

9-1-2020

Balanced Bee Colony Algorithm Based on Fitness Euclidean-distance Ratio

Zhang Ming

1. *School of Internet of Things, Jiangnan University, Wuxi 214122, China;;*

Tian Na

2. *School of Humanities, Jiangnan University, Wuxi 214122, China;*

Zhicheng Ji

1. *School of Internet of Things, Jiangnan University, Wuxi 214122, China;;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Balanced Bee Colony Algorithm Based on Fitness Euclidean-distance Ratio

Abstract

Abstract: According to the power exploration and poor exploitation ability of artificial bee colony (ABC), a balanced bee colony (FER-ABC) was proposed. *This algorithm modified the search equation based on "fitness Euclidean-distance ratio" and differential algorithm (DE). The FER strategy is useful for multi-optimization and the DE is beneficial to single-optimization. In order to exploit the advantages to full, a new search structure was proposed which balanced the exploitation and exploration.* For continuous problems, the simulations on twelve benchmark functions indicate that this FER-ABC algorithm can improve the accuracy effectively and increase the convergence rate apparently. For the discrete problem, this proposed algorithm is proved to be feasible and advantageous on the simulation of four standard flexible job shop scheduling module.

Keywords

artificial bee colony, balanced bee colony, chaotic strategy, "fitness Euclidean-distance ratio" strategy, differential algorithm

Recommended Citation

Zhang Ming, Tian Na, Ji Zhicheng. Balanced Bee Colony Algorithm Based on Fitness Euclidean-distance Ratio[J]. Journal of System Simulation, 2015, 27(5): 980-989.

基于适应值欧式距离比的均衡蜂群算法

张明¹, 田娜², 纪志成¹

(1. 江南大学物联网工程学院, 无锡 214122; 2. 江南大学人文学院, 无锡 214122)

摘要: 针对人工蜂群算法探索能力强但开发能力弱等特性, 提出一种均衡蜂群算法。该算法根据“适应值欧式距离比”策略和差分算法改进更新公式, “适应值欧式距离比”策略有助于多峰问题的优化, 而差分算法善于优化单峰问题, 为发挥两者的优势, 提出了一种新的搜索结构, 有利于探索与开发能力达到平衡。在初始化时引入混沌策略提高种群多样性。在连续域内, 12 个标准测试函数的仿真结果表明, 本算法能有效地提高最优解的精度, 加快收敛速度。在离散域内, 采用 4 个标准柔性作业车间调度模型, 验证了本算法在解决实际问题中的可行性和优越性。

关键词: 人工蜂群算法; 均衡蜂群算法; 混沌策略; “适应值欧式距离比”策略; 差分算法

中图分类号: TP301.6 文献标识码: A 文章编号: 1004-731X (2015) 05-0980-10

Balanced Bee Colony Algorithm Based on Fitness Euclidean-distance Ratio

Zhang Ming¹, Tian Na², Ji Zhicheng¹

(1. School of Internet of Things, Jiangnan University, Wuxi 214122, China; 2. School of Humanities, Jiangnan University, Wuxi 214122, China)

Abstract: According to the power exploration and poor exploitation ability of artificial bee colony (ABC), a balanced bee colony (FER-ABC) was proposed. This algorithm modified the search equation based on “fitness Euclidean-distance ratio” and differential algorithm (DE). The FER strategy is useful for multi-optimization and the DE is beneficial to single-optimization. In order to exploit the advantages to full, a new search structure was proposed which balanced the exploitation and exploration. For continuous problems, the simulations on twelve benchmark functions indicate that this FER-ABC algorithm can improve the accuracy effectively and increase the convergence rate apparently. For the discrete problem, this proposed algorithm is proved to be feasible and advantageous on the simulation of four standard flexible job shop scheduling module.

Keywords: artificial bee colony; balanced bee colony; chaotic strategy; “fitness Euclidean-distance ratio” strategy; differential algorithm

引言

人工蜂群算法^[1](Artificial bee colony, ABC)是一种基于蜜蜂群体智能行为的优化算法, 由

Karaboga 在 2005 年提出。ABC 算法与其他群算法相比是很有竞争力的, 尤其是控制参数少、机制简单、鲁棒性强等特点。Karaboga 在文献[2-4]中采用 ABC 算法优化多维数值问题, 仿真结果表明 ABC 算法能解决多维工程问题。在文献[5-7]中, ABC 算法已经被应用于解决多种实际问题, 如带树叶约束最小化平移树的问题^[6]等。然而 ABC 算法也有局限性, 比如在处理多峰问题时容易陷入局部最优和停滞状态, 处理单峰问题时收敛速度较慢



收稿日期: 2014-09-23 修回日期: 2015-01-08;
基金项目: 国家高技术研究发展计划课题(2013AA040405);
作者简介: 张明(1990-), 女, 江苏盐城, 硕士生, 研究方向为柔性作业车间调度问题、智能算法优化; 田娜(1983-), 女, 河北石家庄, 博士, 副教授, 研究方向为柔性作业车间调度问题、智能算法优化; 纪志成(1959-), 男, 浙江宁波, 博士, 教授, 博导, 研究方向为高性能电机驱动, 绿色制造物联应用技术等。

<http://www.china-simulation.com>

等问题。原因在于 ABC 算法探索能力强而开发能力弱。探索和开发是互相矛盾的, 前者表示搜索未知解空间得到可行解, 后者表示对当前种群做细致搜索。而 ABC 算法的更新公式随机性较强且只做单维处理, 使得其擅长于探索而非开发。因此, 使探索和开发达到平衡是 ABC 算法的研究重点。

近年来, 许多学者对 ABC 算法做了改进^[8-10]。在文献[8]中, Bansal 对全局最优值进行局部搜索, 并利用黄金分割搜索算法控制搜索步长。Karaboga 在文献[9]中给出了 ABC 算法在不同控制参数下的性能。文献[10]中 Gao 等人改进了搜索公式, 使得蜂群始终在前次迭代的最优解附近开发, 此外该算法还提出新的搜索机制和初始化方法。然而, 到目前为止, 平衡探索和开发能力仍是难点。比如文献[10]所提出的新的搜索机制去除了跟随蜂和侦察蜂阶段, 在一定程度上失去了人工蜂群算法的特色。

文献[11]中 Liang 等人将“适应值-欧式距离比”策略(Fitness Euclidean-distance Ratio, FER)应用到差分算法(Differential Algorithm, DE)中, 对多峰模型极值点的搜索取得良好效果, 但并未考虑到算法的收敛性问题。用 FER 策略和差分算法改进 ABC 算法的更新公式, 使得种群中的每个解与其对应的“最佳邻域解”进行交叉, 平衡开发和探索能力, 此外运用混沌策略完善初始化过程, 增加初始种群的多样性。

由于 FER 对多峰问题的求解效果显著, 因此将该算法离散化后解决柔性作业调度问题^[12-15](Flexible Job-shop Scheduling Problem, FJSP)。FJSP 包含工序排序和机器分配两个问题。工序排序必须遵守工序先后次序的约束, 而机器分配指每道工序可在不同的机器上加工, 并且加工时间各不相同。因此 FJSP 是 NP(Non-deterministic Polynomial)问题。算法在 4 个柔性模型算例上测试取得较好的结果, 验证了本算法解决实际问题的可行性和优越性。

1 人工蜂群算法

ABC 算法主要包含 3 个部分: (1)食物源: 表

示优化问题的解, 每个可行解都由一个优化参数集组成; (2)适应值: 表示食物源的质量, 即可行解的目标函数值; (3)蜂群, 蜂群包含引领蜂、观察蜂和侦察蜂, 引领蜂和食物源的数量相等, 一个引领蜂有且只有一个食物源。

根据公式(1)随机产生初始食物源, 并计算适应值:

$$X_{ij} = X_j^{\min} + \text{rand}(0,1) * (X_j^{\max} - X_j^{\min}) \quad (1)$$

$$f_i = f(X_i), X_i \in R^D, i \in \{1,2,\dots,SN\} \quad (2)$$

X_{ij} 为食物源 (或问题的解), 维度为 D ; X_{ij} 为食物源 X_i 的 j 维位置; X_j^{\max} , X_j^{\min} 分别表示解空间的上限值和下限值; f_i 为 X_i 的适应值。SN 是种群规模。引领蜂在食物源的附近搜索, 并对其质量评价和记忆, 如果 V_i 优于原解 X_i , 那么 V_i 将替换 X_i , 更新公式如下:

$$V_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \quad (3)$$

其中: X_{kj} 为解 X_k 的 j 维, $k = 1,2,\dots,SN$, 并且 $k \neq j$, $j = 1,2,\dots,D$; ϕ_{ij} 是 $[-1,1]$ 之间的随机数。引领蜂将食物源的信息与观察蜂分享, 观察蜂根据食物源质量决定是否跟随。食物源的质量越好被观察蜂选中的概率越高。食物源被选中的概率由下式表示:

$$P_i = f_i / \sum_{n=1}^{SN} f_n \quad (4)$$

观察蜂根据公式(3)对食物源进一步更新。如果食物源的质量经过连续 $Limit$ 次更新仍未得到改善, 该食物源将被抛弃, 引领蜂变为侦察蜂寻找新的食物源。循环迭代直到满足终止条件, 此时算法得到最优解。

2 均衡蜂群算法

2.1 初始化方法

种群初始化在进化算法中非常重要, 它对算法的多样性起重要作用。本论文结合混沌搜索提出一种初始化方法。

2.1.1 混沌搜索

混沌是自然界中广泛存在的一种非线性现象, 具有随机性、遍历性及规律性等特点, 混沌序列指在一定范围内能按其自身规律不重复地遍历所有状态^[16]。因此混沌策略的引入可以提高初始化种群的多样性。混沌搜索策略的主要思想是通过某特定映射方式产生混沌序列, 运用载波方式将混沌序列调整到优化问题的取值空间内。常用的 Logistic 映射就是一个典型的混沌系统, 其方程如下:

$$C(t) = \mu \times C(t-1) \times (1 - C(t-1)) \quad (5)$$

将式(5)看成一个动力学系统, μ 为控制参数, 当 $\mu=4$ 时处于完全混沌状态。由初值 $C_0 = rand(0,1) \cap C_0 \neq \{0, 0.25, 0.5, 0.75, 1.0\}$ 迭代出混沌序列 $[C(1), C(2), C(3), \dots]$ 。

2.1.2 初始化方法

算法 1: 设置混沌迭代次数为 $\max_{chao} = 350$, 种群大小为 SN , 维度为 D 。

初始化方法伪代码如下:

```

for i = 1 : D
    for k = 1 : max_chao
        C_0 = mu * C_0 * (1 - C_0)
    end
    chao_0i = LB + (UB - LB) * C_0
end
for i = 1 : SN
    for j = 1 : D
        chao_ij = mu * chao_(i-1)_j * (1 - chao_(i-1)_j)
        X_ij = LB + (UB - LB) * chao_ij
    end
end

```

2.2 改进的更新公式

2.2.1 适应值欧式距离比

“适应值欧式距离比”^[11]是由 Li 提出, 它是“适应值距离比(Fitness Distance Ratio, FDR)”的衍生。FER 是指邻域解和最差解的适应值之差与目

标解和邻域解间的欧式距离的比值(针对最小化问题)。2014 年 Liang 等人将 FER 引入到差分算法中, 提出一种改进差分算法 FER-DE, 用于寻找多峰问题的极值点。FER 的计算见公式(6)。

$$FER_j = \frac{|f_j - f_w|}{X_i - X_j} \quad (6)$$

其中: f_w 为最差适应值; X_j 是种群中的任意一个解, 将欧式距离大于平均值的解的 FER 值设置为 0, 如

$$FER_j = \begin{cases} 0, & i = j \text{ or } X_i - X_j \geq \frac{1}{SN} \sum_{j=1}^{SN} (X_i - X_j) \\ FER_j, & \text{otherwise} \end{cases} \quad (7)$$

2.2.2 更新公式

ABC 算法通过公式(3)更新解, 然而邻域解 X_k 是任意选择的, 因此生成的新解有助于保持多样性但是收敛速度过慢。若用全局最优解 G_{best} 代替 X_k , 虽然能提高收敛速度, 但是如果 G_{best} 与 X_i 的欧式距离过大, 不利于优化多峰函数。本文基于均衡的思想并借鉴差分算法提出 2 种更新公式 ABC/current-to-best/1 和 ABC/rand/2, 见公式(8)和(9):

$$V_{ij} = X_{ij} + \phi_{ij} \times (G_{best,j} - X_{ij}) \quad (8)$$

$$V_{ij} = X_{1j}^{nBest} + \omega_1 \times (X_{2j}^{nBest} - X_{3j}^{nBest}) - \omega_2 \times (X_{4j}^{nBest} - X_{5j}^{nBest}) \quad (9)$$

$$\phi_{ij} = \left(0.2 + 0.8 \times \frac{\max FE - FE}{\max FE} \right) \times (2rand - 1) \quad (10)$$

其中: ω_1, ω_2 均为 $[-1,1]$ 之间随机数; $G_{best,j}$ 为 G_{best} 的 j 维; $\{X_{pj}^{nBest} | p=1,2,\dots,5\}$ 分别为 X_j 的“最佳邻域解” $\{X_p^{nBest} | p=1,2,\dots,5\}$ 的 j 维。

蜂群算法搜索空间的大小由参数 ϕ_{ij} 控制, 如果参数 ϕ_{ij} 过大, 则搜索空间增大, 最优值可能被轻易忽略, 不易于收敛; 如果参数 ϕ_{ij} 过小, 大部分个体在局部机制附近运行, 算法容易出现过早收敛和停滞现象, 难以突破局部极值, 甚至陷入局部最优点。因此在公式(10)中 ϕ_{ij} 随着迭代次数的增加递减, 在迭代过程前期种群向全局最优解靠拢, 而在迭代过程后期种群侧重于对自身的进一步优化。其中 FE 表示当前函数评估次数(Function Evaluation, FE), $\max FE$ 表示函数最大评估次数。

2.2.3 “最佳邻域解”的选择

FER 值越大证明目标函数值越小且距离 X_i 越近。公式(8)采用 G_{best} 来指引全局搜索方向, 这可以提高单峰函数的收敛速度和精度, 然后却容易使得多峰函数陷入局部最优点。此时需要增加种群的多样性; 如果公式(9)中的解是从种群中的解随机选择其他解进行交叉, 虽然提高了种群的多样性, 却减弱了收敛能力。因此公式(9)采用 FER 策略选“最佳邻域解” X^{nbest} 进行更新, 使得解 X_i 向其邻域内的最优解靠拢, 这样在保证收敛精度的同时, 又不失多样性。本文运用锦标赛策略选出 5 个“最佳邻域解”。以下是伪代码:

算法 2: 首先计算解 X_i 的 FER 值;

for $i = 1:5$

 运用锦标赛策略选出解 X_i ;

$X_{ij}^{nbest} = X_k$;

$FER_k = 0$;

end

2.3 均衡蜂群算法流程

根据“适应值欧式距离比”策略和差分算法改进更新公式, “适应值欧式距离比”策略有助于多峰问题的优化, 而差分算法善于优化单峰问题, 为了使两者达到平衡, 提出了一种新的搜索结构和选择概率 on_para , on_para 的大小影响算法的收敛速度。算法内容如下(图 1 是算法流程图):

Step 1: 设置算法参数, 包括最大迭代次数 $\max FE$ 、种群规模 SN , $Limit$ 等;

Step 2: 利用算法 1 得出初始化种群 $\{X_i | i = 1, 2, \dots, SN\}$, 计算适应值 $\{f_i | i = 1, 2, \dots, SN\}$;

Step 3: 引领蜂阶段:

(1) 对种群中的任意解 X_i , 随机选择一个解 $\{\forall X_k | k \in \{1, 2, \dots, SN\} \cap k \neq i\}$, 选择任意维度 j , 生成随机数 ϕ_{ij} ;

(2) 根据式(8)对 X_i 进行邻域搜索, 得出新解 V_i , 并计算其适应值;

(3) 运用贪婪算法更新解;

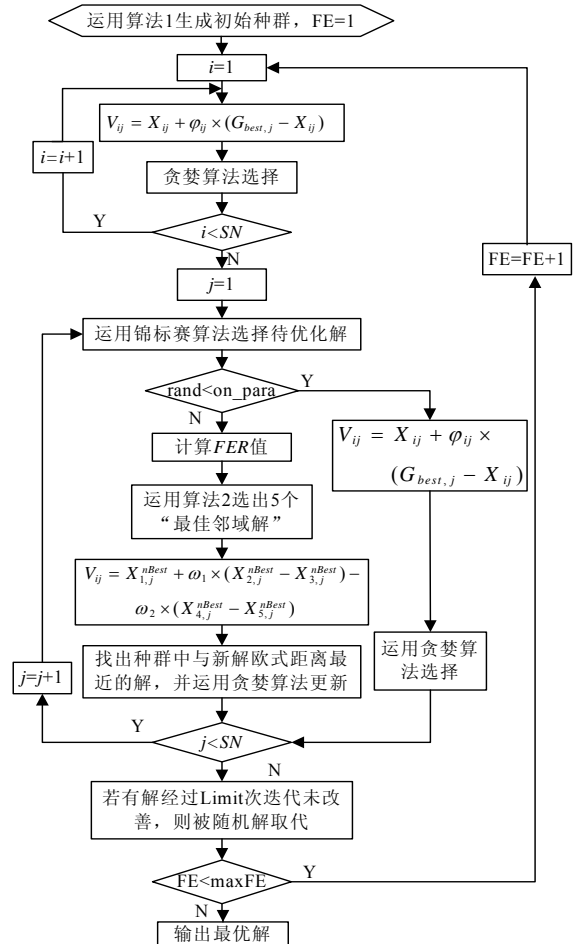


图 1 FER-ABC 算法流程

Step 4: 观察蜂阶段

(1) 利用式(4)计算种群选择的概率 P_i ; 设置选择参数 on_para ;

(2) 若满足循环条件则执行 Step 4.(3), 否则执行 step4.(7);

(3) 根据锦标赛策略选择解 X_i ;

(4) 如果 $rand(0,1) < on_para$, 执行 Step 4.(5), 否则执行 step4.(6-7);

(5) 根据式(8)对 X_i 进行邻域搜索, 得出新解 V_i , 运用贪婪算法更新解, 执行 Step 4.(8);

(6) 计算 FER 值, 运用算法 2 选出“最佳适应值”解;

(7) 根据式(9)对 X_i 进行邻域搜索, 得出新解 V_i , 在种群中找出与 V_i 的欧式距离最近的解, 并用贪婪算法更新, 执行 Step 4.(8);

(8) 判断循环是否结束, 若是, 则跳到 Step 5,

否则执行 Step 4.(3);

Step 5: 侦察蜂阶段

判断是否有解经过连续 $Limit$ 次的函数估计后仍没有改善, 如果有, 则根据式(1)产生新解替换该解。

Step 6: 返回 Step 3, 直至满足终止迭代次数。

2.4 仿真结果与分析

实验种群规模 SN 设为 75, 维度 D 为 30 维和 60 维, 最大函数评估次数 $\max FE$ 分别为 150 000 和 300 000, 运行次数 $Nrun$ 为 30。Logistic 混沌搜索中 $\mu=4$, $on_para=0.5$, $Limit=D \times SN$ 。FER-ABC 算法在 12 个 benchmark 函数上进行测试, 如表 1 所示。在表 1 中 $f_1 \sim f_6$ 为连续单峰模型函数, f_7 是噪声四次函数。 f_8 是 Rosenbrock 函数, 它在 $D=2$ 或者 $D=3$ 是单峰模型, 当维数增加时, 多峰极小点也会增加。 $f_9 \sim f_{11}$ 是多峰模型函数, 且多峰局部极值点随着维度的增加成指数增长。

表1 12个标准测试函数

函数	取值范围	最小值
$f_1(X) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0
$f_2(X) = \sum_{i=1}^D (10^6)^{(i-1)/(D-1)} x_i^2$	$[-100, 100]^D$	0
$f_3(X) = \sum_{i=1}^D ix_i^2$	$[-10, 10]^D$	0
$f_4(X) = \sum_{i=1}^D x_i ^{(i+1)}$	$[-10, 10]^D$	0
$f_5(X) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0
$f_6(X) = \max_i \{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^D$	0
$f_7(X) = \sum_{i=1}^D ix_i^4 + rand[0,1]$	$[-1.28, 1.28]^D$	0
$f_8(X) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	0
$f_9(X) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10]$	$[-5.12, 5.12]^D$	0
$f_{10}(X) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0
$f_{11}(X) = 418.9829 \times D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	$[-500, 500]^D$	0
$f_{12}(X) = \sum_{i=1}^D (x_i + 0.5)^2$	$[-100, 100]^D$	0

为检验 FER-ABC 算法的性能, 将 FER-ABC 算法、MABC^[10](Modified Artificial Bee Colony Algorithm)、标准 ABC 算法的仿真结果进行比较。表 2 记录了 3 种算法在不同函数和不同维数下的均值和方差。从实验对比结果看出, FER-ABC 算法明显优于 ABC 算法, 在函数 f_9 (仅限 30 维)、 f_{10} 和 f_{12} 上算法可以收敛到最小值 0。与 MABC 算法比较时, FER-ABC 占据明显优势, 尤其是对函数 f_6 , f_7 的优化, 仅在对函数 f_9 (仅限 $D=60$) 以及函数 f_{11} 的优化结果较劣于 MABC。FER-ABC 算法的收敛精度和收敛速度明显得到提高。

图 2~5 显示了 FER-ABC 和 ABC 的收敛曲线对比, FER-ABC 在速度和精度两方面都远胜于 ABC。值得注意的是, FER-ABC 在收敛前期与 ABC 相比不占明显优势, 尤其是图 3; 但是在收敛后期, 除了图 3 趋于缓慢之外, FER-ABC 的收敛曲线近似于直线。这充分说明 FER-ABC 能够在提高收敛精度的同时, 保证种群的多样性。

3 FER-ABC 在 FJSP 问题中的应用

3.1 柔性作业车间调度模型

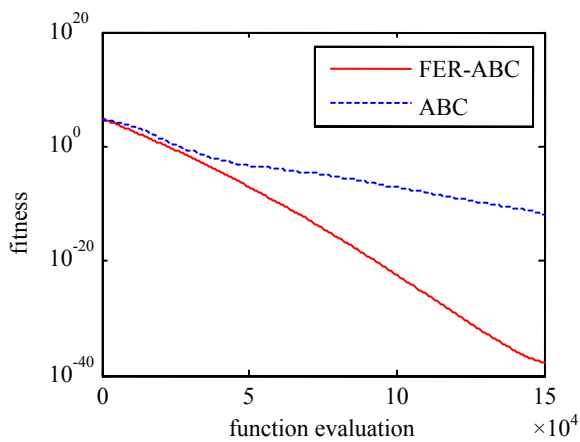
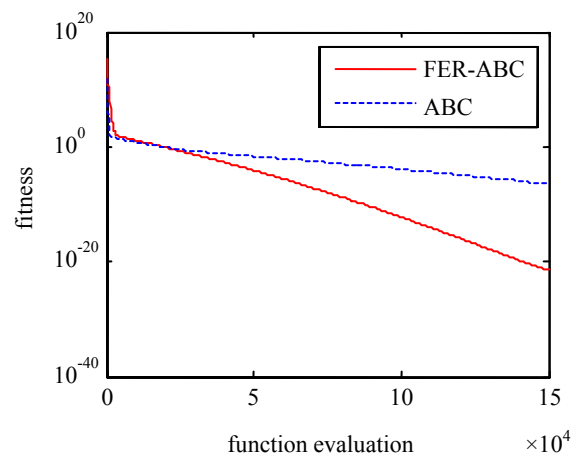
设 N 个工件在 M 台机器上加工, 工件集 $J = \{J_i\}_{1 \leq i \leq n}$, 任意工件 J_i 包含工序集 $O_{i1}, O_{i2}, \dots, O_{ij}, \dots, O_{in_i}$, n_i 为工件 J_i 的工序数。每个工件必须按照给定的工序顺序进行加工, O_{ij} 为工件 J_i 的第 j 步操作; 机器集合为 $M = \{M_k\}_{1 \leq k \leq m}$, 每台机器只能同时加工一道工序, M_{ij} 表示工件 J_i 的 j 步操作的可行机器集, 任意工序 O_{ij} 只能从可行机器集 M_{ij} 中选取一台机器加工, 且加工过程不可间断; S_{ij} 和 C_{ij} 分别表示工序 O_{ij} 的开始时间和完工时间, P_{ijk} 表示工序 O_{ij} 在机器 M_k 上的加工时间, 相同工序在不同机器上的加工时间不同, 所有的工件和机器在时间为 0 时都是可用的。

FJSP 研究的目的是为每一道工序分配一个合适的机器, 并且对所有工序进行排序, 使得加工时间最小化, 即完成所有工件的时间最少:

$C_M = \max_{1 \leq i \leq n} \{C_{i,n_i}\}$ 。在 ABC 算法中, 本文将 行表示工序, 列表示机器, 数值表示每道工序在不同机器上的操作时间。表 3 中,

表2 算法性能比较

函数	维度	ABC		MABC		FER_ABC	
		均值	方差	均值	方差	均值	方差
f_1	30	1.64e-12	1.66e-24	9.43e-32	6.67e-32	2.345e-38	8.902e-76
	60	6.135e-12	1.465e-23	6.03e-29	4.31e-29	7.741e-36	7.96e-71
f_2	30	5.543e-09	1.971e-17	3.66e-28	5.96e-28	3.607e-33	7.525e-65
	60	1.5682E-08	1.438e-16	3.51e-25	2.72e-25	7.582e-31	2.247e-60
f_3	30	2.963e-13	2.21e-26	2.1e-32	1.56e-32	1.943e-38	7.313e-76
	60	2.42411E-12	3.325e-24	1.39e-29	8.84e-30	4.93e-36	2.376e-71
f_4	30	9.503e-17	1.32e-34	2.7e-69	5.38e-69	1.119e-79	4.13e-158
	60	5.8103E-15	7.94e-29	3.00e-62	3.87e-62	1.687e-68	2.560e-135
f_5	30	3.606e-07	5.34e-15	2.40e-17	9.02e-18	3.32e-22	1.036e-44
	60	1.26E-06	7.87e-14	6.96e-16	1.20e-16	1.53e-23	3.11e-46
f_6	30	16.796	5.092	10.2	1.49e-00	0.00105	1.58e-06
	60	42.022	4.321	37.7	3.14e-00	2.7576	1.056
f_7	30	2.84e-16	4.12e-33	0.0371	8.53e-03	2.59e-63	1.26e-125
	60	0.219	0.00133	0.114	1.16e-02	0.0068	1.31e-06
f_8	30	4.10e-01	4.13e-01	0.611	4.55e-01	0.1618	2.53e-01
	60	1.71e-00	1.57e-00	1.51	1.34e-00	1.10	1.23e-00
f_9	30	0.103	0.090	0	0	0	0
	60	1.48e-00	8.9e-01	0	0	5.2	6.16
f_{10}	30	8.15e-17	4.593e-09	0	0	0	0
	60	5.35E-10	1.478E-18	0	0	0	0
f_{11}	30	8.90e+01	8.12e+01	1.21e-13	4.53e-13	0.0241	7.82e-04
	60	5.21e+02	1.32e+02	3.56e-11	2.18e-12	0.024	0.0015
f_{12}	30	0	0	0	0	0	0
	60	0	0	0	0	0	0

图2 f_1 在 30 维的进化曲线图3 f_5 在 30 维进化曲线

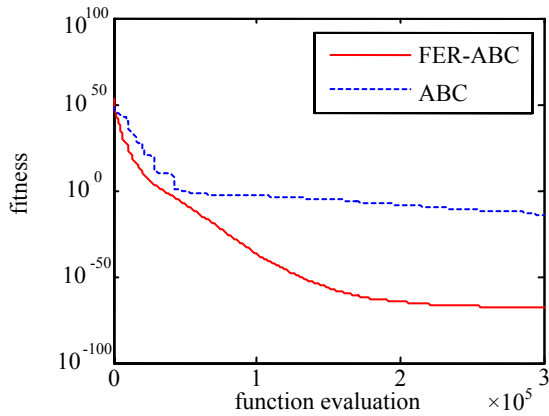
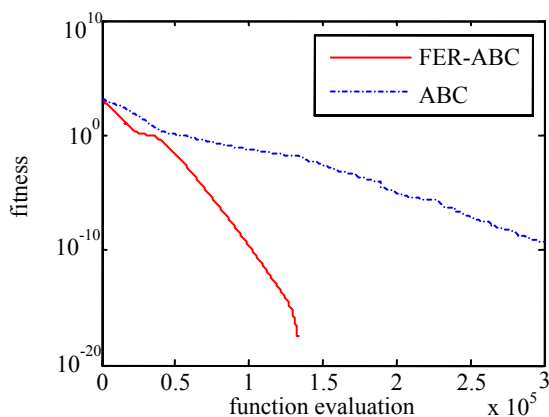
图4 f_4 在 60 维进化曲线图5 f_{13} 在 60 维的进化曲线

表3 工序操作时间表

工件	工序	M_1	M_2	M_3	M_4
J_1	Q_{11}	1	3	4	1
	Q_{12}	3	8	2	1
	Q_{13}	3	5	4	7
J_2	Q_{21}	4	1	1	4
	Q_{22}	2	3	9	3
	Q_{23}	9	1	2	2
J_3	Q_{31}	8	6	3	5
	Q_{32}	4	5	8	1

3.2 编码

FJSP 对解 X_i 的编码包含两部分: A, 工序排序矢量; B, 机器分配矢量(见表 4), A, B 向量的长度相等, 等于总工序数。A 向量的各个元素是工件号, 每个工件的所有工序都由该工件的工件号表示, 工件号的出现次数代表工件的工序数; B 向量的各元素代表对应工序的机器号。每道工序在解中的维度表示该工序的加工位置, 如工序 O_{11} 位置为 1, 工序 O_{21} 位置为 2 等。将加工位置按照表 3 第

二列的工序顺序(即 $O_{i1}, O_{i2}, \dots, O_{ij}, \dots, O_{im_i}$) 进行排列组成加工位置矢量 P_i $P_i^T = \{1, 3, 5, 2, 4, 7, 6, 8\}$ 。

表4 工序排序和机器分配

位置	1	2	3	4	5	6	7	8
A	1	2	1	2	1	3	2	3
B	1	3	2	4	1	2	3	4

3.3 初始化

3.3.1 机器分配方法

(1) 全局机器分配方法^[17]

Kacem 等人^[17]提出的定位方法。首先要在工序操作时间表(见表 3)中找出全局最短加工时间以及对应的工序和机器, 将该机器分配给该工序, 且该机器所在列的所有工序需加上加工时间(即机器的负载更新), 按照上述步骤, 直到所有工序的机器分配完毕。

(2) 有序机器分配方法^[18]

首先需要确定工序加工顺序, 按照加工顺序从第一道工序开始为每道工序分配加工时间最小的机器, 机器所在列加上加工时间。

全局机器分配的机器分配顺序由全局最小加工时间确定, 有序机器分配的机器顺序由工序的加工顺序确定。

3.3.2 工序排序方法

在对工序的排序中, 通过 3.2 节编码可知, 工序的排序与加工位置矢量是一一对应的关系, 因此将加工位置矢量作为初始化输入, 通过 2.1 节内容对加工位置矢量进行计算和处理和排序后, 将排序索引结果作为新的加工位置矢量, 然后将其转换为工序排序矢量。

3.3.3 初始化过程

Step 1: 运用 2.2 节的初始化方法生成初始加工位置矢量种群, 并将其转换为工序排序矢量;

Step 2: 分别采用有序机器分配方法和有序机器分配方法为工序分配机器;

Step 3: 运用贪婪算法选择出 SN 个解。

3.4 FER-ABC 算法在 FJSP 中的应用

为了解决 FJSP 问题需要将 FER-ABC 算法更新公式离散化。求解 FJSP 问题时涉及计算 FER 值, 将最大加工工时作为适应值, 加工位置矢量作为欧式距离计算的输入。

Step 1: 雇佣蜂阶段

(1) 对任意解 X_i , 从种群中随机选择一个解 X_k , 对 X_i 和 X_k 进行 POX(precedence preserving order based crossover)交叉生成解 V_i , 具体见文献[18];

(2) 评估 X_i 和 V_i 的适应值, 进行贪婪选择;

(3) 对任意解 X_i 的机器分配矢量, 运用改进均匀交叉策略 MUP(Modified Uniform Crossover)更新机器生成新解 W_i ;

(4) 评估 X_i 和 W_i 的适应值, 进行贪婪选择;

POX 交叉: 是指从工件集中随机生成一个工件子集 $U \in J$, 对解 X_i 和 X_j , 将 X_i 中属于工件子集 U 的所有工序按照既定位置完全复制到新解 X_i^{new} 中, X_j 中不属于 U 的工件工序依次填入到的 X_i^{new} 剩余位置。

MUP 交叉: 是指为工序排序矢量随机生成一组由 0 和 1 组成的序列, 对数值为 1 的所有工序, 在其机器集内选择加工时间最小的机器更新机器分配矢量。

Step 2: 观察蜂阶段

(1) 利用式(4)计算种群的选择概率 P_i ; 设置选择参数 on_para ;

(2) 若满足循环条件则执行 Step2.(3), 否则执行 Step2.(7);

(3) 根据锦标赛策略选择解 X_i ;

(4) 随机生成小数 $rand(0,1)$, 如果 $rand < on_para$, 执行 Step2.(5), 否则执行 Step2.(6)-(7);

(5) 对 X_i 进行采用 Neighboring Structure Se2^[19]更新和邻域智能结构 II^[19], 得出新解 V_i , 运用贪婪算法更新解, 执行 Step 2.(8);

(6) 计算 FER 值, 运用算法 2 选出“最佳适应值”解;

(7) 对 X_i 进行 MUP 交叉和 POX 交叉, 得出新解 V_i , 在种群中找出与 V_i 的欧式距离最近的解, 用贪婪算法更新, 执行 Step 2.(8);

(8) 判断循环是否结束, 若是, 则跳到 Step 3, 否则执行 Step 2.(3);

Step 3: 侦察蜂阶段

判断是否有解经过连续 $Limit$ 次的函数估计后仍没有改善, 如果有, 则产生新解替换该解;

Step 4: 返回 step1 直至满足终止条件。

3.5 仿真与分析

为验证算法效果, 采用 MATLAB2013a 开发环境, 在标准算例集 XWdata^[20]中选取 4 个算例进行测试。表 5 为算例的比较结果, 其中给出定位与控制遗传算法(AL+GCA)^[17]、改进遗传算法(改进 GA)^[21], ABC 算法、混合蜂群算法^[22]以及均衡蜂群算法(FER-ABC)在 XWdata 算例中的比较。从对比结果上看, FER-ABC 的效果远远好于 AL+CGA 算法和 ABC 算法, 与混合蜂群算法的结果一致, 虽然改进 GA 算法 15×10 模型的结果优于 FER-ABC 算法, 但是考虑到它的种群规模和迭代次数均比 FER-ABC 大, 因此改进 GA 算法不占优势。图 6 给出了 FER-ABC 算法在 15×10 问题上所得最优解的甘特图(横坐标表示加工时间, 纵坐标表示机器号)。

表 5 XWdata 算例集对比

问题	AL+GCA	改进 GA		ABC	混合蜂群		FER-ABC		
	最优结果	最优结果	种群规模/迭代次数	最优结果	种群规模/迭代次数	最优结果	种群规模/迭代次数	最优结果	种群规模/迭代次数
4×5	-	11	300/200	11	100/50	-	-	11	100/50
8×8	16	14	300/200	14	100/50	14	100/50	14	100/50
10×10	7	7	300/200	8	100/50	7	100/50	7	100/50
15×10	23	11	300/200	14	100/50	12	100/50	12	100/50

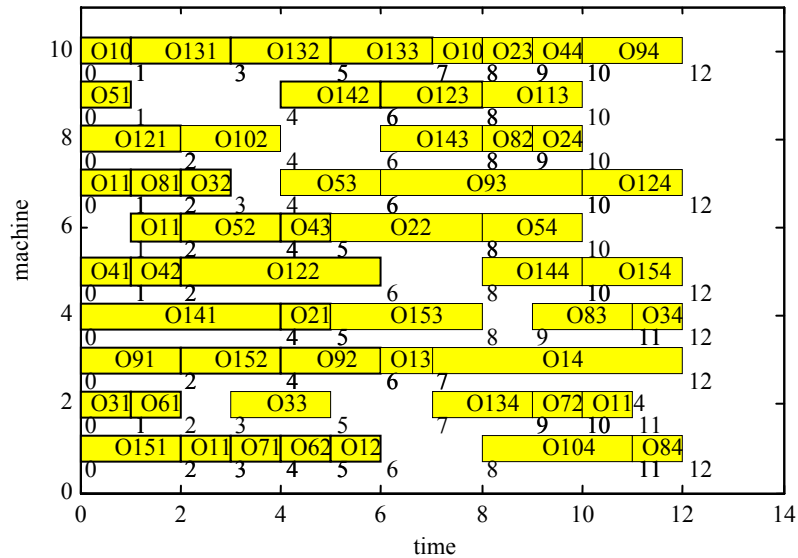


图6 15×10模型甘特图

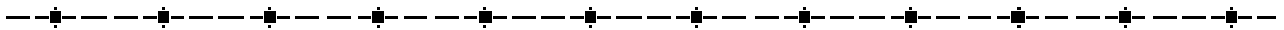
4 结论

针对 ABC 算法的特性, 提出了一种均衡蜂群算法。结合“适应值-欧式距离比”和差分算法对算法的更新公式进行改进。并运用混沌序列进行初始化, 使得算法的开发与探索能力达到平衡。在 12 个 benchmark 函数和 4 个标准的柔性作业车间调度模型上的仿真结果表明, 在函数优化问题上, FER-ABC 算法的收敛速度和收敛精度都优于基本 ABC 算法, 而且其全局搜索与跳出局部最优的能力均强于基本 ABC 算法。在与 MABC 算法的比较中, FER-ABC 算法占据很大优势。此外本文将 FER-ABC 算法求解 FJSP 问题, 在与其他算法的比较中, 也有一定的优势。在未来的工作中, 还将对 FJSP 问题进行深入的研究。

参考文献:

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization [D]. Turkey: Technical Report TR06, Computer Engineering Department, Erciyes University, 2005.
- [2] Karaboga D, Basturk B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm [J]. Journal of Global Optimization (S0925-5001), 2007, 39(3): 459-471.
- [3] Karaboga D, Basturk B. On the Performance of Artificial Bee Colony (ABC) Algorithm [J]. Applied Soft Computing (S1568-4946), 2008, 8(1): 687-697.
- [4] Karaboga D, Akay B. A Comparative Study of Artificial Bee Colony Algorithm [J]. Applied Mathematics and Computation (S0096-3003), 2009, 214(1): 108-132.
- [5] Karaboga D, Gorkemli B, Ozturk C, et al. A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications. Artificial Intelligence Review (S0269-2821), 2014, 42(1): 21-57.
- [6] Draa A, Bouaziz A. An Artificial Bee Colony Algorithm for image contrast enhancement [J]. Swarm and Evolutionary Computation (S2210-6502), 2014, 16: 69-84.
- [7] Sabats L, Udgata S K, Abraham A, et al. Artificial Bee Colony Algorithm for Small Signal Model Parameter Extraction of MESFET [J]. Engineering Applications of Artificial Intelligence (S0952-1976), 2010, 23(5): 689-694.
- [8] Bansal J C, Sharma H. Memetic Search in Artificial Bee Colony Algorithm [J]. Soft Computing (S1432-7643), 2013, 17(10): 1911-1928.
- [9] Akay B, Karaboga D. A Modified Artificial Bee Colony Algorithm for Real Parameter Optimization [J]. Information Sciences (S0020-0255), 2012, 192: 120-142.
- [10] Gao W F, Liu S Y. A Modified Artificial Bee Colony Algorithm [J]. Computers & Operations Research (S0305-0548), 2012, 39(3): 687-697.
- [11] Liang J J, Qu B. Y, Mao X B, et al. Differential

- Evolution Based on Fitness Euclidean-distance Ratio for Multimodal Optimization [J]. *Neurocomputing* (S0925-2312), 2014, 137: 252-260.
- [12] Arit T, Ajchara P A. A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem [J]. *Procedia Computer Science* (S1877-0509), 2013, 20: 96-101.
- [13] Wang L, Zhou G, Xu Y, *et al.* An Effective Artificial Bee Colony Algorithm for the Flexible Job-Shop Scheduling Problem [J]. *The International Journal of Advanced Manufacturing Technology* (S0268-3768), 2012, 60(1-4): 303-315.
- [14] Li J Q, Pan Q K, Tasgetiren F. A Discrete Artificial Bee Colony Algorithm for the Multi-Objective Flexible Job-Shop Scheduling Problem with Maintenance Activities [J]. *Applied Mathematical Modeling* (S0307-904X), 2014, 38(3): 1111-1132.
- [15] Li J Q, Pan Q K, Gao K Z. Pareto-based Discrete Artificial Bee Colony Algorithm for Multi-Objective Flexible Job Shop Scheduling Problems [J]. *The International Journal of Advanced Manufacturing Technology* (S0268-3768), 2011, 55(9-12): 1159-1169.
- [16] Liao G C, Tsao T P. Application Embedded Chaos Search Immune Genetic Algorithm for Short-term Unit Commitment [J]. *Electric Power Systems Research* (S0378-7796), 2004, 71(2): 135-144.
- [17] Kacem I, Hammadi S, Borne P. Approach by Localization and Multi-objective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems [J]. *IEEE Transactions on Systems Man and Cybernetics Part C* (S1094-6977), 2002, 32(1): 1-13.
- [18] Pezzella F, Morganti G, Ciaschettig. A Genetic Algorithm for the Flexible Job-shop Scheduling Problem [J]. *Computers & Operations Research* (S0305-0548), 2008, 35(10): 3202-3212.
- [19] Amiri M, Zandieh M, Yazdani C, *et al.* A Variable Neighborhood Search Algorithm for the Flexible Job-Shop Scheduling Problem [J]. *International Journal of Production Research* (S0020-7543), 2010, 48(19): 5671-5689.
- [20] Kacem I, Hammadi S, Borne P. Pareto- optimization Approach for Flexible Job Shop Scheduling Problems: Hybridization of Evolutionary Algorithm and Fuzzy Logic [J]. *Mathematics and Computers in Simulation* (S0378-4754), 2002, 60(3): 245-276.
- [21] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. *机械工程学报*, 2009, 45(7): 145-151.
- [22] 李修琳, 鲁建厦, 柴国钟, 等. 混合蜂群算法求解柔性作业车间调度问题[J]. *计算机集成制造系统*, 2011, 17(7): 1495-1500.



《系统仿真学报》荣获“2014 中国最具国际影响力学术期刊”证书

由中国学术期刊（光盘版）电子杂志社与清华大学图书馆联合成立的中国学术文献国际评价研究中心，以美国汤森路透 Web of Science 收录的 1.2 万余种期刊为引文统计源，首次研制发布了 2012 《中国学术期刊国际引证年报》（CAJ-IJCR 年报）。第一次给出了我国 5600 余种中外文学学术期刊总被引频次、影响因子、半衰期等各项国际引证指标，并采用了新的国际影响力综合评价指标 CI 对期刊排序，发布了“中国最具国际影响力学术期刊”（排序 TOP5%）和“中国国际影响力优秀学术期刊”（排序 TOP5-10%），在国内外学术界产生了较大反响。之后，2013 年版年报，将引文统计源期刊扩展到 1.44 万多种。目前，2014 版国际、国内年报与 TOP5%和 TOP5-10%期刊的遴选业已完成，《系统仿真学报》被列入“2014 中国最具国际影响力学术期刊”行列。

我学报连续 2 年被列入 TOP5%国内一流的中国最具国际影响力学术期刊，走向世界，进入国际一流，指日可待！