

改进人工蜂群算法求解多目标连续优化问题

葛 宇¹ 梁 静² 王学平³ 谢小川¹

(四川师范大学基础教学学院 成都 610068)¹ (成都工业学院网络中心 成都 610031)²

(四川师范大学数学与软件科学学院 成都 610068)³

摘 要 针对多目标连续优化问题,依据人工蜂群算法原理给出其求解流程,并指出算法中更新策略存在盲目搜索和丢失优秀个体的不足,随后提出改进方案。改进方案包含两部分:首先,设计一种自适应搜索算子,使算法在运行过程中能根据个体质量自动调节搜索范围,让算法搜索行为准确高效;其次,利用外部集合记录下新产生的个体,一次迭代完成后结合外部集合重新构造种群,让算法能有效地保存进化过程中产生的优秀个体。实验中将改进人工蜂群算法与NSGA2算法、改进前算法以及文献报道的同类优秀算法进行了比较,结果说明:改进人工蜂群算法在求解多目标连续优化问题中具有良好的收敛性和均匀性。

关键词 人工蜂群算法,多目标连续优化,更新策略,自适应搜索算子

中图法分类号 TP18 文献标识码 A

Improved Artificial Bee Colony Algorithms for Multi-objective Continuous Optimization Problem

GE Yu¹ LIANG Jing² WANG Xue-ping³ XIE Xiao-chuan¹

(College of Fundamental, Sichuan Normal University, Chengdu 610068, China)¹

(Network Center, Chengdu Technological University, Chengdu 610031, China)²

(College of Mathematics and Soft Science, Sichuan Normal University, Chengdu 610068, China)³

Abstract In order to solve multi-objective continuous optimization problem, this paper gave the solving process according to artificial bee colony algorithm theory, and pointed out that the updating strategy in the algorithm has defect of blind searching and missing good individuals, thus proposed an improved strategy. The improved strategy has two parts. First, a self-adapting searching operator is designed to enable the algorithm to adjust the searching range automatically according to individual quality during the iterative process, leading to a more accurate and efficient algorithm searching process. Second, the newly produced individuals are recorded by external archive, and external archive is combined to reconstruct the colony after a iteration, which can save good individuals in the iterative process. The experiment compares improved artificial bee colony algorithm with NSGA2 algorithm, artificial bee colony algorithm and superior algorithm alike in papers. The comparison result indicates the improved artificial bee colony algorithm has good convergence and uniformity in solving multi-objective continuous optimization problem.

Keywords Artificial bee colony algorithm, Multi-objective continuous optimization problem, Updating strategy, Self-adapting searching operator

1 引言

工程应用与决策中的大量问题可归结为多目标连续优化问题,这类问题不存在唯一的全局最优解,而是存在一个理想最优解集合^[1]。近年来不少研究成果表明:智能优化算法是解决多目标连续优化问题的有效手段^[1],目前已有不少学者对一些典型的智能优化算法进行改进,以提高它们在求解连续多目标优化问题中的性能,如:文献[2]改进了差分进化算法的选择策略,同时增加了反向学习机制以对初始种群中个

体执行搜索;文献[3]在多目标遗传算法的搜索过程中加入了混沌细化搜索方法对种群中前10%的个体执行搜索;文献[4]在多目标粒子群算法基础上使用增广Lagrange乘子法对非支配个体作局部搜索,并用混合多样性策略对非支配个体集合进行维护。

人工蜂群算法^[5]是一种新兴的智能优化算法,在多目标优化领域已逐渐受到国内外学者的关注,如:文献[6]改进了人工蜂群算法的邻域搜索策略,并将其应用到多目标服务组合优化问题求解中;文献[7]对人工蜂群算法初始种群以及进

到稿日期:2013-08-15 返修日期:2013-11-30

本文受四川省教育厅项目:人工蜂群算法及其在多目标优化问题中的应用研究(12ZB112)资助。

葛 宇(1981—),男,硕士,讲师,CCF会员,主要研究方向为计算智能,E-mail:geyufly@yahoo.com.cn;梁 静(1979—),女,硕士,讲师,CCF会员,主要研究方向为图形图像;王学平(1965—),男,博士,教授,博士生导师,主要研究方向为不确定性的数学理论及算法;谢小川(1975—),男,硕士,实验师,主要研究方向为计算智能。

化策略进行改进,并将其用于求解多目标作业调度组合优化问题;文献[8]提出一种基于向量估计的人工蜂群算法,并将其应用到复合材料设计多目标组合优化问题求解中;文献[9]为人工蜂群算法引入交叉算子和外部档案,并用其求解多目标柔性作业调度组合优化问题。

从以上列举的研究成果可以看出,为了在求解多目标连续优化问题时取得较好效果,文献[2-4]提出的改进智能优化法都在原算法基础上增加了一定的个体搜索步骤,这必然会增加算法的实现复杂度。另外,文献[6,7]也主要是将人工蜂群算法应用到了多目标组合优化问题中。为此,本文将人工蜂群算法应用到多目标连续优化问题中,在不增加算法搜索步骤的基础上提出改进方案,以提高人工蜂群算法的求解性能。以下首先描述了多目标优化问题的相关概念,随后根据人工蜂群算法原理^[5]给出其求解多目标连续优化问题的基本流程,然后分析了算法更新策略的不足,并重新设计了一种基于自适应搜索算子和外部集合的更新策略,最后通过实验结果说明本文提出的改进人工蜂群算法在求解多目标连续优化问题中具有良好的收敛性和均匀性。

2 多目标连续优化问题描述

与组合优化问题不同,多目标连续优化问题的解空间是一连续区域,具体描述如下:

$$\text{Min}(f(X))=[f_1(X), f_2(X), \dots, f_m(X)], X \in \Omega \quad (1)$$

其中, Ω 表示连续解空间; $X=(x_1, x_2, \dots, x_D)$ 为式(1)的候选解(也称个体); f_j 为目标函数, $j=1, 2, \dots, m$; m 表示目标个数; D 表示个体的维数; 集合 $W=\{X_1, X_2, \dots, X_Q\}$ 表示包含 Q 个个体的种群。

参考文献[1],对式(1)描述的最小值多目标连续优化问题有如下定义:

定义 1 X_1 支配 X_2 , 记为 $X_1 > X_2$ 。当且仅当 $f_j(X_1) \leq f_j(X_2)$, $j=1, 2, \dots, m$ 且 $f_j(X_1) < f_j(X_2)$, $\exists j \in \{1, 2, \dots, m\}$ 。

定义 2(非支配个体) 如果个体 X_i 是种群 W 中的非支配个体,当且仅当 $\neg \exists X_j \in W, X_j > X_i$ 。

定义 3(非支配个体集合) 所有非支配个体的集合 $P=\{X_i | \neg \exists X_j > X_i, X_i, X_j \in W\}$ 。

定义 4(pareto 等级) 对于种群 W 中个体 X_i , 支配 X_i 的个体数量记为 NP_i 。 NP_i 值相同的个体视为同一 pareto 等级, pareto 等级按 NP_i 值从小到大表示为 $1, 2, 3, \dots, NP_i$ 值为 0 的个体 pareto 等级记为 1。

定义 5(拥挤距离) 拥挤距离用来估计相同 pareto 等级中一个个体周围其它个体的密集程度。相同 pareto 等级的个体对每个目标函数,按该目标函数值的大小进行排序,然后对每个个体 X_i , 计算由个体 X_{i+1} 和 X_{i-1} 构成的立方体的平均边长,并记为 X_i 的拥挤距离。

定义 6(个体优劣比较规则) 规则由两个方面构成: (1)pareto 等级, pareto 等级越低个体越优秀。(2)拥挤距离, 同一 pareto 等级下, 拥挤距离越大个体越优秀。

多目标优化问题要同时使每个目标达到最优是不可能的, 只能找到一组非支配个体达到或逼近问题的理想最优集合^[1]。

3 求解多目标连续优化问题的人工蜂群算法流程

人工蜂群算法中个体进化的主要动力来自其更新策略。以下针对多目标连续优化问题, 先按文献[5]中描述的人工蜂群算法原理给出算法的更新策略, 再给出求解流程。

3.1 更新策略

令 $W=\{X_1, X_2, \dots, X_Q\}$ 表示包含 Q 个个体的种群, 搜索算子式(2)对个体 X_i 的邻域进行搜索产生一个新个体 X_i' , 然后根据贪婪选择算子式(3)决定是否用 X_i' 更新 X_i , 并按式(4)记录个体未更新次数 $trial_i$ 。

$$x_{id}' = x_{id} + R \times (x_{ik} - x_{kd}) \quad (2)$$

其中, x_{id} 表示个体 X_i 的第 d 维分量; x_{kd} 表示个体 X_k 的第 d 维分量; d, k 均随机选择, 且 $k \neq i$; R 为 $[-1, 1]$ 间的随机数。 x_{id}' 表示新产生的第 d 维分量, 其对应的新个体记为 X_i' 。

$$X_i = \begin{cases} X_i', & X_i' > X_i \\ X_i, & \text{NOT}(X_i' > X_i) \end{cases} \quad (3)$$

$$trial_i = \begin{cases} 0, & X_i \text{ 被更新} \\ trial_i + 1, & X_i \text{ 未被更新} \end{cases} \quad (4)$$

值得说明的是, 按照人工蜂群算法原理^[5], 算法搜索到的新个体 X_i' 要能比原个体 X_i 更接近理论最优时才执行 $X_i = X_i'$ 。对于单目标最小值优化问题只需要 $f(X_i') < f(X_i)$ 就能表示 X_i' 比 X_i 更接近理论最优。而对于式(1)描述的连续多目标问题, 结合定义 1 可以看出: 只有当 $X_i' > X_i$ 时才能说明个体 X_i' 比 X_i 更接近理想最优集合。因此, 式(3)中个体的更新条件设置为 $X_i' > X_i$ 。

3.2 人工蜂群算法求解多目标连续优化问题流程

第一步 构造初始种群 W 。

第二步 算法对种群 W 中的每个个体执行更新策略(即式(2)一式(4))。

第三步 算法选择种群 W 中优秀个体执行更新策略。具体地: 算法对种群中的个体按定义 6 规则进行个体优劣排序, 然后采用轮盘赌方式执行 Q 次选择, 每次选出一个较优秀的个体执行式(2)一式(4)。

第四步 个体淘汰。算法淘汰多次未被更新的个体: 根据条件 $\text{MAX}(trial_i) > \text{limit}$ 选择个体 X_i , 随机产生新个体代替 X_i (limit 为指定的阈值)。

第五步 转回第二步, 直到算法达到指定的最大迭代次数后输出非支配个体集合。

4 算法分析及改进

4.1 算法更新策略分析

(1)搜索算子存在盲目性。

为产生新个体, 式(2)对个体 X_i 的第 j 维分量 x_{id} 执行邻域搜索得到 x_{id}' 。式(2)的搜索范围是以 x_{id} 为原点、 $R \times (x_{ik} - x_{kd})$ 为步长的连续区域, 而 R, k, j 均为随机数, 说明 X_k 可以是种群中的任意个体, 其一维分量 x_{kd} 可以对应解空间 Ω 中的任意位置, 因此, 算法每次迭代通过式(2)产生新个体都要对整个解空间 Ω 进行无规律搜索。文献[10]指出, 无规律的随机搜索算子会使算法的进化过程存在盲目性, 可能导致算法出现退化。

(2)贪婪选择算子可能导致优秀个体丢失。

贪婪选择算子(式(3))依据 X_i' 能否支配 X_i 来决定是否

保留 X_i' , 这样的机制不能准确反映个体 X' 的优劣程度, 如图 1 所示。

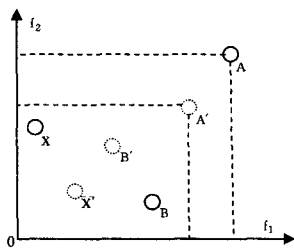


图 1 以两目标优化问题为例的贪婪选择算子分析

图 1 中 X' 、 A' 和 B' 分别为个体 X 、 A 和 B 通过式(2)产生的新个体, 结合图 1 和定义 1 可知: $A' > A$, 而 B' 与 B 、 X' 与 X 互不支配。按照贪婪选择算子(式(3))的规则, 个体 X 、 B 和 A' 会被保留。而图 1 中个体 B' 和 X' 的 pareto 等级均小于 A' , 据定义 6 可知: 个体 X' 和 B' 都比 A' 优秀, 丢弃 X' 和 B' 显然未能准确反映个体在种群中的优劣程度, 会导致优秀个体被丢弃, 种群中优秀个体的数量减少。

4.2 改进后的更新策略

针对前文分析, 本文重新设计个体更新策略为: 自适应搜索算子产生新个体、外部集合记录新个体和重新构造种群 3 个部分, 具体描述如下。

4.2.1 自适应搜索算子

个体的适应度值反映了个体的质量, 为使算法搜索时能根据个体在每个目标上的适应度值(即个体质量)自适应调整搜索范围, 避免盲目搜索, 本文定义个体适应度调节项 T , 如式(5), 并设计了自适应搜索算子, 如式(6):

$$T = \sum_{j=1}^m \frac{f_j(X_i) - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \quad (5)$$

式中, m 表示目标个数, $f_j^{\min}$ 和 $f_j^{\max}$ 分别表示当前种群中个体对应的第 j 个目标的最优、最差适应度值; $f_j(X_i)$ 表示个体 X_i 对应第 j 个目标的适应度值。

$$x_{id}' = x_{id} + R \times T \times \left(\sum_{j=1}^m (\omega_j x_d^j) - x_{id} \right) \quad (6)$$

式中, R 为 $[-1, 1]$ 内的随机数; $\sum_{j=1}^m (\omega_j x_d^j)$ 用于引导搜索方向, ω_j 为随机权重 ($\sum_{j=1}^m \omega_j = 1$), j 表示多目标问题的第 j 个目标, 进化过程中每次迭代都需要随机产生一次 ω_j ; x_d^j 表示边界个体 X^j 的第 d 维分量; 边界个体 X^j 如图 2 所示, 表示当前迭代的种群中第 j 个目标适应度值最优的个体, 即对于第 j 个目标, $\exists X^j \in W, f_j(X^j) = \text{Min}(f_j(X_i)), i=1, 2, \dots, Q$ 。

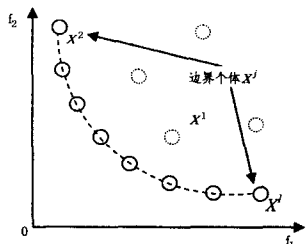


图 2 以两目标优化问题为例的种群中边界个体示意图

4.2.2 外部集合记录新个体

式(6)通过更新个体 X_i 的一维分量产生新个体 X_i' , 为了能有效保留优秀个体, 本文用外部集合 W' 记录通过式(6)产生的新个体 X_i' , 如式(7):

$$W' = X_i' \cup W' \quad (7)$$

4.2.3 重新构造种群

在算法一次迭代中所有个体更新操作(式(6))执行完后, 算法按定义 4 对 $W \cup W'$ 中个体进行 pareto 分级, 根据定义 5 对相同 pareto 等级的个体进行拥挤距离排序。然后按照 pareto 等级从小到大(如遇到相同 pareto 等级的个体则按拥挤距离从大到小取)的规则取出 Q 个个体更新种群 W 。

重新构造种群 W 后, 本文采用式(8)计算个体未更新次数:

$$\text{trial}_i = \begin{cases} 0, & X_i \in W \text{ and } X_i \in W' \\ \text{trial}_i + 1, & X_i \in W \text{ and } X_i \notin W' \end{cases} \quad (8)$$

4.3 改进算法流程

改进后的人工蜂群算法流程如下:

步骤 1 算法初始化。随机产生 Q 个个体构成种群 W , 每个个体 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ 是一个 D 维向量; 指定算法结束条件(最大迭代次数)。

步骤 2 对 W 中每一个体执行更新并记录新个体。对于个体 X_i 随机选择其一维分量 x_{ij} 利用式(6)进行搜索, 产生新个体 X_i' , 并将 X_i' 由式(7)记录进外部集合 W' 。

步骤 3 选择 W 中较优秀的个体执行更新并记录新个体。采用轮盘赌方式执行 Q 次选择, 由定义 6 的规则每次选出一个 pareto 等级较低、拥挤距离较大的优秀个体执行式(6), 产生的新个体由式(7)记录进外部集合 W' 。

步骤 4 按 4.2.3 节描述的策略重新构造种群 W , 并按式(8)记录下个体未被更新次数 trial_i 。

步骤 5 淘汰 W 中多次未被更新的个体。选择 trial_i 值最大的且 pareto 等级大于 1 的个体 X_i , 随机产生新个体代替 X_i 。

步骤 6 判断是否达到算法结束条件, 若达到, 输出非支配个体集合(即 $\text{pareto}=1$ 的个体集合), 否则转到步骤 2。

值得说明的是: 为配合改进后的更新策略, 步骤 5 调整了个体淘汰机制, 不再需要与阈值比较, 直接选择 pareto 等级不为 1 的最久未被更新的个体做更新。这样不但减少了算法参数, 而且让算法在选择被淘汰个体时避开了 pareto 等级为 1 的个体, 防止优秀个体被淘汰。

4.4 改进方案讨论

(1)新搜索算子能自适应调节搜索范围, 避免了盲目搜索。

对于最小值优化问题, 种群中适应度值大的个体距离理想最优个体集合较远, 应执行大范围邻域搜索, 可使其加速靠近理想最优。相反, 适应度值小的个体距离理想最优个体集合较近, 需要执行小范围邻域搜索, 这样有利于发掘和保护优秀个体^[11]。式(6)的搜索范围能随个体 X_i 的适应度值 $f_j(X_i)$ (即个体质量)作自适应调整, 分析如下。

为了描述方便, 令 $B = \sum_{j=1}^m (\omega_j x_d^j) - x_{id}$, 式(6)可简化为:

$$x_{id}' = x_{id} + R \times T \times B \quad (9)$$

因式(9)是对个体的一维分量进行操作, 且 R 是 $[-1, 1]$ 的随机数, 所以式(9)的搜索范围是 $[-T \times B, T \times B]$, 如图 3 所示。

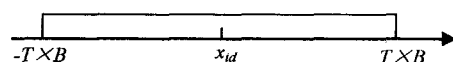


图 3 改进后搜索算子的搜索范围

从图3不难看出,搜索算子对个体 X_i 的搜索范围与 B 、 T 有关。其中 $B = \sum_{j=1}^m (\omega_j x_d^j) - x_{id}$, 据式(6)和改进算法流程步骤2、3的描述可知 $\omega_j x_d^j$ 在一次算法迭代中是不变的,故式(9)在算法一次迭代中针对个体 X_i 的一维分量 x_{id} , B 可看作固定值。以下就 T 作讨论。

设式(5)中 $m=2$, 因本文讨论的是连续函数优化问题,故式(5)表示成如下连续函数:

$$T(f_1(X_i), f_2(X_i)) = \frac{f_1(X_i) - f_1^{\min}}{f_1^{\max} - f_1^{\min}} + \frac{f_2(X_i) - f_2^{\min}}{f_2^{\max} - f_2^{\min}} \quad (10)$$

对式(10)求偏导数得:

$$\begin{cases} \frac{\partial T(f_1(X_i), f_2(X_i))}{\partial f_1(X_i)} = \frac{1}{f_1^{\max} - f_1^{\min}} \\ \frac{\partial T(f_1(X_i), f_2(X_i))}{\partial f_2(X_i)} = \frac{1}{f_2^{\max} - f_2^{\min}} \end{cases} \quad (11)$$

显然式(11)中导数的结果都大于0,故式(10)是关于个体适应度值的单调递增函数,用同样的方法可以说明当 $m>2$ 时上述结论仍然成立。故可得: $f_j(X_i)$ 减小, T 相应变小,图3中的搜索范围随之缩小;相反 $f_j(X_i)$ 增大,图3中的搜索范围会随之变大。即可说明式(6)的搜索范围能随个体 X_i 的适应度值 $f_j(X_i)$ (个体质量)作自适应调整,从而能有效避免盲目搜索。

(2)改进后更新策略能有效保留优秀个体。

改进后策略将新产生的个体记录在集合 W' 中,再把 W' 和当前种群合并($W \cup W'$),保留优秀个体。针对图1描述的情况, $\{X', B', A', X, B, A\} \in W \cup W'$ 。按照4.2.3节中描述的规则重构种群时,改进前被淘汰的个体 X' 和 B' 需要和 $W \cup W'$ 中个体进行优劣比较后才能决定其是否保留,并且根据图1和定义6优劣比较规则可知 X' 和 B' 要优于 A' ,因此它们被保留的优先顺序是: X', B', A' 。从以上分析可以看出,改进后的更新策略能准确反映个体在种群中的优劣程度,有效保留优秀个体。

值得说明的是:该机制引入外部集合 W' 会增加算法运行所需的存储空间,但目前常规的计算机内存配置都在1G以上,故此处增加的存储空间对计算性能的影响可以忽略。

4.5 改进前后算法的计算量分析

与基本人工蜂群算法相比,本文提出的改进方案修改了更新策略,以下对改进前后更新策略的计算量作分析。

设一次加法(或减法)的计算量为 C_1 ,一次乘法(或除法)的计算量为 C_2 ,求解问题的目标个数为 m 。改进前后个体更新策略的计算量分析如下:

(1)改进前个体更新策略执行式(2)时要做2次加法、1次乘法;执行式(4)时最坏可能要执行1次加法。而式(2)、式(4)在改进前算法的一次迭代中会被重复执行2次,因此改进前算法一次迭代中更新策略的最大计算量为:

$$O = 6 \times C_1 + 2 \times C_2 \quad (12)$$

(2)改进后个体更新策略执行式(5)时要做 $3m-1$ 次加法、 m 次乘法;执行式(6)时要做 $m+1$ 次加法、 $m+2$ 次乘法;执行式(8)时最坏可能要执行1次加法。根据改进后算法流程,式(5)、式(6)会在步骤2和步骤3被执行,式(8)会在步骤4被执行。因此改进后算法一次迭代中更新策略的最大计算量为:

$$O' = (8m+1) \times C_1 + (4m+4) \times C_2 \quad (13)$$

因此,本文改进方案在算法一次迭代中更新策略增加的计算量为: $(8m-5) \times C_1 + (4m+2) \times C_2$ 。

另外,除以上增加的计算量外,在4.2.3节重构种群的操作也会导致算法计算量有一定增加。增加的计算量对求解效率的影响在下文实验中作相应讨论。

5 数值实验

5.1 实验设计

本文实验选择如下6个具有代表性的多目标连续优化问题进行测试。

(1)SCH

$$\begin{cases} \min f_1(x) = x^2 \\ \min f_2(x) = (x-2)^2, x \in [-10^3, 10^3] \end{cases}$$

(2)FON

$$\begin{cases} \min f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - \frac{1}{\sqrt{3}})^2) \\ \min f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + \frac{1}{\sqrt{3}})^2) \end{cases}, x_i \in [-4, 4], i = 1, 2, 3$$

(3)ZDT1

$$\begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - \sqrt{\frac{x_1}{g}}), x_i \in [0, 1], i = 1, 2, \dots, D \\ g = 1 + 9(\sum_{i=2}^D x_i / (D-1)) \end{cases}$$

(4)ZDT2

$$\begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - (\frac{x_1}{g})^2), x_i \in [0, 1], i = 1, 2, \dots, D \\ g = 1 + 9(\sum_{i=2}^D x_i / (D-1)) \end{cases}$$

(5)ZDT3

$$\begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(1 - \sqrt{\frac{x_1}{g}} - \frac{x_1 \sin(10\pi x_1)}{g}), \\ g = 1 + 9(\sum_{i=2}^D x_i / (D-1)) \end{cases}$$

$$x_i \in [0, 1], i = 1, 2, \dots, D$$

(6)ZDT6

$$\begin{cases} \min f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ \min f_2(x) = g(1 - (\frac{x_1}{g})^2) \\ g = 1 + 9(\sum_{i=2}^D x_i / (D-1))^{0.25} \end{cases},$$

$$x_i \in [0, 1], i = 1, 2, \dots, D$$

实验在CPU2.6G、内存2G的计算机上进行。实验采用文献[12]中的收敛指标 γ 、多样性指标 Δ 来评价算法结果接近理想最优集合的程度和均匀分布程度。以下将本文改进算法记为IMABC,第3节中描述的基本人工蜂群算法记为MABC。实验从如下4个方面进行:

(1)测试算法参数对IMABC性能的影响。

(2)测试问题在不同维数下,对比IMABC、MABC以及经典算法NSGA2的性能。

(3)比较 IMABC 与 MABC 达到规定的指标精度需要的迭代次数、运行时间和求得的非支配个体数量,评估改进算法增加的计算量对求解效率的影响。

(4)IMABC 与文献报道的优秀算法性能对比。

5.2 算法参数对 IMABC 性能的影响

从 4.3 节中改进算法流程看出 IMABC 的参数有进化代数 G 和种群中个体数 Q ,故 IMABC 需要讨论的参数为 G 和 Q 。测试函数中: SCH 维数 $D=1$, FON 维数 $D=3$, ZDT1、ZDT2、ZDT3 维数 $D=30$, ZDT6 维数 $D=10$ 。表 1 列出了 $Q=200$ 时不同进化代数下的实验结果,表 2 列出了 $G=400$ 时不同种群中个体数对应的实验结果。表 1、表 2 中数据为算法独立运行 10 次的平均结果,括号外的数据表示收敛性指标 γ ,括号内的数据表示多样性指标 Δ 。

表 1 不同进化代数(G)下 IMABC 的结果

	$G=400$	$G=600$	$G=800$	$G=1000$
SCH	8.1E-3 (7.3E-1)	7.2E-3 (2.6E-1)	6.2E-3 (3.6E-1)	3.1E-3 (2.3E-1)
FON	5.1E-4 (2.5E-1)	3.1E-4 (3.7E-1)	4.8E-4 (4.5E-1)	5.6E-4 (3.5E-1)
ZDT1	9.2E-4 (1.3E-1)	9.5E-4 (2.6E-1)	9.8E-4 (3.1E-1)	9.1E-4 (1.6E-1)
ZDT2	3.8E-4 (3.7E-1)	7.0E-4 (3.5E-1)	6.6E-4 (2.6E-1)	6.9E-4 (4.2E-1)
ZDT3	2.6E-3 (2.8E-1)	3.6E-3 (2.5E-1)	2.6E-3 (4.3E-1)	1.7E-3 (1.2E-1)
ZDT6	7.2E-4 (4.5E-1)	7.6E-4 (5.3E-1)	8.1E-4 (4.6E-1)	7.8E-4 (3.9E-1)

表 2 种群个体数(Q)变化时 IMABC 的结果

	$Q=100$	$Q=200$	$Q=300$	$Q=400$
SCH	8.2E-3 (4.7E-1)	8.1E-3 (7.3E-1)	3.8E-3 (4.5E-1)	2.6E-3 (3.6E-1)
FON	6.5E-4 (3.9E-1)	5.1E-4 (2.5E-1)	6.7E-4 (3.3E-1)	6.9E-4 (2.3E-1)
ZDT1	9.9E-4 (4.6E-1)	9.2E-4 (3.2E-1)	9.1E-4 (3.7E-1)	8.8E-4 (3.3E-1)
ZDT2	6.9E-4 (4.3E-1)	3.8E-4 (3.7E-1)	7.3E-4 (3.2E-1)	6.7E-4 (2.8E-1)
ZDT3	7.8E-3 (3.6E-1)	2.6E-3 (2.8E-1)	2.1E-3 (2.7E-1)	7.4E-4 (3.3E-1)
ZDT6	7.7E-4 (5.2E-1)	7.2E-4 (4.5E-1)	7.5E-4 (4.3E-1)	7.9E-4 (4.8E-1)

从表 1 可以看出:IMABC 在不同进化代数下得到的两个指标都在同一数量级。从表 2 可以看出:除 ZDT3 问题 $Q=400$ 的结果外,IMABC 在种群个数变化时得到的结果也处于同一数量级。

以上数据说明:进化代数(从 400 代开始)和种群个体数(从 100 个开始)增长不会对 IMABC 的性能造成明显影响。

5.3 测试问题不同维数下 IMABC、MABC、NSGA2 性能对比

实验在测试问题不同维数的情况下,将 IMABC 与 MABC、经典算法 NSGA2 作性能对比。IMABC 的参数设置为:进化代数 $G=400$,种群个体数 $Q=200$;MABC 的 G 、 Q 同 IMABC,阈值 $limit$ 参考文献[13]设置为 $Q \cdot D$;NSGA2 的 G 、 Q 同 IMABC,其余参数设置参考文献[12]。表 3 中数据为算法独立运行 10 次的平均结果,图 4—图 9 为 3 种算法针对相同维数下的测试问题得到的非支配个体集合效果对比。值得说明的是:测试问题 SCH 和 FON 本身限定了维数分别是 1 和 3,故实验对 SCH 和 FON 问题只做了一种维数的对比。

表 3 测试问题不同维数下的算法结果对比

函数	算法	D	γ 均值	γ 方差	Δ 均值	Δ 方差
SCH	NSGA2	1	8.2E-3	6.2E-8	4.6E-1	4.9E-5
	MABC	1	9.2E-3	1.1E-7	9.2E-1	4.2E-4
	IMABC	1	8.1E-3	3.8E-9	7.3E-1	5.7E-6
FON	NSGA2	3	6.8E-3	7.2E-7	3.7E-1	3.4E-5
	MABC	3	3.2E-2	1.9E-6	1.3	3.3E-3
	IMABC	3	5.1E-4	1.4E-8	2.5E-1	3.2E-6
ZDT1	NSGA2	40	2.3E-1	4.8E-4	4.3E-1	3.7E-6
		60	9.9E-1	3.9E-4	7.9E-1	2.6E-5
	MABC	40	2.7E-2	3.1E-5	9.8E-1	3.4E-5
		60	7.7E-2	6.2E-4	1.2	6.2E-4
	IMABC	40	3.8E-3	3.1E-6	3.3E-1	6.4E-6
		60	7.9E-3	6.3E-6	3.4E-1	4.4E-6
ZDT2	NSGA2	40	4.6E-1	3.5E-4	8.5E-1	5.4E-4
		60	6.1E-1	2.4E-3	1.1	7.6E-3
	MABC	40	1.6E-1	2.6E-4	1.3	2.4E-3
		60	3.7E-1	7.1E-5	1.7	8.2E-2
	IMABC	40	8.4E-4	3.1E-6	3.8E-1	4.1E-5
		60	3.2E-3	1.5E-6	4.6E-1	9.2E-6
ZDT3	NSGA2	40	9.1E-2	6.6E-5	6.7E-1	5.7E-4
		60	1.6E-2	8.2E-4	7.1E-1	3.4E-4
	MABC	40	2.2E-2	3.2E-6	1.4	6.2E-4
		60	3.4E-2	4.8E-5	1.6	5.8E-4
	IMABC	40	5.8E-3	2.7E-7	3.3E-1	2.5E-6
		60	7.2E-3	3.4E-8	4.2E-1	6.4E-5
ZDT6	NSGA2	10	2.3E-1	3.1E-3	1.2	3.7E-3
		20	2.8	4.2E-3	1.4	4.3E-4
	MABC	10	5.3E-2	3.6E-5	1.7	8.6E-4
		20	4.3E-1	1.8E-4	2.3	8.2E-4
	IMABC	10	7.2E-4	1.9E-7	4.5E-1	1.5E-6
		20	8.1E-4	7.4E-8	4.6E-1	3.3E-5

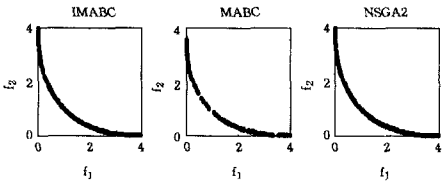


图 4 SCH 函数(1 维)对比结果

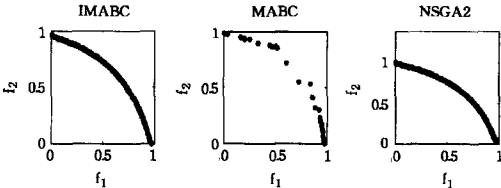


图 5 FON 函数(3 维)对比结果

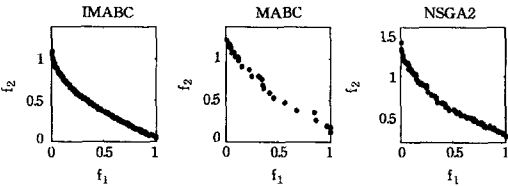


图 6 ZDT1 函数(60 维)对比结果

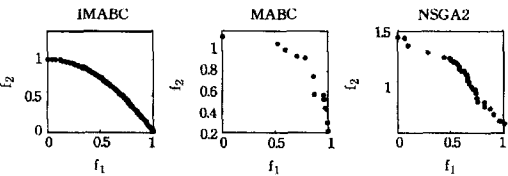


图 7 ZDT2 函数(60 维)对比结果

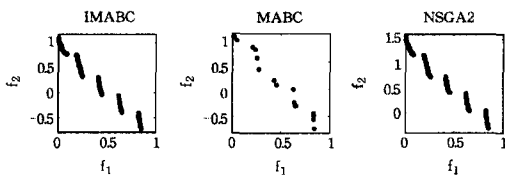


图8 ZDT3 函数(60 维)对比结果

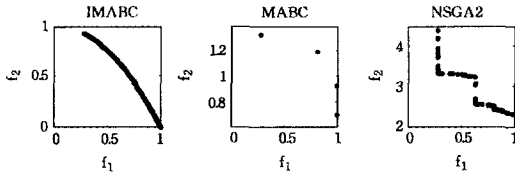


图9 ZDT6 函数(20 维)对比结果

从表3中数据可以看出,除SCH问题外,IMABC的 γ 指标及方差都比另外两种算法领先1个数量级以上,IMABC的 Δ 指标及方差比NSGA2和MABC领先或处于相同数量级。从图5—图9可以发现,IMABC求得的非支配个体集合相比另外两种算法更接近各个目标的理论最优,且分布均匀,尤其是ZDT1、ZDT2、ZDT3和ZDT6问题,IMABC的优势更加明显。

综合比较3种算法,IMABC性能处于领先;并且随问题的维数变大,求解复杂度增加,IMABC的各项指标仍能领先于MABC与NSGA2,并且性能稳定(方差值最小)。

5.4 指定指标精度时对比IMABC与MABC

为了评估IMABC增加的计算量对算法效率的影响,实验选择了5.1节中4个ZDT测试问题,在限定指标精度下对比改进前后算法(IMABC与MABC)的进化代数、运行时间以及求得的非支配个体数。实验指定测试问题的指标精度为 $\gamma=0.09$ 、 $\Delta=0.9$ 。其中,ZDT1、ZDT2、ZDT3取30维,ZDT6取10维,算法设置种群个体数为200,最大进化代数为500,如果500代进化算法都没有达到指定精度,就结束程序。另外,MABC的阈值 $limit$ 参考文献[13]设置为 $Q \times D$ 。表4中数据为算法独立运行10次的平均结果,括号外的数据表示算法IMABC结果,括号内的数据表示算法MABC结果。

表4 指定指标精度时的算法效率对比

	ZDT1	ZDT2	ZDT3	ZDT6
非支配个体数	91 (21)	187 (12)	78 (11)	192 (26)
进化代数	80 (204)	159 (243)	72 (148)	45 (357)
运行时间(s)	0.48 (0.71)	1.04 (1.53)	0.39 (0.46)	0.37 (1.16)

从表4可以看出,IMABC仅需要较少进化代数就能达到规定的指标精度,并且得到的非支配个体数和运行时间都优于MABC。尤其是对ZDT6问题,IMABC不仅进化代数在100代以内,而且得到的非支配个体数达到了192,占种群个体数的95%以上。

从以上数据可以看出:虽然IMABC较改进前增加了计算量,但改进后算法在保证非支配个体均匀分布的同时提高了收敛速度,使IMABC只需较少的进化次数就可以达到指定指标精度,从而缩短了算法运行时间。这足以说明:IMABC增加的计算量(4.5节中分析)不会对算法求解效率

造成明显影响。

5.5 IMABC与文献报道的优秀算法性能对比

为了进一步验证本文改进算法的性能,将IMABC算法在10次实验中求出的平均结果与文献中性能优秀算法结果作对比,如表5所列。其中参与比较的算法有:文献[2]提出的改进多目标差分进化算法、文献[3]提出的改进多目标遗传算法、文献[4,14,15]提出的改进多目标粒子群算法。需要说明的是,IMABC的进化代数为400,种群个体数为200,测试问题设置与被比较文献相同。

表5 IMABC与文献结果对比(括号外数据为 γ ,括号内数据为 Δ)

	IMABC	文献[4]	文献[2]	文献[14]	文献[15]	文献[3]
SCH	8.1E-3 (7.3E-1)	—	5.4E-4 (3.7E-1)	—	1.1E-2 (7.6E-1)	—
FON	5.1E-4 (2.5E-1)	—	1.6E-3 (2.8E-1)	—	—	—
ZDT1	9.2E-4 (3.2E-1)	2.1E-3 (4.1E-1)	4.7E-4 (3.3E-1)	3.2E-3 (2.1E-1)	1.3E-3 (6.8E-1)	1.0E-3 (3.0E-1)
ZDT2	3.8E-4 (3.7E-1)	2.7E-3 (3.8E-1)	5.2E-4 (3.2E-1)	4.6E-3 (2.6E-1)	8.9E-4 (6.3E-1)	6.1E-4 (3.2E-1)
ZDT3	2.6E-3 (2.8E-1)	5.9E-3 (5.6E-1)	6.9E-4 (7.3E-1)	1.1E-3 (2.3E-1)	4.2E-3 (8.3E-1)	4.2E-3 (3.2E-1)
ZDT6	7.2E-4 (4.5E-1)	—	1.4E-3 (3.1E-1)	—	—	—

由表5可以看出:算法的多样性指标 Δ 和文献中的结果处于同一数量级。对于算法的收敛性指标 γ ,IMABC在ZDT3和SCH问题中比文献[2]的结果差1个数量级,但在ZDT6和FON问题中IMABC领先文献[2]1个数量级,其余情况下,IMABC的 γ 指标比文献中结果领先或处于相同数量级。

综合以上数据说明,IMABC有较好的均匀性和收敛性。

结束语 本文将人工蜂群算法应用于多目标连续优化问题求解中。为提高其性能,本文在分析了算法更新策略不足的基础上提出改进方案,具体设计了自适应搜索算子、外部集合和种群重构机制。实验结果表明:在多目标连续优化问题求解中,本文改进算法具有良好的收敛性和分布性,达到了预期效果。

参考文献

- [1] 公茂果,焦李成,杨咚咚. 进化多目标优化算法研究[J]. 软件学报,2009,20(2):271-289
- [2] Ali M, Siarry P, Pant M. An efficient Differential Evolution based algorithm for solving multi-objective optimization problems[J]. European Journal of Operational Research,2012,217:404-416
- [3] 王瑞琪,张承慧,李珂. 基于改进混沌优化的多目标遗传算法[J]. 控制与决策,2011,26(9):1391-1397
- [4] 贾树晋,杜斌,岳恒. 基于局部搜索与混合多样性策略的多目标粒子群算法[J]. 控制与决策,2012,27(6):813-818
- [5] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri:Erciyes University,2005
- [6] 周清雷,陈明昭,张兵. 多目标人工蜂群算法在服务组合优化中的应用[J]. 计算机应用研究,2012,29(10):3625-3628

(下转第286页)

用,对于第一幅图像,本文提出的算法适当地提升了匹配对数目,而对于第二幅图像,普通算法和文献[7]的算法都得到大量的特征点,本文的算法通过阈值的二次调节在保持匹配效果的同时适当减少了特征点数目,使运算量降低,运算效率提高。

为了考察本文阈值调整的有效性,在同一场景不同光照的两幅图像(见图5)中选取4个对应位置的特征点,其阈值变化和DoG响应于表2中。由表2可以看出,对于普通图像,本文采用的初始阈值较Lowe提出的固定阈值无太大变化,对匹配效果影响不大,改进算法主要通过二次调整对特征点分布及数目进行调节。另一方面,文献[7]提出的自适应阈值算法仅仅对光照不足图像有较好的效果,对模糊图像改进不大;并且在计算过程中,由于整幅图像采用了同一阈值,簇效应现象没有得到改善。而本文提出的自适应阈值算法针对光照不足和模糊情况都能获得更多的正确匹配对,且特征点分布较文献[7]提出的算法更加均匀。针对这两种特殊图像,本文提出的初始阈值对匹配效果有较明显的影响,可以有效地提升匹配对数目,而后续的二次阈值调整也可以在一定程度上对特征点数目进行改善调节。

表1实验结果可以表明,本文提出的自适应算法针对模糊图像和光照不足图像都能自动地保留更多数目的特征点,后续匹配过程中能够得到更多的正确匹配对;而对于普通图像,其效果与传统SIFT相当,但由于二次阈值调整的影响,对特征点数目有一定的调节作用,簇效应也会得到改善,特征点分布更加均匀。本文自适应阈值SIFT采用二次阈值调整策略,使阈值能够根据图像自身信息与周围特征点分布信息进行调整,证明了本算法的有效性。

结束语 本文分析了传统SIFT算法中固定阈值产生的影响,指出传统SIFT算法在光照不足和模糊情况下会出现效果减弱以及簇效应等缺陷,然后结合图像的对比度信息和特征点分布信息提出了自适应阈值二次调整算法,并且采用结合特征点分布信息的方法对误匹配点进行剔除。最后本文将改进算法应用到图像匹配中,并且与其他算法进行比较,证实了改进算法在光照不足和模糊情况下的有效性,同时对不

同种类的图像都能够自动地调节特征点数目及分布,证明了算法的有效性。今后的研究将侧重于对误匹配剔除算法以及实时性的改进,使算法能够满足实时匹配的需求。

参考文献

- [1] Lowe D. Object recognition from local scale-invariant features [C]//Proceeding of ICCV, Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers Inc., 1999, 2: 1150-1157
- [2] Lowe D. Distinctive image features from scale-invariant key-point [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110
- [3] Bay H, Tuytelaars T, Gool L V. SURF: Speeded up Robust Features [C]//European Conference on Computer Vision, Berlin, Heidelberg: Springer-Verlag, 2006: 404-417
- [4] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors [C]//Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2004: 511-517
- [5] Morel J M, Yu G S. A-SIFT: A new framework for fully affine invariant image comparison [J]. Society for Industrial and applied Mathematics Journal on Image Sciences, 2009, 2 (2): 438-469
- [6] Song R, Szymanski J. Well-distributed SIFT features [J]. Electronics Letters, 2009, 45 (6): 308-310
- [7] Zhai You, Zeng Luan. A SIFT matching algorithm based on adaptive contrast threshold [C]//International Conference on Consumer Electronics, Communications and Networks, Xianing, China, 2011: 1934-1937
- [8] Fischler M, Bolles R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography [J]. Communications of the ACM, 1981, 24 (6): 381-395
- [9] Li Peng, Yan Han-bing, Cui Gang, et al. Image local invariant features matching using global information [C]//International Conference on Information Science and Technology, Hubei, China, 2012: 627-633
- [10] 施展, 陈庆伟. 基于 QPSO 和拥挤距离排序的多目标量子粒子群优化算法 [J]. 控制与决策, 2011, 26 (4): 540-547
- [11] 毕晓君, 王艳娇. 加速收敛的人工蜂群算法 [J]. 系统工程与电子技术, 2011, 33 (12): 2756-2761
- [12] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Transaction on Evolutionary Computation, 2002, 6 (2): 182-197
- [13] Akay B, Karaboga D. Parameter tuning for the artificial bee colony algorithm [C]//International Conference on Computer and Computational Intelligence, 2009: 608-619
- [14] 刘衍民, 赵庆祯, 牛奔. 基于 E 占优的自适应多目标粒子群算法 [J]. 控制与决策, 2011, 21 (1): 90-95
- [15] Yen G G, Leong W F. Dynamic multiple swarms in multiobjective particle swarm optimization [J]. IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans, 2009, 39 (4): 890-911

(上接第259页)

- [7] Wang L, Zhou G, Xu Y, et al. An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling [J]. The International Journal of Advanced Manufacturing Technology, 2012, 60 (9-12): 1111-1123
- [8] Omkar S N, Senthilnath J, Khandelwal R. Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures [J]. Applied Soft Computing, 2011, 11: 489-499
- [9] Junqing L, Quanke P, Kaizhou G. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems [J]. Int J Adv Manuf Technol, 2011, 55: 1159-1169
- [10] 施展, 陈庆伟. 基于 QPSO 和拥挤距离排序的多目标量子粒子群优化算法 [J]. 控制与决策, 2011, 26 (4): 540-547