

动态环境的人工免疫网络多Agent优化策略

史旭华^{1,2}, 钱 锋¹

(1. 华东理工大学 化学工程联合国家重点实验室, 上海 200237; 2. 宁波大学 电气自动化研究所, 浙江 宁波 315211)

摘要: 基于生物免疫网络的核心思想及多Agent技术, 提出了动态环境下的人工免疫网络多Agent优化策略(Dmaopt-aiNet). 该策略以搜索动态环境中的全局最优解为目标, 引入了邻域克隆选择、邻域竞争和协作操作, 并对Agent自信度状态作自动调整, 在优化策略中采用了双重Agent网络结构、双重变异及动态环境检测策略. 理论分析了Dmaopt-aiNet算法具有全局收敛性, 实验结果表明该算法对高维动态优化问题具有较突出的优越性, 能准确定位动态环境下的最优解, 具有较好的搜索效果和效率.

关键词: 免疫网络; 多Agent; 动态环境; 优化

中图分类号: TP273 **文献标识码:** A

Artificial immune network multi-agent optimization strategy for dynamic environment

SHI Xu-hua^{1,2}, QIAN Feng¹

(1. State-Key Laboratory of Chemical Engineering, East China University of Science and Technology, Shanghai 200237, China;

2. Research Institute of Electric Automatic Control, Ningbo University, Ningbo Zhejiang 315211, China)

Abstract: Based on the idea of biological immune network and multi-agent technology, an artificial immune network multi-agent optimization strategy for dynamic environment(Dmaopt-aiNet) is proposed. The strategy with the target of global optimization introduces neighborhood clonal selection, neighborhood competition and neighborhood collaborative operators. Simultaneously, self-confidence of each agent can be automatically adjusted. In the optimizing process, some strategies such as double-agent network structure, double-mutation strategy and dynamic environmental monitoring are involved. Theoretical analysis shows that Dmaopt-aiNet algorithm is global convergence. Experimental results and comparison illustrate that Dmaopt-aiNet in dealing with high-dimensional dynamic optimization problems is more superior and can accurately determines the location of the optimum with good effectiveness and efficiency.

Key words: immune network; multi-agent; dynamic environment; optimization

1 引言(Introduction)

动态环境优化(dynamic environment optimization, DEO)指目标函数、约束及定义域中至少一种随时间变化, 如工程领域中化工装置操作优化、机器人路径规划、车辆动态调度等, 这类问题的寻优方案常随时间的变化需作相应的调整, 由此最优化技术面临新的挑战. 尽管已有大量的智能优化技术用于静态优化, 但它们解决DEO却极为困难^[1]. 为此, 基于遗传的进化优化逐渐从确定环境转向不确定环境^[2], 此方面研究可追溯到20世纪60年代, 直到80年代才成为众多学者的研究热点^[3]. 在策略上基于遗传的动态环境优化大体分为以下4类: 1) 环境变化后修改进化算子^[3,4]; 2) 始终保持种群的多样性^[5,6]; 3) 引入记忆策略, 重用以前的信息^[7,8]; 4) 采用多种群搜索

策略^[9,10]. 然而从本质上, 此类进化算法较难把握局部与全局, 微观与宏观搜索的平衡^[11].

与遗传进化相比, 人工免疫更适合于处理复杂动态环境优化^[11]. 遗传进化能使种群逐渐收敛并获得满意解, 但搜索中会使种群失去多样性, 人工免疫由于具有多样性维持能力在动态优化方面取得了成果. Gasper等最先尝试将简单人工免疫系统(SAIS)用于动态优化^[12], 算法采用两种自适应变化的抗体激励度评价方案, 用于模式跟踪, 结果表明SAIS在鲁棒性与反应性方面超过了遗传算法; Walker等^[13]比较了克隆选择和进化策略在处理动态优化时参数的敏感性及各种变化环境下算法的有效性; 文献[14]基于免疫生发中心的进化与学习机制, 获得一种解决非约束动态优化算法, 并用5维移动峰问题测试其

收稿日期: 2010-01-04; 收修改稿日期: 2010-10-04.

基金项目: 国家杰出青年科学基金资助项目(60625302); 国家“973”计划资助项目(2009CB320603); 国家科技支撑计划资助项目(2007BAF22B05); 国家自然科学基金资助项目(20876044); 宁波市自然科学基金资助项目(2011A610173); 浙江省自然科学基金资助项目(Y1090548).

性能; de FranCa等^[15]在Opt-aiNet免疫网络的基础上增加了记忆种群并改进了变异方式,提出了Dopt-aiNet算法,在多种动态环境下寻找优化解。然而,由于免疫机理复杂,免疫现象的描述还较困难,不论在模型建立、算法设计等方面都存在不完善之处^[11]。随着对复杂分布式问题求解需求的增长,Agent及多Agent系统被认为是求解复杂分布式问题的高层指导方法^[11]。将多Agent与免疫网络结合,可以利用多Agent的现有研究成果,为新颖的人工免疫网络构建提供有益的思路。

2 人工免疫网络多Agent动态优化策略 (Strategies of artificial immune network multi-agent dynamic optimization)

2.1 多Agent动态优化系统(Multi-agent dynamic optimization system)

DEO数学模型可描述为

$$\min_{\mathbf{x} \in \mathcal{S}(t) \subset \mathbb{R}^{q(t)}} F(\mathbf{x}, t), \quad (1)$$

其中: $\mathcal{S}(t) = \prod_{i=1}^{q(t)} \langle \alpha_{i-}(t), \alpha_{i+}(t) \rangle$ 为定义域, $\mathbf{x} = (x_1, x_2, \dots, x_{q(t)}) \in \mathcal{S}(t)$, $q(t)$ 为环境变量的可变维数值, $F(\mathbf{x}, t)$ 为目标函数。对给定时间 t , 对任意的 $\mathbf{x} \in \mathcal{S}(t)$, 均有 $F(\mathbf{x}^*, t) \leq F(\mathbf{x}, t)$, 则称 $\mathbf{x}^* \in \mathcal{S}(t)$ 为问题的最优解。

Agent能感知环境并反作用于环境,所有由多个Agent构成的系统可表达为多Agent系统。根据Russel对Agent系统特性的描述^[16],需要定义:

- 1) Agent的环境,包括整体环境、局部环境;
- 2) Agent的目标,即竞争力;
- 3) Agent的状态,即自信度。

本文中,一个Agent表示一个可能解,即一个实值向量 $\mathbf{x} = (x_1, x_2, \dots, x_{q(t)})$ 。Agent可被置于矩阵网格 L 上^[17],其规模为 $\text{size}(L) = L_s \times L_s$,每个Agent均赋予一格点并能感知局部环境。

定义1 Agent $A_{m,n}$ 的局部环境定义为

$$\text{Loc. } A_{m,n} = \{A_{m-,n}, A_{m,n-}, A_{m+,n}, A_{m,n+}\},$$

m, n 分别表示Agent所在的行与列,其中:

$$m^- = \begin{cases} m-1, & m \neq 1, \\ L_s, & m = 1, \end{cases} \quad n^- = \begin{cases} n-1, & n \neq 1, \\ L_s, & n = 1, \end{cases}$$

$$m^+ = \begin{cases} m+1, & m \neq L_s, \\ 1, & m = L_s, \end{cases} \quad n^+ = \begin{cases} n+1, & n \neq L_s, \\ 1, & n = L_s. \end{cases}$$

每个Agent占据一网格点,且只和其局部环境的4个Agent发生作用。

定义2 Agent $A_{m,n}$ 代表函数的一个可能解,其竞争力 $C(A_{m,n})$ 定义为

$$C(A_{m,n}) = -F(A_{m,n}), A_{m,n} \in \mathcal{S}(t). \quad (2)$$

在多Agent网格中,每个Agent以一定的概率存在,并具有一定的竞争力,通过邻域克隆选择、竞争与协作来提高自身的竞争力。

定义3 多Agent网格的状态即自信度,用矩阵 $B_{L_s \times L_s}$ 来描述,其元素对应为网格中各个体的自信度,如个体 $A_{m,n}$, 则自信度为 $B(A_{m,n})$, 取值为 $[0,1]$, 1对应最大的自信度, 0反之。高自信度的Agent更有可能成为竞争力强的个体,为了能更好地平衡Agent在动态环境中局部与全局的搜索能力,根据网格中Agent的自信度大小划分开发和搜索Agent。

2.2 免疫网络多Agent操作(Immune network multi-agent operators)

De Castro的Opt-aiNet体现了免疫进化过程,其核心思想为^[18]:

- 1) 确定了抗体的网络化进化环境;
- 2) 通过克隆选择、免疫抑制及随机抗体引入,动态调整进化中的抗体个数,权衡局部与全局的搜索能力;
- 3) 借助免疫记忆细胞保存优良个体,提高进化效率。

将Opt-aiNet的核心思想融入到多Agent动态优化中,提出了免疫网络多Agent动态优化算法Dmaopt-aiNet。竞争和协作作为多Agent进化的重要操作算子,已被普遍采用^[17~19],为此,Dmaopt-aiNet采用了网格环境下的邻域克隆选择、竞争和协作来提高自身的竞争力,同时各Agent的自信度通过自激励及邻域学习操作自动调整。设Dmaopt-aiNet算子作用在个体 $A_{m,n} = (a_1, a_2, \dots, a_{q(t)})$ 上, $A_{\max,mn}^C, A_{\max,mn}^B$ 分别为 $A_{m,n}$ 邻域内竞争力和自信度最大的个体。

2.2.1 搜索算子(Search operators)

邻域克隆选择:

从免疫应用来说,邻域下的克隆选择进而扩充到整个网格使操作更接近生物的自然进化。选择 $A_{m,n}$ 及其邻域Loc. $A_{m,n}$ 中自信度或竞争力最大的个体,分别进行克隆,对克隆后的个体按变异率 p_m 进行变异,选择变异后竞争力或自信度最大的替代原个体,并生存在网格上,直到作用于所有网格。邻域克隆选择算子可表达为

$$A_{\max,mn}^{C,B} \leftarrow T_s \circ T_m \circ T_c(A_{\max,mn}^{C,B}), \quad (3)$$

$$A_{\max,mn}^{C,B} = A_{\max,mn}^C \cup A_{\max,mn}^B. \quad (4)$$

这里: T_c 为克隆操作,

$$A_{\max,mn}'^{C,B} = \mathbf{I} \times A_{\max,mn}^{C,B},$$

$\mathbf{I}$ 为 w 维行向量; T_m 为变异操作:

$$A_{\max,mn}''^{C,B} = m_{\Theta m}(A_{\max,mn}'^{C,B}),$$

$m_{\Theta m}$ 为变异运算; T_s 为选择操作:

$$A_{\max, mn}^{C,B} = \max(A_{\max, mn}^{C,B}, A_{\max, mn}^{C,B}).$$

通过邻域克隆选择能提高个体竞争力和自信度。

邻域竞争:

邻域竞争代替了Opt-aiNet的免疫抑制及随机抗体引入, 可以在维持个体多样性的基础上加快进化速度. 网格经过邻域克隆选择后, 若新个体 $A_{m,n}$ 不低于 $A_{\max, mn}^C$ 的竞争力或 $A_{\max, mn}^B$ 的自信度, 则可继续存活在网格上; 否则将死亡, 空出的格点将被新个体 $\beta_{m,n} = (b_1, b_2, \dots, b_{q(t)})$ 占据. 邻域竞争算子可表述为:

如果 $C(A_{m,n}) \geq A_{\max, mn}^C$ 或 $B(A_{m,n}) \geq A_{\max, mn}^B$, 则 $\text{Loc}(\cdot) \leftarrow \text{Loc}.A_{m,n}$; 否则, $\text{Loc}(\cdot) \leftarrow \beta_{m,n}$.

$$b_k = \begin{cases} \alpha_{k-}(t), & \text{如果 } a_k + r(-1, 1) \times (a_k - \alpha_{k-}(t)) < \alpha_{k-}(t), \\ \alpha_{k-}^-(t), & \text{如果 } a_k + r(-1, 1) \times (a_k - \alpha_{k-}(t)) > \alpha_{k-}^-(t), \\ a_k + r(-1, 1) \times (a_k - \alpha_{k-}(t)), & \text{其他,} \\ k = 1, 2, \dots, q(t), \end{cases} \quad (5)$$

$r(-1, 1)$ 为均匀分布的随机数.

邻域协作:

邻域中的Agent, 通过交换自信度、竞争力信息, 促进协作, 权衡动态环境下的局部与全局搜索能力. 邻域协作算子为:

1) 在邻域空间中, 选择自信度最高、最小及竞争力最强的个体, 进行以下操作:

$$\hat{A}_{m,n}^T = A_{\max, mn}^C + 0.5(A_{\max, mn}^B - A_{\min, mn}^B). \quad (6)$$

这里 $A_{\max, mn}^C$, $A_{\max, mn}^B$, $A_{\min, mn}^B$ 分别为当代邻域中竞争力最强和自信度最高、最小个体.

2) 为了增大邻域个体间的离散程度, 对新个体 $\hat{A}_{m,n}^T = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_{q(t)})$ 和当前个体 $A_{m,n}^T = (a_1, a_2, \dots, a_{q(t)})$, 进行两项分布交叉, 生成新个体 $\bar{A}_{m,n}^T = (\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{q(t)})$, 即

$$\bar{a}_k = \begin{cases} a_k, & r(0, 1) > c_r, \\ \hat{a}_k, & \text{其他,} \end{cases} \quad k = 1, 2, \dots, q(t), \quad (7)$$

其中: $c_r \in [0, 1]$ 为杂交参数, $r(0, 1)$ 是 $[0, 1]$ 间的随机数.

3) 将 $\bar{A}_{m,n}^T$ 与 $A_{m,n}^T$ 进行竞争选择, 保留优良个体到下一代, 其操作为

$$A_{m,n}^{T+1} = \begin{cases} \bar{A}_{m,n}^T, & \text{如果 } C(\bar{A}_{m,n}^T) \geq C(A_{m,n}^T), \\ A_{m,n}^T, & \text{其他.} \end{cases} \quad (8)$$

2.2.2 自信度调整操作(Confidence adjustment operation)

个体的自信度反映各个体的进化水平, 高自信度

个体经邻域克隆选择更有机会增加自身的竞争力, 可有效防止种群退化. 网格中的各Agent自信度通过自激励与邻域学习自动调整.

自激励操作:

如果子代的竞争力强于父代, 则子代的自信度作正反馈调整, 否则作负反馈调整, 表示为

$$\begin{cases} B(A_{m,n}) = \frac{B(A_{m,n}) + \gamma}{1 + \gamma}, & \text{如果 } C(A_{m,n}^T) \geq C(A_{m,n}^{T-1}), \\ B(A_{m,n}) = B(A_{m,n})(1 - \gamma), & \text{其他.} \end{cases} \quad (9)$$

这里: γ 为激励系数, $C(A_{m,n}^{T-1})$, $C(A_{m,n}^T)$ 为子代、父代个体竞争力.

邻域自信度学习:

本文规定: 自信度大的个体用于进一步开采搜索区域内更好的解, 称搜索Agent; 而自信度小的个体用作环境开发操作, 称开发Agent. 设 $A_{\max, mn}^C$ 为邻域中竞争力最强的个体, 其前一代竞争力为 $C(A_{\max, mn}^{C,T-1})$, 邻域自信度学习算子为

$$\begin{cases} B(A_{m,n}^T) = \min(1, B(A_{m,n}^T) + \exp(-B(A_{m,n}^T))/(T+1)^2), & \text{如果 } C(A_{m,n}^T) \geq C(A_{\max, mn}^{C,T-1}); \\ B(A_{m,n}^T) = \max(0, B(A_{m,n}^T) - \exp(-B(A_{m,n}^T))/(T+1)^2), & \text{其他.} \end{cases} \quad (10)$$

2.2.3 动态优化策略(Dynamic optimization strategy)

为了能快速跟踪环境的变化, 准确定位最优解位置, 本文采用了以下的动态优化策略: 双重Agent网络结构、双重变异及动态环境检测策略.

1) 双重Agent网络结构.

双重Agent网络结构能同时运行两类Agent, 即搜索Agent($B(A_{m,n}^T) \geq \zeta$)和开发Agent($B(A_{m,n}^T) < \zeta$), 免疫记忆库可以记录一定时期内算法获得的最好解. 当环境发生了变化, 此时的Agent需借助记忆库进化, 搜索Agent负责保证算法的探索能力, 每经过一段时间将竞争力最强的个体传递给记忆库; 开发Agent能周期性地从记忆库中引入最佳个体, 同时将自信度 $B(A_{m,n}^T) \geq \zeta$ 的个体转换为搜索Agent.

2) 动态环境检测策略.

环境发生变化后, 个体的竞争力也发生了变化. 当某一Agent进化前后代位置接近, 但其对应的竞争

力发生显著变化,基本可以推测环境发生了变化.令记忆库中最佳个体的竞争力前后代与其所对应距离的比值为

$$\delta(T) = \frac{C(M^T) - C(M^{T-1})}{d(T)},$$

这里

$$d(T) = \sqrt{\sum_{k=1}^{q(t)} (M^T - M^{T-1})}.$$

令

$$\Delta\delta(T) = |\delta(T) - \delta(T-1)|,$$

取

$$\begin{aligned} \chi(T) &= \min(\max(1, \Delta\delta(T)), \\ &\quad (\max(\alpha_k^-(t)) - \min(\alpha_{k-}(t))), \\ &\quad k = 1, 2, \dots, q(t)). \end{aligned} \quad (11)$$

这里: T 代表进化代数, $C(M^T)$ 代表记忆库中的最佳个体 M 在 T 时刻对应的竞争力, $\delta(T)$ 代表Agent前后代竞争力变化与距离变化的比值, $\Delta\delta(T)$ 代表 M 的前后两代 $\delta(T)$ 的变化率.一般地,当环境未发生变化时, $\Delta\delta(T)$ 接近于0;相反,当环境发生变化时, $\Delta\delta(T)$ 产生跃变, $\chi(T)$ 为环境变化系数.

3) 双重变异策略.

开发 Agent 与搜索 Agent 的主要区别在于开发 Agent 用于探索整个解空间,即求泛操作,要求个体能均匀分布于整个搜索空间;而搜索 Agent 用于逼近当前的局部最优解,即求精操作,要求个体能相对集中分布于各局部解周围.在动态环境中,为了更好地平衡 Agent 的求泛和求精能力,采用了双重变异策略,搜索 Agent 采用了受控高斯变异和精英学习结合的Q-变异^[20]操作,开发 Agent 采用了适合免疫网络动态环境的 μ -变异操作^[21].

在变异率为 P_m 下的Q-变异操作为^[20]

$$\begin{aligned} A'_{m,n} &= \\ &A_{m,n}^T + r(0,1) \cdot (c_1 \cdot \exp(-\text{norm}(f(A_{m,n}^T)/\eta) + \\ &c_2 \cdot (A_{m,n}^T - A_{\max}^C)), \end{aligned} \quad (12)$$

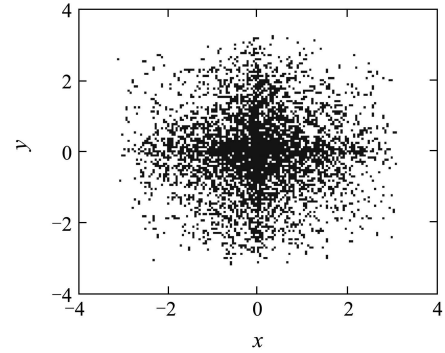
这里: $A_{m,n}^T, A'_{m,n}$ 分别为Q-变异前后的个体, c_1 和 c_2 为学习率,满足约束条件 $c_1 + c_2 = 1$, $\text{norm}(\cdot)$ 为归一化操作函数, η 是控制因子, $r(0,1)$ 为(0,1)内的随机数, $A_{\max}^C$ 是全局内最佳个体.将1000个体从(0,0)出发经Q-变异的子代分布如图1(a)所示.

在变异率为 P_m 下的 μ -变异操作可描述为^[21]

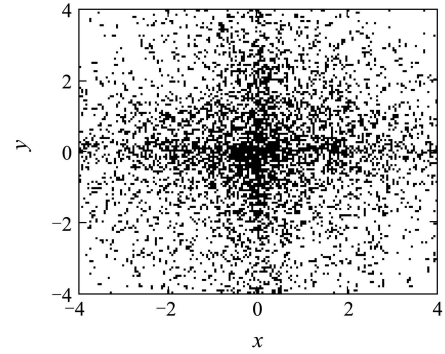
$$a_k' =$$

$$\begin{cases} a_k + \chi(T) \cdot u(0, \alpha_k^-(t) - a_k) \cdot r(0,1), \\ \text{如果 } u(0,1) > 0.5; \\ a_k - \chi(T) \cdot u(0, a_k - \alpha_{k-}(t)) \cdot r(0,1), \\ \text{其他;} \\ k = 1, 2, \dots, q(t). \end{cases} \quad (13)$$

$u(a,b)$ 为 $[a,b]$ 内均匀分布的随机数.将1000个体从(0,0)出发经 μ -变异的子代分布如图1(b)所示.取不同的环境变化系数 $\chi(T)$, μ -变异的子代分布如图1(b)、图2(a)和图2(b)所示,可见,随着环境变化系数的逐渐增大,其子代分布更显均匀化,这就是动态环境变化的要求.

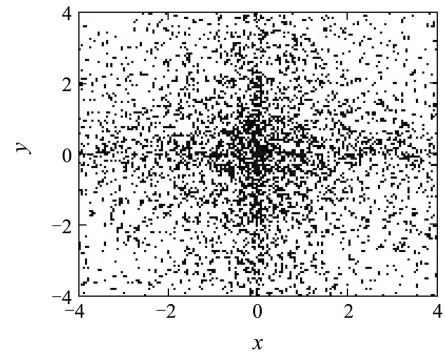


(a) Q-变异的子代分布

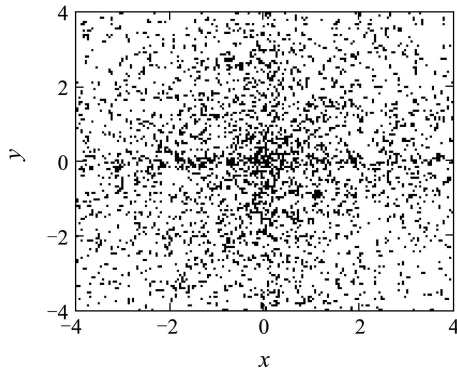


(b) μ -变异的子代分布($\chi(T) = 1$)

图1 两种变异自(0,0)点出发子代分布图
Fig. 1 Spatial distribution of the offspring derived from the point (0,0) of two kinds of mutation



(a) $\chi(T) = 1.5$

(b) $\chi(T) = 2$ 图2 不同环境变化系数 $\chi(T)$ 的子代分布图Fig. 2 Distribution of the offspring of different $\chi(T)$

可见, Dmaopt-aiNet算法充分利用了开发与搜索两种Agent在动态环境中的协调能力. 开发Agent用于探索整个解空间, 通过 μ -变异使个体均匀分布于解空间, 并通过克隆选择、邻域竞争增加自身的竞争力, 当开发Agent的自信度 $B(A_{m,n}^T) \geq \zeta$ 时, 就自动转换为搜索Agent; 搜索Agent用于搜索局域解, 通过Q-变异使个体能够相对集中分布于各局部解周围, 也通过克隆选择、邻域竞争增加自身的竞争力, 当搜索Agent的自信度退化使 $B(A_{m,n}^T) \leq \zeta$ 时, 就自动转化为开发Agent; 搜索与开发Agent共存于同一个免疫网络中, 通过邻域协作算子平衡两种Agent的整体探索与局部搜索能力.

2.3 Dmaopt-aiNet算法(Dmaopt-aiNet algorithm)

Dmaopt-aiNet描述如下:

Step 1 初始化网格系统 $L(0)$, 设置自信度阈值 ζ , 开辟记忆库空间 M , 置 $T = 0$;

Step 2 计算 $L(T)$ 中各个体的竞争力 $C(A_{m,n})$ 与自信度 $B(A_{m,n})$;

Step 3 根据网格 $L(T)$ 中个体的自信度 $B(A_{m,n})$ 和阈值 ζ 划分开发Agent和搜索Agent, 满足 $B(A_{m,n}^T) > \zeta$ 的为搜索Agent, $B(A_{m,n}^T) \leq \zeta$ 的为开发Agent;

Step 4 对网格 $L(T)$ 中的个体进行邻域克隆选择操作, 开发Agent采用 μ -变异, 搜索Agent采用Q-变异, 更新得到网络 $L^1(T)$;

Step 5 对 $L^1(T)$ 中每个个体执行邻域竞争操作, 并通过自信度自激励和邻域学习完成自信度自调整, 更新得到网络 $L^2(T)$;

Step 6 执行邻域协作操作, 更新得到网络 $L^3(T)$;

Step 7 更新记忆库中的个体, 将记忆库中个体的自信度 $B(A_{m,n}^T)$ 设为 ζ , 执行环境监测操作, 并

判断是否满足停止准则, 如满足, 则停止, 否则 $T = T + 1$, 转Step 2;

Step 8 记忆库中最好个体即为所解.

3 Dmaopt-aiNet收敛性分析(Analysis of Dmaopt-aiNet convergence performance)

定义 4 在环境 Ψ_0 下定义全局最优解集为

$$A^* \triangleq \{A \in \mathcal{S}(t) : F(A) = F^* \triangleq \min(F(A') : A' \in \mathcal{S}(t))\}. \quad (14)$$

对于网格上的群体 A , 设 $\psi(A) \equiv |A \cap A^*|$ 表示群体 A 中包含最优解的数目, 称 $\mathcal{S}^*(t) = \{A \in \mathcal{S}(t) | \psi(A) \geq 1\}$ 为最优Agent的种群空间.

定义4表明, 最优Agent的种群空间中至少含有1个最优个体.

定义 5 假如对于任意的初始状态 A_0 , 均有

$$\lim_{T \rightarrow \infty} P\{A(T) \cap A^* \neq \emptyset | A(0) = A_0\} = \lim_{T \rightarrow \infty} P\{A(T) \in \mathcal{S}^*(t) | A(0) = A_0\} = 1, \quad (15)$$

即 $\lim_{T \rightarrow \infty} P\{\psi(A(T)) \geq 1 | A(0) = A_0\} = 1$.

则称算法以概率1收敛到最优种群集.

定理 1 在环境 Ψ_0 下, 免疫网络多Agent算法以概率1收敛到最优种群集.

证 在环境 Ψ_0 下, 设 $P_0(T) = P\{\psi(A(T)) = 0\} = P\{A(T) \cap A^* = \emptyset\}$, 由Bayes条件概率公式有

$$\begin{aligned} P_0(T+1) &= P\{\psi(A(T+1)) = 0\} = \\ &P\{\psi(A(T+1)) = 0 | \psi(A(T)) \neq 0\} \times \\ &P\{\psi(A(T)) \neq 0\} + \\ &P\{\psi(A(T+1)) = 0 | \psi(A(T)) = 0\} \times \\ &P\{\psi(A(T)) = 0\}. \end{aligned} \quad (16)$$

由Dmaopt-aiNet的邻域克隆选择、邻域竞争及协作操作容易得出: 在一次迭代下, 以上3个算子操作后获得的最优解数不少于操作前; 经邻域竞争及协作操作后的最优解数不少于操作前. 因此, $P\{\psi(A(T+1)) = 0 | \psi(A(T)) \neq 0\} = 0$, 则

$$\begin{aligned} P_0(T+1) &= P\{\psi(A(T+1)) = 0\} = \\ &P\{\psi(A(T+1)) = 0 | \psi(A(T)) = 0\} \times P_0(T) = 0, \end{aligned} \quad (17)$$

且

$$\begin{aligned} &P\{\psi(A(T+1)) = 1 | \psi(A(T)) = 0\}_{\min} = \\ &\min\left\{\frac{1}{L_s \cdot L_s} \left[\frac{1}{\lambda - 1} P_m^{d(a(T+1), a_i(T))} (1 - P_m)^{l-d(a(T+1), a_i(T))} \right]^w\right\} > 0. \end{aligned}$$

其中本文中采用实数编码的变异策略, 上式中 $\lambda = 10$, l 为编码长度, $d(\cdot, \cdot)$ 为个体间的欧氏距离.

设

$$\begin{aligned} \vartheta &= \min_T P\{\psi(A(T+1)) = 1 | \psi(A(T)) = 0\}_{\min}, \\ T &= 0, 1, 2, \dots, \\ P\{\psi(A(T+1)) = 1 | \psi(A(T)) = 0\} &\geq \vartheta > 0, \end{aligned} \quad (18)$$

则

$$\begin{aligned} P\{\psi(A(T+1)) = 0 | \psi(A(T)) = 0\} &= \\ 1 - P\{\psi(A(T+1)) \neq 0 | \psi(A(T)) = 0\} &= \\ 1 - P\{\psi(A(T+1)) \geq 1 | \psi(A(T)) = 0\} &\leq \\ 1 - P\{\psi(A(T+1)) = 1 | \psi(A(T)) = 0\} &\leq \\ 1 - \vartheta < 1, \end{aligned}$$

因此

$$\begin{aligned} 0 &\leq P_0(T+1) \leq (1-\vartheta)P_0(T) \leq \\ (1-\vartheta)^2 \times P_0(T-1) \dots &\leq (1-\vartheta)^{T+1} \times P_0(0). \end{aligned}$$

因为 $\lim_{T \rightarrow \infty} (1-\vartheta)^{T+1} = 0$, $1 \geq P_0(0) \geq 0$, 则

$$0 \leq \lim_{T \rightarrow \infty} P_0(T) \leq \lim_{T \rightarrow \infty} (1-\vartheta)^{T+1} P_0(0) = 0, \quad (19)$$

显然

$$\lim_{T \rightarrow \infty} P_0(T) = 0,$$

所以

$$\begin{aligned} \lim_{T \rightarrow \infty} P\{A(T) \cap A^* \neq \phi | A(0) = A_0\} &= \\ 1 - \lim_{T \rightarrow \infty} P_0(T) &= 1, \end{aligned}$$

即 $\lim_{T \rightarrow \infty} P\{A(T) \cap A^* \in S^*(t) | A(0) = A_0\} = 1$.

于是定理 1 得证, 即在环境 Ψ_0 下, 免疫网络多 Agent 算法以概率 1 收敛.

4 仿真实验与分析(Simulation experiment and analysis)

为了验证 Dmaopt-aiNet 的性能, 采用 Angeline 的动态测试试验^[22]:

线性变化函数为 $\Delta \mathbf{x} = \tau \cdot [\mathbf{I}]_{q(t) \times 1}$;

圆变化函数为 $\Delta \mathbf{x} = \tau \sin(2\pi t/25) \cdot [\mathbf{I}]_{q(t) \times 1}$, f 为偶数;

$\Delta \mathbf{x} = \tau \cos(2\pi t/25) \cdot [\mathbf{I}]_{q(t) \times 1}$, f 为奇数;

Gaussian 变化函数为 $\Delta \mathbf{x} = r(0, 1) \cdot [\mathbf{I}]_{q(t) \times 1}$, τ 表示函数的变化量, f 表示函数的变化周期.

选用如下典型的测试函数, 并将本算法结果与 ICS^[23], Opt-aiNet^[24] 和 Dopt-aiNet^[15] 的进行比较:

$$\text{Sphere} = \sum_{i=1}^{q(t)} x_i^2, \quad x \in [-1.28, 1.28], \quad q(t) = 30;$$

$$\text{Rosenbrock} = \sum_{i=1}^{q(t)-1} (100(x_i - x_{i+1})^2 + (x_i - 1)^2),$$

$$x \in [-100, 100], \quad q(t) = 30;$$

$$\text{Griewank} = \frac{1}{4000} \sum_{i=1}^{q(t)} x_i^2 - \prod_{i=1}^{q(t)} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$x \in [-600, 600], \quad q(t) = 30;$$

$$\text{Rastrigin} = \sum_{i=1}^{q(t)} (x_i^2 - 10 \cos(2\pi x_i) + 10),$$

$$x \in [-5.12, 5.12], \quad q(t) = 30.$$

定义动态环境的平均误差 Acc 为: 每次环境变化后最后一代记忆库中竞争力最大个体对应的函数值和相应理论最小值之差的平均. 可表示为

$$\text{Acc} = \frac{f}{Rn \cdot T} \sum_{r=1}^{Rn} \sum_{i=1}^{T/f} (F(A_{c_{\max_i}}) - \text{opt}_i), \quad (20)$$

上式: Rn 表示算法的运行次数, T 表示算法的迭代次数, f 表示函数的变化周期, $F(A_{c_{\max}})$ 表示每次环境变化后所获最后一代记忆库中竞争力最大个体对应的函数值, opt 表示函数的理论最小值.

ICS 的初始群体规模为 100, 增殖系数为 0.04, 每代随机抗体更新数为 10; Opt-aiNet 的初始群体规模为 20, 抗体抑制阈为 0.5, 每个抗体的克隆数为 4, 每代抗体更新率为 40%; Dopt-aiNet 的初始群体规模为 20, 抗体抑制阈为 0.5, 每个抗体的克隆数为 4, 每代抗体更新率为 40%, 最大的抗体个数为 40; Dmaopt-aiNet 的规模选择为 $Ls \times Ls$ 为 5×5 , 克隆规模为 4, 激励系数 $\gamma = 0.5$, 控制系数 $\eta = 0.5$, 自信度阈值 $\zeta = 0.5$. 以上算法的最大迭代次数 $N_{\max}$ 都为 1000, 选择 $\tau = 0.1$, $f = 1$. 在以上参数设置下, 各算法分别独立执行 50 次, 统计结果见表 1, 其中 Dopt-aiNet 算法的运行结果直接取自文献[15].

由表 1 可知, 在 50 次统计值中, Dmaopt-aiNet 在各动态函数下获得的最小目标函数值比其他算法的小, 总体上获得的最大函数值比 ICS 的略小, 且在动态函数下获得的平均误差及方差均比其他算法的小, 此结果可获知, Dmaopt-aiNet 能较好地跟踪环境变化且寻优效果较为稳定. 尽管 Opt-aiNet 和 Dopt-aiNet 在某些环境下所获最差结果比 Dmaopt-aiNet 的好(通过表 1 的最大值比较), 但结合所获最小值、平均误差特别是方差比较可知, Opt-aiNet 和 Dopt-aiNet 在动态环境下所获效果稳定性较差. Dmaopt-aiNet 在免疫网络基础上采用了 Agent 的竞争、协作技术, 在保持个体多样性基础上能较好地权衡全局与局部搜索能力, 因此总体搜索结果的稳定性较好.

表 1 4种算法性能比较
Table 1 Performance comparison of 4 algorithms

函数	算法	线性变化函数	圆变化函数	Gaussian变化函数
Sphere	平均误差±方差	ICS	2.02 ± 0.92	2.61 ± 0.87
		Opt-aiNet	2.35 ± 1.47	1.96 ± 0.99
		Dopt-aiNet	0.02 ± 2.22	0.02 ± 0.17
		Dmaopt-aiNet	0.02 ± 0.66	0.02 ± 0.06
	最大值	ICS	62.20	2.37
		Opt-aiNet	32.20	3.42
		Dopt-aiNet	46.76	0.75
		Dmaopt-aiNet	53.14	2.14
	最小值	ICS	0.86	1.08
		Opt-aiNet	1.02	1.35
		Dopt-aiNet	—	—
		Dmaopt-aiNet	4.2×10^{-3}	5.6×10^{-3}
Rosenbrock	平均误差±方差	ICS	4.16 ± 0.58	3.92 ± 0.92
		Opt-aiNet	4.33 ± 0.47	3.61 ± 0.19
		Dopt-aiNet	0.50 ± 0.17	0.57 ± 0.24
		Dmaopt-aiNet	0.22 ± 0.66	0.37 ± 0.15
	最大值	ICS	4.62	57.23
		Opt-aiNet	3.17	37.18
		Dopt-aiNet	2.59	43.79
		Dmaopt-aiNet	5.28	41.14
	最小值	ICS	1.09	0.86
		Opt-aiNet	1.22	1.02
		Dopt-aiNet	—	—
		Dmaopt-aiNet	7.9×10^{-2}	6.2×10^{-2}
Rastrigin	平均误差±方差	ICS	2.17 ± 1.12	3.11 ± 0.97
		Opt-aiNet	1.09 ± 0.87	2.11 ± 1.08
		Dopt-aiNet	0.03 ± 0.16	0.38 ± 0.58
		Dmaopt-aiNet	0.02 ± 0.07	0.12 ± 0.26
	最大值	ICS	44.10	72.15
		Opt-aiNet	32.27	62.21
		Dopt-aiNet	39.56	74.05
		Dmaopt-aiNet	38.14	73.14
	最小值	ICS	0.91	1.26
		Opt-aiNet	0.73	1.01
		Dopt-aiNet	—	—
		Dmaopt-aiNet	5.1×10^{-3}	2.8×10^{-3}
Griewank	平均误差±方差	ICS	2.02 ± 2.32	3.01 ± 0.87
		Opt-aiNet	1.23 ± 1.97	2.77 ± 0.73
		Dopt-aiNet	0.03 ± 1.76	0.33 ± 0.17
		Dmaopt-aiNet	0.02 ± 0.86	0.17 ± 0.12
	最大值	ICS	6.20	6.20
		Opt-aiNet	4.20	3.20
		Dopt-aiNet	1.48	1.02
		Dmaopt-aiNet	7.14	5.14
	最小值	ICS	1.06	0.86
		Opt-aiNet	0.92	1.02
		Dopt-aiNet	—	—
		Dmaopt-aiNet	6.7×10^{-4}	5.4×10^{-4}

5 Dmaopt-aiNe的性能分析(Dmaopt-aiNet performance analysis)

5.1 操作算子对动态寻优的影响(Influence of operators on dynamic optimization)

多样性是动态优化的重要评估指标, 文献[25]引入了表现型多样性(phenotypical diversity, PDM)和基因型多样性(genotypical diversity, GDM)测量指标, 本文将用于Dmaopt-aiNet的多样性测量, 定义如下:

$$PDM = C(A_{m,n})_{avg} / C(A_{m,n})_{max}, \quad (21)$$

$$GDM = (\bar{E} - E_{min}) / (E_{max} - E_{min}), \quad (22)$$

其中: $C(A_{m,n})_{avg}$ 和 $C(A_{m,n})_{max}$ 分别为当代网格系统个体的平均竞争力和最大竞争力, $PDM \in [0, 1]$, 当算法收敛时, 其值趋于1; $\bar{E}$, E_{max} 和 E_{min} 分别为当代系统中所有个体与最佳个体间的平均欧氏距离、最大和最小欧氏距离. $GDM \in [0, 1]$, 当GDM趋于0时, 表明个体趋于一致, 算法呈收敛状态; 而当GDM较大时, 则个体差异较大. 通常, 如果 $PDM \geq 0.9$ 且 $GDM < 0.1$, 认为算法已趋收敛; 如果 $0 < PDM < 0.9$ 且 $GDM \geq 0.1$, 则算法处搜索阶段. Dmaopt-aiNet采用了网格环境下的邻域克隆选择、竞争和协作操作来提高个体的竞争力, 为研究各操作算子对动态寻优的影响, 设计了动态测试实验: 采用高斯变化的Sphere函数, 以100代为单位, 交替独立运行各操作算子, 比较50次的PDM、GDM平均结果如图3所示, 图中: “※”点为环境变化点, 其中: $\tau = 0.1$, $f = 50$; 编号1, 3, 5为Dmaopt-aiNet; 编号2为邻域克隆选择, 编号4为邻域竞争, 编号4为邻域协作.

图3明显可以看出, 运行Dmaopt-aiNet时, 一旦环境发生变化(图中“※”点开始), PDM由小以较快的速度上升, 而GDM会有大的跳变, 然后逐渐减小, 表明算法能很好地由搜索逐渐过渡到收敛阶段, 搜索阶段GDM都能维持在0.1~0.45范围内, 表明算法多样性较好; 环境变化后, 邻域克隆选择算子的PDM一直维持较小的水平, 而GDM一直能维持较大水平, 表明系统一直运行在搜索阶段, 而多样性较好; 与邻域克隆选择相反, 邻域竞争在环境变化后PDM由小很快增大, 而GDM维持较低水平, 表明邻域竞争能快速收敛, 但由于多样性不够, 容易早熟; 与以上两种算子不同的是邻域协作算子的PDM和GDM对环境变化不敏感, 都能维持较大的水平, 表明经个体间的信息交流能较好地维持搜索中的动态平衡. 综上, 通过以上3个操作算子的共同作用, 能使Dmaopt-aiNet在搜索中保持多样性的基础上, 快速收敛, 并能维持局部与全局的

平衡.

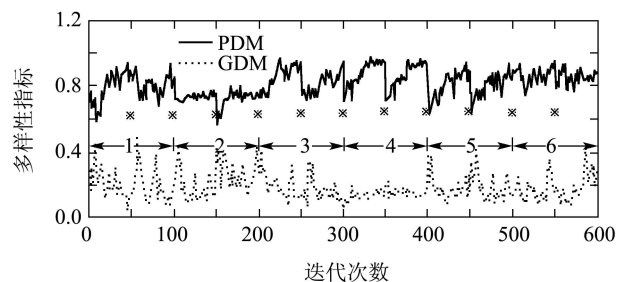


图3 各操作算子的多样性指标

Fig. 3 Diversity measurements for each operators

5.2 控制参数对寻优性能的影响(Influence of control parameters on the optimization performance)

Q-变异受控因子 η 在搜索过程中起了重要的作用, 在Dmaopt-aiNet算法中, Q-变异受控因子 η 取值为 $\eta \in \{0.1, 0.2, 0.3, 0.5, 1, 3, 10\}$, 表2为不同 η 下的50次平均结果. 不同 η 意味着不同的变异水平, η 越大对应的变异就越大. 从表2中看出, η 取0.1时对线性变化函数效果较好, $\eta = 0.2$ 到0.3时对圆变化效果较好, 线性变化其次; 而 η 取0.5~1时, 高斯变化函数寻优效果有明显改善; η 取3~10时, 以上3种函数变化的寻优效果明显恶化. 由此可见, 受控因子的选取与函数变化程度有关, 线性变化要求采用较小的受控因子, 圆变化所需的受控因子略大, 而复杂的高斯随机变化需采用稍大的受控因子, 且太大的变异受控因子容易产生逃逸现象, 使搜索效果变差.

5.3 函数的维数对评价次数的影响(Influence of dimensions on the number of evaluations)

函数的评价次数是影响动态性能的关键. 本文还对上述4个典型测试函数按维数为2~100变化进行评估次数的分析. 将Dmaopt-aiNet在每个采样的维数上按一种动态运动随机运行20次, 记录记忆库个体10次内不变化下的平均评估次数. 图4(a)~(d)给出了平均评价次数随维数变化的关系, 由图可知, 对于Sphere, Griewank和Rastrigin函数, 尽管维数在2~40变化内评价次数随维数的增长较明显, 但以后随着维数的增长, 评价次数增长速度降低, 当维数大于45时, 评价次数分布在4000~5000间; 对于Rosenbrock函数, 在维数达到20后, 对评价次数没有明显的影响, 函数各种变化下的评价次数在6000左右波动. 可见, 随着优化函数维度的增加, 所用的评价次数增加得不是很明显. 可见, Dmaopt-aiNet在高维函数优化上有着十分优越的性能.

表 2 η 对Dmaopt-aiNet寻优的影响(平均误差 $\pm$ 方差)

Table 2 Infuence of η on the performance of Dmaopt-aiNet(mean error $\pm$ standard deviation)

η	Sphere			Rosenbrock		
	线性变化函数	圆变化函数	Gaussian变化函数	线性变化函数	圆变化函数	Gaussian变化函数
0.1	0.01 $\pm$ 0.13	0.86 $\pm$ 0.37	1.01 $\pm$ 0.28	0.05 $\pm$ 0.18	0.75 $\pm$ 0.88	1.23 $\pm$ 0.68
0.2	0.01 $\pm$ 0.37	0.42 $\pm$ 0.12	1.07 $\pm$ 0.44	0.03 $\pm$ 0.48	0.31 $\pm$ 0.48	1.11 $\pm$ 0.51
0.3	0.01 $\pm$ 0.42	0.39 $\pm$ 0.10	1.03 $\pm$ 0.36	0.02 $\pm$ 0.38	0.29 $\pm$ 0.37	1.05 $\pm$ 0.43
0.5	0.02 $\pm$ 0.66	0.18 $\pm$ 0.09	0.02 $\pm$ 0.06	0.22 $\pm$ 0.66	0.37 $\pm$ 0.15	0.11 $\pm$ 0.12
1	0.02 $\pm$ 0.36	0.13 $\pm$ 0.19	0.11 $\pm$ 0.16	0.12 $\pm$ 0.46	0.22 $\pm$ 0.25	0.26 $\pm$ 0.22
3	3.12 $\pm$ 2.16	2.73 $\pm$ 1.64	3.57 $\pm$ 1.29	3.69 $\pm$ 2.77	4.86 $\pm$ 3.01	3.61 $\pm$ 2.22
10	6.61 $\pm$ 4.21	9.25 $\pm$ 4.21	11.24 $\pm$ 8.19	7.28 $\pm$ 5.11	5.32 $\pm$ 2.14	8.17 $\pm$ 5.31

η	Sphere			Rosenbrock		
	线性变化函数	圆变化函数	Gaussian变化函数	线性变化函数	圆变化函数	Gaussian变化函数
0.1	0.02 $\pm$ 0.10	0.87 $\pm$ 0.75	1.31 $\pm$ 0.65	0.02 $\pm$ 0.16	0.86 $\pm$ 0.54	8.13 $\pm$ 2.74
0.2	0.02 $\pm$ 0.10	0.41 $\pm$ 0.21	1.43 $\pm$ 0.36	0.03 $\pm$ 0.22	0.38 $\pm$ 0.32	7.33 $\pm$ 2.86
0.3	0.02 $\pm$ 0.06	0.31 $\pm$ 0.11	1.17 $\pm$ 0.16	0.02 $\pm$ 0.15	0.29 $\pm$ 0.27	6.81 $\pm$ 2.66
0.5	0.02 $\pm$ 0.07	0.12 $\pm$ 0.26	0.02 $\pm$ 0.08	0.02 $\pm$ 0.86	0.17 $\pm$ 0.12	5.76 $\pm$ 2.18
1	0.03 $\pm$ 0.17	0.24 $\pm$ 0.46	0.07 $\pm$ 0.19	0.05 $\pm$ 0.71	0.23 $\pm$ 0.31	4.12 $\pm$ 1.01
3	3.33 $\pm$ 2.09	3.19 $\pm$ 2.16	4.12 $\pm$ 1.69	3.97 $\pm$ 2.14	2.73 $\pm$ 1.97	12.32 $\pm$ 4.53
10	9.19 $\pm$ 5.47	13.29 $\pm$ 8.12	6.92 $\pm$ 7.36	9.92 $\pm$ 4.21	7.76 $\pm$ 5.32	13.38 $\pm$ 2.76

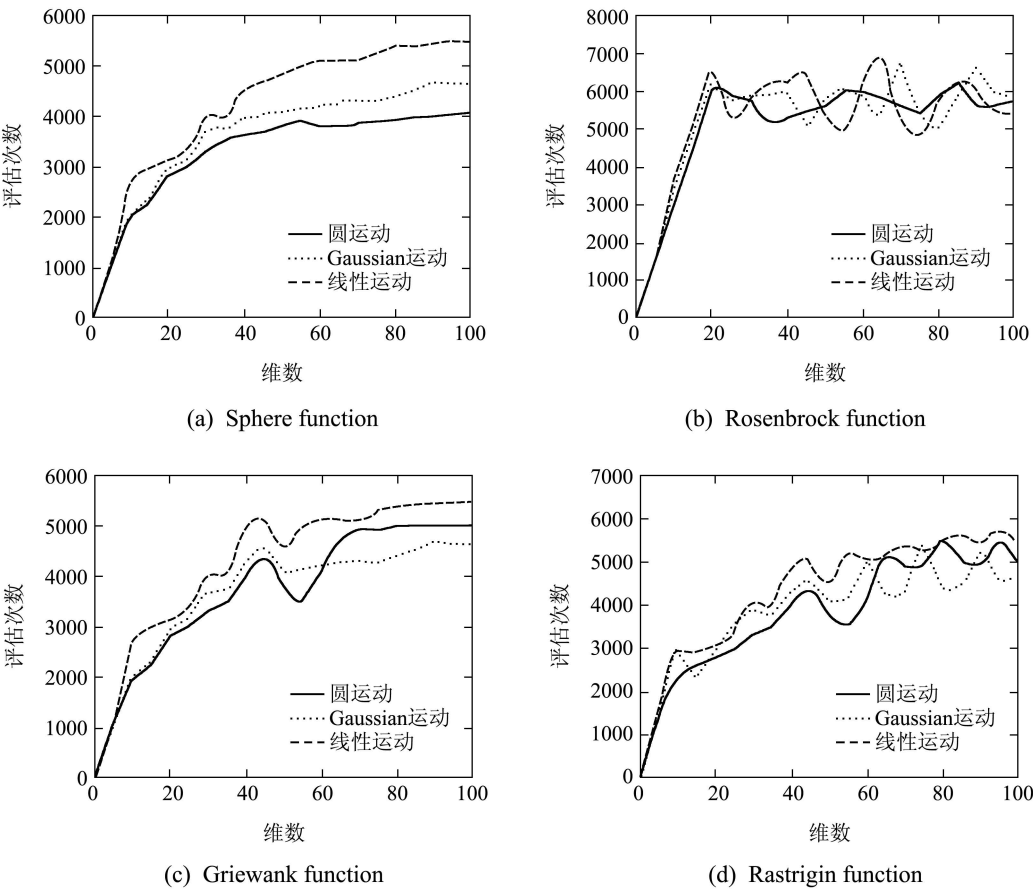


图 4 算法评价次数与函数维数的关系

Fig. 4 Relationship between the number of evaluations and the dimension of the functions

6 结论(Conclusions)

借助生物免疫网络的核心思想及多Agent的竞争、协作技术,设计一种动态环境下的免疫网络多Agent优化算法.为了能快速跟踪环境的变化,准确定位动态环境下的最优解的位置,本文采用了以下的动态优化策略:双重Agent网络结构、双重变异及动态环境检测策略,并用于动态环境下高维多峰值函数优化.理论分析了Dmaopt-aiNet算法具有全局收敛性,实验结果表明该算法在处理高维动态优化问题具有较突出的优越性,并能准确定位动态环境下优化问题的最优解的位置,具有较好的搜索效果和效率.另外,本文还对算法的性能作了研究,研究了各搜索算子及算法控制参数对动态寻优的影响,尤其是平均评价次数随维数的变化规律,结果表明,随着优化函数维度的增加,所用的评价次数增加得不是很明显.关于该算法的广泛应用和理论研究,需待进一步开展.

参考文献(References):

- [1] JIN Y C, BRANKE J. Evolutionary optimization in uncertain environments—A survey[C] // *Proceedings of IEEE Congress on Evolutionary Computation*. New York: IEEE, 2005, 9(3): 303 – 317.
- [2] MORRISON R W, JONG DE K A. Triggered hypermutation revisited[C] // *Proceedings of IEEE Congress on Evolutionary Computation*. Piscataway: IEEE, 2000: 1025 – 1032.
- [3] GOLDBERG D E, SMITH R E. Nonstationary function optimization using genetic algorithms with dominance and diploidy[C] // *Proceedings of the 2nd International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1987: 59 – 68.
- [4] SIMOES A, COSTA E. Using GAs to deal with dynamic environments: a comparative study of several approaches based on promoting diversity[C] // *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco: Morgan Kaufmann Publishers, 2002: 698 – 702.
- [5] MORI N, KITA H, NISHIKAWA Y. Adaptation to a changing environment by means of the thermo-dynamical genetic algorithm[C] // *Parallel Problem Solving from Nature*. Berlin: Springer Publishers, 1996: 513 – 522.
- [6] YANG S. Non-stationary problem optimization using the primal-dual genetic algorithm[C] // *Proceedings of the 2003 Congress on Evolutionary Computation*. Piscataway: IEEE, 2003: 2246 – 2253.
- [7] BRANKE J. Memory enhanced evolutionary algorithms for changing optimization problems[C] // *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway: IEEE, 1999: 1875 – 1882.
- [8] BENDTSEN C N, KRINK T. Dynamic memory model for non-stationary optimization[C] // *Proceedings of the 2002 Congress on Evolutionary Computation*. Piscataway: IEEE, 2002: 145 – 150.
- [9] BRANKE J, KAUBLER T, SCHMIDT C. A multi-population approach to dynamic optimization problems[M] // *Adaptive Computing in Design and Manufacturing*. Berlin: Springer-Verlag, 2000: 299 – 308.
- [10] OH S K, LEE C Y, LEE J J. A new distributed evolutionary algorithm for optimization in nonstationary environments[C] // *Proceedings of the 2002 Congress on Evolutionary Computation*. Piscataway: IEEE, 2002: 1875 – 1882.
- [11] 焦李成, 杜海峰, 刘芳, 等. 免疫优化计算、学习与识别[M]. 北京: 科学出版社, 2007: 316 – 325.
(JIAO Licheng, DU Haifeng, LIU Fang, et al. *Immune Optimization, Learning and Recognition*[M]. Beijing: Science Press, 2007: 316 – 325.)
- [12] GASPAR A, COLLARD P. Two models of immunization for time dependent optimization[C] // *IEEE International Conference on Systems, Man and Cybernetics*. New York: IEEE, 2000, 1: 113 – 118.
- [13] WALKER J H, GARRETT S M. Dynamic function optimization: Comparing the performance of clonal selection and evolution strategies[C] // *Proceedings of the 2nd International Conference on Artificial Immune Systems*. Berlin: Springer Verlag, 2003: 273 – 284.
- [14] 罗印升, 李人厚, 张维玺. 基于免疫机理的动态函数优化算法[J]. 西安交通大学学报, 2005, 39(4): 384 – 388.
(LUO Yinsheng, LI Renhou, ZHANG Weixi. Immune-based mechanism of dynamic function optimization algorithm[J]. *Journal of Xi'an Jiaotong University*, 2005, 39(4): 384 – 388.)
- [15] FRANCA D, ZUBEN V, CASTRO D. An artificial immune network for multimodal function optimization on dynamic environments[C] // *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. New York: ACM Press, 2005: 289 – 296.
- [16] RUSSELL S, NORVIQ P. *Artificial Intelligence: a modern Approach*[M]. New York: Prentice-Hall, 1995: 375 – 382.
- [17] 钟伟才, 薛明志, 刘静, 等. 多智能体遗传算法用于超高维函数优化[J]. 自然科学进展, 2005, 15(10): 1078 – 1083.
(ZHONG Weicai, XUE Mingzhi, LIU Jing, et al. Multi-agent genetic algorithm for high dimensional function optimization[J]. *Progress in Natural Science*, 2005, 15(10): 1078 – 1083.)
- [18] HOEN P J, TUYLS K, PANAIT L, et al. An overview of cooperative and competitive multiagent learning[M] // *STYLS K, HOEN P, VERBEECK K, et al. Lecture Notes in Computer Science: Learning and Adaption in Multi-Agent Systems*. Berlin: Springer-Verlag, 2006, 3898: 1 – 46.
- [19] LAU H Y K, WONG V W K. An immunity approach to strategic behavioral control[J]. *Engineering Applications of Artificial Intelligence*, 2006, 6(2): 1 – 18.
- [20] 李中华, 张雨浓, 谭洪舟, 等. 一类具有精英学习能力的增强型人工免疫网络优化算法[J]. 控制理论与应用, 2009, 26(3): 283 – 290.
(LI Zhonghua, ZHANG Yunong, TAN Hongzhou, et al. An enhanced artificial immune network with elitist-learning capability for optimization problems[J]. *Control Theory & Applications*, 2009, 26(3): 283 – 290.
- [21] TROJANOWSKI K, WIERZCHON S T. Immune-based algorithms for dynamic optimization[J]. *Information Sciences*, 2009, 179(10): 1495 – 1515.
- [22] ANGELINE P J. Tracking extrema in dynamic environments[C] // *Proceedings of the 6th International Conference on Evolutionary Programming*. Berlin: Springer, 1997, 1213: 13 – 16.
- [23] CASTRO D, ZUBEN V. Learning and optimization using the clonal selection principle[J]. *IEEE Transactions on Evolutionary Computation*. Piscataway, 2002, 6(3): 239 – 251.
- [24] CASTRO D, TIMMIS J. An artificial immune network for multimodal function optimization[C] // *Proceedings of the 2002 Congress on Evolutionary Computation*. Piscataway: IEEE, 2002: 699 – 704.
- [25] HERRERA F, LOZANO M. Adaptation of genetic algorithm parameters based on fuzzy logic controllers[M] // *Genetic Algorithms and Soft Computing*. Berlin: Springer-Verlag, 1996: 95 – 125.

作者简介:

史旭华 (1967—), 女, 副教授, 主要研究领域为计算智能理论及其在系统建模和优化中的应用, E-mail: shixuhua@nbu.edu.cn;

钱锋 (1961—), 男, 教授, 博士研究生, 博士生导师, 主要从事工业过程先进控制、优化与故障诊断、人工智能在流程工业过程建模、控制和优化中的应用等方面的研究, E-mail: fqian@ecust.edu.cn.