

## 合作型协同演化算法研究进展\*

张凯波, 李 斌

(中国科学技术大学电子科学与技术系, 安徽 合肥 230027)

**摘要:**合作型协同演化算法是近年来计算智能研究的热点。它运用生物协同演化的思想,通过构建两个或者多个种群,建立它们之间的合作关系。两个或多个种群通过相互合作来提高各自的性能,适应复杂系统的动态演化环境以及大规模演化环境,从而达到种群优化的目的。主要介绍了合作型协同演化算法的研究状况以及国内外研究进展,详细介绍了它的基本结构及对应的研究、基本算法及一些新兴算法,同时介绍了一些在现实生活中的应用,展望了合作型协同演化算法的发展前景。

**关键词:**合作型协同演化算法;问题分解;子空间相关性

**中图分类号:**TP18

**文献标志码:**A

**doi:**10.3969/j.issn.1007-130X.2014.04.018

## Research overview of cooperative coevolutionary algorithms

ZHANG Kai-bo, LI Bin

(Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027, China)

**Abstract:** Cooperative coevolution algorithm is a hot research topic in computational intelligence in recent years. Inspired by the principle of natural selection, cooperative coevolution algorithm constructs two or more groups and establishes the cooperative relationship among the groups. Two or more groups cooperates together to improve their performance, be adapted to dynamic evolutionary circumstance of complicated systems and large-scale evolutionary circumstance, thus achieving the goal of population optimization. The research state and advances of cooperative coevolution algorithms in domestic and foreign are discussed and surveyed. The paper introduces the three main aspects of cooperative coevolution algorithm: basic structure and corresponding studies, basic and improved algorithms, and some applications in the real life. Finally, research prospects are indicated.

**Key words:** cooperative coevolution algorithm; problem decomposition; subspace correlation

### 1 引言

合作型协同演化算法是近年来计算智能研究的热点。它基于 divide-and-conquer 思想,求解问题主要分为以下三步:(1)问题分解:将大规模问题分解成若干小规模子问题;(2)子问题求解:用一种特定的演化算法求解每一个子问题;(3)子问题合并:合并各个子问题的解,构成原问题的解。

合作型协同演化算法最早是由 Potter A M<sup>[1]</sup>结合 GA 算法提出的,即 CCGA。随着 CCGA 的提出,对于合作型协同演化算法的研究逐渐分为以下三类:(1)对于合作型协同演化算法结构的探究。包括问题分解、子空间的相关性、协同个体适应度值分配以及群体多样性等。(2)将合作型协同演化算法结构和不同的演化算法或者算子相结合,得到更多的新的合作型协同演化算法。(3)运用合作型协同演化算法解决不同类型的问题,包括数值优

\* 收稿日期:2012-07-12;修回日期:2012-12-19

基金项目:国家自然科学基金资助项目(61071024, U0835002);教育部基本科研业务费专项资金资助项目

通信地址:230027 安徽省合肥市蜀山同区潜山路 209 号华邦光明世家 2# 801

Address: Room 801, Building 2, Huabang Guangmingshijia, 209 Qianshan Rd, Shushan Distric, Hefei 230027, Anhui, P. R. China

化、动态优化问题、对偶系统等。

在后文的叙述中,CC (Cooperative Coevolution)表示合作型协同演化。

## 2 合作型协同演化算法结构研究

合作型协同演化算法是普通演化算法的拓展。随着自变量维数的增加,普通演化算法的性能急剧下降。因此,合作型协同演化算法采用分而治之的方法,通过将自变量分解到两个或多个子空间中去,子空间之间通过协作来提高各自的性能,从而达到种群优化的目的。本文将从CC算法结构、一个个具体的CC算法以及CC算法的应用三个方面分别介绍CC算法。

### 2.1 问题分解

Potter A M 在 CCGA<sup>[1]</sup>中使用的是 relaxation 方法,主要思想是为了优化一个  $n$  维自变量的问题,保持  $n-1$  维变量不变,只优化剩下的 1 维向量,即单维分解。这种方法也被用在文献[2]中。在 CCGA 的基础上,Potter A M 进行了进一步的研究<sup>[3,4]</sup>。他发现了两种新的问题分解方法:(1)文献[3]首先让第一个子空间进行演化,直到演化结果改进小于阈值;然后让第二个子空间和第一个子空间并行演化。这里其他的子空间选择的是当前最优个体协同演化,直到演化结果改进小于阈值;然后添加第三个子空间进行并行演化,这样直到最后一个子空间。(2)基于对子空间和小生境的进一步研究<sup>[4]</sup>,Potter A M 发现了基于小生境的新的问题分解方法。随着对问题分解方法的进一步探究,Potter A M 认为问题分解必须面对四个问题<sup>[5]</sup>:(1)子空间能否定位和覆盖多个小生境环境?(2)一般来说,子空间能否演化到一个合适的水平层次上?(3)当子空间的数目与作用发生变化时,CC 能否很好地适应这个变化,对结果影响大不大?(4)能否得到子种群的合适的数目?针对这四个方面的问题,Potter A M 和 De Jong 根据一系列实验作了回答:(1)每个子空间定位一个或是两个小生境,让其他的子空间定位其他的小生境;两个或是更多的子空间能够覆盖一个普通小生境;但是有些子空间对于定位和覆盖小生境没有明显作用。(2)随着子空间的数目增加到超过小生境的数目时,结果越来越好,然而需要耗费更多的计算代价。(3)随着迭代次数的增加,子空间的作用将稳定下来。随着子空间数目和作用的改变,结果会呈

现出通适性,能够很好地适应子群体数目和作用的改变。(4)合适数目的子空间是能够得到的。

在 Potter A M 的研究工作的基础之上,多种不同的问题分解的方法相继产生。其中包括将问题的自变量均分为两半,每一半作为一个子空间进行优化的折半分解<sup>[6]</sup>;每一维自变量都有相同的几率被分配到任意一个子空间中去的随机分组<sup>[7]</sup>;在随机分组中,随着随机分组次数的增加,多个相关变量被分到一个子空间的概率大大增加<sup>[8]</sup>;保证正在优化的子空间中的自变量是当前方差最大的自变量的方差分组<sup>[9]</sup>。

随着对自变量之间相关性研究的深入,不仅仅可以将相关程度较高的自变量合并到一个子空间<sup>[10]</sup>,同时可以将一个子空间分解<sup>[11,12]</sup>。当自变量之间独立时,子空间分解;当自变量之间相关时,子空间合并。当搜索局部最优区域,子空间合并比子空间分解能够更有效地逃离局部最优。当搜索全局最优区域时,子空间分解能够更快地找到全局最优。如果合并后的子空间在一定时间内不能让结果变好,该子空间会被分解。为了更有效地解决不可分问题,一系列新的基于自变量相关性的分组方法相继被提出。Ray T<sup>[13]</sup>通过计算自变量之间的相关系数,提出对自变量进行分组的方法——CCEA-AVP。Chen Wen-xiang<sup>[14]</sup>提出了一种新的将相关自变量分配到同一个分组中去的分组方法——CCVIL。

近年来,新的分组方法层出不穷。动态分组规模<sup>[15]</sup>方法是通过一种反馈学习机制动态调整分组的规模。Chandra R 等<sup>[16]</sup>基于神经元的功能性质提出一种新的子空间分解的编码方案(NSP),虽然相比 CoSyNE<sup>[17]</sup>,它的模块化层次低,但是这让它初始化阶段计算代价较小。Shi Min 等<sup>[18]</sup>采用盲目分解的方法解决问题。

### 2.2 子空间相关性

文献[19,20]在以下几个方面对子空间的相关性做了细致的分析和深入的研究。

#### 2.2.1 协同选择压力

协同选择压力是指如何选择每个子空间中的个体进行协同演化。根据个体选择的贪婪程度可以分为五类,分别是:(1)每个子空间只选择最优个体;(2)每个子空间只选择随机个体;(3)每个子空间只选择最差个体;(4)每个子空间选择上述三种个体中的任意两种个体;(5)每个子空间选择上述三种个体。实验发现,只要子空间选择了最优个体

进行协同演化,结果就会比较好。选择两种个体结果(包括最优个体)和选择三种个体结果差不多。当不知道变量相关性时,使用(最优+随机)这种个体选择方法会成为第一选择。根据个体选择顺序可以分为以下三类:(1)顺序演化:每次迭代只选择一个子空间进行演化,只更新这个子空间;(2)并行演化:每次迭代所有的子空间都进行演化,等所有的子空间演化结束再统一进行更新;(3)重排方法:随机打乱每个子空间中的个体顺序,然后让个体配对分组。实验发现,并行方法比顺序方法效果好,重排方法在比较复杂的问题上结果更好。

### 2.2.2 协同个体规模

当选择一个或是两个协同个体,结果已经很好了。对于变量相关度很大的复杂问题,协同选择压力的重要性远远小于协同个体规模,如果计算允许的话,增加协同个体规模,通常会使得结果更好。Panait L 等<sup>[21]</sup>用一种简单的方法随时间改变协同选择个体的数目。在演化早期,用多个协同选择个体确定最有希望的空间。这是因为初始化个体之间有较高的多样性。当群体开始收敛时则减少个体。因为当其他群体开始收敛的时候,选用更多的协同个体不能明显改进解的质量。最终结果表明,随时间减少协同个体数目比固定协同个体数目结果要好。

### 2.2.3 子空间的适应度值分配

任一个子空间中的不同个体有不同的适应度值,这些不同的适应度值必须通过某种方法得到该子空间一个特定的适应度值。这里有三种方法:(1)将子空间所有个体中最好的个体对应的适应度值作为该子空间的适应度值;(2)将子空间所有个体中适应度值的平均值作为该子空间的适应度值;(3)将子空间所有个体中最差的个体对应的适应度值作为该子空间的适应度值。实验结果表明方法(1)的结果最好。

### 2.2.4 子空间的优化策略和时间策略

子空间优化策略主要包括同步策略和异步策略<sup>[22]</sup>。同步策略是指只有当所有的子空间都演化结束之后才更新全局群体;异步策略是指每演化一个子空间就更新一次全局群体。实验证明异步策略和同步策略结果基本差不多,在部分问题上异步策略结果会稍好。子空间的演化时间的不同对结果也会产生一定的影响<sup>[21]</sup>。子空间演化时间的改变,可能会改善结果,也可能让结果变差。

## 2.3 协同个体适应度值分配

协同个体不同的子空间都有不同的适应度值,这些不同的适应度值必须通过某种方法得到该协同个体特定的适应度值。这里有三种方法<sup>[19,20]</sup>:(1)将个体最好的协同子空间对应的适应度值作为个体的适应度值;(2)将个体的协同子空间的适应度值的平均值作为个体的适应度值;(3)将个体最差的协同子空间对应的适应度值作为个体的适应度值。实验结果表明方法(1)的结果最好。为了逃离局部最优,一些新的适应度值分配方案相继被提出,包括全局适应度最优值(全局适应度用类似 CCGA2<sup>[1]</sup>的方法得到)和局部适应度最优值(局部适应度是对每个子空间的单独特征来计算,只有训练数据中目标函数对应的部分特征会被用于计算,其他特征不会对局部适应度有贡献)取平均值<sup>[23]</sup>;全局适应度最优值和局部适应度最优值平均值根据一定的概率产生个体适应度<sup>[24]</sup>。

## 2.4 群体多样性

增强群体多样性,可以解决合作型协同演化中早熟收敛的问题。增强多样性的方法<sup>[25]</sup>主要包括增加群体规模、使用基于小生境的模型和岛模型、近邻搜索等。

## 2.5 其他方面

对于 CC 算法结构的研究和分析除了以上四个方面外,在其他的一些方面也相继展开。例如,基于参考共享协同模型,同时用 Pareto 支配原则来衡量个体的表现,从而解决显性基因的问题<sup>[26]</sup>;CCGA 在单目标优化问题的多适应度衡量上出现了三种新的方法<sup>[27]</sup>:贪婪、非支配排序以及 even-distributed 排序。同时,适度的变异能够明显提升 CCEA 对可分解问题的处理<sup>[28]</sup>,自适应变异的引入<sup>[29]</sup>则让性能进一步改进。子空间之间信息交互频率对于结果会产生一定的影响<sup>[30]</sup>,降低交互频率会让结果更精确,但是会导致早熟收敛。采用动态群体规模、精英策略等方法<sup>[31,32]</sup>则能进一步改进 CCEA 的性能。

# 3 合作型协同演化算法

## 3.1 合作型协同演化算法结构和演化算法结合

随着 Potter A M 将 GA 作用于合作型协同演化算法结构并提出了 CCGA<sup>[1]</sup>开始,不断有新的演化算法或者结合新的演化算子的基本演化算法应用于合作型协同演化算法结构的框架中,出现了很

多新的合作型协同演化算法。CC 算法结构和演化算法相结合,主要包括以下几类。

### 3.1.1 CC 和基本演化算法相结合

CCGA 对解空间进行分解,如果自变量之间存在相关性会导致早熟收敛,同时 CCGA 对参数存在严重的依赖。为了解决这两个方面的问题,更多的基本演化算法和 CC 相结合形成了多种不同的 CC 算法。

演化规划应用于 CC 框架中去得到的 FEPC<sup>[2]</sup>极大地加快了 CC 算法解决大规模问题时的收敛速度。

演化策略和 CC 结合得到的 CCES<sup>[33]</sup>对于解决自变量之间具有相关性的高维问题效果更好。

CPSO<sup>[34]</sup>是由 CC 算法结构和 PSO 算法结合得到的。因为标准 PSO 应用到 CC 中去得到的 CPSO-S<sub>k</sub> 仍然可能陷入局部最优,因此文献[34]提出了 CPSO-H<sub>k</sub>。在 CPSO-H<sub>k</sub> 中,CPSO-S<sub>k</sub> 和标准 PSO 交替运行,共享它们的全局最优解信息,最终找到全局最优解。

差分演化算法结合 CC 框架得到的 CCDE<sup>[35]</sup>尝试将解空间分解成每一维自变量一个子空间。实验结果表明这种分解并不总是比解空间分为两部分的结果好。

### 3.1.2 群体觅食算法和 CC 结合

近年来随着群体觅食算法的兴起,不断有 CC 与群体觅食算法相结合,扩展 CC 算法的范围,人工蜂群算法就是其中一种。人工蜂群算法 ABCA (Artificial Bee Colony Algorithm)和 CC 算法框架结合得到 CABCA<sup>[36]</sup>,改善了 ABCA 的实验结果、收敛速度和鲁棒性。

ABCA 是 Karaboga D<sup>[37]</sup>于 2005 年通过模拟蜜蜂群的智能采蜜行为提出的一种群智能随机优化算法。在该算法中蜜蜂根据各自的分工进行不同的活动,并实现蜂群信息的共享和交流,从而找到要求解的问题的最优解。

在 ABCA 中,人工蜂群包含采蜜蜂、观察蜂和侦察蜂三个组成部分,每个蜜源的位置代表优化问题的一个可能解,蜜源的花蜜量对应于相应解的质量或适应度。

首先,ABCA 随机产生初始群体即个初始解(为采蜜蜂数也为蜜源数目)。每个解为一个维的向量( $d$  为优化自变量的个数)。经过初始化之后,首先是采蜜蜂工作阶段:采蜜蜂根据记忆中的位置信息产生一个变化的位置并检查新位置的花蜜量。

若新位置的花蜜量比原来的多,则该蜜蜂记住新位置并忘记原位置。等到所有的采蜜蜂完成搜索过程后,采蜜蜂将蜜源信息与观察蜂共享进入到观察蜂工作阶段。观察蜂根据与不同位置的花蜜量相关的概率选择一个蜜源位置,像采蜜蜂那样对记忆中的位置做一定的改变,并检查新候选位置的花蜜量。若新位置优于记忆中的位置,则用新位置替换原位置。一个观察蜂选择某个蜜源的概率为:

$$p_i = fitness_i / \sum_{j=1}^{n_e} fitness_j \quad (1)$$

其中,  $fitness_i$  是基于第  $i$  个个体的适应度值(花蜜量)。根据记忆位置产生一个新的位置:

$$v_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}) \quad (2)$$

其中,  $k \in \{1, 2, \dots, s_n\}$  和  $j \in \{1, 2, \dots, d\}$  是随机选择的下标,  $k \neq i$ ;  $\phi$  是在  $[-1, 1]$  的随机数。控制了  $x_{ij}$  邻域内新解的产生并代表蜜蜂对两个可视范围内蜜源位置的比较。随着参数的  $x_{ij} - x_{kj}$  值越来越小,对于位置  $x_{ij}$  的扰动也越来越小。

假设一个蜜源经过限定的循环次数之后不能被改进,则该蜜源的采蜜蜂转变成成为侦察蜂。该蜜源的位置会被侦察蜂在解空间内随即发现的新位置替代。侦察蜂发现新位置的操作为:

$$x_i^j = x_{\min}^j + \text{round}(0, 1)(x_{\max}^j - x_{\min}^j) \quad (3)$$

和 ABCA 相比,CABCA 处理问题的结果更好,收敛速度更快,鲁棒性也更好。

### 3.1.3 多种演化算法和 CC 结合

Vanneschi L<sup>[38]</sup>根据 GP 能够优化结构、GA 能够优化数值的特点,提出了基于 GA 和 GP 混合的协同演化:根据 GA、GP 不同的运行阶段主要有四种结构:

(1)基本算法:GA 和 GP 交替进行。

(2)延迟算法:在 GA 和 GP 交替进行之前,GP 先单独迭代一段时间,找到较好的优化结构。

(3)CINI 算法:首先让 GP 单独迭代直到结果没有显著的改进,然后进行 GA 和 GP 交替优化。

(4)AUTO 算法:根据 GA 和 GP 对于结果的改进大小决定到底是运行 GA 还是 GP。

实验结果表明,以上四种算法中 CINI 算法性能最好。

### 3.1.4 层次 CC 算法

Maniadakis M<sup>[39]</sup>提出并分析了层次合作型协同演化算法(HCCE),HCCE 不仅突出不同层次的局部部分的独立的作用,而且强调它们作为一个整体系统的作用。局部部分由自身动态性驱使,满足

每个部分的特殊的目标;同时,高层次的演化过程探索完整的结构,协调低层次部分的演化。

### 3.1.5 微演化算法与 CC 框架相结合

两者结合可以得到快速收敛的微协同演化算法,主要包括:基于 DE 算法的 CC-Micro-DE<sup>[40]</sup>以及基于 PSO 算法的 CC-Micro-PSO<sup>[41]</sup>。

微演化算法和普通演化算法的区别是群体规模较小,这会限制它的搜索能力,特别是在复杂环境中的搜索能力。同时,群体规模小会导致群体早熟收敛,在早期迭代中降低了群体的多样性。对于 CC-Micro-DE 和 CC-Micro-PSO,群体规模为 6 是一个合适的选择。通常  $p = n(\text{自变量维数})/N(\text{群体规模})$  暗示一个算法解决给定问题的困难度。大多数情况下  $p$  的值越大,微演化算法越难解决问题。

比较基本的微演化算法,微演化算法和 CC 框架结合后的算法解决高维问题的能力变强,性能也更好,较为有效地克服了微演化算法解决高维问题时遇到的早熟收敛问题。

### 3.1.6 CC 结构和 Memetic 算法相结合

CC 结构和 Memetic 算法相结合用于解决神经网络问题。主要包括结合 CC 结构以及通过交叉的局部细化的 XLCC 算法<sup>[42,43]</sup>,基于交叉的局部细化的局部搜索可以在合适的时机产生优势个体,提升算法性能。XLCC 中,不是对每个子群体都进行局部搜索,而是当一次循环完成之后,再对子群体合并后的个体进行局部搜索。每次循环完成之后,所有子群体中的最优个体合并成一个 meme。然后用局部搜索的方法对该 meme 进行交叉操作,用交叉后的 meme 代替每个子群体中的最差个体。

XLCC 的性能比标准的 CC 算法要好,这为进一步利用其他的局部搜索策略打下了良好的基础。但是, XLCC 最大的缺点在于参数设置的计算代价太大。

### 3.1.7 CC-Rank

CC-Rank<sup>[44]</sup>用并行 CC 学习排名问题。它用两阶段的方法从训练数据集中学习排名函数。首先,CC-Rank 进入问题分解阶段:随机在整个特征空间初始化用树结构表示的  $L$  个解之后,将每一个解分解成  $N$  个子树,这样就生成了  $N$  个群体,每个群体包含  $L$  个个体。问题分解阶段结束后,进入循环的演化阶段。在该阶段, $N$  个群体并行地协同演化。每个群体中适应度最好的个体被选

出进行协同。每次循环结束后,并行操作将暂停,每个群体中被选出的适应度最好的  $N$  个个体将合并构成完整解,并且将这个解放入一个解决方案池作为排名函数的候选解。当演化阶段的循环全部结束的时候,验证数据集将从所有的候选解中选取最好的解作为最终的排名函数。

CC-Rank 成功地运用于复杂优化问题的解决。与当前最好的算法相比,在解决排名问题的时候能够同时提高算法的精确度和效率。

### 3.1.8 pCCEA

Bucci A 等<sup>[45]</sup>提出一种用 Pareto 支配机制的 CCEA-pCCEA。下面以两个种群为例介绍基于非支配排序的 Pareto 协同演化算法。它的具体步骤如下:

(1)根据问题域划分为两个种群(种群 A 和种群 B),所有的种群构成一个进化系统。

(2)初始化两个种群,在两个种群中分别选择最优的染色体,并随机选择另外一条染色体作为两个种群各自的种群代表。

(3)对种群 A 的所有个体和种群 B 的代表个体采用遍历组合法,构成新的个体。

(4)对于多个目标的每个目标函数,分别计算所有个体的目标函数值。

(5)计算个体的非支配水平和密集距离,实现非支配排序。

(6)取前  $n$  个经过非支配排序的个体,将其分解为两个种群的下一代父个体。

(7)根据目标函数值选择两个种群的代表,对两个种群的染色体分别进行独立的演化操作产生子代种群。

(8)重复(1)~(7),直到满足迭代终止条件为止。

pCCEA 对个体之间信息差异很敏感,相比于标准的 CCEA,对于解决相关性问题的性能更好。

### 3.1.9 CC 结构和新兴演化算法结合

近年来,随着对合作型协同演化算法的深入研究,一些新兴的演化算法应用到 CC 的框架中去,例如,CC 算法框架和文化算法的结合<sup>[46]</sup>,和分布估计算法 EDA 的结合<sup>[47]</sup>。

文化算法<sup>[45]</sup>的主要思想是明确地从进化种群中获得求解问题的知识,并用于搜索过程。它是由群体空间和信念空间两部分组成。群体空间是基于传统群体的进化。信念空间是基于信念文化的进化,用于知识经验的形成、存储和传播。两个空

间之间既相互独立又有着联系。文化算法基本步骤如下:

- (1) 初始化种群空间和信念空间。
- (2) 演化种群空间的个体。
- (3) 搜集种群空间中优秀个体的经验知识,用于更新信念空间。
- (4) 利用解决问题的知识指导群体空间的进化。
- (5) 根据规则从新生成个体中选择一部分个体作为下一代个体的父个体。
- (6) 重复(2)~(5),直到达到迭代终止条件。

文化算法和 CC 结合能够有效地加快文化算法的收敛速度,而且知识空间的信息迁移也比文化算法中个体迁移策略性能要好。

分布估计算法<sup>[47]</sup>作为一种新型的进化算法,主要有以下几方面的特点。首先,从生物进化的数学模型上来看,分布估计算法与传统进化算法不同:传统进化算法是基于对种群中的各个个体进行遗传操作(交叉、变异等)来实现群体的进化的,是对生物进化“微观”层面上的数学建模;而分布估计算法则是基于对整个群体建立数学模型,直接描述整个群体的进化趋势,是对生物进化“宏观”层面上的数学建模。其次,分布估计算法给人类解决复杂的优化问题提供了新的工具,它通过概率模型可以描述变量之间的相互关系,从而对解决非线性、变量耦合的优化问题更加有效。

分布估计算法(EDA)和 CC 结合得到的 CE-DA 能够有效地逃离局部最优解,特别是在协同个体数目较少的时候,但是和标准 EDA 相比,CE-DA 耗费的计算代价更大。

### 3.2 合作型协同演化算法结构和改进演化算子结合

还有一些算法是通过已有的算法中的算子进行修改或增减,然后结合 CC 结构来改善性能。

#### 3.2.1 随机分组策略和自适应权值策略

杨振宇等<sup>[7]</sup>采用随机分组策略和自适应权值策略改进 CCDE,提出了新的更加有效的 DECC<sup>[7]</sup>。随机分组策略的“随机”指的是每个变量被分配到任何一个子问题中的机会相同,它增加相关的变量分在同一分组的概率,算法在优化的过程中会自动更新分组结构;自适应加权机制在 CC 框架中每个优化周期结束后,对每个组构成的决策变量分量增加一个权值,然后使用指定演化算法优化这  $m$  个权值构成的权值向量,从而进一步提高当前解的质量。自适应加权机制对于处理不可分

问题效果很明显。

随后李晓东等<sup>[48]</sup>用这两种策略改进 CPSO,得到 CCPSO。实验结果表明,这两种策略能够明显改进先前的 CC 算法解决大规模问题的性能,CCPSO 的性能较之 CPSO 也有了大幅的提升。

然而,DECC 算法框架中仍然存在一个在实际应用中很难确定的参数,即分组大小。这主要是因为最优的分组大小往往是问题相关的,一般来说子问题规模越小越容易优化,所以对于可分问题,分组大小越小越好;而另一方面,对于不可分问题,则分组大小越大越好,因为它能提供更高的概率将存在相关性的决策变量分在同一组。另外,即使对于同一问题的不同优化阶段,最优的分组大小也不易确定,比如在优化初期,小的分组大小能更快地帮助算法找到一些有潜力的搜索子空间;而在优化后期,大的分组大小能在子问题中包含更多的全局信息,这些全局信息对算法收敛到全局而非局部极值点很有用。所以,如果能根据问题特性及不同的优化阶段自动确定合适的分组大小,将有望使算法发挥出更好的性能。

因此,杨振宇等<sup>[15]</sup>采用动态分组规模的方法进一步改进 DECC,提出 MLCC 算法。在 MLCC 算法中,首先根据不同分组大小设计一组问题分解器,然后将它们置于一个问题分解器池中,池中的每个分解器都代表了它所能涵盖的决策变量相关层次。当使用 MLCC 算法优化一个给定的问题时,演化过程仍然被划分为若干个演化周期,在每个周期的开始,算法首先根据其历史性能表现选择一个问题分解器,之后该问题分解器被用来将原大规模问题分解成若干个子问题;然后 MLCC 算法将采用一种指定的演化算法分别对每个子问题求解,当每个演化周期结束后,选择使用的问题分解器的性能记录将用它在当前周期的性能表现替代。根据这种工作机制,MLCC 算法可以在演化过程中根据需要自动选择合适的问题分解器,达到分组大小层面的自适应。

李晓东等<sup>[49]</sup>用动态分组规模的策略改进 CPSO,得到 CCPSO2。

#### 3.2.2 根据自变量的方差大小进行分组

王瑜<sup>[9]</sup>提出的 VP-DECC<sup>[9]</sup>中,方差较大的自变量将被选出构成当前要优化的子群体。首先计算  $d$  维自变量的方差,按大小排序,按从大到小的顺序分别将个体分配到子群体中去,保证正在优化的子群体中的自变量是当前方差最大的一批自变

量。VP-DECC 能够进一步提升 DECC 处理复杂高维问题的能力。

### 3.2.3 基于贡献的分组方法

基于贡献的 CC 算法——CBCC<sup>[50]</sup> 基于所有子空间对全局适应度的贡献选择一定数目的子空间进行进一步优化,减轻了不同子空间之间的不平衡,允许更有效地利用计算资源。CBCC 的算法流程如下:

(1) 随机初始化群体,计算群体适应度,得到最优适应度和最优个体。

(2) 根据每个子空间对全局适应度的贡献,按照循环赛模式选择子群体。

(3) 选择最大贡献的子空间进一步优化。

(4) 重复(2)和(3),直到达到终止条件。

CBCC 的主要缺点是对于局部改变反应慢,且依赖演化早期的信息积累,这需要用自适应的方法维持局部贡献和全局贡献之间的平衡。

### 3.2.4 一些变异因子作用于演化算法,和 CC 结构相结合

一些变异因子作用于演化算法,和 CC 结构相结合,同样能够改善 CC 算法的性能。

自适应的方法已经广泛地应用于标准演化算法中改善算法性能,因此一种 CCEA<sup>[51]</sup> 用自适应变异因子去更新搜索方向和搜索步长,极大地改善了标准 CCEA 的性能。

Aichour M<sup>[52]</sup> 第一次将选择变异因子作用于经典 GP,设计出 CCGP 来解决符号回归问题。主要思想是用变异去关注没有正确演化的基因组区域。选择表现最差的子树进行变异,它将被新的随机子树替代。子树的表现是通过局部演化来衡量的,局部演化衡量子树的表现,是将子树用一个自然数替代计算这个修改个体的适应度,它和原个体的适应度差值就是子树的表现。

### 3.2.5 PCA 策略用于 CC 结构

Omidvar M<sup>[53]</sup> 用 PCA(主成分分析)降低问题维数和用演化过程中得到的信息改进变量分组的概率这两种策略改进 DECC。PCA 的主要步骤如下:

(1) 减去平均值:该阶段所有的数据项减去群体的平均值,得到平均调整数据。

(2) 基于给定的群体计算协方差矩阵。

(3) 计算协方差矩阵的特征值和特征向量。

(4) 将协方差矩阵的特征向量按特征值从大到小排序,特征值最高的特征向量是群体的主成分。

因此通过选择前  $n$  个最大的特征向量可以降低数据集的维数。

(5) 创建新的数据集。即(4)中得到的特征向量集乘以(1)中得到的调整数据集。

PCA 能够计算特征向量任意两维之间的比率,如果该值在提前设定的范围中,这两个自变量将放入同一个子空间。这种方法极大地提升了 DECC 解决高维相关性问题的性能。

### 3.2.6 修改已有的算法中的部分算子

Lee Ching-Hung<sup>[54]</sup> 修改 CCDE 中 DE 的变异策略增强多样性,运用自适应缩放因子策略进一步提升算法精确度和收敛速度。同时用 BP 方法进行局部搜索,有效地克服早熟收敛的缺点,改善 CCDE 的性能。

## 3.3 非 CC 算法结构的解决大规模问题的演化算法

还有一些其他的算法并不是利用 CC 算法结构去有效地解决大规模问题,例如,SOUPDE 算法<sup>[55]</sup> 的主要思想是重排策略结合自适应 DE 中的扩展因子;基于动态维数交叉<sup>[56,57]</sup> 的 PSO 算法;利用自适应协方差矩阵的旋转不变性有效地解决不可分问题的 CMA-ES 算法<sup>[58]</sup>,不仅允许每一维自变量对应不同的变异步长,而且用一个旋转矩阵保证变异沿着解决不可分问题的优化目标的方向前进。

## 4 合作型协同演化算法的应用

合作型协同演化算法已经广泛应用于诸多领域,如电子工程、模式识别和交通运输规划等。

### 4.1 CC 算法解决网络问题

Potter A M 将 CC 算法成功地运用于概念学习和神经网络结构中去<sup>[59]</sup>。Garcia-Pedrajas N 等<sup>[60]</sup> 提出一种基于协作型协同演化的演化人工神经网络的方法,称为 COVNET。Kimura S 等<sup>[61]</sup> 将 CC 算法用于推断大规模基因网络的 S-system 模型。Chandra R 等<sup>[62]</sup> 用 Memetic 框架结合 CC 结构、用 NSP(Neuron Based Subpopulation)将网络分解到神经元级别。Ruela A S 等<sup>[63]</sup> 将 CC 算法用于 WSN 的物理拓扑的快速设计,目标是设计出能够体现高聚类系数和小的平均最短路径长度的物理拓扑,改善基于路由的最小化能量消耗和延迟的树结构。Carvalho A<sup>[64]</sup> 用 CCGA 算法学习贝叶斯网络的结构,将问题分解为两个子部分,一个寻找节点的顺序,一个寻找优化连接矩阵。Supu-

domchok S 等<sup>[65]</sup>在 CC 结构中通过研究最大化 ATP 以及最大化所有光通量值之和的 ATP 来进行枯草杆菌代谢网络的流量平衡分析。其中最大化 ATP 是传统的线性目标函数,最大化所有光通量值之和的 ATP 是非线性目标函数。

#### 4.2 CC 算法作用于模型设计 and 应用问题

曹先彬等<sup>[66]</sup>提出了一种基于协同演化的行人检测系统。滕弘飞等<sup>[25]</sup>提出基于对偶系统的协同演化的卫星模型布局设计。孙伟等<sup>[67]</sup>提出将 CCGA 用于全断面岩石掘进机圆盘刀具布置设计。Nebti S 等<sup>[68]</sup>用 CC 改善径向基函数神经网络(RBFNN)用于识别手写阿拉伯数字。Izzah 等<sup>[69]</sup>用 CC 算法解决云计算中 SaaS(软件经营)安置问题,即 SaaS 组成部分之间的交互以及 SaaS 数据部分之间的交互。梁昌洪等<sup>[70]</sup>基于 CC 提出一种新的分解策略和协同方法解决无功优化问题。分解策略是基于电压母线的敏感性和动力系统中母线的衍生电力距离。Sim 等<sup>[71]</sup>运用 CCGA 解决炼油厂调度的全局优化。Wu A 等<sup>[72]</sup>用一种基于遗传性的去耦自适应 CC 算法进行电力电子电路优化设计。周翔等<sup>[73]</sup>提出一种结合了 GA 和离散 PSO 的 CC 算法解决模糊处理时间和模糊交货日期条件下的多处理路径的车间作业调度。

#### 4.3 CC 算法解决其他问题

Panait L<sup>[74,75]</sup>提出了一种在协作型协同演化框架下将传统演化计算技术应用到多智能体行为学习领域的方法。Au C K<sup>[76,77]</sup>用 CC 算法解决动态优化问题。Nema S 等<sup>[78]</sup>结合了粒子群算法、梯度搜索和 CC 框架平衡探索和搜索,快速、准确、高效地解决约束优化问题。Boonlong K 等<sup>[79]</sup>用 CCGA 鉴定粒子图像测速中的错误速度矢量。李伟<sup>[80,81]</sup>用基于 CC 结构的免疫诊断框架定义及分析误差诊断,改进了误差诊断方法的精确度,克服动态环境中误诊的缺陷,由免疫细胞之间的相互促进和抑制的方法来实现误差诊断。

## 5 结束语

合作型协同演化算法与普通演化算法的根本差别在于,合作型协同演化算法中一个个体的适应度的计算是通过和其他个体相互交互的过程中合作进行的。本文从合作型协同演化算法结构(包括问题分解、子空间的相关性、适应度值分配以及群体多样性等几个重要方面)的探究、具体的一个

合作型协同演化算法的介绍以及利用合作型协同演化算法解决不同类型的问题这三个方面介绍了合作型协同演化算法目前的研究进展,从目前的发展情况来看,作者认为合作型协同演化算法在以下几个方面有待进一步深入研究。

#### 5.1 问题分解策略

对于合作型协同演化算法结构,除了自然分解之外,问题分解并没有普遍性的比较好的分解方法。近年来,对于问题分解策略的研究是一个热点,但是如何能够得到比较理想的结果、比较好的分组,还需要进一步研究。

#### 5.2 子空间相关性

当前对于子空间相关性的研究,主要是通过统计计算两个子空间的相关系数,然后对子空间进行分解和合并。但是,除了统计计算相关性之外,是否还存在其他的合理的方法得到子空间的相关性关系亟待探索。

#### 5.3 合作型协同演化算法结构和高维优化算法相结合

目前合作型协同演化算法主要是将合作型协同演化算法结构和基本的演化算法相结合。除了合作型协同演化算法结构之外,还有一些别的高维优化算法也能够较为有效地解决大规模优化问题,是否能将二者结合起来,更加有效地解决高维优化问题,也是当前的研究热点。

#### 5.4 合作型协同演化算法结构和新兴演化算法相结合

近年来涌现了很多新兴的演化算法,主要包括觅食算法(细菌觅食算法、蜂群觅食算法、人工蜂群算法等)、分布估计算法等。对于如何更好地将合作型协同演化算法结构和新型演化算法相结合,提升这些算法的效果,这方面还需要深入的探索。

#### 5.5 合作型协同演化算法的应用

合作型协同演化算法应用主要涉及到数值优化、调度问题、博弈策略设计、模式识别以及数据挖掘等,在这些领域合作型协同演化算法有着极为广阔的研究前景,等着我们一步步探究。

#### 参考文献:

- [1] Potter A M, De Jong K A. A cooperative co-evolutionary approach to function optimization [C]//Proc of the 3rd International Conference on Parallel Problem Solving from Nature, 1994:249-257.
- [2] Liu Y, Yao X, Zhao Q, et al. Scaling up fast evolutionary

- programming with cooperative coevolution [C]//Proc of the 2001 Congress on Evolutionary Computation, 2001; 1101-1108.
- [3] Potter A M , De Jong K A. Evolving neural networks with collaborative species [C]//Proc of 1995 Summer Computer Simulation Conference, 1995;340-345.
- [4] Potter A M, De Jong K A, Grefenstette J J. A co-evolutionary approach to learning sequential decision rules [C]//Proc of the 6th International Conference on Genetic Algorithms, 1995;366-370.
- [5] Potter A M , De Jong K A. Cooperative coevolution: An architecture for evolving co-adapted subcomponents [J] . Evolutionary Computation, 2000, 8(1):1-29.
- [6] Shi Y, Teng H, Li Z. Cooperative co-evolutionary differential evolution for function optimization [C]//Proc of the 1st International Conference on Natural Computation, 2005; 1080-1088.
- [7] Yang Zhen-yu, Tang Ke, Yao Xin. Differential evolution for high dimensional function optimization [C]//Proc of IEEE Congress on Evolutionary, 2007;1.
- [8] Omidvar M N, Li X, Yang Z, et al. Cooperative co-evolution for large scale optimization through more frequent random grouping [C]//Proc of IEEE World Congress on Computational Intelligence ,2010;1.
- [9] Wang Y, Li B, Lai X X. Variance priority based cooperative co-evolution differential evolution for large scale global optimization [C]//Proc of the 2009 IEEE Congress on Evolutionary Computation, 2009;1232-1239.
- [10] Weicker K, Weicker N. On the improvement of coevolutionary optimizers by learning variable interdependencies [C]//Proc of the 1999 Congress on Evolutionary Computation, 1999;1627-1632.
- [11] Kim M W, Ryu J W. Species merging and splitting for efficient search in coevolutionary algorithm [C]//Proc of the 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2004), 2004;332-341.
- [12] Kim M W, Ryu J W. An efficient coevolutionary algorithm using dynamic species control [C]//Proc of the 3rd International Conference on Natural Computation, 2007;431-435.
- [13] Ray T, Yao Xin. A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning [C]//Proc of 2009 IEEE Congress on Evolutionary Computation, 2009;983-989.
- [14] Chen Wen-xiang, Weise T, Yang Zhen-yu, et al. Large-scale global optimization using cooperative coevolution with variable interaction learning [C]//Proc of Parallel Problem Solving from Nature, 2010;300-309.
- [15] Yang Z, Tang K, Yao X. Multilevel cooperative coevolution for large scale optimization [C] // Proc of the 2008 IEEE Congress on Evolutionary Computation, 2008; 1663-1670.
- [16] Chandra R, Freaan M, Zhang Meng-jie. Building subcomponents in the cooperative coevolution framework for training recurrent neural networks [R]. Technical Report, New Zealand; Victoria University of Wellington, 2009.
- [17] Gomez F, Schmidhuber J, Miikkulainen R. Accelerated neural evolution through cooperatively coevolved synapses [J]. Journal of Machine Learning Research, 2008(9):937-965.
- [18] Min Shi. Empirical analysis of cooperative coevolution using blind decomposition [C]//Proc of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, 2011;141-142.
- [19] Wiegand R P, Liles W C, De Jong K A. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms [C]//Proc of the 2nd Annual Conference Companion on Genetic Evolutionary Computation Conference, 2001;1235-1242.
- [20] Luke S, Sullivan K, Abidi F. Large scale empirical analysis of cooperative coevolution [C]// Proc of the 13th Annual Conference Companion on Genetic and Evolutionary Computation Conference, 2011;151-152.
- [21] Panait L, Luke S. Time-dependent collaboration schemes for cooperative coevolutionary algorithms [C]//Proc of the AAAI Fall Symposium on Coevolutionary and Coadaptive Systems, 2005;1.
- [22] Crăciun C, Nicoară M, Zaharie D. Enhancing the scalability of metaheuristic by cooperative coevolution [C]//Proc of the 7th International Conference on Large-Scale Scientific Computing, 2010;310-317.
- [23] Zhu F, Guan S. Cooperative co-evolution of GA-based classifiers based on input decomposition [J]. Journal of Engineering Applications of Artificial Intelligence, 2008, 21(8): 1360-1369.
- [24] Wang Chao, Gao Jing-huai. A new differential evolution algorithm with cooperative coevolutionary selection operator for waveform inversion [C]//Proc of 2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2011;688-690.
- [25] Teng Hong-fei, Chen Yu, Zeng Wei, et al. A dual-system variable-grain cooperative coevolutionary algorithm; Satellite-module layout design [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(3):438-455.
- [26] Shi M, Wu H. Pareto cooperative coevolutionary genetic algorithm using reference sharing collaboration [C]//Proc of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009;867-874.
- [27] Shi M. Comparison of Sorting Algorithms for Multi-fitness Measurement of Cooperative Coevolution [C]//Proc of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009;2583-2588.
- [28] Jansen T, Wiegand R P. Exploring the explorative advantage of the cooperative coevolutionary (1+1) EA [C]//Proc of the 5th Annual Conference on International Conference on Genetic and Evolutionary Computation, 2003; 310-321.
- [29] Au C-K, Leung H-F. Biasing mutations in cooperative coevolution [C] // Proc of IEEE Congress on Evolutionary Computation, 2007;828-835.
- [30] Popovici E, De Jong K A. The effects of interaction frequency on the optimization performance of cooperative coev-

- olution [C]//Proc of the 8th Annual Conference on Genetic and Evolutionary Computation Conference, 2006;353-360.
- [31] Popovici E, De Jong K. Understanding cooperative co-evolutionary dynamics via simple fitness landscapes [C] //Proc of the 2005 Conference on Genetic and Evolutionary Computation, 2005;507-514.
- [32] Brest J, Zamuda A, Boskovic B, et al. High-dimensional real parameter optimization using self-adaptive differential evolution algorithm with population size reduction [C] // Proc of IEEE Congress on Evolutionary Computation, 2008;2032-2039.
- [33] Sofge D, De Jong K, Schultz A. A blended population approach to cooperative coevolution for decomposition of complex problems[C]// Proc of the Congress on Evolutionary Computation, 2002;413-418.
- [34] van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization [J]. IEEE Transactions on Evolutionary Computation, 2004,8(3):225-239.
- [35] Shi Y, Teng H, Li Z. Cooperative co-evolutionary differential evolution for function optimization [C]//Proc of the 1st International Conference on Natural Computation, 2005; 1080-1088.
- [36] Zou Wen-ping, Zhu Yun-long, Chen Han-ning, et al. Cooperative approaches to artificial bee colony algorithm[C]// Proc of the 2010 International Conference on Computer Application and System Modeling (ICCA SM), 2010;44-48.
- [37] Karaboga D. An idea based on bee swarm for numerical optimization[R]. Technical Report-TR06, Kayseri:Erciyes University,2005.
- [38] Vanneschi L, Mauri G, Valsecchi A, et al. Heterogeneous cooperative coevolution; strategies of integration between GP and GA [C]//Proc of the 8th Annual Conference on Genetic and Evolutionary Computation,2006;361-368.
- [39] Maniadakis M, Trahanias P. Assessing hierarchical cooperative coevolution [C]//Proc of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-07), 2007;391-398.
- [40] Parsopoulos K E. Cooperative micro-differential evolution for high-dimensional problems[C]//Proc of the 11th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2009), 2009;531-538.
- [41] Parsopoulos K E. Cooperative micro-particle swarm optimization [C]//Proc of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009;467-474.
- [42] Chandra R, Freat M, Zhang Meng-jie. A memetic framework for cooperative coevolution of recurrent neural networks [C]//Proc of the 2011 International Joint Conference on Neural Networks (IJCNN 2011),2011;673-680.
- [43] Chandra R, Freat M, Zhang M. A memetic framework for cooperative co-evolutionary feedforward neural networks [R]. Technical Report ECSTR10-22, Wellington;School of Computing and Engineering, Victoria University of Wellington, 2010.
- [44] Wang Shuai-qiang, Byron J G, Wang Ke, et al. CCRank: Parallel learning to rank with cooperative coevolution [C]// Proc of the 25th AAAI Conference on Artificial Intelligence, 2011;1249-1254.
- [45] Bucci A, Pollack J B. On identifying global optima in cooperative coevolution[C]//Proc of the 2005 Conference on Genetic and Evolutionary Computation, 2005;539-544.
- [46] Guo Yi-nan, Cheng Jian, Lin Yong. Cooperative interactive cultural algorithms adopting knowledge migration [C] // Proc of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009;193-200.
- [47] Vo C, Panait L, Luke S. Cooperative coevolution and univariate estimation of distribution algorithms [C] // Proc of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms,2009;141-150.
- [48] Li Xiao-dong, Yao Xin. Tackling high dimensional non separable optimization problems by cooperatively coevolving particle swarms [C]//Proc of IEEE Congress on Evolutionary Computation, 2009;1546-1553.
- [49] Li Xiao-dong, Yao Xin. Cooperatively coevolving particle swarms for large scale optimization [J]. IEEE Transactions on Evolutionary Computation, 2012,16(2):210-224.
- [50] Omidvar M N, Li Xiao-dong, Yao Xin. Smart use of computational resource based on contribution for cooperative co-evolutionary algorithms [C] // Proc of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11), 2011;1115-1122.
- [51] Au C-K, Leung H-f. Guided mutations in cooperative coevolutionary algorithms for function optimization [C] // Proc of the 19th IEEE International Conference on Tools with Artificial Intelligence, 2007;407-414.
- [52] Aichour M,Lutton E. Cooperative co-evolution inspired operators for classical GP schemes [C]//Proc of International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO'07), 2007;169-178.
- [53] Omidvar M N, Li Xiao-dong, Yang Zhen-yu, et al. Cooperative co-evolutionary algorithm for large scale optimization through more frequent random grouping [C]//Proc of the 2010 IEEE Congress on Evolutionary Computation,2009;1-8.
- [54] Lee Ching-Hung, Chung Pei-yuan, Chang Hao-han. A novel hybrid differential evolution algorithm; A comparative study on function optimization [C]// Proc of the International MultiConference of Engineers and Computer Scientists, 2011;1-6.
- [55] Weber M, Neri F, Tirronen V. Shuffle or update parallel differential evolution for large-scale optimization [J]. Soft Computing, 2011,15(11);2089-2107.
- [56] Hu Cheng-yu, Wang Bo. Particle swarm optimization with dynamic dimension crossover for high dimensional problems [J]. Computing Technology and Automation, 2008,28(1): 92-95. (in Chinese)
- [57] Hu Cheng-yu, Yan Xue-song, Li Chuan-feng. Particle swarm optimization with dynamic dimension crossover for high dimensional problems [C] // Proc of ISICA '08, 2008; 750-759.
- [58] Omidvar M, Li X. A comparative study of CMA-ES on

- large scale global optimization[C]//Proc of the 23rd Australasian Joint Conference on Advances in Artificial Intelligence, 2010;303-312.
- [59] Potter M A. The design and analysis of a computational model of cooperative coevolution[D]. Virginia:George Mason University, 1997.
- [60] Garcia-Pedrajas N, Hervás-Martínez C, Muñoz-Pérez J. COVNET: A cooperative coevolutionary model for evolving artificial neural networks [J]. IEEE Transactions on Neural Networks, 2003;14(3):575-596.
- [61] Kimura S, Ide K, Kashihara A, et al. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm[J]. Journal of Bioinformatics, 2005, 21(7):1154-1163.
- [62] Chandra R, Freen M, Zhang Meng-jie. A memetic framework for cooperative co-evolutionary feedforward neural networks[R]. Technical Report ECSTR10-22, Wellington: School of Computing and Engineering, Victoria University of Wellington, 2010.
- [63] Ruela A S, Aquino A L L, Guimaraes F G. A cooperative coevolutionary algorithm for the design of wireless sensor networks Track name: Bio-inspired solutions for wireless sensor networks [C]//Proc of GECCO'11, 2011;607-614.
- [64] Carvalho A. A cooperative coevolutionary genetic algorithm for learning Bayesian network structures[C]//Proc of GECCO'11, 2011;1131-1138.
- [65] Supodomchok S, Chaiyaratana N, Phalakomkule C. Cooperative coevolutionary approach for flux balance in bacillus subtilis[C]//Proc of the 2008 IEEE World Congress on Computational Intelligence, 2008;1226-1231.
- [66] Cao X, Qiao H, Keane J. A low-cost pedestrian-detection system with single optical camera [J]. IEEE Transactions on Intelligent Transportation Systems, 2008,9(1):58-67.
- [67] Sun Wei, Huo Jun-zhou, Chen Jing, et al. Disc cutters' layout design of the full-face rock tunnel boring machine (TBM) using a cooperative coevolutionary algorithm [J]. Journal of Mechanical Science and Technology, 2011, 25(2):415-427.
- [68] Nebti S, Boukerram A. An improved radial basis function neural network based on a cooperative coevolutionary algorithm for handwritten digits recognition [C]//Proc of the 2010 International Conference on Machine and Web Intelligence (ICMWI), 2010;464-468.
- [69] Yusoh Z I M, Tang Mao-lin. A cooperative coevolutionary algorithm for the composite SaaS placement problem in the cloud [C]//Proc of ICONIP'10, 2010;618-625.
- [70] Liang C H, Chung C Y, Wong K P, et al. Parallel optimal reactive power flow based on cooperative coevolutionary differential evolution and power system decomposition [J]. IEEE Transactions on Power System, 2007, 22(1):319-323.
- [71] Dias L M S D M, Pacheco M A I C. Refinery scheduling optimization using genetic algorithms and cooperative coevolution [C]//Proc of the 2007 IEEE Symposium on Computational Intelligence in Scheduling(CI-Sched 2007), 2007;151-158.
- [72] Wu A, Zhang Jin, Chung H. Decoupled optimal design for power electronic circuits with adaptive migration in coevolutionary environment [J]. Journal of Applied Soft Computing, 2011,11(1):23-31.
- [73] Zhou Xiang, Bao Zhen-qiang, Wang Gui-jun, et al. Optimization of fuzzy job-shop scheduling with multi-process routes and its co-evolutionary algorithm [C]//Proc of the 4th International Conference on Intelligent Computation Technology and Automation, 2011;866-870.
- [74] Panait L, Luke S. Cooperative multi-agent learning: The state of the art [J]. Journal of Autonomous Agents and Multi-Agent Systems, 2005,11(3):387-434.
- [75] Panait L, Luke S, Wiegand R P. Biasing coevolutionary search for optimal multi-agent behaviors [J]. IEEE Transactions on Evolutionary Computation, 2006,10(6):629-645.
- [76] Au C-K, Leung H-F. On the behavior of cooperative coevolution in dynamic environments[C]//Proc of the 2008 IEEE Congress on Evolutionary Computation, 2008;2827-2836.
- [77] Au C-K, Leung H-F. Investigating collaboration methods of random migrant scheme in cooperative coevolution[C]//Proc of the 2009 IEEE Congress on Evolutionary Computation, 2009;2701-2707.
- [78] Nema S, Goulermas J Y, Sparrow G, et al. A hybrid cooperative search algorithm for constrained optimization [J]. Structural and Multidisciplinary Optimization, 2011,43(1):107-119.
- [79] Boonlong K, Maneeratana K, Chaiyaratana N. Determination of erroneous velocity vectors by cooperative coevolutionary genetic algorithm [C]//Proc of the 2006 IEEE Conference on Cybernetics and Intelligent Systems, 2006;1-6.
- [80] Li Wei. Research on the diagnosis learning technology based on the cooperative evolution mechanism[C]//Proc of the 2009 International Conference on Computational Intelligence and Software Engineering, 2009;1-4.
- [81] Li Wei. Research on the application analysis technology for cooperative fault diagnosis [C]//Proc of the 8th World Congress on Intelligent Control and Automation, 2010;5790-5793.

#### 附中文参考文献:

- [56] 胡成玉, 王博. 基于动态维度交叉的粒子群高维函数优化 [J]. 计算技术与自动化, 2009,28(1):92-95.

#### 作者简介:



张凯波(1988-),男,安徽六安人,硕士生,研究方向为合作型协同演化和大规模优化。E-mail: kbzhang@mail.ustc.edu.cn

ZHANG Kai-bo, born in 1988, MS candidate, his research interests include cooperative coevolution, and large scale optimization.