

求解置换流水线调度问题的改进萤火虫优化算法

张丽红 余世明

(浙江工业大学信息工程学院 杭州 310023)

摘要 针对最小化最大完成时间的置换流水线调度问题,提出了一种改进的离散萤火虫优化算法。在传统萤火虫优化算法的基础上,采用基于升序排序的随机键编码方式对萤火虫种群进行离散化处理,使用 NEH 算法对萤火虫种群进行初始化处理,结合遗传算法的交叉变异思想改进位置更新策略,采用个体变异方式解决孤立个体问题,提高算法的寻优能力。最后通过典型算例对改进算法进行仿真测试,实验结果表明该算法求解置换流水线调度问题时具备很强的寻优能力和鲁棒性,明显优于传统萤火虫优化算法和遗传算法,是解决置换流水线调度问题的一种有效算法。

关键词 置换流水线调度,萤火虫优化算法,NEH 算法,位置更新策略

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.8.048

Improved GSO Algorithm for Solving PFSP

ZHANG Li-hong YU Shi-ming

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract An improved discrete glowworm swarm optimization (DGSO) was proposed for solving the permutation flow shop scheduling problem (PFSP) with the objective of minimizing makespan. On the basis of the traditional glowworm swarm optimization algorithm, this paper used a random key coding method based on the ascending order to discretize the glowworm swarm populations. Then it used the NEH algorithm to initialize the glowworm swarm populations, the idea of crossover and mutation thought in genetic algorithm was used to improve the location updating strategy, and individual mutation was used to resolve individual problems in isolation, improving the capability of algorithm optimization. Finally, the simulation test of the improved algorithm was carried out through typical examples. Simulation results show its efficiency and superiority for solving the PFSP. It is obviously better than the traditional glowworm swarm optimization and genetic algorithm, and it is an effective algorithm for solving flow shop scheduling problem.

Keywords Permutation flow shop scheduling problem, Glowworm swarm optimization algorithm, NEH algorithm, Location updating strategy

1 引言

生产调度就是实现资源的合理配置和优化,是很多企业实现产品的合理化、集成化和自动化及提高生产效益和企业收益的一种有效手段。在企业的生产制造过程中,有 80% 的时间会消耗在工件的运输过程、工件的排队或者工件没有到达机器的等待等非加工过程。因此,研究流水线调度问题具有一定的理论价值和实践意义。

置换流水线调度问题(Permutation Flow Shop Scheduling Problem, PFSP)^[1]是混合流水线调度问题中的一类经典的车间调度问题,也是一个 NP 难题。有关研究资料表明^[2]: 约有 1/4 的组装线或者生产制造系统可以简化成置换流水线调度(PFSP)模型。自从 Johnson 发表了第一篇以流程时间为目标的二阶流水线调度问题的精确算法^[3]后,国内外学者提出了一系列的算法来求解流水线调度问题,同时也让流水线调度问题得到了各界学者的关注,成为了调度优化领域的

研究热点。文献[4]提出并证明了以最大完成时间为评价指标的流水线调度问题是一类 NP 难题。文献[5]在 Johnson 算法和 Plumer 算法的基础上提出了 RA 算法来求解流水线调度问题,提升了原有算法的性能,取得了一定的效果。文献[6]根据工件的总加工时间来确定工件处理的优先级,提出了 NEH 算法来求解流水线调度问题,取得了较好的结果。文献[7]将禁忌搜索算法应用到求解流水线调度问题中,证明该算法的效果优于启发式算法。文献[8]成功应用蚁群算法求解最小化最大完成时间的置换流水线调度算法,取得了较好效果。

萤火虫优化算法(Glowworm Swarm Optimization, GSO)^[9]是一种新兴的群智能优化算法,由印度学者 Krishnanand 等人于 2005 年在基于多声源位置探测的基础上提出的,是一种逻辑简单、参数少、收敛速度快、寻优精度高的随机优化算法。目前,萤火虫优化算法在函数优化领域应用广泛^[10-12],国内文献中尚没有出现在其生产调度方面的应用。

到稿日期:2015-07-13 返修日期:2015-09-19

张丽红(1991-),女,硕士生,主要研究方向为智能优化算法及其应用与仿真,E-mail:zlh562011966@163.com;余世明(1962-),男,博士,教授,博士生导师,主要研究方向为模型预测控制、系统辨识、大系统建模与优化、信号与信息处理、嵌入式系统应用等,E-mail:ysm@zjut.edu.cn.

本文针对复杂置换流水线调度问题,提出解决 PFSP 问题的离散萤火虫优化算法(Discrete Glowworm Swarm Optimization, DGSO),其采用基于升序排序的随机键编码方式对萤火虫种群进行离散化处理,使用 NEH 算法对萤火虫种群进行初始化处理,结合了遗传算法的交叉变异思想改进位置更新策略。通过上述改进,提高了算法解决 PFSP 问题的寻优能力,同时确保算法具备较好的鲁棒性。最后通过标准测试算例验证了改进的 GSO 算法的有效性和优越性。

2 PFSP 数学模型

置换流水线调度问题一般描述为: n 个加工工件要在 m 台机器上加工,每个加工工件需要经过 m 道工序,每道工序要求在不同的机器上加工,每台机器在同一时间只能加工一个工件,其后工件只能在上道工序完成后才能开始工作, n 个加工工件在 m 台机器上的加工顺序相同, n 个加工工件在每台机器上的加工顺序也相同。问题的求解目标是确定 n 个加工工件在每台机器上的最优加工顺序,该顺序能够最小化最大完成时间。本文以最大完成时间为评价指标建立数学模型,每个加工工件在每台机器上的完成时间的数学公式可描述为:

$$T(P_1, 1) = t_{p_1, 1} \quad (1)$$

$$T(P_1, k) = T(P_1, k-1) + t_{p_1, k}, k=2, \dots, m \quad (2)$$

$$T(P_i, 1) = T(P_{i-1}, 1) + t_{p_i, 1}, i=2, \dots, n \quad (3)$$

$$T(P_i, k) = \max\{T(P_{i-1}, k), T(P_i, k-1)\} + t_{p_i, k}, \quad (4)$$

$$i=2, \dots, n; k=2, \dots, m$$

$$T_{\max}(P) = T(P_n, m) \quad (5)$$

在上述数学模型中, $t_{i,j}$ 和 $T(i,j)$ 分别表示加工工件 i 在机器 j 上的加工时间和加工完成时间, $P=(P_1, P_2, \dots, P_n)$ 为 n 个加工工件的一个排序, Π 是所有工件排序的集合。

3 萤火虫优化算法的基本原理

3.1 算法的仿生原理

自然界中大多数萤火虫都会发出短促而有节奏的光,通过光的强度和频率来确定其他个体的存在和吸引力强度,从而进行捕食或者完成求偶、交配和繁殖的使命。萤火虫优化算法就是模拟自然界中萤火虫发光的行为构造出的随机优化算法,算法的执行基于以下 3 点:

(1) 所有萤火虫没有性别之分;

(2) 假定每个萤火虫个体都携带荧光素,荧光素浓度由萤火虫所处位置决定,位置越佳,适应度值越好,对其周围的萤火虫的吸引力越大;

(3) 假定每个萤火虫个体都具备特定的感知半径,萤火虫只能在其感知半径内识别荧光素。

3.2 算法的数学描述与分析

GSO 算法总体可分为 4 大阶段:荧光素值更新阶段、寻找最亮个体阶段、萤火虫位置更新阶段及感知半径更新阶段。

(1) 荧光素值更新的计算公式为:

$$l_i(t) = \max\{0, (1-\rho)l_i(t-1) + \gamma J[X_i(t)]\} \quad (6)$$

其中, ρ 表示荧光素衰减因子, $0 < \rho < 1$; γ 表示荧光素更新率,用于增强荧光素值; $J[X_i(t)]$ 表示萤火虫 i 在时刻 t 所处位置对应的目标函数值。

(2) 用轮盘赌方法寻找最亮个体,其中萤火虫 i 的领域集 $N_i(t)$ 的定义如下:

$$N_i(t) = \{j; d_{ij} < r_d^i; l_i(t) < l_j(t)\} \quad (7)$$

其中, d_{ij} 表示萤火虫 i 和 j 之间的欧几里得距离。若萤火虫 i 有多个邻居,它就会根据荧光素值的概率比例来随机选择邻域集中的一个邻居,其选择概率的公式为:

$$p_{ij}(t) = \frac{l_i(t)}{\sum_{k \in N_i(t)} l_k(t)} \quad (8)$$

(3) 萤火虫位置更新方式为:

$$X_i(t+1) = X_i(t) + s \left[\frac{X_j(t) - X_i(t)}{\|X_j - X_i\|} \right] \quad (9)$$

其中, s 表示移动的步长。

(4) 感知半径的更新公式为:

$$r_d^i(t+1) = \min\{r_d^i, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\} \quad (10)$$

其中, β 是邻域变化率, n_i 表示邻域阈值。

算法的寻优过程是:萤火虫种群随机分布在解空间中,根据式(6)计算每个个体的荧光素浓度;对于每一个个体,通过式(7)的方式选择其邻域集;萤火虫个体向其邻域内优秀个体移动的概率通过式(8)计算;再根据轮盘赌法选择优秀个体移动,式(9)是其位置更新方法,最后根据式(10)修正个体感知半径。

4 求解 PFSP 的改进 GSO 算法

标准的 GSO 算法用来求解连续空间中的优化问题,其位置矢量是实数编码,对于求解 PFSP 问题等离散问题并不合理,效果不佳。本文提出一种 DGSO 算法求解 PFSP 问题。

4.1 编解码方式

标准 GSO 算法的求解空间是连续实数。PFSP 问题的每一个可行解都是离散变量,算法无法实现对离散变量的更新。所以, DGSO 算法首先需要构建从连续求解空间到离散工件排序的映射关系。

本文针对 PFSP 问题的特点,采用基于升序排列的随机键编码的方式。根据随机初始萤火虫个体 i 的位置,得到空间矢量是 $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$,将 $x_{i,1}$ 到 $x_{i,n}$ 进行升序排序,重新得到的 1 到 n 的序列为 $\pi = [j_1, j_2, \dots, j_n]$,这个序列就是工件的加工顺序,这种编解码方式能够有效降低算法的求解难度,提高算法的求解效率。

4.2 基于 NEH 构造启发式算法的初始化

对于置换流水线调度问题,NEH 算法是目前最有效的启发式算法之一。NEH 算法的基本思想是赋予总加工时间越长的工件排序中优先的插入选择权。首先计算每个工件在所有机器上的加工时间的总和,按照递减顺序对其进行排序,对前两个工件进行排序调度,后面工件依次插入到已完成调度的工件序列中的某个位置,直到所有工件调度完成,得到一个次优调度序列。为了使初始种群具备比较好的质量,本文利用 NEH 启发式算法来构造一个初始解,再以 DGSO 算法对其进行优化。

4.3 个体距离计算方法

在标准的 GSO 中,萤火虫个体之间的距离为欧几里得距离,但是在 PFSP 中,在离散空间中欧氏距离已不再适用,所

以需要重新定义个体之间的距离。在 DGSO 中,采用工件序列中的差异来表示个体距离,如式(11)所示:

$$d_{ij} = \frac{\sum_{s=1}^n |x_{j,s} - x_{i,s}|}{N} * c \quad (11)$$

其中, x_{iN} 和 x_{jN} 示萤火虫个体 i, j 在第 N 维的编码; c 为常数,这里取 0.8; N 为分子项取值上限,按照式(12)计算:

$$N = \begin{cases} (n^2 - 1)/2, & n = \text{奇数} \\ n^2/2, & n = \text{偶数} \end{cases} \quad (12)$$

4.4 位置更新策略

在 DGSO 中,萤火虫个体位置更新借鉴遗传算法交叉变异的思想,交叉操作采用遗传算法中应用广泛的 OX(Order Crossover)法^[13],它是 Davis 提出的用于解决车间调度的交叉方式。在两个父串中随机选择一个固定的匹配区域,如:

A=1 2 3|4 5 6 7|8 9

B=3 5 6|9 7 4 2|1 8

首先将匹配区域复制到子代中,得到:

A'=× × ×|4 5 6 7|× ×

B'=× × ×|9 7 4 2|× ×

将匹配序列置于末尾,交换非匹配序列和父串,得到:

A''=1 8 3 5 6 9 7 4 2

B''=8 9 1 2 3 4 5 6 7

获取 A'' 串和 B'' 串中除去匹配序列外的子序列 1-8-3-9-2 和 8-1-3-5-6,将此子序列交换后填充到子代 A' 和 B', 得到最终交叉操作后的子序列:

A* = 3 9 2 4 5 6 7 1 8

B* = 3 5 6 9 7 4 2 8 1

变异采用文献[14]提出的目标导向变异,这是一种基于贪心思想的启发式变异操作。若在一个串中两个相邻码的距离是整个串中相邻码距离最大的,那么认为这两个码相邻的不合理性较大。选择其中一个码与串中任意非相邻码对换,若适应度值未得到优化,则重复此操作,若仍然得不到更好的适应度值,则取两个码中的另一个重复之前的操作,若适应度值依然没有得到优化,则认为这两个码相邻是较合理的,随机选取两个码交换操作;否则,用变异后的串代替原串。

位置更新公式定义如下:

$$x_i^{t+1} = \begin{cases} x_i^t \otimes x_j^t, & rand \leq p_c \\ Mu(x_i^t), & rand > p_c \end{cases} \quad (13)$$

其中, x_i^t 表示萤火虫 i 在第 t 次迭代时所处的位置, x_j^t 表示萤火虫 j 在第 t 次迭代时所处的位置, $rand$ 表示介于 0 和 1 之间的随机数, p_c 是交叉变异阈值概率。

具体的更新过程为:萤火虫 i 在第 $t+1$ 次迭代过程,邻域集内存在多个萤火虫个体,通过轮盘赌法选择萤火虫 j 作为目标,向萤火虫 j 移动,其位置更新策略为产生一个随机数,若 $rand \leq p_c$,则萤火虫 i 选择 $x_i^t \otimes x_j^t$ 即与萤火虫 j 的位置进行交叉,获取萤火虫 j 位置的优良基因序列;若 $rand > p_c$,则萤火虫 i 对当前的位置进行变异操作($Mu(x_i^t)$),产生新的个体位置。最后对新的位置进行判定,若萤火虫 i 所处的新位置的适应度函数优于原位置的,则萤火虫 i 位移至新位置,否则舍弃新位置而保留原位置。

DGSO 采用式(13)的方式同时应用交叉变异对萤火虫个体的位置进行更新的原因是,遗传算法中交叉操作能够使个

体 i 获得个体 j 的优良基因片段,但是仅使用交叉操作容易使整个萤火虫种群陷入局部最优解,降低收敛速度,因此需要使个体 i 在一个较小的概率下出现变异,提高整个种群的抗早熟能力。

4.5 孤立个体的改进

标准 GSO 算法将萤火虫个体感知范围内荧光素浓度高于自身的萤火虫作为它的邻域集,再计算萤火虫个体向它们移动的概率并通过轮盘赌法选择优秀个体。但是,标准 GSO 算法并未给出萤火虫个体不存在邻域集的处理方式,在这种情况下,萤火虫个体将成为孤立个体,不再参与后续的算法操作,降低了算法的效率和精度。DGSO 算法采用个体变异的方式增加孤立个体的搜索能力,增加个体间交叉变异的概率,提高算法的寻优能力。

4.6 DGSO 算法流程

图 1 给出了求解置换流水线车间调度问题的 DGSO 算法的流程,具体如下:

- 1) 初始化算法的基本参数:设置迭代次数 x , 萤火虫种群数量 m , 荧光素衰减因子 ρ , 初始荧光素浓度 $l(0)$, 初始感知半径 r_s , 邻域变化率 β , 邻域阈值 n_s , 交叉变异阈值概率 P_c 等。
- 2) 采用 4.1 节的方式随机初始化萤火虫的位置,其中一个萤火虫的位置采用 4.2 节的方式生成。
- 3) 计算萤火虫个体的适应度函数并更新荧光素浓度。
- 4) 通过 4.3 节的方式计算个体之间的距离,并根据式(7)计算每一个萤火虫个体的邻域集,若当前个体不存在邻域集,采用 4.4 节提出的方式对该个体进行变异操作;若存在邻域集,则采用轮盘赌的方式选择优秀个体。
- 5) 通过 4.4 节的方式对萤火虫个体的位置进行更新操作,更新个体的感知半径。
- 6) 若算法已经达到最大迭代次数,则转步骤 7), 否则转步骤 3)。
- 7) 输出最优调度序列和对应的适应度函数值,算法结束。

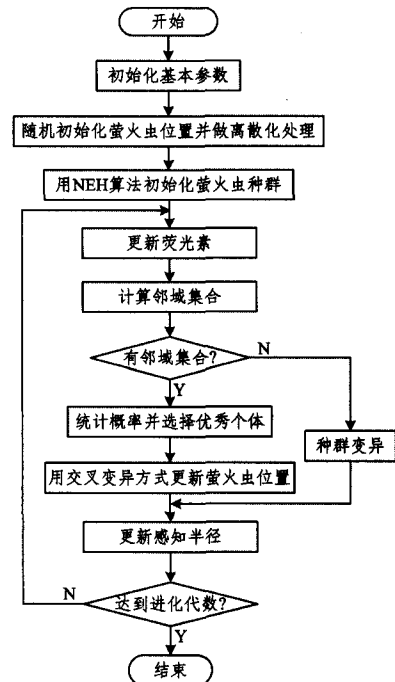


图 1 改进萤火虫优化算法求解 PFSP 问题流程图

5 仿真结果分析

为了验证改进的萤火虫优化算法在求解 PFSP 问题时的性能,本文选取部分 Car 类^[15]标准测试算例和 Taillard 标准算例^[16]进行仿真测试,并与标准萤火虫优化算法、遗传算法进行对比验证。

仿真环境为 2.13GHz 主频 Inter Core i3 处理器,3GB 内存,Windows XP 操作系统,MATLAB R2014 编译软件。DG-

SO 算法参数设置为:最大迭代次数 $x=50$,萤火虫数量 $m=50$,荧光素衰减因子 $\rho=0.4$,荧光素更新率 $\gamma=0.6$,邻域变化率 $\beta=0.08$,初始荧光素浓度 $l(0)=5$,初始感知半径 $r_s=0.5$,邻域阈值 $n_r=5$,交叉变异阈值概率 $P_c=0.75$ 。上述参数是参阅大量文献并经过多次实验后所得的可靠结果。遗传算法参数设置参考文献^[17]。所有算法独立运行 20 次,实验结果如表 1 所列,图 2 给出了 Car5 算法仿真结果。

表 1 标准测试算例实验结果

P	C*	n * m	DGSO			GSO			GA		
			C* _{DGSO}	BRE(%)	PRD(%)	C* _{GSO}	BRE(%)	PRD(%)	C* _{GA}	BRE(%)	PRD(%)
Car1	7038	11 * 5	7038	0	0	7038	0	1.80	7038	0	0.52
Car2	7166	13 * 4	7166	0	0	7166	0	2.38	7166	0	1.62
Car3	7312	12 * 5	7312	0	0.24	7376	0.88	3.61	7312	0	2.68
Car4	8003	14 * 4	8003	0	0.12	8030	0.34	3.27	8003	0	2.51
Car5	7720	10 * 6	7720	0	0.44	7736	0.21	3.60	7720	0	2.33
Car7	6590	7 * 7	6590	0	0	6590	0	2.53	6590	0	1.34
TA002	1359	20 * 5	1367	0.59	0.97	1446	6.40	9.20	1369	0.74	1.57
TA011	1582	20 * 10	1626	2.78	3.56	1875	18.52	25.78	1636	3.41	4.89
TA021	2297	20 * 20	2343	2.00	3.14	2621	14.11	23.67	2369	3.13	4.51
TA031	2724	50 * 5	2730	0.22	0.74	2963	8.77	13.33	2799	2.75	4.02
TA042	3064	50 * 10	3239	5.71	8.02	3793	23.79	32.62	3321	8.39	13.18

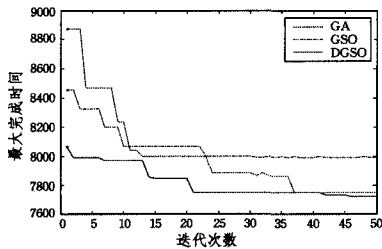


图 2 Car5 仿真结果

图 2 给出的 Car5 是整个 Car 类问题中相对较为复杂的一个实例,采用 NEH 算法对种群进行初始化的 DGSO 算法明显在算法初始时刻就获得了较好的初始解,且在迭代 46 次时发现 Car5 实例理论下界值,优于 GA 算法和 GSO 算法。

表 1 中 DGSO 代表离散萤火虫优化算法,GSO 代表普通萤火虫优化算法,GA 代表遗传算法。除各算法得出的最优解外,采用最优相对误差 (BRE) 和平均相对百分比偏差 (PRD) 衡量各算法性能。 C^* 代表对应的算例的理论值,BRE 表示各个算法求得的最优解与理论解相对偏差;PRD 的计算如式(14)所示:

$$PRD = \frac{1}{n} \sum_{i=1}^n \frac{C_{\min best}^i - C^*}{C^*} \times 100\% \quad (14)$$

其中, n 为独立测试实验次数, $C_{\min best}^i$ 为每次独立实验获得的最优值。根据表 1 的实验数据可知,GSO 算法虽然具备一定的解决小规模 PFSP 问题的能力,但仅获得了 Car1、Car2 和 Car7 的理论解,且 PRD 较大,说明该算法寻优成功率较低,且算法面对大规模的 Taillard 算例寻优偏差较大,精度较低,这是由于标准萤火虫优化算法是一种针对连续求解空间的数学优化算法,不是求解 PFSP 问题的理想算法。本文针对 PFSP 问题建立离散数学模型,对标准 GSO 算法进行离散化处理,并改进位置更新策略和个体距离计算方式等得出的 DGSO 算法相对 GSO 算法在 BRE 和 PRD 指标性能上得到明显提升,相比 GA 算法也具有整体优势。实验结果表明,DGSO 算法具备较好的寻优能力和鲁棒性,是解决 PFSP

问题的一种有效算法。

结束语 针对最小化最大完成时间的置换流水线调度问题,在分析传统 GSO 算法原理的基础上,结合 NEH 算法和遗传算法的特点,提出了全新的 DGSO 算法,利用标准算例进行了测试,结果表明 DGSO 算法处理置换流水线问题的效果优于传统 GSO 算法和遗传算法,是解决该问题的一种有效算法。今后将进一步提高 DGSO 算法的性能并应用其解决更复杂的生产调度问题。

参考文献

- [1] Wang Ling, Wu Hao, Tang Fang. A Hybrid Quantum-Inspired Genetic Algorithm for Flow Shop Scheduling[J]. Lecture Notes in Computer Science, 2005, 3654: 636-644
- [2] Wang L, Zhong D Z. An effective hybrid heuristic for flow shop scheduling[J]. International Journal of Advanced Manufacturing Technology, 2003, 21: 38-44
- [3] Johnson S. Optimal two-and-three stage production schedules with setup time include[J]. Naval Research Logistics Quarterly, 1954, 1: 61-68
- [4] Gupta J N. Two-stage hybrid Flow shop scheduling problem [J]. Operational Research Society, 1988, 39(4): 359-364
- [5] Dannenbring D G. An evaluation of flow shop sequencing heuristics[J]. Management Science, 1977, 23(11): 1174-1182
- [6] Nawaz M, Enscore J E, Ham I A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem[J]. Omega, 1983, 11(1): 91-95
- [7] Gupta J N D, Chen C L, Yap L Y, et al. Designing a tabu search algorithm to minimize total flow time in a flow shop[J]. Arabian Journal for Science and Engineering, 2000, 25(1C): 79-94
- [8] Rajendran C, Zieglerb H. Amt colony algorithms for permutation flow-shop scheduling problem to minimize makespan/total flowtime of jobs[J]. European Journal of Operational Research, 2004, 155(2): 426-438

(下转第 266 页)

通过对 6 个典型的多目标函数的测试可知,本文提出的 IFEMOPSO 算法能更好地收敛于真实 Pareto 前端,且其所产生的非劣解集的分布也更加均匀。

结束语 本文提出的基于直觉模糊集熵的粒子群多目标优化算法 IFEMOPSO 通过基于直觉模糊熵的动态惯性权重调整、种群变异操作以及外部档案使用,很好地控制了种群进化过程中的多样性以及收敛速度,避免了算法的早熟,使其能更有效地在多目标优化问题的空间中进行搜索。实验结果表明,与已有的 MOPSO 和 NSGA-II 算法相比,IFEMOPSO 算法所产生的非劣解集具有更好的收敛性和分散性,能有效地解决多目标优化问题。

参 考 文 献

- [1] Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms[C]//Proc. of the 1st IEEE International Conference on Genetic Algorithms. 1985:93-100
- [2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Trans. on Evolutionary Computation, 2002, 6(2): 182-197
- [3] Kennedy J, Eberhart R C. Particle swarm optimization [C] // Proc. of IEEE International Conference on Neural Networks. 1995:1942-1948
- [4] Coello C A C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization [C] // Proc. IEEE Int. Conf. on Evolutionary Computation. Piscataway: IEEE Service Center, 2002, 2: 1051-1056
- [5] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization [J]. IEEE Trans. on Evolutionary Computation, 2004, 8(3): 256-279
- [6] Wang Hui, Qian Feng. Improved PSO-based multi-objective optimization by crowding with mutation and particle swarm optimization dynamic changing [J]. Control and Decision, 2008, 23

(11): 1238-1242 (in Chinese)

王辉, 钱峰. 基于拥挤度与变异的动态微粒群多目标优化算法 [J]. 控制与决策, 2008, 23(11): 1238-1242

- [7] Zadeh L A. Fuzzy sets [J]. Information and Control, 1965, 8(3): 338-356
- [8] Burillo P, Bustice H. Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets [J]. Fuzzy Sets and Systems, 1996, 78(3): 305-316
- [9] Wang Yi, Lei Ying-jie. A technique for constructing intuitionistic fuzzy entropy [J]. Control and Decision, 2007, 12(22): 1390-1394 (in Chinese)
王毅, 雷英杰. 一种直觉模糊熵的构造方法 [J]. 控制与决策, 2007, 12(22): 1390-1394
- [10] Wang Yu-zhe, Lei Ying-jie, Zhou Lin, et al. Intuitionistic fuzzy discrete particle swarm algorithm [J]. Control and Decision, 2012, 27(11): 1735-1740 (in Chinese)
汪禹喆, 雷英杰, 周林, 等. 直觉模糊离散粒子群算法 [J]. 控制与决策, 2012, 27(11): 1735-1740
- [11] Zhang Kun, Wang Xue. Evaluating and sequencing of air target threat based on IFE and dynamic intuitionistic fuzzy sets [J]. Systems Engineering and Electronics, 2014, 36(4): 697-701 (in Chinese)
张堃, 王雪. 基于 IFE 动态直觉模糊法的空战目标威胁评估 [J]. 系统工程与电子技术, 2014, 36(4): 697-701
- [12] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results [J]. Evolutionary Computation, 2000, 8(2): 173-195
- [13] Van Veldhuizen D A, Lamont G B. Multi-objective evolutionary algorithm research: A history and analysis [R]. Ohio: Air Force Institute of Technology, 1998
- [14] Schott J. Fault tolerant design using single and multicriteria genetic algorithm optimization [D]. Cambridge: Massachusetts Institute of Technology, 1995

(上接第 243 页)

- [9] Krishnanand K N, Ghose D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics [C] // Proc of IEEE Swarm Intelligence Symposium. Piscataway: IEEE Press, 2005: 84-91
- [10] Huang Zheng-xin, Zhou Yong-quan. Self-adaptive step glowworm swarm optimization algorithm for optimizing multimodal functions [J]. Computer Science, 2011, 32(10): 220-224 (in Chinese)
黄正新, 周永权. 自适应步长萤火虫群多模态函数优化算法 [J]. 计算机科学, 2011, 38(7): 220-224
- [11] Liu Jia-kun, Zhou Yong-quan. Glowworm swarm optimization algorithm based on max-min luciferin [J]. Application Research of Computers, 2011, 28(10): 3662-3664 (in Chinese)
刘佳昆, 周永权. 一种最大最小萤光素值人工萤火虫算法 [J]. 计算机应用研究, 2011, 28(10): 3662-3664
- [12] Wu Wei-min, Kang Shao-jiang, Lin Zhi-yi, et al. Multimodal function optimisation based on improved glowworm swarm optimisation [J]. Computer Applications and Software, 2014(1): 283-285, 302 (in Chinese)
吴伟民, 亢少将, 林志毅, 等. 基于改进萤火虫算法的多模态函数优

化 [J]. 计算机应用与软件, 2014, (1): 283-285, 302

- [13] Davis L. Job shop scheduling with genetic algorithms [C] // International Conference on Genetic Algorithms and Their Applications. Mahwah, NJ, Lawrence Erlbaum Associate, 1985: 136-140
- [14] Ge Hong-wei. The research of some optimization problems based on computational intelligence [D]. Jilin: Jilin University, 2006 (in Chinese)
葛宏伟. 基于计算智能的若干优化问题研究 [D]. 吉林: 吉林大学, 2006
- [15] Carlier J. Scheduling with disjunctive constraints [J]. Operations Research, 1978, 12(4): 333-350
- [16] Taillard E. Scheduling instances [OL]. (2015-06-03). <http://mistic.heg-vd.ch/taillard/problems.dir/ordonnancement.dr/ordonnancement.html>
- [17] Wang Shu-ting. Studies on permutation flow shop scheduling using genetic algorithm variable neighborhood search [D]. Changsha: Huazhong University of Science and Technology, 2013 (in Chinese)
王书婷. 基于遗传变邻域算法的置换流水车间调度问题研究 [D]. 长沙: 华中科技大学, 2013