

多智能体系统中子域适应度评估的合作协进化协作

张晓勇, 彭军, 李哲琴

(中南大学 信息科学与工程学院, 湖南 长沙, 410083)

摘要: 为了克服合作协进化算法在解决复杂多智能体系统协作问题时存在的适应度函数难以建立和协作行为难以达到全局最优等问题, 提出 1 种子域适应度评估的合作协进化算法来实现异构多智能体系统中智能体的自适应协作。该算法将复杂问题域模型分解成相互影响较小、较易求解的子问题域模型, 在子问题域模型之间并行使用合作协进化算法来完成智能体协作行为的进化, 有效降低适应度评估的复杂度。在子问题域进行合作协进化时, 在适应度函数中引入环境因子影响矩阵, 将其他子问题域的影响信息映射到该子问题域中的个体适应度评估中, 从而引导种群向全局优化方向进化。ECJ 系统中的仿真实验结果验证了其有效性。

关键词: 自适应协作; 合作协进化; 适应度评估; 子问题域

中图分类号: TP391.9

文献标志码: A

文章编号: 1672-7207(2010)02-0572-06

Cooperative co-evolutionary collaboration algorithm based on sub-domain fitness evaluation in multi-agent system

ZHANG Xiao-yong, PENG Jun, LI Zhe-qin

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: In order to overcome the difficulties of establishing the fitness evaluation function and obtaining the global optimal decision when the cooperative co-evolutionary algorithm is applied to solve the collaboration in complex multi-agent system, a cooperative co-evolution algorithm with sub-domain fitness evaluation was introduced to deal with the adaptive collaboration of heterogeneous multi-agent system. In the proposed algorithm, the complex problem domain model was decomposed into some sub-domain models with less interaction. The results show that the evolution of agents' behavior is completed by parallel cooperative co-evolution among the sub-domains, which reduces the complexity of fitness evaluation function effectively. While the cooperative co-evolution is executed in a sub-domain, a matrix of environmental impact is introduced to map other sub-domains' influence information to the fitness evaluation function in this sub-domain, to achieve the global optimization of population evolution. The proposed algorithm is proven to be effective in ECJ simulation platform.

Key words: adaptive collaboration; cooperative co-evolutionary; fitness evaluation; sub-domain

协作是多智能体系统中的 1 个关键问题, 多智能体系统中的自主智能体要完成系统的目标必须进行有效的协作。智能体处在一个高度动态变化的环境中, 状态空间极大且难以准确预测, 所以, 难以预先为智能体设定完整的行为策略和控制参数。智能体必须具

有自适应动态环境的能力, 在运行过程中动态调整自身行为策略, 实现多智能体系统更强的环境适应性和鲁棒性, 才能更好地完成系统的复杂任务。多智能体系统协进化算法是近年来进化理论发展的热点, 它为解决复杂系统动态自适应、机器学习等问题提供了 1

收稿日期: 2009-07-15; 修回日期: 2009-09-20

基金项目: 国家自然科学基金资助项目(60874042)

通信作者: 张晓勇(1980-), 男, 山西原平人, 博士研究生, 从事计算机应用与多智能体系统等研究; 电话: 0731-82655576; E-mail: zxyong@gmail.com

种新手段^[1-6]。协进化算法基于生态学的种群协同理论,运用种群间的自动调节、自适应原理,是复杂动态变化环境中的智能体群体产生适应性协作行为的新途径。对于协进化算法在多智能体系统协作的应用,目前已有成功的范例^[7-9]。Uchibe等^[10]用合作型协进化算法获取足球机器人的协作行为;Luke等^[11]用协进化方法训练得到了1个完整的机器人足球队;Puppala等^[12]提出了1种“共享记忆”方法,用于存储协进化过程中成功合作的个体对,并将这种方法应用到2个房屋粉刷机器人的协调控制中,获得了较好的效果。但是,这些方法都只适用于任务简单的同构智能体系统中。在异构问题域中,Fukuda等^[13]采用1种细菌感染协进化算法,解决智能机器人运动规划的决策问题;Thomas等^[14]对著名的“捕食者-猎物”问题进行了全面研究,并采用合作型协进化方法研究了由 k 个异类智能体组成的团队如何获取合作策略的问题。这些方法不适用于复杂、高维的问题域中。采用合作协进化算法对异构多智能体系统的协作行为策略进行优化、寻求最优协作策略时,存在如下问题:适应度函数难以建立;协作难以达到全局最优。为此,本文作者针对非结构化的环境和多样化复杂任务的异构多智能体系统协作效用最大化问题,提出1种基于子域适应度评估的合作协进化协作算法,克服了合作协进化算法在解决复杂多智能体系统协作问题时存在的通信量过于繁重、适应度函数难以建立等问题。

1 合作协进化算法

合作协进化算法模拟自然界种群之间的协进化机制,对所有种群进行并行进化,优化种群之间的合作。在多智能体系统中采用合作协进化算法对智能体之间的行为进行自适应协作协调,定义如下。

(1) 问题域:整个多智能体系统所要解决的问题或者需要完成的任务。

(2) 种群:1个智能体所拥有的行为决策集。

(3) 个体:1个状态环境下通过规则推理生成的1个行为决策。

(4) 适应度函数:对问题域中任务完成的优劣程度进行度量的函数。

合作协进化算法的核心部分是个体适应度的评估。个体的适应度是根据其与其他种群中个体一起完成任务的优劣通过适应度函数计算得到赋值的。对那些有利于种群间协调协作的行为赋予较高的适应度,而对于不利于协作的策略则赋予较低的适应度,使种

群朝着相互协调适应的方向进化,产生全局协调协作行为。

2 子域适应度评估的合作协进化算法

2.1 问题域的划分

在异构多智能体系统的实际问题中,不同种类的智能体完成的任务不同,对某智能体来说,其他类智能体的行为决策对其子任务的完成影响较小。因此,在评估该智能体行为决策的适应度时,若不从异构种群中选取代表个体参与评估,则可减少通信负担。根据智能体的异构特征和整个系统的任务规划,将问题域空间分解成相互影响较小、较易求解的子问题域,这样,在子问题域进行合作协进化,将大大降低协进化算法中适应度评估函数的维数,简化适应度函数的建立过程。异构多智能体系统的自适应协作问题描述如下。

定义1:问题域。问题域 F 由异构多智能体系统中所有的种群组成,能完成多智能体系统需要完成的总任务。按照动态任务规划方法可以将总任务划分成 N 个子任务,则问题域 F 可以表示为:

$$F = \{F_1, F_2, \dots, F_i\}, i \quad (1, N) \quad (1)$$

定义2:子问题域。子问题域 F_i 由 M 个同构种群构成,能完成问题域 F 总任务中的一部分(子任务)。其中, M 表示子问题域 F_i 中的智能体个数。同构种群结构相同,行为功能相同并且联系紧密,以完成同一子任务为目标,则子问题域 F_i 可以表示为:

$$F_i = \{P_1^i, P_2^i, \dots, P_j^i\}, j \quad (1, M) \quad (2)$$

其中: P_j^i 表示子问题域 F_i 中的第 j 个种群,是该智能体的所有 L 个行为决策的集合。则 P_j^i 可以表示为:

$$P_j^i = \{p_1^{ij}, p_2^{ij}, \dots, p_k^{ij}\}, k \quad (1, L) \quad (3)$$

复杂动态环境下的异构多智能体协作问题便可归结为系统中所有异构智能体寻找最优合作策略来完成的任务。

2.2 子问题域模型中适应度函数的设计

把复杂问题域划分成子问题域后,如何将子问题域中合作协进化算法寻求的最优合作策略朝着全局优化方向发展成为1个需要解决的关键问题。问题域分解后子问题域中的个体适应度评估需要考虑3个因素:

(1) 考虑其所在的子问题域的环境适应性;

(2) 考虑其与所在的子问题域中其他种群中个体协调协作的表现;

(3) 考虑其对来自其他子问题域中异构种群的环境适应性的影响。

由于子问题域内每个种群的感知范围有限,只能获得有限范围内的局部状态信息,对其他子问题域内的状态信息未知。在子问题域中采用合作进化算法,个体通过适应度函数计算得到的结果是不准确的,因为它只考虑了前面 2 个因素而没顾及到全局,从而使得进化难以收敛于全局优化;因此,需要对子问题域模型中的适应度评估函数进行合理设计,使其能够考虑其他子问题域的影响,以得到 1 个更准确的评估值。

矩阵法是环境影响综合评价中的一种基础方法和重要手段,广泛应用于环境影响评价中。矩阵法是把各项活动和受影响的各项环境因子分别表示为矩阵的列与行,在两者之间建立直接的因果关系,以表示哪些活动对哪些环境因子产生影响和影响的程度。子域适应度评估的合作进化算法采用环境因子影响矩阵法,将其对来自其他子问题域中异构种群的环境适应性的影响映射到适应度函数中评估适应度,进行如下定义。

定义 3:环境因子影响矩阵。假设任意一个子域中种群的个体对其他子域中种群的环境因子 $e_q(q=1, \dots, C)$ 产生影响,共有 C 个环境因子受到影响。矩阵 $H_{N \times N}(e_q)$ 为各个子域的环境因子 e_q 影响矩阵; $h_{iw}(e_q)$ 表示在对第 i 个子域中种群的个体进行评估时,来自第 w 个子域中种群对其环境因子 e_q 的影响,当 $i=w$ 时,定义 $h_{iw}(e_q)=0$;当 $i \neq w$ 时, $h_{iw}(e_q)$ 根据实际问题域中各子域中的种群对其他子域的环境因子 e_q 影响程度来定义。则 $H_{N \times N}(e_q)$ 可以表示成如下形式:

$$H_{N \times N}(e_q) = \begin{pmatrix} 0 & \dots & h_{1N}(e_q) \\ \vdots & 0 & \vdots \\ h_{N1}(e_q) & \dots & 0 \end{pmatrix} \quad (4)$$

定义 4:贡献度。 N 个子域的贡献度构成贡献度矢量 Z_N , 为:

$$Z_N = \{z_1, z_2, \dots, z_N\} \quad (5)$$

其中: $z_1+z_2+\dots+z_N=1$; $z_i = \frac{Value(F_i)}{Value(F)}$; z_i 表示子域 F_i

在整个问题域中的贡献度,定义为第 i 个子域 F_i 中完成的子任务价值 $Value(F_i)$ 在完成整个复杂问题域 F 中的总任务价值 $Value(F)$ 所占的比例。

定义 5:适应度函数。在子问题域 F_i 中有 M 个同构种群,种群 P_j^i 中的 1 个个体 p_k^{ij} 的适应度函数为:

$$f(p_k^{ij}) = \tau(\theta_1^i, \theta_2^i, \dots, \theta_{M-1}^i, \omega(p_k^{ij})) + \lambda Z_N \sum_{q=1}^E h_i(e_q) = \tau(\theta_1^i, \theta_2^i, \dots, \theta_{M-1}^i, \omega(p_k^{ij})) + \lambda(z_1, z_2, \dots, z_N) \sum_{q=1}^E (h_i(e_q), h_{2i}(e_q), \dots, h_{Ni}(e_q))^T \quad (6)$$

式中: $\theta_1^i, \theta_2^i, \dots, \theta_{M-1}^i$ 为来自子问题域 F_i 中参与评估的其他 $M-1$ 个种群的代表个体的信息; $\omega(p_k^{ij})$ 为个体 p_k^{ij} 自身的信息,它们可以是 1 个常值、1 个函数或者矢量; $\tau(\theta_1^i, \theta_2^i, \dots, \theta_{M-1}^i, \omega(p_k^{ij}))$ 表示个体 p_k^{ij} 在子域环境下评价其对该子域环境的适应度; $h_i(e_q)$ 为 $H_{N \times N}(e_q)$ 的第 i 列; λ 为协调因子,可依据子问题域之间影响程度定义。

协进化算法中问题域模型的功能是将需要评价的个体与从问题域的其他种群中选出的代表个体组合形成协作行为,应用在问题域模型中,通过维护随时更新的状态信息,采用适应度评估函数评价其适应度,并赋予该个体协作的适应度返回其种群。在改进后的算法中,不仅子问题域模型继承了问题域模型的功能,子问题域模型之间还能通过信息通信交互获得环境因子影响矩阵,采用改进的适应度函数评价个体适应度。

2.3 子域适应度评估的合作进化算法

为了提高算法的效率,每个子问题域独立进行协进化。子域适应度评估的合作进化算法描述如下:

- 1 $t=0$
- 2 对于每个子域 F_i 中的所有种群,进行如下操作:
- 3 对种群进行随机初始化操作;
- 4 计算初始种群中每个个体的适应度;
- 5 结束
- 6 直到满足终止条件之前,进行如下操作:
- 7 $t=t+1$
- 8 对子域 F_i 中的每个种群 P_j^i ,进行如下操作:
- 9 基于适应度从上一代种群中选取新一代种群;
- 10 将遗传算子(交叉、变异)应用到种群的个体中;
- 11 对种群的每个个体,评价其适应度;
- 12 结束
- 13 个体 p_k^{ij} 适应度评价方法:

- 14 从子域 F_i 中的其他种群中选择代表个体;
- 15 对于种群 P_j^i 中的每个需要评价的个体 p_k^{ij} , 进行如下操作:
- 16 将个体 p_k^{ij} 与子域 F_i 中的其他种群中选择代表个体组合形成协作行为;
- 17 通过将这种协作行为应用到该子问题域中;
- 18 参考来自其他子问题域中的环境因子影响信息评价适应度;
- 19 对个体 p_k^{ij} 赋予该协作的适应度。
- 20 结束

子域适应度评估的合作进化算法协作模型如图 1 所示。图 1 中整个复杂的问题域有 6 个种群, 分成 2 个子问题域, 则 6 个种群分成 2 类, 子问题域之间可并行进化。为了计算种群 3 中个体的适应度, 该个体必须与该子问题域中的其他种群(种群 1、种群 2)中的代表个体结合组成 1 个子问题的可能解决方案, 然后, 考虑对子问题域 2 中种群的环境适应性的影响来评估此子问题域方案的适应度, 并将该个体的适应度反馈到该种群的进化中。

子域适应度评估的合作进化算法通过复杂问题域按照动态任务分解方法划分成子域, 缩小了种群进化规模, 简化了适应度函数的建立过程; 通过设计子域模型中的适应度评估函数, 子问题域中的种群进化寻求最优合作策略朝着全局优化方向发展; 同时, 在评估某种群的个体适应度时需要的代表个体数量减少, 使得种群间的通信量大量减少, 减轻了系统通信负担。

3 算法仿真和结果分析

为了验证子域适应度评估合作进化算法的有效性, 在 ECJ(Evolutionary Computation Journal)^[15] 系统上进行仿真。

3.1 算法仿真

根据 Potter 等^[8] 的实验结果, 合作进化算法对于典型的多元函数优化问题都能得到很高的优化效率, 如 Rosenbrock, Rastrigin, Rotated Griewank 和 Rotated Expanded Scaffer 多元函数, 这些函数的共同特点是各个变量之间存在交联关系。如对于 Rosenbrock 函数: $f = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$, 按照合作进化算法, 选择 2 个进化种群 A 与 B, 种群 A 中的每个个体对应于变量 x_1 的不同取值, 种群 B 中的每个个体对应于变量 x_2 的不同取值, 该函数在 $x_1 = x_2 = 1$ 时取全局最小值 0。

依据异构复杂问题域模型, 将 Rosenbrock, Rastrigin, Rotated Griewank, Rotated Expanded Scaffer 多元函数组合起来, 每个函数取 2 个自变量, 构建如下异构问题域函数:

$$F = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 + [y_1^2 - 10 \cos(2\pi y_1) + 10] + [y_2^2 - 10 \cos(2\pi y_2) + 10] + \frac{z_1^2}{4000} - \cos\left(\frac{z_1}{\sqrt{1}}\right) + 1 + \frac{z_2^2}{4000} - \cos\left(\frac{z_2}{\sqrt{2}}\right) + 1 + 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 - 0.001(x^2 + y^2))^2} - \frac{1}{2} |x_1 y_1 + x_2 y_2 + x_1 z_1 + x_2 z_2 + x_1 x + x_2 y + y_1 z_1 + y_2 z_2 + y_1 x + y_2 y + z_1 x + z_2 y|$$

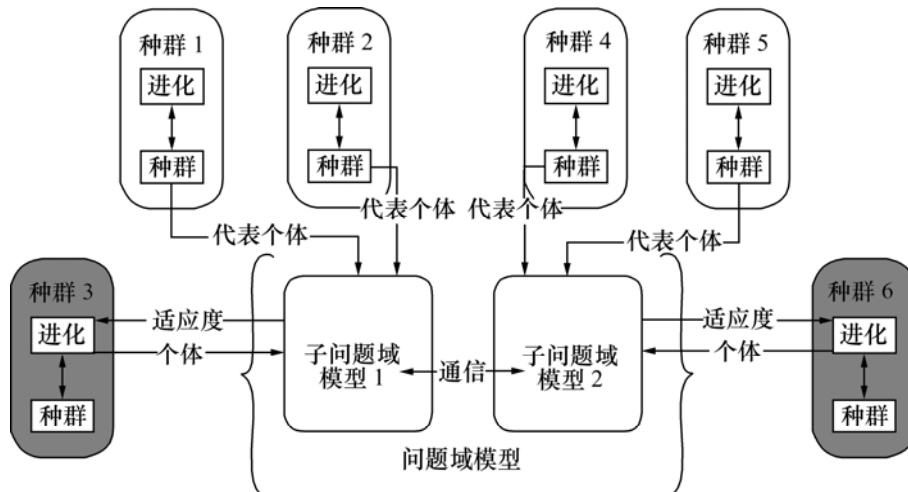


图 1 子域适应度评估的合作进化算法协作模型

Fig.1 Collaboration model based on cooperative co-evolution algorithm with sub-domain fitness evaluation

其中： $x_1, x_2, y_1, y_2, z_1, z_2, x$ 和 y 这 8 个自变量可以看成是 8 个智能体对应的种群中的个体， F 为问题域函数，采用改进的算法来求解 F 函数的全局最优值 0。可以证明：当 x_1 和 x_2 为 1， y_1, y_2, z_1, z_2, x 和 y 均为 0 时， $F=0$ 。

3.2 仿真结果及分析

在 ECJ 系统上采用 Rosenbrock 函数作为问题域进行合作进化算法仿真，然后，在此系统基础上进行修改编程，将问题域扩展成异构问题域 F ，仿真子域适应度评估的合作进化算法。为了使结果具有对比性，还进行了 6 个种群的子域适应度评估的合作进化算法仿真。

3.2.1 算法时间代价分析

在仿真过程中，采用合作进化算法，2 个种群进化到 75 代时，函数值接近于 0，耗时约 2 s。采用子域适应度评估的合作进化算法，基于 6 个种群同时进化 1 代耗时约 1 s；当进化到 23 代时，函数值接近于 0 时共耗时约 23 s；基于 8 个种群同时进化 1 代需要约 3 s，当进化到 8 代时，函数值接近于 0，共耗时约 24 s。

从仿真时间可知：进化的种群越多，完成进化所需时间越多。其主要原因是：在 ECJ 系统中，依据改进后算法编程，采用的是单线程，程序在一个种群进化完成后，才调用下一种群的进化，而没有实现改进后算法中所示种群之间并行协进化。在实际 MAS 系统中，种群的进化是在种群内部并行进行；因此，若实现了改进后算法中所示的种群之间并行协进化，则基于 6 个种群同时进化 1 代约需 0.17 s，基于 8 个种群同时进化 1 代需要约 0.37 s。但是，采用合作进化算法 2 个种群同时进化 1 代只需 0.027 s。这是因为算法改进后，种群之间并行进化，子域模型之间进行通信来交互环境因子影响信息，适应度评估相对于原来的 2 个种群的合作进化更复杂，增加了算法的时间。划分的子问题域越多，所需时间便越多。

3.2.2 算法收敛性分析

采用合作进化算法仿真基于 2 个种群的进化结果如图 2 所示；图 3 所示为采用子域适应度评估的合作进化算法仿真基于 8 个种群的进化结果。

由图 2 可知：2 个种群合作进化算法进化比较缓慢，进化到第 75 代后问题域的值收敛于 0。由图 3 可知：8 个种群采用子域适应度评估的合作进化算法进化较快。

由仿真结果可知：当种群数量越多、问题域越复杂时，子域适应度评估的合作进化算法越显示其优

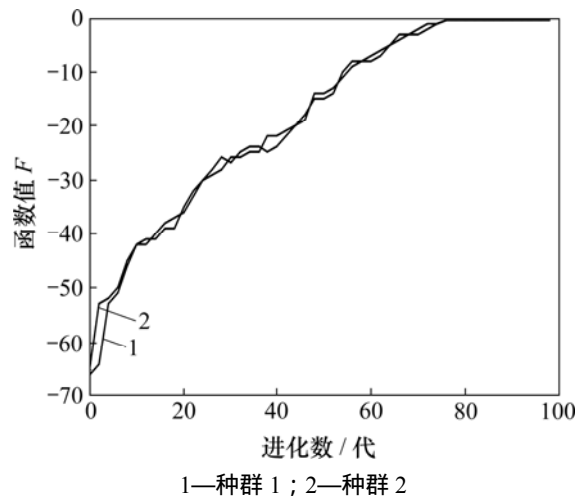
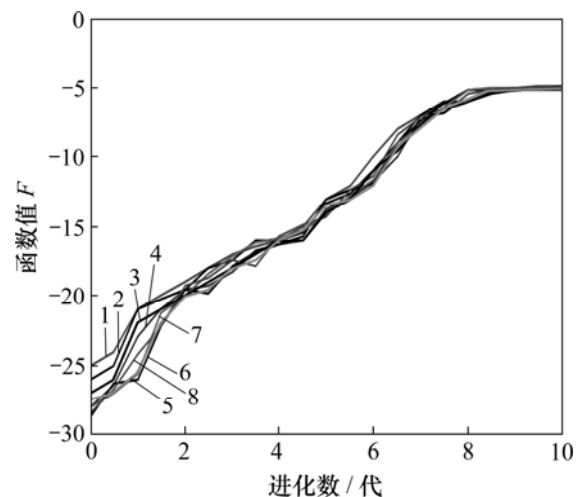


图 2 合作进化算法基于 2 个种群进化结果

Fig.2 Results of cooperative co-evolution algorithm for 2 populations



1—种群 1；2—种群 2；3—种群 3；4—种群 4；5—种群 5；
6—种群 6；7—种群 7；8—种群 8

图 3 子域适应度评估的合作进化算法基于 8 个种群进化结果

Fig.3 Results of cooperative co-evolution algorithm with sub-domain fitness evaluation for 8 populations

越的收敛性；通过采用环境因子影响矩阵法设计适应度评估函数，能有效引导种群更快速地寻求全局最优合作策略；种群越多，则需进化到最优的代数越少，验证了子域适应度评估的合作进化算法的有效性。

4 结论

(1) 根据问题域的异构特征和求解的需要，将问

题域空间分解成子问题域,缩小了种群进化规模,简化了适应度函数的建立过程,同时,在评估某种群的个体适应度时需要的个体数量减少,使得种群间的通信量大量减少,减轻了系统通信负担。

(2) 子问题域模型之间可以进行信息通信交互来获得环境因子影响信息,通过设计子域模型中的适应度评估函数,采用环境因子影响矩阵法将其他子域影响信息映射到该子域中种群的个体适应度评估中,从全局引导种群之间的协作进化。

(3) 在子问题域之间并行进行协进化时,子问题域之间需要交换环境因子影响信息,在一定程度上提高了算法的复杂度,但算法能够大大减少复杂问题域的进化代数,加快协进化的收敛速度。

参考文献:

- [1] Li X, Sol L K. Applications of decision and utility theory in Multi-agent systems[R]. Lincoln: University of Nebraska-Lincoln, 2004.
- [2] 范波,潘泉,张洪才.一种基于分布式强化学习的多智能体协调方法[J].计算机仿真,2005,22(6):115-117.
FAN Bo, PAN Quan, ZHANG Hong-cai. A method for multi-agent coordination based on distributed reinforcement learning[J]. Computer Simulation, 2005, 22(6): 115-117.
- [3] Huang J, Yang B, Liu D Y. A distributed Q-learning algorithm for multi-agent team coordination[C]//Proceedings of the Fourth International Conference on Machine Learning and Cybernetics. Guangzhou: IEEE Press, 2005: 108-109.
- [4] Tehrani A M, Kamel M S, Khamis A M. Fuzzy reinforcement learning for embedded soccer agents in a multi-agent context[J]. International Journal of Robotics and Automation, 2006, 21(2): 110-119.
- [5] Tan K C, Yang Y J, Goh C K. A distributed cooperative coevolutionary algorithm for multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2006, 10 (5): 527-549.
- [6] Lu H, Yen G. Rank-density-based multiobjective genetic algorithm and benchmark test function study[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(4): 325-343.
- [7] 罗杰,段建民,陈建新.一种引入局部交互的群体协作行为协同进化机制[J].机器人,2007,29(4):313-319.
LUO Jie, DUAN Jian-min, CHEN Jian-xin. A mechanism of cooperative coevolution with local interaction for collective cooperation behaviors[J]. Robot, 2007, 29(4): 313-319.
- [8] Potter M A, De Jong K A. Cooperative co-evolution: An architecture for evolving co-adapted subcomponents[J]. Evolutionary Computation, 2000, 8(1): 1-29.
- [9] Yang Z Y, Tang K, Yao X. Large scale evolutionary optimization using cooperative co-evolution[J] Information Sciences, 2008, 178(15): 2985-2999.
- [10] Uchibe E, Nakamura M, Asada M. Cooperative behavior acquisition in a multiple mobile robot environment by co-evolution[C]//Robocup-98: Robot Soccer World Cup II, Lecture Notes In Computer Science. London: Springer-Verlag, 1999: 273-285.
- [11] Luke S, Hohn C, Farris J, et al. Co-evolving soccer softbot team coordination with genetic programming[C]//RoboCup-97: Robot Soccer World Cup I, Lecture Notes in Computer Science. London: Springer-Verlag, 1998: 398-411.
- [12] Puppala N, Sen S, Gordin M. Shared memory based cooperative co-evolution[C]//Proceedings of the IEEE International Conference on Evolutionary Computation. New Jersey: IEEE Press, 1998: 570-574.
- [13] Fukuda T, Kubota N. Learning, adaptation and evolution of intelligence robotic system[C]//Proceedings of 1998 IEEE International Symposium on Computational Intelligence in Robotics and Automation. New Jersey: IEEE Press, 1998: 2-7.
- [14] Thomas D, Haynes, Sen S. Co-adaptation in a team[J]. International Journal of Computational Intelligence and Organizations, 1997, 1(4): 1-4.
- [15] Luke S. A Java EC research system[EB/OL]. 2008-09. <http://cs.gmu.edu/~eclab/projects/ecj/>.

(编辑 陈灿华)