i = 1, 2, \dots, n$, 每个工件 i 包含 O_i 道工序, 用 O_{ij} 表示工件 i 第 j 道工序 ($1 \leq j \leq O_i$). m 台设备的编号用 M_k 来表示, $k = 1, 2, \dots, m$, 对每道工序 O_{ij} 都至少有 1 台设备可完成它的加工. 定义 P_{ijk} 设备分配矩阵, 当 $P_{ijk} = 1$ 时, 表示工件 i 的第 j 道工序在设备 M_k 上加工, 否则 $P_{ijk} = 0$. 定义 T_{ijk} 为加工时间矩阵, 当设备 M_k 不能加工此工序时, 则定义 $T_{ijk} = \infty$, 矩阵 S 代表工件的开工时间矩阵, 其中 S_{ij} 为工件 i 的第 j 道工序的开工时间, 矩阵 C 代表工件的完工时间矩阵, 其中 C_{ij} 为工件第 j_i 道工序的完工时间, 本文设定调度的目标是求最小化最大完工时间 C_{\max} , 如公式(1)所示.

$$C_{\max} = \min(\max\{C_{ij} \mid i = 1, 2, \dots, n; j = 1, 2, \dots, m\}) \quad (1)$$

$$\text{s. t. } C_{ij} = S_{ij} + \sum_{k=1}^m (P_{ijk} \times T_{ijk}) \quad (2)$$

$$S_{ij} \geq C_{i(j-1)} \quad (3)$$

$$P_{ijk} \in \{1, 0\}, \sum_{k=1}^m P_{ijk} = 1 \quad (4)$$

$$\sum_{i=1}^n \sum_{j=1}^{O_i} \sum_{k=1}^m P_{ijk} = \sum_{i=1}^n O_i \quad (5)$$

公式(2)~公式(5)表示问题的约束条件, 其中公式(2)表示一个工序的完工时间等于其开工时间加上它在某一具体加工设备上的加工时间之和;公式(3)说明每道工序都必须等其前驱工序加工完成才能开始加工;公式(4)说明对应每个工序,只能将它分配到一台可加工设备上;公式(5)说明必须给所有的工序分配加工设备.

2 基于柔性作业车间调度的混合蜂群优化算法

2.1 种群编码与解码

ABC 算法具有连续性本质,为了解决 FJSP 问题,采用独特的编码方式将一个用浮点数表达的

个体序列转换成工件序列,这样每个个体可以通过它对应的工件序列计算适应值。

采用双层编码来解决 FJSP 调度问题中每道工序的加工设备不唯一的问题,第一层基于设备分配编码,第二层基于工序排序编码。

1) 工序排序编码

一个 D 维的可行解 X 与一个调度方案一一对应,该调度方案是基于所有工序的排序编码,比如编号为 i 的工件共有 j 道工序,在一个调度中将出现 j 次,其第 k 次出现代表该工件第 k 道工序的加工顺序。

2) 设备分配编码

对所有待加工的 S 道工序初始化设备编码进行随机分配,设备分配编码向量中的第 i 个元素对应编号为 O_{ij} 道工序的加工设备编号,那么对于每道工序,一个设备加工选择方案就设定好了。这样编码便于后面可行解位置的更新操作,避免产生不可行的加工设备。

解码时,由编码段可得出各工件的加工序列,由可行解的第二层编码根据工序表可得出每个工序的加工设备,根据时间约束和设备约束将每道工序布置在合适的时间,生成调度方案。

2.2 算法设计

初始化阶段分别将食物源的个数、雇佣蜂数目和观察蜂数目设置为 $n/2$,在此阶段设备分配及工序排序编码方式与第一阶段相同,食物源以及各种蜂也都采用两层编码方式。

1) 雇佣蜂阶段

雇佣蜂在每个食物源周围的局部搜索操作。针对食物源 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 中的一个随机位置 j ,利用雇佣蜂式(6)进行变异操作,生成新的候选食物源 $V_i = (x_{i1}, x_{i2}, \dots, v_{ij}, \dots, x_{id})$;再根据求各个设备对应的工序排列求其适应度值,在候选食物源 V_i 与原食物源 X_i 之间进行选择操作,从而生成 X_i 的新食物源位置。

$$v_{ij} = w \otimes h(x_{ij}) \quad (6)$$

随机产生一个随机数 r ,对于粒子 X_i 每维上的元素值,如果 $r < w$ (w 为粒子 X_i 的变异概率), $h(x_{ij})$ 对食物源的 2 个向量分别调整,操作步骤如下:对选中的食物源 X_i 的两个向量分别进行调整,在 A 向量中随机选择 1 个与其不等的设备号进行替换,在 B 向量中随机交换 2 个不同元素的位置;否则, A 、 B 向量的值保持不变。

2) 观察蜂阶段

根据式(7)计算每个食物源的适应度,拥有

较优目标函数的食物源被选中的概率较大。进而观察蜂在较优的食物源周围进行局部搜索操作。由式(8)计算出每个食物源的选择概率 $prob_i$,进而利用随机轮盘赌法选出一个食物源 X_i ,雇佣蜂执行相同的变异操作。

$$Fitness_i = \frac{1}{1 + f_i} \quad (7)$$

$$prob_i = \frac{Fitness_i}{\sum_{i=1}^{n/2} fitness_i} \quad (8)$$

3) 混沌局部搜索 (CLS)

将食物源适应度函数按照从大到小排列,利用混沌序列的随机性、遍历性和规律性。具体操作如下:CLS 操作只对设备向量 A 进行调整,使向量 A 分配到更合适的机器,从而更好地平衡各机器负载。由于 CLS 的加入必然会增加算法搜索时间,因此只针对最优食物源 X_{best} 的设备 A 向量,进行邻域搜索操作。

步骤 1 在食物源 X_i 中,选出当前种群最优食物源 X_{best} 的机器设备分配向量。

步骤 2 令 $k=0$,利用式(9),将变量 x_i^k 映射成为 0 到 1 之间的混沌变量 cx_i^{k+1}

$$cx_i^k = x_i^k - x_{\min,i}/x_i^k - x_{\max,i}, i = 1, 2, \dots, n \quad (9)$$

其中: $x_{\max,j}$ 和 $x_{\min,j}$ 分别表示定义域的下界和上界。

步骤 3 利用式(10),生成 Logistic 混沌变量 cx_i^{k+1} 。

$$cx_i^{k+1} = 4cx_i^k(1 - cx_i^k), i = 1, 2, \dots, sn/2 \quad (10)$$

其中: cx_i^k 为第 i 个混沌变量, k 表示迭代步数, $cx_i^0 \in (0, 1)$,且 $cx_i^0 \neq \{0.25, 0.5, 0.75\}$,同时 cx_i^k 是 $(0, 1)$ 内均匀分布的随机数。

步骤 4 将混沌变量 cx_i^{k+1} 转化为决策变量 x_i^{k+1} 。

$$x_i^{k+1} = x_{\min,i} + cx_j^{k+1}(x_{\max,i} - x_{\min,i}), i = 1, 2, \dots, n \quad (11)$$

步骤 5 根据决策变量 x_i^{k+1} ,对新解进行性能评价,若新解优于初始解,或者混沌搜索已达到预先设定的最大迭代次数,将新解作为混沌局部搜索的结果,否则令 $k = k + 1$ 并返回步骤 4。

4) 侦察蜂阶段

该阶段只针对选择出来的食物源的工序排序 B 向量进行,对于工序排序方案在参数 limit 范围内没有得到提升,则被放弃,利用式(12)重新生成新位置。

$$x_{ij} = x_{\min,j} + rand(0, 1) \times (x_{\max,j} - x_{\min,j}) \quad (12)$$

其中 $x_{\max,j}$ 和 $x_{\min,j}$ 分别是 X_i 中第 j 维的下限和上限, $rand(0,1)$ 是 $0 \sim 1$ 的随机数.

2.3 时间复杂度分析

本部分针对本文提出的 HABC 算法与基本 ABC 算法的平均计算时间复杂度进行理论分析和比较. 假设种群大小为 n , 迭代总次数为 $\max gen$, 每个粒子每一次迭代需要的运算时间 $T_1 = T_2 = \dots = T_n = T_{abc}$, 则可以得出基本 ABC 算法进行优化所需要的总的运行时间为 $T = \sum_{i=1}^{\max gen} \sum_{j=1}^N (T_{ABC})$, 与之相比, HABC 算法在 ABC 的基础上, 增加了混沌局部搜索算子, 混沌局部搜索算子时间复杂度取决于混沌序列的长度 K , 由于 $K = N$, 故 HABC 算法进行优化所需要的总运行时间为 $T = \sum_{i=1}^{\max gen} \sum_{j=1}^N (T_{ABC} + T_{chaotic})$, 与基本 ABC 算法在同一个数量级, 是比较理想的状态. HABC 算法以较少的算法复杂度为代价换取精度的大幅

提高是值得的, 符合算法改进的基本原则.

3 实验与分析

为了验证本文的 HABC 算法对于求解 FJSP 问题的性能, 选取 OR-library 公布的两类典型算例, 即 FT 类和 LA 类, 各问题规模见表 6, 采用 matlab 2012(a) 编程, 运行环境为 P5 CPU, 主频 2.66 GHz, 内存 4 GB, Windows 7 操作系统, 通过上述分析, 算法参数设置如下, 混沌序列长度 $K = 100$, 算法独立运行 30 次, 最大迭代次数为 5 000, 雇佣蜂数目 = 观察蜂数目 = 50, 侦查蜂参数 $limit = 100$, 最终结果取多次独立运行最优解 (best), 最差解 (worst), 平均值 (mean) 以及平均 CPU 时间 (time) 作为算法对比的尺度. 为了进一步验证混合 HABC 算法的可行性和有效性, 分别将该算法与基本 ABC 算法, 文献 [11] 提出的混合粒子群算法 (HPSO) 测试的最优值进行比较, 结果如表 1 ~ 表 3 所示.

表 1 FT10 和 FT20 的计算结果

Table 1 The result of FT10 and FT20

FT10	HABC	HPSO	ABC	FT20	HABC	HPSO	ABC
最优值	930	930	930	最优值	1 165	1 178	1 165
最差值	946	968	956	最差值	1 170	1 198	1 194
平均值	937.4	956.6	954.5	平均值	1 167.4	1 184.5	117
标准差	3.12	12.42	13.73	标准差	2.93	6.38	8.42

表 2 LA21 和 LA31 的计算结果

Table 2 The result of LA21 and LA31

LA21	HABC	PSO	ABC	LA36	HABC	PSO	ABC
最优值	1 046	1 086	1 067	最优值	1 784	1 792	1 789
最差值	1 088	1 137	1 095	最差值	1 796	1 804	1 802
平均值	1 066.3	1 096.4	1 083.5	平均值	1 786.3	1 797.5	1 798.7
标准差	3.62	13.31	6.42	标准差	7.73	14.52	8.48

表 3 HABC 与其它算法进行比较

Table 3 The result of HABC compared with other algorithms

问题	n, m	$C(\text{best})^*$	HABC		HPSO		ABC	
			Best	Time/s	Best	Time/s	Best	Time/s
FT06	6, 6	55	55	6.8	55	7.0	55	6.7
FT10	10, 10	930	930	14.9	930	19.9	930	14.8
FT20	20, 5	1 165	1 165	37.2	1 178	44.8	1 165	32.7
LA01	10, 5	666	666	5.3	666	14.7	666	5.4
LA06	15, 5	926	926	16.6	926	22.3	926	16.4
LA11	20, 5	1 222	1 222	37.2	1 228	45.5	1 224	39.5
LA16	10, 10	945	945	28.4	946	49.7	952	30.2
LA21	15, 10	1 046	1 046	32.7	1 047	35.6	1 049	32.2
LA26	20, 10	1 218	1 218	43.6	1 223	48.4	1 227	45.5
LA31	30, 10	1 784	1 784	58.5	1 786	69.2	1 788	58.8
LA36	15, 15	1 268	1 269	46.4	1 269	49.9	1 272	46.5

比较表1~表3,可知:HABC最优值、最差值、平均值和标准差均优于PSO和ABC,经历第一阶段广度搜索,随着迭代的进行,带混沌算子的蜂群算法带领种群进入深度搜索,表明带混沌搜索算子的ABC算法能实现两种算法的优势互补,进而提高求解质量,而且该算法具有更高的稳定性和可靠性.HABC算法所获得的最优解等于或者优于其他2个算法所得到的最优解,对于FT类和LA类问题均具有更好的寻优能力,并有效提高解的质量.对于以上11个算例,HABC均能稳定地寻找到最优值或近似最优值,并且在较短时间内得到满意解.说明HABC具有较高的求解精度和鲁棒性.

4 结 论

利用蜂群算法自身优点,制定了一种新型种群编码方法,以避免进化过程中非法解的出现,针对FJSP调度问题的特点,提出了一种新型混合蜂群算法,增大了可行解的搜索空间,提高了算法的搜索精度,仿真实验结果证明了算法的有效性和可行性.

参考文献:

[1] 宋存利. 生产调度问题及其智能优化算法研究[D]. 大连:大连理工大学,2011.

[2] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayser:Erciyes University,2005.

[3] Pan Q K, Tasgetiren M F, Suganthan P N, et al. A discrete artificial bee colony algorithm for the lot-streaming

flow shop scheduling problem[J]. Information Sciences, 2010,181(12):2455-2468.

- [4] Tasgetiren M F, Pan Q K, Suganthan P N, et al. A discrete artificial bee colony algorithm for the total flow-time minimization in permutation flow shops[J]. Information Sciences, 2011,181(16):3459-3475.
- [5] Han Y, Duan J, Zhang M. Apply the discrete artificial bee colony algorithm to the blocking flow shop problem with makespan criterion[C]//Control and Decision Conference(CCDC), 2011 Chinese, IEEE, 2011:2131-2135.
- [6] Pan Q K, Fatih Tasgetiren M, Suganthan P N, et al. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem[J]. Information Sciences, 2011,181(12):2455-2468.
- [7] 桑红燕,潘全科,任立群. 求解批量流水线调度问题的蜂群算法[J]. 计算机工程与应用, 2011,47(21):35-39.
- [8] Banharsakun A, Sirinaovakul B, Achalakul T. Job shop scheduling with the best-so-far ABC[J]. Engineering Applications of Artificial Intelligence, 2012, 25(3):583-593.
- [9] Zhang R, Wu C. An artificial bee colony algorithm for the job shop scheduling problem with random processing times[J]. Entropy, 2011,13(9):1708-1729.
- [10] Zhou G, Wang L, Xu Y, et al. An effective artificial bee colony algorithm for multi-objective flexible job-shop scheduling problem[J]. Advanced Intelligent Computing Theories and Applications, 2012,6839:1-8.
- [11] Xia W J, Wu Z M. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems[J]. Computers and Industrial Engineering, 2005,48(2):409-425.