第 34 卷 第 11 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.11 Computer Engineering June 2008
·博士论文· 文章编号：1000—3428(2008)11—0012—03 文献标识码：A 中图分类号：TP301.6

基于迁徙差分进化算法集成的模体识别

胡桂武
（广东商学院数学与计算科学系，广州 510320）

摘 要：为了克服微分进化的局部收敛问题，通过模拟游牧民族的迁徙机制，提出一种迁徙策略，将其与差分进化算法相结合，得到一种迁徙差分进化算法新范式，利用集成技术，发挥各种差分进化算法的优点，提高算法的全局搜索能力。通过生物序列模体识别实验，验证了该算法的有效性。

关键词：迁徙策略；模体识别；差分进化算法；协同进化

Motif Detection Based on Migration Differential Evolution Ensemble

HU Gui-wu
(Department of Mathematics & Computational Science, Guangdong University of Business Studies, Guangzhou 510320)

【Abstract】 In order to solve the problem of local convergence in differential evolution, this paper proposes migration strategy by simulating nomadic migration, and gets a novel migration differential evolution model by merging migration strategy. The algorithm with ensemble technique sufficiently exerts the advantages of different differential evolution, and its global search capability is enhanced badly. The algorithm is used to deal with biological sequence motif detection, and experiments show that it is effective

【Key words】 migration strategy; motif detection; differential evolution; harmonious evolution

1 概述

早期游牧民族为了寻找新的水源和牧场或者躲避其他部落的侵略，在部落首领的带领下进行群体移动，通过迁徙来克服恶劣的自然环境和适应复杂的社会环境，使经济更富裕、生存能力更强。对优化问题和游牧民族迁徙的比较研究表明，两者存在相似之处，本文提出了一种模拟游牧民族迁徙机制的迁徙策略。

差分进化算法(Differential Evolution, DE)[1]于 1996 年被提出，目前已有许多改进的DE模型[1]，其与EA 类似，易收敛到局部最优解。针对这个情况，本文提出了一种基于迁徙策略的迁徙差分进化算法。为了充分利用不同微分进化算法的优点，本文从另一个角度提出了一种迁徙差分进化算法集成，提高了算法的全局搜索能力。

生物序列模体识别是当今生物信息学面临的一个复杂问题，近年来，有代表性的算法有EM算法、Gibbs采样算法、GA、人工神经网络等[2-4]。但所有的算法都没有达到满意的结果，而使用本文的迁徙差分进化算法集成求解，能取得比较好的结果。

2 基于迁徙策略的差分进化算法集成

2.1 迁徙策略

游牧民族的迁徙本质上是在部落首领的带领下进行的群体移动。表 1 给出了游牧民族的迁徙与迁徙策略之间的对应关系。

表 1 游牧民族的迁徙与迁徙策略的对应关系

游牧民族的迁徙	迁徙策略
部落群体	种群$\{\boldsymbol{X}_1\boldsymbol{X}_2\cdots\boldsymbol{X}_n\}$
部落成员	种群个体$\boldsymbol{X}_k$
部落首领	种群最优个体$\boldsymbol{P}_g$
部落迁徙	种群个体按相同方向移动$\boldsymbol{X}_k=\boldsymbol{X}_k+\boldsymbol{X}^0(k=1,2,\cdots,n)$

定义 1 迁徙策略 已知在D维目标搜索空间中，n个个体组成一个群落，每个个体i包含一个D维的位置向量$\boldsymbol{X}_i$，群体最优位置向量为$\boldsymbol{P}_g$，按一定的机制生成一个目标向量$\boldsymbol{X}^*$。迁徙策略是按式(1)调整自身位置：

$$\boldsymbol{X}_i=\boldsymbol{X}_i+\lambda\frac{(\boldsymbol{X}^*-\boldsymbol{P}_g)}{\left|\boldsymbol{X}^*-\boldsymbol{P}_g\right|} \tag{1}$$

其中，λ是[0, 1]之间的随机数。

2.2 迁徙差分进化算法

DE算法是一种基于群体差异的进化算法，也有类似于GA的变异、交叉和选择等操作[1]。

为了提高DE摆脱局部极值的能力，本文把迁徙策略应用于DE，如果连续一定的代数后，群体中最优个体没有变得更优，即陷入局部最优，则由适应值最大的粒子带领群体向同一个目标$\boldsymbol{X}^*$迁徙，最后用迁徙前最优个体$\boldsymbol{P}_g$替换迁徙后的最差个体。这就是迁徙差分进化算法(Migration Differential Evolution, MDE)，它既能保证群体中个体的多样性，又可避免陷于局部极值，并且可以与任何改进的差分进化算法结合。下面以基本DE为例说明MDE。

Step1 随机生成初始种群。

Step2 变异操作。

Step3 交叉操作。

Step4 选择操作。

Step5 通过以上DE的变异、交叉和选择操作使种群进化到下一代，如果满足收敛条件，则转Step6；如果连续一定的

基金项目：国家自然科学基金资助项目(30230350)；广东省自然科学基金资助项目(06301003)

作者简介：胡桂武(1970－)，男，副教授、博士，主研方向：计算智能，生物信息学

收稿日期：2008-01-30 **E-mail：**pophu998@sohu.com

代数，群体中最优个体没有变得更优，则随机生成一个目标点$\boldsymbol{X}^*$，按定义 1 执行迁徙操作，同时用迁徙前的最优个体$\boldsymbol{P}_g$替换迁徙后的最差个体，转Step2。

Step6 输出全局最优，算法结束。

2.3 迁徙差分进化算法集成

传统差分进化算法没有考虑任何一种改进型差分进化算法自身的优点，且在单一模式下，算法容易陷入局部极值。为了利用不同差分进化算法的优势，本文提出了MDE集成(Migration Differential Evolution Ensemble, MDEE)，基本思想是利用$N(N>1)$个独立的MDE进行并行搜索求解，经过一定的代数后，集中各独立MDE求得的最优解$\boldsymbol{P}_i$ (i=1,2,⋯,N)，从中选出最优的作为 ***WP***，如果***WP***满足要求，则作为解并输出结果，否则，用***WP***取代第N个MDE$_N$中的$\boldsymbol{P}_N$，从第i(i=2,3,⋯,N)个MDE$_i$中选择部分个体取代第i-1 个MDE$_{i-1}$中的部分个体，实现不同DE之间的信息交流，然后开始新一轮的并行搜索。这不仅扩大了种群规模，还充分发挥了各种DE的优势。图 1 给出了N个MDE集成的模型。

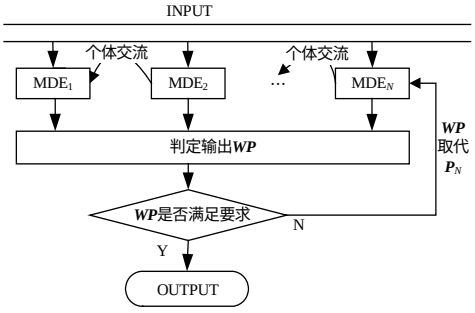

图 1 迁徙 DE 集成模型

3 面向模体识别的迁徙差分进化算法集成

定义 2 模体识别[4] 给定一组字母表A中的序列$x^1,x^2,\cdots,x^m$及整数ω，在每一个x^i中找出长度为ω的子序列，使得这m个子序列间的相似度最大。

3.1 算法中的编码方式、速度、位置

个体的编码方式是用整数编码，对于 DNA, RNA 序列用 0, 1, 2, 3 分别代表相应的碱基，对于蛋白质序列用 0~19 代表相应的 20 个碱基。

定义 3 状态空间中的位置即模体的编码序列，状态空间即搜索空间，记为$\Omega=\{(x_1x_2\cdots x_N)\,|\,x_i\in\{0,1,\cdots,N\}\}$。其中，$N$为模体的长度。对于RNA 或DNA序列，$N$=3；对于蛋白质序列，$N$=19；$X_{\max}$显然是$N$。

定义 4 速度向量 $\boldsymbol{V}$ 的定义与位置一样，区别是在于最大速度的初始值：$V_{\max}=k\cdot X_{\max}$ $(0.1\le k\le 1.0)$。

定义 5 速度与位置的加法 假设 $\boldsymbol{X}$ 为某个位置向量，$\boldsymbol{V}$ 为速度向量，令向量$\boldsymbol{W}=\boldsymbol{X}+\boldsymbol{V}$, $\boldsymbol{X}=(x_1x_2\cdots x_k)$, $\boldsymbol{V}=(v_1v_2\cdots v_k)$, $\boldsymbol{W}=(w_1w_2\cdots w_k)$，其中，

$$w_i=\begin{cases} x_i+v_i & x_i+v_i\le X_{\max} \\ [\tilde{w}_i] & x_i+v_i\ge X_{\max}\end{cases} \tag{2}$$

其中，$0\le\tilde{w}_i\le X_{\max}$, $x_i+v_i=\tilde{w}_i+lX_{\max}$；$[x]$为取整函数；$l$ 为自然数。速度的最小值$V_{\min}$定义为 0。

定义 6 位置与位置的减法 位置 $\boldsymbol{X}$ 与位置 $\boldsymbol{Y}$ 的减法为速度 $\boldsymbol{V}$，令 $\boldsymbol{X}$-$\boldsymbol{Y}$=$\boldsymbol{V}$：

$$v_i=\begin{cases} \bar{v}_i & 0\le x_i-y_i \\ \tilde{v}_i & x_i-y_i\le 0\end{cases} \tag{3}$$

其中，$\bar{v}$ 是 x_i-y_i 除以$V_{\max}$得到的余，$0\le\tilde{v}_i\le V_{\max}$，$\tilde{v}_i=(x_i-y_i)+lV_{\max}$。

定义 7 速度的乘法 $\boldsymbol{V}=(v_1v_2\cdots v_k)$，令$c\boldsymbol{V}=(w_1w_2\cdots w_k)$，其中，

$$w_i=\begin{cases} cv_i & 0\le cv_i\le V_{\max} \\ V_{\max} & cv_i\ge V_{\max}\end{cases} \tag{4}$$

定义 8 速度与速度的加法 假设$\boldsymbol{V}_1$，$\boldsymbol{V}_2$ 为速度，令$\boldsymbol{V}=\boldsymbol{V}_1+\boldsymbol{V}_2$, $\boldsymbol{V}_1=(v_{11}v_{12}\cdots v_{1k})$, $\boldsymbol{V}_2=(v_{21}v_{22}\cdots v_{2k})$, $\boldsymbol{V}=(v_1v_2\cdots v_k)$，其中，

$$v_i=\begin{cases} v_{1i}+v_{2i} & v_{1i}+v_{2i}\le V_{\max} \\ V_{\max} & v_{1i}+v_{2i}\ge V_{\max}\end{cases} \tag{5}$$

3.2 迁徙差分进化算法中的适应值函数

模体识别是给定一组字母表 A 中的序列$S_1,S_2,\cdots,S_m$及整数ω，在每一个S_i中找出长度为ω的子序列，使得这 m 个子序列间的相似度最大，即允许子序列中有部分位置误配，误配个数可以事先设置。

定义 9 模体 $\boldsymbol{P}_n$ 与长度相等的子序列序列 S_m^i 的适应值为

$$Fitness(S_m^i,\boldsymbol{P}_n)=\Big(\sum_{j=1}^{\omega}f(S_m^{ij},P_n^j)+\frac{L}{\omega}\Big)\Big/\omega \tag{6}$$

其中，$f(S_m^{ij},P_n^j)=\begin{cases}1 & \text{if } S_m^{ij}=P_n^j \\ 0 & \text{if } S_m^{ij}\ne P_n^j\end{cases}$；$\omega$是 $\boldsymbol{P}_n$ 的长度；S_m^{ij}是子序列 S_m^i 第 j 个位置上的元素；P_n^j 是模体 $\boldsymbol{P}_n$ 第 j 个位置上的元素；L 等于连续不匹配子段的长度减一再相加。

定义 10 模体 $\boldsymbol{P}_n$ 与序列 S_m 的适应值为 $Fitness(S_m,\boldsymbol{P}_n)$。假设序列 S_m 中与 $\boldsymbol{P}_n$ 长度相等的所有子序列为 $S_n^{\,1},S_n^{\,2},\cdots,S_n^{\,K}$，则

$$Fitness(S_m,\boldsymbol{P}_n)=\max_{l=1}^{K}\{Fitness(S_m^l,\boldsymbol{P}_n)\} \tag{7}$$

定义 11 模体 $\boldsymbol{P}_n$ 与序列组 $S=\{S_1,S_2,\cdots,S_m\}$ 的适应值为

$$Fitness(S,\boldsymbol{P}_n)=\frac{\sum_{l=1}^{m}Fitness(S_l,\boldsymbol{P}_n)}{m} \tag{8}$$

显然，适应值越大，模体 $\boldsymbol{P}_n$ 与真正的模体越接近，$\boldsymbol{P}_n$ 越好。

3.3 面向模体识别的迁徙差分进化算法集成

算法总体框架如图 1 所示，整个算法由 N 个子块组成，每一子块由独立的迁徙 DE 算法执行并行操作，MDEE 详细过程如下：

Step1 输入待求模体的 n 个序列 $x^1,x^2,\cdots,x^n$、模体的长度 ω 及误配数 $Num(error)$。

Step2 输入进行模体识别的N个独立IDE$_i$(i=1,2,⋯,N)子模块。

Step3 对每一个独立的IDE$_i$并行进行如下的操作：

(1)初始化初始种群中的个体，令 $\boldsymbol{P}_i$ 为其中最好的个体。

(2)对于每个个体进行变异操作。

(3)对于每个个体进行交差操作。

(4)对于每个个体进行选择操作。

(5)计算每个个体的适应值，用最优个体替换 $\boldsymbol{P}_i$，判断算法收敛准则是否满足：满足则执行Step5；如果连续一定的代数后，群体中 $\boldsymbol{P}_i$ 没有变得更优，则随机生成一个目标点$\boldsymbol{X}^*$，按定义 1 执行迁徙操作，同时用迁徙前的最优个体$\boldsymbol{P}_i$替换迁徙后的最差个体。如果迁徙次数超过预定的值，转Step4；否则，转(2)。

Step4 输出子模块IDE$_i$中的最优个体 $\boldsymbol{P}_i$ (i=2,3,⋯,N)，集中各独立IDE求得的全局最优解***WP***，用***WP***取代底层第N个

MDE_N中的 $\boldsymbol{P}_N$，并且从第$i(i=2,3,\cdots,N)$个MED_i中选择部分个体取代第i-1 个MED_{i-1}中的部分个体，执行Step3。

Step5 输出最优结果，算法结束。

根据 2.3 节、3.3 节可知，MDEE 是一种集成模型，随迁徙策略和 DE 的不同而不同，并且可以灵活应用于不同的问题中。

4 实验与分析

为了验证MDEE的性能，令模型的底层DE数目N=2，IDE_1中的DE(DE/best/1/exp)为[5]

$$\boldsymbol{v}_{i,G}=\boldsymbol{x}_{best,G}+F\cdot(\boldsymbol{x}_{r2,G}-\boldsymbol{x}_{r3,G})$$

IDE_2中的DE(DE/randtobest/1/exp)为

$$\boldsymbol{v}_{i,G}=\boldsymbol{x}_{i,G}+F\cdot(\boldsymbol{x}_{best,G}-\boldsymbol{x}_{i,G})+F\cdot(\boldsymbol{x}_{r1,G}-\boldsymbol{x}_{r2,G})$$

整个算法用C++编程实现，算法中连续代数定义为 300，迁徙次数定义为 100，实验数据来自于数据库SwissProt中的OTCase 家族，它们是：OTC_AQUAE, OTC_ANASP, OTC2_PSESF, OTC2_ECOLI, OTC2_LACLA, OTC1_PSESH, OTC1_ECOLI，长度都在 300~345 之间。下列选出部分包含Motif：VKFMHCLPAFHDDETTE的数据进行说明：

OTC_AQUAE:VKVMHCLPAKKGQEITEEVFEKNADFIFTQ
OTC_ANASP:AIVLHCLPAHRGEEITEEVIEGSQSRVWQQA
OTC2_PSESF:TIFMHCLPAFHDLDTEVARETPDLVEVEDS
OTC2_ECOLI:VKRLHCLPAFHDDQTTLGKQMAKEFDLHG
OTC2_LACLA:TKFMHCLPASRGEEVVDAVIDGPNSICFDE
OTC1_PSESH:DVLFMHCLPAHRGEEISVDLLDDSRSVAWD
OTC1_ECOLI:VKFLHCLPAFHDDQTTLGKKMAEEFGLHG

总共进行了 2 次实验：(1)模体长度设置为 8，允许误配数目是 2；(2)模体长度设置为 11，允许误配数目是 3。为了评价算法的有效性，表 2 列出了 MDEE, MDE, DE 的实验结果。可以看出，MDEE 的计算精确度明显好于 MDE 和 DE。

表 2 MDEE, MDE, DE 的实验比较

Algorithm	模体长度	Detected Motif	模体适应值
MDEE	8	KMHCLPAF	0.810
MDE_1	8	FLHCLPAS	0.767
DE_1	8	FHCLPAFK	0.714
MDEE	11	KFMHCLPAFHD	0.711
MDE_2	11	LFLHCLPASFH	0.701
DE_2	11	IFFHCLPAKHG	0.607

5 结束语

本文提出了一种迁徙差分进化算法的集成模型，实现了不同差分进化算法之间的交流，发挥了不同模型的优点，提高了算法的全局搜索能力。用该算法进行生物序列模体识别，取得了比较理想的效果，这也是差分进化算法在计算生物学中的一种新的应用尝试。

参考文献

[1] Storn R, Price K. Minimizing the Real Functions of Contest by Differential Evolution[C]//Proc. of IEEE Conference on Evolutionary Computation. [S. l.]: IEEE Press, 1996.

[2] Lo N W, Chang Changchen. Human Promoter Prediction Based on Sorted Consensus Sequence Patterns by Genetic Algorithms[C]//Proceedings of the International Congress on Biological and Medical Engineering. Singapore: [s. n.], 2002.

[3] Lawrence C E, Altschul S F, Boguski M S, et al. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment[J]. Science, 1993, 262(5131): 208-214.

[4] Jonathan M. A Simulated Annealing Algorithm for Finding Consensus Sequences[J]. Journal of Bioinformatics, 2002, 18(11): 1494-1499.

[5] Kaelo P, Ali M M. A Numerical Comparison of Some Modified Differential Evolution Algorithms[J]. European Journal of Operations Research, 2006, 169(3): 1176-1184.

~~~~~~

（上接第 11 页）

门限值时，如 2(=0.5r^2)，称该节点可以被较好的定位。在不使用非凸约束的情况下，如果使用 4 个(或 6 个)多边形顶点，68%(或 79%)的节点可以被较好地定位。然而，使用 6 个多边形顶点，且同时使用凸约束和非凸约束的情况下，90%的节点可以被较好地定位。此外，在使用非凸约束之后，网络中节点可行地理区域面积的最大值减小了 30%。因此，使用非凸约束能较大幅度地提高传感器节点定位精度。

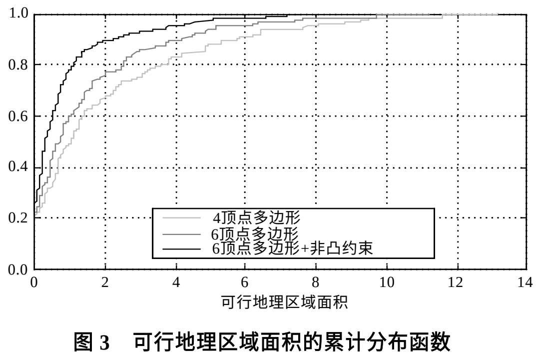

图 3 可行地理区域面积的累计分布函数

5 结束语

本文提出了一种传感器节点的定位方法，能将传感器节点的真实位置限定于可行地理区域内。该算法仅需网络节点的邻接信息，使用外接多边形近似传感器节点不规则的可行地理区域。仿真结果表明，增加多边形的定点数目以及利用网络中的非凸约束信息，都能提高节点的定位精度。

参考文献

[1] Chen Weipeng, Jennifer C H. Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks[C]//Proc. of IEEE International Conference on Network Protocols. Atlanta, Georgia, USA: IEEE Press, 2003: 284-294.

[2] Niculescu D. Ad Hoc Positioning System[C]//Proc. of IEEE Global Telecommunications Conf.. San Antonio, TX, USA: IEEE Press, 2001.

[3] Niculescu D, Nath B. Ad Hoc Positioning System Using AoA[C]//Proc. of INFOCOM’03. San Francisco, USA: [s. n.], 2003.
~~~~~~