

约束优化进化算法综述*

李智勇^{1,2}, 黄滔^{1,2}, 陈少森^{1,2}, 李仁发^{1,2}

¹(湖南大学 信息科学与工程学院, 湖南 长沙 410082)

²(湖南省嵌入式与网络计算重点实验室, 湖南 长沙 410082)

通讯作者: 李智勇, E-mail: zhiyong.li@hnu.edu.cn



摘要: 约束优化进化算法主要研究如何利用进化计算方法求解约束优化问题,是进化计算领域的一个重要研究课题.约束优化问题求解存在约束区域离散、等式约束、非线性约束等挑战,其问题的本质是,如何处理可行解与不可行解的关系才能使得算法更高效.首先介绍了约束优化问题的定义;然后,系统地分析了目前存在的约束优化方法;同时,基于约束处理机制,将这些方法分为罚函数法、可行性法则、随机排序法、 ε -约束处理法、多目标优化法、混合法等6类,并从约束处理技术的角度对约束优化进化算法的最新研究进展进行综述;最后,指出约束优化进化算法需进一步研究的方向与关键问题.

关键词: 进化算法;约束处理技术;进化计算;约束优化;约束优化进化算法

中图法分类号: TP301

中文引用格式: 李智勇,黄滔,陈少森,李仁发.约束优化进化算法综述.软件学报,2017,28(6):1529-1546. <http://www.jos.org.cn/1000-9825/5259.htm>

英文引用格式: Li ZY, Huang T, Chen SM, Li RF. Overview of constrained optimization evolutionary algorithms. Ruan Jian Xue Bao/Journal of Software, 2017, 28(6): 1529-1546 (in Chinese). <http://www.jos.org.cn/1000-9825/5259.htm>

Overview of Constrained Optimization Evolutionary Algorithms

LI Zhi-Yong^{1,2}, HUANG Tao^{1,2}, CHEN Shao-Miao^{1,2}, LI Ren-Fa^{1,2}

¹(College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China)

²(Key Laboratory of Embedded and Network Computing, Hunan Province, Changsha 410082, China)

Abstract: Constrained optimization evolutionary algorithm, which mainly studies how to use evolutionary computation method to solve constrained optimization problems, is an important research topic in evolutionary computation field. Discrete constraint, equality constraint, nonlinear constraints are challenges to solving constraint optimization. The basis of this problem solving is how to handle the relationship between feasible solution and infeasible solution. In this study, the definition of constrained optimization problem is firstly provided, and then, the existing constrained optimization approaches are systematically analyzed. Meanwhile, algorithms are classified into six categories (i.e., penalty function method, feasible rules, stochastic ranking, ε -constraint, multi-objective constraint handling, and hybrid method), and the state-of-art constrained optimization evolutionary algorithms (COEAs) are surveyed with respect to constraint-handling techniques. Research progress and challenges of the six categories of constraint handling techniques are discussed in detail. Finally, the issues and research directions of constraint handling techniques are discussed.

Key words: evolutionary algorithm; constraint handling technique; evolutionary computation; constrained optimization; constrained

* 基金项目: 国家自然科学基金(61173107, 61672215, 91320103, 61672217); 广东省教育部产学研结合重大科技专项(2012A090300003); 广东省科技计划(2013B090700003); 湖南省研究生科研创新项目(CX2016B067)

Foundation item: National Natural Science Foundation of China (61173107, 61672215, 91320103, 61672217); Production Study Research Cooperation Projects of Department of Education of Guangdong Province (2012A090300003); Guangdong Provincial Science and Technology Projects (2013B090700003); Graduate Scientific Research Innovation Foundation of Hunan Province (CX2016B067)

收稿时间: 2016-05-03; 修改时间: 2016-10-11; 采用时间: 2017-01-16; jos 在线出版时间: 2017-02-20

CNKI 网络优先出版: 2017-02-20 14:05:54, <http://www.cnki.net/kcms/detail/11.2560.TP.20170220.1405.017.html>

optimization evolutionary algorithm

在工程设计、作业调度、智能控制、交通优化、金融投资、网络通信等诸多优化领域,常常遇到许多条件限制(约束),这些条件限制给问题的求解带来了极大的挑战^[1,2].这类优化问题,称为约束优化问题(constrained optimization problem,简称 COP).约束优化问题是优化领域的一个重要问题,约束优化问题中的约束条件通常包括决策变量上下界约束条件、等式/不等式约束条件.按照约束条件的数学特性,还可以分为线性/非线性约束条件.由于约束条件的存在,导致在决策变量的搜索空间中产生了不可行域(搜索空间由可行域和不可行域两部分组成).从无约束优化到约束优化,原本单一的优化目标必须同时考虑优化目标和约束两个方面,而有效的无约束优化方法处理约束优化问题时却是一筹莫展.因此,研究约束优化问题具有重要的理论意义和实际价值.

传统的优化算法求解这类问题时都是基于梯度信息,这只适用于目标函数和约束条件可微的情形,而且求得的解多为局部最优解.进化算法是一种模拟自然过程的全局优化方法,它用个体来代表待求解问题的解,将一定数量的不同个体组成种群.从初始种群开始,采用变异、交叉、选择等操作,模仿自然界的进化过程,引导种群进行进化,使得种群中的个体逐步接近问题的最优解.与传统优化算法相比,进化算法是一种基于群体的搜索技术,具有鲁棒性强、搜索效率高、不易陷入局部最优等特点,因此,它更适合于求解约束优化问题.然而,进化算法的本质是一种无约束的优化技术,必须将其与一定的约束处理机制相结合,构成约束优化进化算法,才能更好地求解复杂的约束优化问题.

约束处理方法对约束优化算法的性能具有重大影响.好的约束处理方法应能将约束优化问题转化为无约束优化问题.同时,充分发挥作为搜索机制的进化算法的性能,还应该有较少的参数,以便于实现.为了提高进化算法求解约束优化问题的效率,近年来,研究者提出了大量将约束条件结合到进化算法的约束处理方法.根据约束优化进化算法中约束处理机制从处理约束的方式不同,可以将它们划分为6类^[3](罚函数法^[4]、可行性法则^[5]、随机排序法^[6]、 ϵ 约束处理法^[7]、多目标优化法^[8]、混合法^[9]):罚函数法通过添加惩罚因子的方式将约束优化问题转化为无约束优化问题,是最为传统的约束处理方法之一,该类方法的不足在于惩罚参数的选取比较困难;可行性法则是 Deb 提出的一种约束处理方法,其将可行解和不可行解分离处理,并且可行解的适应度函数值优于不可行解的适应度函数值,因此,该方法使得不可行解难以进入种群,这样就使得该方法在求解最优解位于可行域边界上的约束优化问题时可能存在问题;随机排序法采用冒泡排序的方法来处理约束优化问题; ϵ 约束处理法通过设定 ϵ 值将约束划分为不同的区间,个体之间的比较在不同的约束区间内采用不同的处理方法,该方法是对可行性法则的扩展,它以牺牲时间为前提,对搜索空间进行充分搜索;多目标优化法根据将约束优化问题转换为多目标优化问题,并采用多目标优化技术来处理问题,然而,没有引入偏好的多目标优化方法可能并不像人们所想象的那样高效;混合法求解约束优化问题是近年来研究的热点,该方法通过将不同的约束处理机制相结合的方法来处理约束优化问题.

纵观国内外对约束优化领域的研究情况,Coello 等人的综述主要针对 2002 年以前的约束优化进化算法,对近些年出现的约束优化进化算法的讨论较少;国内在该领域的综述,主要是偏重于 2009 年以前的研究成果.鉴于最近几年约束优化进化算法的迅猛发展,著名的进化优化领域国际会议(IEEE Congress on Evolutionary Computation,简称 CEC)分别于 2006 年和 2010 年进行了专题讨论并组织了约束优化竞赛^[10,11],促进了约束优化问题的研究与发展.国内外的著名学者对约束优化进化算法进行了长期研究,取得了一定的研究成果^[12,13].我们认为:有必要对该领域的最新研究成果做全面的介绍和讨论,并对目前约束优化中存在的挑战和问题进行描述.

本文首先介绍了约束优化问题的定义,然后结合上述对约束优化进化算法中约束处理机制的分类,对各类算法的基本思想、最新的研究状况、存在的主要挑战等方面进行了全面的描述与探讨.最后,本文指出约束优化进化算法需进一步研究的方向与关键问题.

1 约束优化问题的定义

很多实际问题可以归结为决定一组决策变量,使得目标函数取得最优值的问题.同时,这组决策变量必须满足一定的约束条件,这就构成了约束优化问题或者数学规划问题.通常,约束优化问题可以定义为如下形式(不失一般性,本文考虑最小化情形):

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g_j(x) \leq 0, \quad j=1, \dots, q \\ & && h_j(x) = 0, \quad j=q+1, \dots, m \end{aligned} \quad (1)$$

其中, $x=[x_1, x_2, \dots, x_n] \in S$ 是决策变量, S 表示 n 维搜索空间,它满足如下边界约束条件:

$$l_i \leq x_i \leq u_i, 1 \leq i \leq n \quad (2)$$

$f(x)$ 是目标函数, q 表示不等式约束条件的个数, $m-q$ 表示等式约束的个数. $g_j(x)$ 为第 j 个不等式约束条件, 而 $h_j(x)$ 为第 $j-q$ 个等式约束条件. 其中, u_i 和 l_i 分别为分量 x_i 的上界和下界. 满足所有约束条件的解是可行解, 而所有的可行解组成可行域 $\Omega \in S$, 不满足任意一个约束条件的解被称为不可行解, 所有的不可行解组成不可行域, 如果 $g_j(x)=0 (j=1, \dots, q)$ 在一个可行解处成立, 则称该不等式约束条件在该可行解处是活跃的. 显然, 所有的等式约束在可行域内都是活跃的.

在约束优化中, 等式约束在通常情况下被转化为不等式约束, 于是, 个体 x 在第 j 个约束条件上的约束违反程度表示为

$$G_j(x) = \begin{cases} \max\{g_j(x), 0\}, & 1 \leq j \leq q \\ \max\{|h_j(x)| - \delta, 0\}, & q+1 \leq j \leq m \end{cases} \quad (3)$$

上面公式中, δ 为等式约束的容忍参数, 该参数可以定义为一个特定的值(一般设为 0.0001)或采用自适应变化的方式定义. 所以, 个体 x 总的约束违反程度(又称约束违约度)表示为

$$v(x) = \sum_{j=1}^m G_j(x) \quad (4)$$

通过上述定义, 约束优化问题的优化目标就是: 在满足所有不等式约束的前提下, 在变量的定义范围内, 找到使得目标函数 $f(x)$ 达到最优的解(最优解).

值得注意的是: 由于约束条件之间的存在差异, 从而会出现某些约束条件可能对个体的约束违约度 $v(x)$ 起着主导性的作用, 所以, 可通过采用标准化来平等地对待每个约束条件. 在标准化过程中, 首先找出群体中的个体违反每个约束条件的最大值 $G_j^{\max} (j \in \{1, \dots, m\})$:

$$G_j^{\max} = \max_{i=1, \dots, NP} (G_j(x_i)), j \in \{1, \dots, m\} \quad (5)$$

其中, NP 为种群规模, 即群体中所包含的个体数. 于是, 个体 x 的标准化约束违反程度 $v_{norm}(x)$ 定义为该个体的每个约束违反标准值的平均值:

$$v_{norm}(x) = \frac{1}{m} \sum_{j=1}^m \frac{G_j(x_i)}{G_j^{\max}}, i \in \{1, 2, \dots, NP\} \quad (6)$$

另外, 有时为了均衡目标函数和约束条件之间的差异, 可以通过标准化目标函数来使目标函数和约束违反程度的数量级相同, 最后, 个体 x 的标准化目标函数 $f_{norm}(x)$ 如下显示:

$$f_{norm}(x) = \frac{f(x_i) - f_{\min}}{f_{\max} - f_{\min}}, i \in \{1, 2, \dots, NP\} \quad (7)$$

其中, f_{\min} 和 f_{\max} 分别定义为当前种群中所有个体的目标函数的最小值与最大值:

$$f_{\max} = \min_{i=1, \dots, NP} f(x_i), f_{\min} = \max_{i=1, \dots, NP} f(x_i) \quad (8)$$

通过以上标准化之后, 每个个体的目标函数和约束违反程度的值将在 $[0, 1]$ 区间内变化.

2 基于约束处理的进化算法

进化算法因其优良的性能被用于处理约束优化问题,但其本质上是一种无约束优化进化算法,必须要将其与一定的约束处理机制结合,才能对约束优化问题进行求解.进化算法和约束处理方法相结合构成完整的约束优化进化算法.进化计算领域的研究人员已经提出了大量的约束处理方法和约束优化进化算法,下面根据近年来进化算法中约束处理技术的研究趋势,将约束优化算法间接地划分为以下 6 类:(1) 罚函数法;(2) 可行性法则;(3) 随机排序法;(4) ε 约束处理法;(5) 多目标优化法;(6) 混合法.

2.1 罚函数法

罚函数法因为执行十分简单而得到了广泛的应用,尤其是应用到进化算法里处理约束优化问题.其主要思想是:基于个体的约束违反程度构造惩罚项,通过对目标函数 $f(x)$ 增加惩罚项 $p(x)$ 来构造惩罚适应度函数 $fitness(x)$,将约束优化问题转换为无约束优化问题进行处理.通用的惩罚适应度函数可以定义为如下形式:

$$fitness(x) = f(x) + \alpha \sum_{i=1}^m r_i G_i(x), \sum_{i=1}^m r_i = 1 \quad (9)$$

其中, r_i 表示为用户对每个约束条件定义的约束违反水平数, α 为惩罚系数.为了在目标函数和约束之间起到平衡,罚函数法通过调整惩罚项中惩罚系数来实现.显然, α 的值过小,则对不可行解的惩罚不够,不利于算法搜索到可行解; α 的值过大时,惩罚程度大,则不利于算法收敛于最优解.此外, α 值的设定,面向不同的目标优化函数设定也不同.所以,如何合理设置惩罚系数,是利用罚函数法求解约束优化问题的一个瓶颈,也是约束优化领域的关键问题.下面围绕惩罚系数的选取来介绍几种应对策略.

静态罚函数法是指惩罚项中的惩罚系数不依赖于进化代数,在种群的整个进化过程中设置为定值,但算法通常因数值的单一性而效果不佳.文献[14]提出了一种分层惩罚参数法,采用如下的惩罚适应度函数:

$$fitness(x) = f(x) + \sum_{i=1}^m R_{k,i} \times G_i(x) \quad (10)$$

其中, $R_{k,i}$ ($k=1, \dots, q, i=1, \dots, p$) 为惩罚系数, q 为用户对每个约束条件定义的约束违反水平数.该方法将约束违反的程度分为多个等级,不同等级采用不同的惩罚系数 $R_{k,i}$, 约束违反程度越大,则违反等级越高,对应的惩罚力度也就越大.分级处理机制相对单一数值法更为合理,但该方法需要一定的先验信息来确定各等级的惩罚系数值,另外还产生了较多需要调节的参数.这类方法实现相对简单且多用于求解简单的约束优化问题,不足之处在于它的通用性,因为它对每一类问题,都需要进行惩罚系数的反复实验,寻找到合适的惩罚系数.

动态罚函数法指惩罚系数随着进化代数的变化而变化,惩罚系数的数值是时变的,通常它是随着进化代数增大而增大.其理由是:在进化初期采用较小的惩罚系数,因对不可行解的惩罚较小,算法将有可能对可行域之外进行一定程度的搜索;而进化后期采用较大的惩罚系数,使得算法的搜索集中在可行域,寻找目标更优的可行解.文献[15]提出了如下的方式构造惩罚适应度函数:

$$fitness(x) = f(x) + (C \times t)^\alpha \sum_{i=1}^m G_i^\beta(x) \quad (11)$$

其中, t 为进化代数, C, α, β 为事先设定的常数.虽然该方法不需要设定惩罚系数值,但其他参数仍需要设定.与静态罚函数法相比,虽然动态罚函数法在性能方面上更具优越性,但是找到理想的惩罚系数动态变化规律(初始化惩罚系数的选取)仍然是一个棘手的问题.

死罚函数法^[16]是最简单的罚函数法,它总是拒绝不可行解,完全忽略不可行个体所提供的任何有价值的信息.在死罚函数法中,不可行解的惩罚适应度值定义为 0,这样,当初始种群不包含可行解,进化过程趋于停滞,因为种群中的所有个体的惩罚适应度值都一样,此时需要重新生成初始可行性种群.死罚函数法仅适合于可行域为凸形或可行域占搜索空间比例较大的约束优化问题.

自适应罚函数法是一种通用性更强的约束处理方法,因为它能够把从种群的进化过程中获取的信息作为反馈动态地调整惩罚系数.如文献[17]中提出了一种自适应惩罚适应度函数形式:

$$fitness(x) = f(x) + \lambda(t) \sum_{i=1}^m G_i(x) \quad (12)$$

其中, $\lambda(t)$ 在每一代按如下方式更新:

$$\lambda(t+1) = \begin{cases} (1/\beta_1)\lambda(t), & \text{if case \#1} \\ \beta_2\lambda(t), & \text{if case \#2} \\ \lambda(t), & \text{otherwise} \end{cases} \quad (13)$$

其中, $\beta_1 > \beta_2 > 1$. case # 1 表示在过去 g (用户定义的参数) 代中找到的最好个体均为可行解, case # 2 表示在过去 g 代中找到的最好个体均为不可行解. 其原理可理解为: 若在此之前所找到的最好个体均为可行解, 则表明惩罚系数已足够大, 可适当减少来降低对不可行解的惩罚压力; 若此前所找到的最好个体均为不可行解, 则表明惩罚系数过小, 需适当增大来增强对不可行解的惩罚力度.

除了上述几种方法外, 广泛使用的罚函数法还有退火罚函数法、隔离遗传算法、协同罚函数法等. 在退火罚函数法^[18]中, 惩罚系数随着温度的降低而不断增加; 隔离遗传算法^[19]通过两个不同的惩罚系数来对群体中的个体进行评价, 其旨在于在过大和过小的惩罚系数之间达到平衡; 协同罚函数法^[20,21]包括两个群体, 第 1 个群体中个体表示问题的解, 第 2 个群体的个体为惩罚系数的集合, 利用它来计算第 1 个群体中个体的适应度值. 文献[22]采用进化策略作为搜索框架, 为保持种群中可行解与不可行解的平衡, 始终允许在种群中保留不可行个体以增强全局搜索能力, 算法根据种群中可行解在整个种群中所占的比例来动态调整惩罚系数. Farmani 和 Wright^[23]提出一种自适应适应度表示法, 该方法将惩罚分为两个阶段进行, 都是通过提取种群中的边界个体的信息来确定惩罚强度. Xiao 等人^[24]设计了一种遗传算法求解多约束优化问题, 它利用 KS 函数的融合特征将多个约束条件融合成一个约束条件, 从而减少了静态罚函数法中惩罚系数数量过多的问题和避免了约束条件权重分配问题.

近年来出现了一些具有收敛性能较优的约束优化进化算法^[25,26], 其中有很多方法的研究重点是如何构造出合理的自适应或适应性的惩罚函数. 因为自适应惩罚函数相比动态、静态等其他方法, 自适应调整参数策略的优点在于不需要引入额外的一些参数并略去参数反复调优的过程, 所以目前自适应策略是惩罚法的研究热点.

Tessema 和 Yen^[27]提出了一种自适应惩罚函数的方法, 该方法采用当前种群中可行解的比例来自适应地控制惩罚的强度. 与其他方法不同的是: 该方法除了引入距离项(把种群中的个体映射到二维坐标系内的点, 而该点到坐标原点的距离)之外, 还为了找到最佳不可行解, 对不可行解采用双惩罚. 该方法旨在于先挖掘出种群中具有较好质量的不可行解, 然后再引导种群向可行域或者最优解逼近. 文献[27]中的惩罚适应度函数:

$$fitness(x) = \begin{cases} f_{norm}(x), & \text{for feasible individual} \\ v_{norm}(x), & \text{if there is no feasible individual} \\ \sqrt{f_{norm}(x)^2 + v_{norm}(x)^2} + [(1-rf)f_{norm}(x) + rf \times v_{norm}(x)], & \text{otherwise} \end{cases} \quad (14)$$

其中, $f_{norm}(x)$ 为标准化(单位化)目标函数, $v_{norm}(x)$ 为标准化(单位化)约束违约度, rf 为种群中可行个体所占的百分比(简称可行比率). 在进化的过程中, 种群中的可行个体的数目是用来指导搜索的方向: 若可行比率较小, 倾向于发现更多的可行个体; 若可行比率较大, 偏向于寻找到问题的最优解.

Lemonge 和 Barbosa 等人^[28]提出了一种自适应惩罚函数方法, 并将其和稳态遗传算法相结合, 用于解决工程优化问题. 该方法为每个约束条件分配各自的惩罚系数, 这样来实现约束条件之间的平衡. 在进化的过程中, 该方法根据种群状态信息(比如种群中是否存在可行解和每个约束的违反程度)自适应地决定每个约束条件的惩罚系数的变化. 惩罚适应度函数的表示形式如下:

$$fitness(x) = \begin{cases} f(x), & \text{if } x \text{ is feasible} \\ h(x) + \sum_{j=1}^m k_j v_j(x), & \text{otherwise} \end{cases} \quad (15)$$

其中, $h(x)$ 函数的表示形式如下:

$$h(x) = \begin{cases} f(x), & \text{if } f(x) > \langle f(x) \rangle \\ \langle f(x) \rangle, & \text{otherwise} \end{cases} \quad (16)$$

$\langle f(x) \rangle$ 是指当前种群的目标函数平均值,而惩罚系数采用如下的公式进行计算:

$$H = \begin{cases} f(x_{\text{worst}}), & \text{if there is no feasible element in the population} \\ f(x_{\text{bestfeasible}}), & \text{otherwise} \end{cases} \quad (17)$$

H 系数的定义如下:

$$k_j = H \left| \frac{\langle v_j(x) \rangle}{\sum_{i=1}^m [\langle v_i(x) \rangle]^2} \right| \quad (18)$$

其中, $\langle v_i(x) \rangle$ 为当前种群的第*i*个约束条件的违反程度均值.当种群发生一定的变化时,惩罚系数和 H 系数按公式(17)、公式(18)重新进行计算.显然,从 H 系数的定义形式来看:如果当前种群中不存在可行个体,在不需要考虑真实目标函数的情况下,此时约束优化问题可以看作个体到可行域之间距离的最小优化问题(简称约束满足问题).然而当种群中出现可行个体,再根据公式(15)来更新种群中所有个体的适应度值,而此可行个体最终成为种群中适应度最大的个体.该方法能容忍部分不可行个体保留下来继续生存.值得注意的是,该方法可以很好地解决全局最优解位于可行域边界上的约束问题.

Melo 和 Iacca^[29]提出了一种自适应罚函数法,并将其与改进的自适应协方差矩阵进化策略(CMA-ES)结合起来,得到了一个新的约束优化算法.在自适应罚函数法中,惩罚系数随着每一代群体中全部不可行个体的约束违反信息的变化而变化;采用如下的惩罚适应度函数:

$$\text{fitness}(x) = \text{bestfeas} + |f(x) - \text{bestfeas}| \times |v(x) / \text{max_penalty}| \quad (19)$$

其中, bestfeas 是当前种群中可行个体的最好的目标函数值,若群体中没有可行解,此时, bestfeas 的值由用户定义; max_penalty 指约束违反程度最大不可行个体的约束违约度.从上式可以发现:该方法对违反约束程度越大的不可行个体施加的惩罚强度也就越大;相对可行个体,违约度小的不可行个体可能具有更好的适应度,但不可行个体的适应度是不可能比种群中最好的可行个体的适应度更好.基于上述约束处理方法,文献[30]提出一种两阶段的文化基因框架:第1阶段将约束优化问题变成无约束优化问题进行处理,其目标是为局部搜索阶段找到好的初始个体,为了实现这个目标,采用自适应惩罚CMA-ES作为全局搜索策略;第2阶段使用局部搜索策略(比如单形法)来进一步改善解的质量,其目标是为了找到全局最优解.该方法用于解决规模大的约束优化问题具有很好的优化效果.

Saha 和 Das 等人^[31]提出了一种模糊惩罚法(fuzzy rule-based penalty function approach,简称FRSP),并将该模糊惩罚法与差分进化算法结合,用来解决约束优化问题,它具有两个惩罚项,其中:第1项惩罚采用了基于条件法则的 Mamdani 模糊推理系统,该模型事先掌握了自适应函数所有必需的条件(作为条件库),然后通过模糊逻辑控制(FLC),将个体的约束违反程度、目标函数和种群中可行解的比例作为模糊输入,经过模糊推理过程后,最终得到模糊输出量,即惩罚值;第2项惩罚项是根据种群中可行解所占百分比来动态地平衡目标函数和约束条件.由前面的两项构成的惩罚适应度函数的形式如下:

$$\text{fitness}(x) = p(x) + d(x) \quad (20)$$

其中, $p(x)$ 函数即 Mamdani 模糊模型的输出函数, $d(x)$ 函数的具体形式如下:

$$d(x) = rf \times f_{\text{norm}}(x) + (1 - rf) \times v_{\text{norm}}(x) \quad (21)$$

在上面的公式中, rf 为当前种群中可行个体占的百分比, $f_{\text{norm}}(x)$ 和 $v_{\text{norm}}(x)$ 分别为个体的标准化目标函数和标准化约束违反程度.

尽管罚函数法是约束优化进化算法里最易于实现的方法,但仍然存在缺陷,最主要的不足是惩罚参数的选取比较困难,而且算法的性能强烈地依赖于参数的选择.若选取的参数过大,则进入可行域可能得以保证,但会降低对可行域的探索,因此失去了由不可行个体提供的一些有价值的信息;若选取的参数过小,相对于目标函数而言,惩罚项将被忽略.这两种情况都会导致得不到高质量的解.

2.2 可行性法则

由于罚函数存在着一定的缺陷,近些年来,研究人员将目光放到了其他约束优化机制上,最终提出了几种新的约束处理方法,其中,Deb^[32]提出的可行性法则是当前应用最受欢迎和最有效的约束处理技术之一,在可行性法则中,当两个解 x_i 和 x_j 进行比较时, x_i 比 x_j 更优,通常按以下准则进行解的比较^[33].

- (1) x_i 是可行解而 x_j 是不可行解;
- (2) x_i 和 x_j 都是可行解, x_i 具有更小的目标函数值;
- (3) x_i 和 x_j 都是不可行解, x_i 具有更小的约束违约度.

显然,上述准则(1)和准则(3)是针对约束的,使搜索方向指向可行域;准则(2)则是针对目标,使搜索方向指向目标值好的区域.但上述比较准则存在主要的不足是难以发挥不可行个体的作用,尤其是当种群中的大部分个体都是可行个体时,不可行个体将很难进入群体.为了保持种群的多样性,该文还引入了一种小生境技术.在文献[32]的基础上,Mezure-Montes 和 Coello^[34]提出了一种简单多成员进化策略,在该算法中,一些有潜力的不可行解被保留到下一代中继续存活.同年,他们^[35]将可行性法则与差分进化算法的贪婪选择方法进行结合,用来解决约束优化问题;在进化的过程中,该方法允许一个父代个体可以产生多个子代个体并从多个子代个体中选择一个可行性更优的个体作为其子代,旨在增加产生更优越子代的概率.另外,该方法通过选择概率 S_i 来控制种群的多样性.选择概率 S_i 指定了群体中基于目标函数进行比较的个体比例.

可行性法则将比较个体仅划分为可行个体和不可行个体,如果面对可行域占整个搜索空间比例较小而且其空间拓扑结构复杂的约束问题,种群可能处于停滞不可行域.于是,为了扩展对于个体的分类,Cui 和 Li 等人^[36]利用个体之间存在的优劣关系,在文中提出个体的相对可行度的概念,并根据个体相对可行度来进行个体的比较.

- (1) 当两个比较的个体 x_i 和 x_j 均为可行个体时,目标函数值小的个体占优;
- (2) 当两个比较的个体中, x_i 为可行个体而 x_j 为不可行个体时,可行个体 x_i 占优;
- (3) 当两个个体 x_i 和 x_j 均为不可行个体时,如果它们之间的相对可行度相等,比较它们的约束违约度,违反约束程度小的个体占优;否则,相对可行度大的个体占优.

对于个体的相对可行度的相关定义如下:

定义 1(约束条件的相对可行度(relative feasibility degree of a constraint)). 给定约束问题,设 F 指整个可行域, F_i 指满足约束条件 i 的空间区域.整个可行域的空间大小与满足第 i 约束条件的可行区间大小的比值 $\rho_i=|F|/|F_i|$ 即为约束条件 i 的相对可行度.

定义 2(相对非共同满足约束集合(relative non-common satisfied constraint set)). 设 x_i 与 x_j 分别是种群中两个不同的个体, G 代表约束问题的约束条件集合, G_i 和 B_i 分别指个体 x_i 满足约束条件集合与违反约束条件集合,对应 G_j 和 B_j 分别是个体 x_j 满足约束条件集合和违反约束条件集合. G_i 和 B_j 的交集表示为: $Q(x_i \rightarrow x_j)$,即 x_i 相对 x_j 的非共同满足约束集合.

定义 3(个体的相对可行度(relative feasibility degree of a solution candidate)). 设 x_i 与 x_j 分别是种群中的两个不同的个体, $Q(x_i \rightarrow x_j)$ 指 x_i 相对 x_j 的非共同满足约束集合, $\rho_k(k \in Q(x_i \rightarrow x_j))$ 指约束条件 k 的相对可行度, x_i 相对 x_j 的可行度:

$$R(x_i \rightarrow x_j) = \begin{cases} 0, & Q(x_i \rightarrow x_j) = \emptyset \\ 1, & Q(x_i \rightarrow x_j) = \{1, 2, \dots, m\} \\ \max_{i \in Q(x_i \rightarrow x_j)} \rho_i, & \text{otherwise} \end{cases} \quad (22)$$

Ullah 和 Sarker 等人^[37]针对很多约束优化问题的可行域占搜索空间比例很小并很难找到可行解的特点,设计了一种缩减搜索空间技术(SSRT),该技术用于引导初始化种群中不可行解向可行域靠近.此外,该文还采用了类似于联赛选择算子(基于可行性法则成对比较个体).该算子从解的领域内寻找到局部最优解,并将它用于进化策略中交叉操作来提高搜索机制的效率.虽然文献[37]提出的约束优化进化算法性能较好,但其实现比较复杂,还引入了额外的参数.

约束优化进化算法性能的好坏取决于两个因素:一是作为搜索引擎的进化算法,二是用来处理约束的约束处理方法.最近几年,研究者们把焦点放在了进化算法的搜索效率上,提出了大量的约束优化算法.下面将对一些代表性的约束优化进化算法进行简要的概述.

Elsayed 和 Sarker 等人^[38]为了提高遗传算法的性能,对遗传算法进行了改进.新算法采用多父体交叉操作(multi-parent crossover,简称 MPC)和随机变异操作:交叉操作首先通过类似锦标赛选择算子的方式从种群中选出 3 个父代,然后根据目标函数或者约束违约度进行个体之间的排序,最后进行 3 种不同序列的交叉,产生 3 个新的子代,其中,前两个子代用来帮助于种群的局部开发,第 3 个子代有助于增强遗传算法的全局搜索能力;随机变异操作能帮助逃离局部最优解,避免过早成熟,该算法能解决大部分约束优化问题,但处理高维离散搜索空间的优化问题上存在不足.

Sarker 和 Elsayed 等人^[39]提出一种具有通用性的 DE-DPS(differential evolution with dynamic parameters selection)算法,用于解决所有的约束问题.在进化的过程中,在给定种群规模的情况下,种群中的个体根据不同的参数组合(F 扩张因子、 CR 交叉概率)进行变异、交叉、选择后产生新一代种群.这样循环经过 t 代数后,根据每组参数的成功率进行排序(成功率:采用特定组合的控制参数产生成功保留到下一代种群后代的个体数目与采用该组控制参数进行进化的个体数目之比),筛选出部分最好的参数组合.当种群经过 $g(g>t)$ 代时,在用户给定的种群规模范围内逐渐减少种群规模.最终,确定种群进化阶段的最优参数组合(包括 NP 种群规模、 F 扩张因子、 CR 交叉概率),直到循环迭代结束时找到不同问题的各个进化阶段的性能最优的参数组合.

Wang 等人^[40]认为解决约束优化问题的本质在于如何有效地均衡目标函数和约束条件,提出了一种 FROFI (the feasible rule with the incorporation of objective function information)方法,该方法通过将目标函数的信息融入到可行性法则来使目标函数和约束条件之间达到一个有效的平衡.在该方法中,种群中个体经过差分进化的变异、交叉操作后产生子代个体,父代个体与子代个体首先基于可行性法则比较来决定子代个体是否直接进入下一代,然后将不能进入下一代的子代个体再根据目标函数与其对应的父代个体比较,把所有目标函数好的实验个体归档保存,然后根据目标函数,将归档中的子代个体替换新的群体中的部分个体进入下一代种群.

可行性法则总是认为可行解优于不可行解,该方法相对其他方法可以更快速地找到可行解,且该方法简单易实现.与其他方法相比,不需要使用定义任何参数,具有较好的普适性(适合与不同的进化算法相结合,而不去过多地改变算法本身),因此其受到广泛的应用.注意:该类约束处理方法忽略了不可行个体所携带有价值的信息,不利于算法更快地收敛于可行区域内的最优解.

2.3 随机排序法

随机排序法(stochastic ranking,简称 SR)是由 Runarsson 和 Yao^[6]于 2000 年提出,随机排序被用来平衡目标函数和约束违约度之间的矛盾关系.在该方法中,采用概率参数(pf)来决定是根据目标函数还是根据个体总的约束违约程度来评判两个个体的优越性,其基本流程如下:

随机排序法
if 可行解 或者 $rand < pf$ then
基于解的目标函数评判先后顺序
else
基于解的约束违约度评判先后顺序
end if

该方法将目标函数和约束违约度之间的平衡问题转换为一个基于某个概率的随机排序问题,它相对传统的随机排序方法具有更好的普适性,没有必要针对不同的目标优化函数和优化阶段设置不同的约束惩罚因子.但又引发了另外一个问题,即,如何设定概率参数 pf 的值.在最基本的随机排序法中, pf 的值设定为 0.475.针对该问题,有较多研究者提出了改进方法,例如:文献[41]提出了一种 pf 值动态变化的方法,其 pf 值由 0.475 线性下降至 0.025;文献[42]采用在随机排序法中的概率参数 pf 随着种群进化代数动态变化的方法,并将该方法与新的多成员差分进化算法进行结合,设计出 DSS-MD 算法.

2.4 ε 约束处理法

Takahama 和 Sakai^[7]在 2006 年 CEC 关于约束处理的专题会议上提出了一种 ε 约束处理法,该方法的核心思想是:通过人为设定 ε 值,基于个体的约束违约度分为不同的区域,在不同的区域,可行解和不可行解分别采用不同的评判方法.该方法相对于可行性法则,利用了不可行区域中具有较好目标函数值的不可行解的信息,具有更好的收敛性能.该方法使用一个显式的分段函数(如公式(23)、公式(24))控制 ε ,然后根据公式(25)(ε 水平比较)来评判个体的优劣:

$$\varepsilon(k) = \begin{cases} \varepsilon(0) \left(1 - \frac{k}{T_c}\right)^{cp}, & 0 < k < T_c \\ 0, & k \geq T_c \end{cases} \quad (23)$$

其中, $\varepsilon(0)$ 的定义为

$$\varepsilon(0) = v(X_\theta) \quad (24)$$

$$(f_1, v_1) < (f_2, v_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{if } v_1, v_2 < \varepsilon \\ f_1 < f_2, & \text{if } v_1 = v_2 \\ v_1 < v_2, & \text{otherwise} \end{cases} \quad (25)$$

其中, X_θ 是种群个体根据约束违约度升序排序后第 θ ($\theta=0.05 \times NP$)个个体, NP 为种群的规模, k 为当前代数或迭代次数以及控制代数 $T_c \in [0.1T_{\max}, 0.8T_{\max}]$;而 cp 满足区间 $[2, 10]$.在文献[7]中,两位作者将 ε 约束处理方法与差分进化算法结合构成 ε 差分进化算法(ε DE),该算法采用了一个基于梯度的变异算子来寻找可行解.虽然基于梯度的变异算子可以避免对不可行域进行无意义搜索,但若对种群中全部的不可行解进行修复,则它很耗时.因此,在文献[7]的基础上,Bu 和 Luo 等人^[43]提出了一种基于群体的修复策略,该策略不是盲目地从种群中选择不可行个体,而是从种群中选择一些具有代表性的不可行解基于梯度修复法进行修复.

ε 约束处理方法的本质是通过用 ε 水平比较方法替换普通的比较方法来评价个体的优劣,将用来求解无约束优化问题的算法转换成求解带约束的优化问题的算法.所以,该类算法的性能主要由进化算法性能的好坏来决定.研究者们已针对如何有效提高该类约束优化进化算法(ε DE)的稳定性、效率等性能问题进行了一系列后续的研究.

Brest 等人^[44]提出了一种自适应差分进化算法(jDEsoco)用于求解单目标实值约束优化问题,该算法除了使用老化机制重新初始化停滞不前的个体来加快种群的收敛速率的同时避免陷入局部最优之外,还采用基于概率选择不同的 DE 策略(“rand/1/bin”|0.9 和“best/1/bin”|0.1),而每种 DE 策略的控制参数根据公式(26)、公式(27)来进行自适应调整:

$$F_{i,gen+1} = \begin{cases} F_i + rand_1 * F_u, & \text{if } rand_2 < \tau_1 \\ F_{i,gen}, & \text{otherwise} \end{cases} \quad (26)$$

$$CR_{i,gen+1} = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ CR_{i,gen}, & \text{otherwise} \end{cases} \quad (27)$$

其中, gen 指遗传代数或称迭代次数, $rand_i$ ($i=1,2,3,4$)取值范围为 $[0, 1]$, τ_1, τ_2, F_i, F_u 的值分别为 0.1,0.1,0.1,0.9.

为了改善 ε DE 的效率与稳定性,必须要解决算法收敛性与种群多样性之间的冲突.Takahama 和 Sakai^[45]基于等级的 ε 约束差分进化算法(ε RDE)来解决约束问题.通过 ε 水平比较为种群的每个个体进行排序并分配等级(等级越低说明解的质量越好).在 DE 操作中,个体根据所选基向量(个体)的等级高低选择不同的参数值(F 扩张因子、 CR 交叉概率),基向量(个体)的等级越低,则选择的参数值(F 扩张因子越小和 CR 交叉概率越大)有助于增加算法的收敛速度;相反(F 扩张因子越大和 CR 交叉概率越小),则有助于保持种群的多样性.

该方法具有较好的收敛性能,已与 GA^[46],PSO^[47]等算法相结合,并取得了较好的优化效果.但该方法需要人为地设定不同的 ε 水平,没有考虑实际的搜索场景,因此,这类约束优化算法的收敛速度方面存在一定的不足.此外,该方法高度依赖 ε 初始值 and 变化函数的选取,这也是 ε 约束处理法需要进一步改进的地方.

2.5 多目标优化法

多目标优化法的主要思想是:将约束优化问题转换为多目标优化问题,然后利用多目标优化技术来处理转换后的问题.所以下面先给出了多目标优化问题的相关描述和多目标优化中的几个重要定义,不失一般性,考虑以下具有 n 个决策变量和 m 个目标函数的多目标优化问题(multi-objective optimization problems,简称 MOPs):

$$\min \text{ or } \max f(x)=(f_1(x), f_2(x), \dots, f_m(x)) \quad (28)$$

其中, $x=(x_1, x_2, \dots, x_n) \in X \subset \mathcal{R}^n$ 为决策变量, X 为决策空间; $f(x) \in Y \subset \mathcal{R}^m$ 为目标向量, Y 为目标空间.

定义 4(Pareto 支配(Pareto dominance)). 决策向量 $x_u \in X$ Pareto 支配决策变量 $x_v \in X$, 记为 $x_u \prec x_v$, 当且仅当:

- $\forall i \in \{1, \dots, m\}$ 满足 $f_i(x_u) \leq f_i(x_v)$;
- $\exists j \in \{1, \dots, m\}$ 满足 $f_j(x_u) < f_j(x_v)$.

此时,也称决策变量 x_v Pareto 劣于(dominated by)决策变量 x_u .若决策变量 x_u 与决策变量 x_v 不存在 Pareto 支配关系,则称它们非劣(non-dominated).

定义 5(Pareto 最优解(Pareto optimality)). 变量 $x_u \in X$ 称为 X 上的 Pareto 最优解,当且仅当 $\neg \exists x_v \in X$ 使得:

$$x_v \prec x_u.$$

定义 6(Pareto 最优解集(Pareto optimal set)). 对于给定的多目标优化问题 $f(x)$, Pareto 最优解集(ρ^*)定义为

$$\rho^* = \{x_u \in X \mid \neg \exists x_v \in X, x_v \prec x_u\}.$$

在将约束优化问题转换为多目标优化问题时,通常存在着两种方式:第 1 种方式将约束优化问题转换为具有两个目标的多目标优化问题,其中,第 1 个目标是目标函数 $f(x)$,第 2 个目标为个体的约束违约度函数 $v(x)$;第 2 种方式将约束优化问题的目标函数和约束条件分别作为不同的目标看待,这样,将优化约束问题转化为具有 $m+1$ 个目标的多目标优化问题.

在多目标技术方面,一般来说,主要有 3 种多目标优化技术运用于处理转换后的问题.

- (1) 使用 Pareto 占优作为一种选择准则;
- (2) 使用 Pareto 排序(ranking)来定义个体适应度值;
- (3) 将种群划分为若干子群体,对子群体中个体的评估或者基于目标函数,或者基于某个约束条件.

多目标优化法不仅成功地避免了目标函数和约束之间的不平衡问题,而且在处理约束优化上取得了较好的性能.在 2009 年,王勇等人^[48]对该类算法进行了系统的分析与论述,所以本文首先选取其中具有代表性的几种算法进行回顾,然后再对该类算法的最新的研究成果进行详细论述.

Surry 等人^[49]提出了 COMOGA(constrained optimization by multi-objective genetic algorithm)算法,在该算法中,约束优化问题被视作约束满足问题或无约束优化问题来处理.当约束优化问题被视为约束满足问题时,目标函数被忽略,此时,个体之间的比较由 Pareto 排序决定,Pareto 排序基于约束违反来定义.当约束优化问题被视为无约束优化问题时,约束条件被忽略,此时,个体之间的比较由目标函数决定.Venkatraman 和 Yen^[50]提出了一种包含两个阶段的通用框架来处理约束问题:第 1 阶段,当种群里没有可行解时,该方法仅使用约束违反程度选择个体;当种群中出现可行解时(第 2 阶段),原来的约束优化问题转化为一个双目标优化问题,其中一个目标是原来的目标函数,而另外一个目标则是约束违反程度,之后,再使用 Pareto 排序的方法来选择个体.

Wang 等人^[8]提出了一种 HCOEA(hybrid constrained optimization evolutionary algorithm)算法,在该算法中,将约束问题转化为双目标优化问题,采用带小生境操作的遗传算法框架,并基于非支配解的概念进行解之间的替换,但只有子代个体在支配父代个体时,才以概率 1 替换父代个体.同时,算法除了引入类似于 MOEA 中不可行解的存储和替换机制外,还引入一种聚类分割和多个父代交叉的局部搜索技术,根据搜索空间中各个个体之间的相互距离,将种群聚类为 M 个子种群,各子种群通过内部交叉产生新个体,进而组合新种群.

Coello^[51]基于向量评估遗传算法求解约束优化问题.将种群分成相等规模的子种群,每个子群对应一个约束条件,其中,某个子群体将目标函数作为适应度函数来评价个体,其余的子群体将相应的单个约束条件作为适应度函数来评价个体,其目的在于:每个子群体试图找到对应约束条件的可行解,通过联合所有的子种群,该方法找到所有约束条件下的可行解.该算法的主要缺陷在于:子群体的个数会随着约束条件的个数呈线性增加;另

外,如何确定子群体的合理规模有待进一步研究.

Gong 和 Cai^[52]提出了一种多目标差分进化算法,该算法引入正交设计法用于种群的初始化以及基于正交的交叉算子来提高算法的局部搜索能力,并采用差分进化作为搜索机制.在该算法中,比较父代个体和子代个体时先优先考虑约束条件.如果子代个体 Pareto 支配父代个体,则立即用子代个体取代父代个体.如果两个比较个体在约束条件上存在非支配关系,则再考虑目标函数和约束违约度,被支配的父代个体则立即被子代个体取代.之后,将非劣的子代个体与父代个体组成一个混合群体,然后选取其中的非劣个体进入下一代.

Reynoso-Meza 和 Blasco 等人^[53]提出了一种 sp-MODE(the multiobjective differential evolution algorithm with spherical pruning)法来解决高约束优化问题,该方法基于双重选择策略.在每一次迭代,首先从种群(P)中随机选择 N 个个体组成的子群体通过 DE 进化产生子代群体(P_n)后,然后利用 Pareto 排序的方法从联合群体($P_n \cup D$)中找出所有的非劣个体,最后采用球形修剪技术从非劣解集中筛选出每个球形网格内具有较小约束违反程度的非劣个体(D).同时,球形裁剪技术维持了种群的多样性.当产生一个新的个体时,Pareto 占优用于比较该个体与其父代个体的优劣.

Singh 和 Ray 等人^[54]针对很多约束优化问题的最优解位于可行域边界的特点,提出了 IDEA(infeasibility driven evolutionary algorithm)法.在该方法中,约束优化问题被转化为两个目标的多目标优化问题来处理:第 1 个目标为原目标函数 $f(x)$;另外一个目标为基于种群成员之间的个体相对约束违约度(根据每个约束条件的违反程度,将种群中的每个个体从低到高进行排序;同时,约束违反度越低的个体分配的等级越高(可行个体对应的等级为 0),个体相对约束违约度是所有约束条件上对应等级的总和).在进化的过程,IDEA 首先根据个体的可行性将种群划分成两个群体(可行解集和不可行解集),并分别对该两个群体使用 Pareto 排序过程的方法来定义个体的等级,然后再按一定比例 u (用户定义),分别从两个群体中选择个体组成下一代种群.

Wang 和 Cai^[55]提出 CMODE(combining multiobjective optimization with differential evolution)算法,该算法将多目标优化策略与差分进化算法结合起来处理约束问题.CMODE 首先从种群中随机选择 n 个个体组成的群体经过差分进化的变异、交叉操作产生子代群体,然后从子代群体找出全部的非劣个体;接着,随机替换父代群体中的任意一个被支配个体(如果该劣于个体存在).此外,该方法采用一种不可行解存档和替换机制,将每一代种群中好的不可行解存档起来,用于在后续进化中随机地替换种群中的个体.一方面,该机制保证了种群的多样性,使算法不会过早收敛到某些可行的局部最优;另一方面,该机制有利于对边界区域进行局部搜索.同年,Wang 和 Cai^[56]再次提出了一种由全局模型和局部模型动态组成的混合框架(dynamic hybrid framework,简称 DyHF),用来求解约束问题.DyHF 根据当前种群中可行解的比例自发地实行全局搜索模型或者局部搜索模型.局部搜索模型首先将种群通过聚类方法划分成多个群体,然后分别从每个子代群体中找出所有的非劣个体,最后选择一个非劣个体随机地替换其父代群体中一个劣于个体(如果该个体存在),使种群从不同方向逼近可行域.全局搜索模型是从整个种群的角度利用 Pareto 优劣关系比较个体,选择父代个体和子代个体中的优胜者.

Li 和 Zhang^[57]提出了一种有偏支配的约束处理技术,使用该有偏支配进行个体的比较.假设第 k 个目标函数 $f_k(x)$ 是偏向的目标函数(有偏目标函数),有偏支配的定义如下:

定义 7(有偏支配或 b -支配(biased dominance or b -dominance)). 决策向量 $x_u \in X$ 有偏支配决策变量 $x_v \in X$, 记为 $x_v \prec_b x_u$, 当且仅当:

$$\begin{cases} x_u \prec x_v, & f(x_u) < b \wedge f(x_v) < b \\ f(x_u) < f(x_v), & \text{otherwise} \end{cases} \quad (29)$$

其中, b 为有偏阈值.当两个比较个体的第 k 个目标函数的函数值小于或等于 b 时,有偏支配等价于 Pareto 支配;当两个比较个体中任何一个的第 k 个目标函数的函数值超过 b 时,有偏支配只考虑第 k 个目标函数而忽视其他目标函数.文献[57]将有偏支配与差分进化算法结合,提出了一种有偏的多目标优化算法(biased multiobjective optimization,简称 BMO).在每一代开始时,使用公式(30)来更新 b 的值,对种群中的每个个体使用差分进化的变异和交叉操作生成一个子代个体.如果新生成的子代个体有偏支配对应的父代个体,则立即用新的子代个体取代父代个体.算法重复上述迭代过程,直到终止条件满足.最后,种群中的最好个体就是算法得到的最终解.

$$b = G_{\max} - G_{\min} - 0.0001 \quad (30)$$

其中, G_{\min} 是种群中个体的最小约束违反程度值, 而 G_{\max} 为种群中个体的最大约束违反程度值. 当种群中所有的个体都远离可行域时, b 取值较小, 进化的目标侧重于减小约束违反程度, 使种群中的个体向可行域靠近; 当种群中部分个体已接近可行域或者种群中存在可行个体时, b 值较大, 使得目标函数和约束违反程度得到同时考虑, 进而找到全局最优解.

Gao 和 Yen^[58]提出了一种双种群协同进化差分算法, 它把原来的约束问题转变成一个双目标优化问题, 其中, 第 1 个目标是目标函数, 第 2 个目标是约束违反度. 在进化过程中, 为了分别对待这两个目标, 整个种群根据个体的可行性被划分成两个子群体, 每个子群体优化对应的目标. 同时, 该文还引用了一种信息共享机制促进两个子群体之间搜索信息的相互交流, 旨在让不可行个体变成可行个体并寻找到最优解.

多目标优化法在求解约束优化问题时表现出来突出的收敛结果和性能, 特别适用于解决可行域空间很小的问题. 但不得不承认的是, 该方法需要耗费较高的计算资源, 这也是多目标优化法应用的瓶颈之一.

2.6 混合法

由于约束问题的复杂性和多样性, 采用单一的约束处理技术往往达不到理想的效果. 目前, 约束优化技术研究的主流方向是设计具有自适应机制和混合机制的高效技术. 混合法就是将两种或者以上不同的约束处理技术相结合用于处理约束问题, 该方法充分利用不同约束处理技术的特点来更有效地解决约束优化问题, 但如何合理地混合不同约束处理技术, 是研究该方法的关键点.

基于上述思想, Mallipeddi 等人^[9]提出了一种约束处理技术的集成框架(ensemble of constraint handling techniques, 简称 ECHT)来综合利用各种已有的约束处理技术. 该方法采用多种群策略, 不同的种群采用不同的约束处理机制, 并通过种群间的相互学习来应对复杂的搜索场景.

Wang 和 Cai 等人^[59]提出了一种自适应均衡模型(adaptive trade-off model, 简称 ATM), 该模型将种群的进化过程分为 3 个阶段, 并根据群体在 3 个不同阶段的特点采取不同的选择策略. 当种群中只存在不可行解时, 该模型首先将约束问题转换成多目标优化问题, 然后通过 Pareto 排序的方法从种群中选择具有较小约束违反度的非劣解进入下一代种群. 当种群中同时包含不可行个体和可行个体时, 该模型采用自适应罚函数法将有约束问题转化成无约束优化问题处理, 并根据转变后的个体适应度函数选择个体. 当种群中仅包含可行解时, 个体之间的比较和选择基于目标函数. 他们还提出一种 ATM-HD(hybrid differential evolution and adaptive trade-off model)算法^[60], 该算法将混合了多种变异策略的差分进化算法和 ATM 相结合用来处理约束问题.

Elsayed 等人^[61]提出了一种差分进化的集成框架, 该框架包含不同的差分进化操作——由 4 种变异操作、两种不同的重组操作和两种约束处理机制(可行性法则和 ϵ 约束处理法)随机组合 16 种不同策略的集成. 在进化的过程中, 种群中的个体选择随机选择一种策略, 每种策略通过它改善个体的能力来决定种群中个体采用该策略的使用比例(概率). 此外, 该文采用了一种局部搜索操作来提高解的精度. Jiao 等人^[62]提出了一种新型的选择进化策略算法, 在该算法中: 首先, 种群的个体全部经过进化策略后产生新的子代种群; 然后将父代种群和子代种群组合, 除去其中具有较大约束违反程度的个体; 最后, 从组合种群中选出所有的非劣个体进入下一代种群, 若非劣个体数少于规定的种群规模, 则使用自适应转换个体的目标函数的方法(罚函数法)从组合群体选择剩余个体进入下一代种群.

Wang 和 Liu 等人^[63]基于正交设计差分进化算法用于处理约束优化问题, 该算法把种群中的个体随机成对采用正交设计产生多个子群体, 然后合并多个子群体成一个群体, 再用 Pareto 排序选出群体中选择所有的非劣个体. 根据所有非劣个体组成的群体中是否存在可行解, 采用两种不同的替换操作. 如果群体中不存在可行个体, 用约束违反程度最小的非劣个体随机替换父代种群中的劣于个体(假如该劣于个体存在). 如果群体中存在可行个体, 以概率 0.8 选择群体中的两个非劣个体随机替换父代群体中的两个个体或者以概率 0.2 基于可行性法则用非劣的可行子代个体替换最差的劣于父代个体.

Cai 和 Wang^[64]提出了 MOEA(multi-objective optimization-based evolutionary algorithm, 简称 MOEA)算法, 算法以实数编码 GA 作为搜索框架, 根据替换操作的差异, 给出 3 种不同模型: 在模型 I 中, 每次迭代获得的所有

非劣个体,均用一对一方式替换父代个体中受支配的劣于个体;在模型 II 中,每次迭代获得的所有非劣个体,仅从中随机选取一个非劣个体用于替换父代个体中任意一个被支配个体;在模型 III 中,采用可行性法则进行替换操作.同时,算法根据种群中可行解与不可行解的比例以及解的目标函数值与约束违约度,适应性地从 3 种模型中选择一种模型进行父代与子代间的替换.另外,该算法还引入一种不可行解的存档和替换机制,其旨在于引导种群向可行域靠近.

在 2010 年的约束优化的国际竞赛中,Tasgetiren 和 Suganthan 等人^[65]提出了 eDE(ensemble of differential evolution algorithms)算法.在该算法中,种群中每个个体采用不同的差分进化策略(比如 DE/best/1/bin,VPS).在算法的不同过程中采用不同的约束处理技术,比如可行性准则,用于选出种群中最好的个体.当产生一个新的个体后, ε 水平比较用于判断父代个体和子代个体的优劣,在 VPS(variable parameter search)操作产生的子代个体和父代个体间之根据自适应惩罚适应度进行比较.

Mani 和 Patvardhan^[66]通过实验和理论分析发现适应性罚函数法和可行性法则之间存在性能互补的关系(存在优势和劣势互补),设计了一种双种群的自适应协同法,该方法使用了两个同等规模的种群.在进化的过程中,第 1 个种群中个体采用自适应罚函数法来计算个体的适应度值,第 2 个种群中个体采用可行法则计算个体的适应度值.同时,第 2 个种群的个体与第 1 个种群进行个体交换(信息交流)来自适应地调整惩罚系数.

Gan 和 Peng 等人^[67]提出了一种自适应决策者(adaptive decision maker,简称 ADM),该 ADM 以自适应惩罚函数的形式决定候选解(子代群体中的非劣个体)是否取代其父代个体.在每一代开始时,先从种群中随机选取 u 个个体组成一个群体,对群体中个体进行单形交叉(simplex crossover,简称 SPX)操作后产生新的子代群体;然后,采用 Pareto 占优的选择准则从子代群体中找出所有的非劣个体;最后,根据转换后的自适应惩罚函数公式(31)来计算个体的适应度值,再从种群中选择一个最优的非劣个体(惩罚适应度最好的个体),并将此非劣个体与种群中最差的劣于个体(惩罚适应度最差的个体)进行比较,如果非劣个体具有更小的目标函数值,则立即用新的非劣个体取代父代个体:

$$F(x)=f(x)+10^{\alpha(1-\rho)} \cdot v(x) \quad (31)$$

其中, $f(x)$ 为目标函数, $v(x)$ 为个体总的约束违反程度, ρ 为当前种群中可行解的百分比, α 满足区间[1,15].该方法存在的主要限制在于,需要反复实验来进行参数的调优.

Deb 和 Datta^[68]提出一种多目标优化和惩罚函数方法相结合的方法来处理约束优化问题.在方法中,约束优化问题被转变成无约束优化问题.在每一次迭代,首先用双目标优化法找到非支配前沿,若满足条件(隔一定代数 t),根据当前非支配前沿与惩罚函数之间存在的关系(在二维坐标系($f-v$)里,非支配前沿中位于约束违约度等于零处的点斜率来估计惩罚函数中惩罚系数)确定优良惩罚系数的估计值,然后利用罚函数法将约束问题转变成无约束优化问题,并用局部搜索模型来求解转换后的无约束优化问题,直到终止条件满足(解在局部搜索前后所对应的目标函数值之间的差值小于 0.000 1),最终得到约束问题的最优解.在文献[68]的基础上,他们加入了一种自适应的约束标准化技术^[69].近年来,他们再次提出一种基于个体惩罚的约束处理方法^[70],与之前提出的方法不同之处在于,该方法利用双目标优化法来确定每个约束条件的惩罚系数的优良估计.

3 亟待解决的问题

虽然约束优化进化算法已经取得了一定的成果,但是仍有很多问题需要解决,主要是:

(1) 等式约束处理

目前,约束优化进化算法对于等式约束优化问题的约束效果并不令人满意,难以收敛算法最优解.目前主要采用动态缩小 δ 值来提高算法的收敛性能,但面对非线性等式约束、离散等式约束等问题,依然具有较大的挑战.

(2) 离散约束优化问题

在该类约束优化问题中,问题的可行区域是离散的,这使得算法容易陷入某个可行区域中的最优解,难以获得全局最优解.如何使得搜索能穿梭于各个可行区域间,是该问题的主要挑战.多种群、分布估计可能是解决该问题的可行方案之一.

(3) 约束条件与优化目标之间的不平衡

约束优化不仅要求所得解的质量,同时要求满足约束条件.对于有些情况,如过分强调满足约束条件,将有可能导致搜索偏离性能好的区域;相反,如果过分强调解的质量,则有可能导致搜索空间偏离可行域.因此,如何根据问题的先验信息和从搜索过程中得到的知识来有效协调约束与目标之间的不平衡非常重要.

(4) 可行解与不可行解的分别对待

约束优化问题相对于普通约束优化问题,其本质区别在于如何对待不可行解携带的信息.在早期的研究中,对该问题的研究取得了较大的进展,但近年来对该基本问题并没有取得实质性的进展.这主要是由于近年来进化算法的搜索效率得到了快速的发展,弥补了约束处理的不足.模糊理论、粗糙理论、 ε 约束处理等技术可能是该问题的研究方向.

(5) 普适的约束处理技术

目前,不同的约束处理技术对应不同的随机搜索技术有着不同的搜索效率.无免费的午餐定理告诉我们,没有任何算法能对所有问题都高效.但我们能不能设计一种对大多数类别算法都高效的约束处理机制?这将会是未来研究的方向之一.

(6) 自适应约束处理技术

正如前面所说,不同的约束处理技术对应不同的随机搜索技术有着不同的搜索效率.设计一种针对不同的搜索状态而采用不同的约束处理机制,自适应地选择不同的约束处理技术可能是可行的方案之一.另外,超启发式框架也具有一定的可行性.

4 结束语

约束优化进化算法是智能优化算法研究领域研究的重要课题之一.虽然经过多年的研究,但依然存在较多的问题亟待解决.本文对现有进化算法中的约束处理技术进行了系统的分析,并将其根据处理约束的方式不同划分为 6 大类,同时,对各方法的研究现状和主要不足进行了逐一分析,最后给出了进化算法中的约束处理技术研究亟待解决的问题与拟解决方案.由于问题所涉及的方面过多,不可能面面俱到,希望与研究者们广泛交流共同促进该领域的发展.

References:

- [1] Michalewicz Z, Schoenauer M. Evolutionary algorithm for constrained parameter optimization problems. *Evolutionary Computation*, 1996,4(1):1–32. [doi: 10.1162/evco.1996.4.1.1]
- [2] Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computation Methods in Applied Mechanics and Engineering*, 2002,191(11-12):1245–1287. [doi: 10.1016/S0045-7825(01)00323-1]
- [3] Mezura-Montes E, Coello CAC. Constraint-Handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 2011,1(4):173–194. [doi: 10/1016/j.swevo.2011.10.001]
- [4] Gong W, Cai ZH, Liang DW. Adaptive ranking mutation operator based differential evolution for constrained optimization. *IEEE Trans. on Cybernetics*, 2015,45(4):716–727. [doi: 10.1109/TCYB.2014.2334592]
- [5] Saha A, Datta R, Deb K. Hybrid gradient projection based genetic algorithms for constrained optimization. In: *Proc. of the Congress on Evolutionary Computation*. Barcelona: IEEE Service Center, 2010. 2851–2858. [doi: 10.1109/CEC.2010.5586303]
- [6] Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. on Evolutionary Computation*, 2000,4(3):284–294. [doi: 10.1109/4235.873238]
- [7] Takahama T, Sakai S. Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites. In: *Proc. of the 2006 IEEE Int'l Conf. on Evolutionary Computation*. Vancouver: IEEE Press, 2006. 372–378. [doi: 10.1109/CEC2006.1688283]
- [8] Wang Y, Cai ZX, Guo GQ, Zhou YR. Multi-Objective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Trans. on Systems, Man, and Cybernetics (B)*, 2007,37(3):560–575.

- [9] Mallipeddi R, Suganthan PN. Ensemble of constraint handling techniques. *IEEE Trans. on Evolutionary Computation*, 2010,14(4): 561–579. [doi: 10.1109/TEVC.2009.2033582]
- [10] Liang JJ, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan PN, Coellon Coello CA, Deb K. Problem definitions and evaluations criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical Report, #2006005, Singapore: Nanyang Technological University, 2006.
- [11] Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2010 competition and special session on single objective constrained real-parameter optimization. Technical Report, #2010004, Singapore: Nanyang Technological University, 2010.
- [12] Wang Y, Cai ZX, Zhou YR, Fan Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*, 2009,37(4):395–413. [doi: 10.1007/s00158-008-0238-3]
- [13] Jia GB, Wang Y, Cai ZX, Jin YC. An improved $(\mu+\lambda)$ -constrained differential evolution for constrained optimization. *Information Sciences*, 2013,222(10):302–322. [doi: 10.1016/j.ins.2012.01.017]
- [14] Homaifar A, Lai SH, Qi X. Constrained optimization via genetic algorithms. *Simulation Trans. of the Society for Modeling & Simulation International*, 1994,62(4):242–254. [doi: 10.1177/003754979406200405]
- [15] Joines J, Houck C. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In: Michalewicz Z, *et al.*, eds. *Proc. of the 1st IEEE Conf. on Evolutionary Computation*. Orlando: IEEE Press, 1994. 579–584. [doi: 10.1109/ICEC.1994.349995]
- [16] Hoffmeister F, Sprave J. Problem-Independent handling of constraints by use of metric penalty functions. In: Fogel LJ, Angeline PJ, Bäck T, eds. *Proc. of the 5th Annual Conf. on Evolutionary Programming (EP'96)*. San Diego: MIT Press, 1996. 289–294.
- [17] Basheed K. An adaptive penalty approach for constrained genetic algorithm optimization. In: Koza JR, *et al.*, eds. *Proc. of the 3rd Annual Conf. on Genetic Programming (GP'98)/Symp. on Genetic Algorithms (SGA'98)*. San Francisco: Morgan Kaufmann Publishers, 1998. 584–590.
- [18] Michalewicz Z, Attia NF. Evolutionary optimization of constrained problems. In: Sebald AV, Fogel LJ, eds. *Proc. of the 3rd Annual Conf. on Evolutionary Programming*. River Edge: World Scientific, 1994. 98–108.
- [19] Le RRG, Knopf-Lenoir C, Haftka RT. A segregated genetic algorithm for constrained structural optimization. In: Eshelman LJ, ed. *Proc. of the 6th Int'l Conf. on Genetic Algorithms*. San Francisco: Morgan Kaufman Publishers, 1995. 558–565.
- [20] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 2000, 41(2):113–127. [doi: 10.1016/S0166-3615(99)00046-9]
- [21] Huang F, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, 2007,186(1):340–356. [doi: 10.1016/j.amc.2006.07.105]
- [22] Hamida SB, Schoenauer M. ASCHEA: New results using adaptive segregational constraint handling. In: Fogel DB, *et al.*, eds. *Proc. of the Congress on Evolutionary Computation 2002 (CEC 2002)*. Piscataway: IEEE Service Center, 2002. 884–889. [doi: 10.1109/CEC.2002.1007042]
- [23] Farmani R, Wright JA. Self-Adaptive fitness formulation for constrained optimization. *IEEE Trans. on Evolutionary Computation*, 2003,7(5):445–455. [doi: 10.1109/TEVC.2003.817236]
- [24] Xiao JH, Xu J, Shao Z, Jiang CF, Pan L. A genetic algorithm for solving multi-constrained function optimization problems based on KS function. In: *Proc. of the 2007 IEEE Congress on Evolutionary Computation*. Singapore: IEEE Press, 2007. 4497–4501. [doi: 10.1109/CEC.2007.4425060]
- [25] Matias J, Correia A, Mestre P, Serodio C, Couto P, Teixeira C, Melo-Pinto P. Adaptive penalty and barrier function based on fuzzy logic. *Expert Systems with Applications*, 2015,42(19):6777–6783. [doi: 10.1016/j.eswa.2015.04.070]
- [26] Lin CH. A rough penalty genetic algorithm for constrained optimization. *Information Sciences*, 2013,241(241):119–137. [doi: 10.1016/j.ins.2013.04.001]
- [27] Tessema B, Yen G. A adaptive penalty formulation for constrained evolutionary optimization. *IEEE Trans. on Systems, Man, and Cybernetics (A)*, 2009,39(3):565–578. [doi: 10.1109/TSMCA.2009.2013333]
- [28] Lemonge ACC, Barbosa HJC, Bernardino HS. Variants of an adaptive penalty scheme for steady-state genetic algorithms in engineering optimization. *Engineering Computations*, 2015,32(8):2182–2215. [doi: 10.1108/EC-07-2014-0158]

- [29] De Melo V, Veloso V, Iacca G. A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization. *Expert Systems with Applications*, 2014,41(16):7077–7049. [doi: 10.1016/j.eswa.2014.06.032]
- [30] De Melo V, Iacca G. A CMA-ES-Based 2-Stage memetic framework for solving constrained optimization problems. In: *Proc. of the 2014 IEEE Symp. on Foundations of Computational Intelligence (FOCI)*. Orlando: IEEE Press, 2014. 143–150. [doi: 10.1109/FOCI.2014.7007819]
- [31] Saha C, Das S, Pal K, Mukherjee S. A fuzzy rule-based penalty function approach for constrained evolutionary optimization. *IEEE Trans. on Cybernetics*, 2014,46(12):2953–2965. [doi: 10.1109/TCYB.2014.2359985]
- [32] Deb K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 2000,186(2-4):311–338. [doi: 10.1016/S0045-7825(99)00389-8]
- [33] Zielinski R, Laur R. Constrained single-objective optimization using differential evolution. In: *Proc. of the 2006 IEEE Int'l Conf. on Evolutionary Computation*. Vancouver: IEEE Press, 2006. 223–230. [doi: 10.1109/CEC.2006.1688312]
- [34] Mezura-Montes E, Coello CAC. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. on Evolutionary Computation*, 2005,9(1):1–17. [doi: 10.1109/TEVC.2004.836819]
- [35] Mezura-Montes E, Velazquez-Reyes J, Coello Coello CA. Modified differential evolution for constrained optimization. In: *Proc. of the 2006 IEEE Int'l Conf. on Evolutionary Computation*. Vancouver: IEEE Press, 2006. 332–336. [doi: 10.1109/CEC.2006.1688286]
- [36] Cui CG, Li YJ, Wu TJ. A relative feasibility degree based approach for constrained optimization problems. *Frontiers of Information Technology and Electronic Engineering*, 2010,11(4):249–260. [doi: 10.1631/jzus.C0910072]
- [37] Ullah A, Sarker R, Cornforth D. Search space reduction technique for constrained optimization with tiny feasible space. In: Keijzer M, ed. *Proc. of the 10th Annual Conf. on Genetic and Evolutionary Computation*. New York: ACM Press, 2008. 881–888. [doi: 10.1145/1389095.1389268]
- [38] Elsayed SM, Sarker RA, Essam DL. GA with a new multi-parent crossover for constrained optimization. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2011)*. New Orleans: IEEE Press, 2011. 857–864. [doi: 10.1109/CEC.2011.5949708]
- [39] Sarker RA, Elsayed SM, Ray T. Differential evolution with dynamic parameters selection for optimization problems. *IEEE Trans. on Evolutionary Computation*, 2014,18(5):689–707. [doi: 10.1109/TEVC.2013.2281528]
- [40] Wang Y, Wang BC, Li HX, Yen GG. Incorporating objective function information into the feasibility rule for constrained evolutionary optimization. *IEEE Trans. on Cybernetics*, 2015,46(12):2938–2952. [doi: 10.1109/TCYB.2015.2493239]
- [41] Runarsson TP, Yao X. Search biases in constrained evolutionary optimization. *IEEE Trans. on Systems, Man, Cybernetics (C)*, 2005,35(2):233–243. [doi: 10.1109/TSMCC.2004.841906]
- [42] Zhang M, Luo WJ, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 2008,178(15):3043–3074. [doi: 10.1016/j.ins.2008.02.014]
- [43] Bu C, Luo W, Zhu T. Differential evolution with a species-based repair strategy for constrained optimization. In: *Proc. of the 2014 IEEE Congress on Evolutionary Computation*. Beijing: IEEE Press, 2014. 967–974. [doi: 10.1109/CEC.2014.6900526]
- [44] Brest J, Borko B, Zumer V. An improved self-adaptive differential evolution algorithm in single objective constrained real-parameter optimization. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2010)*. Barcelona: IEEE Service Center, 2010. 1073–1078. [doi: 10.1109/CEC.2010.5585931]
- [45] Takahama T, Sakai S. Efficient constrained optimization by the ϵ constrained rank-based differential evolution. In: *Proc. of the 2012 IEEE Congress on Evolutionary Computation*. Brisbane: IEEE Press, 2012. 1–8. [doi: 10.1109/CEC.2012.6256111]
- [46] Takahama T, Sakai S, Iwane N. *AI 2005: Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2005. 389–400. [doi: 10.1007/11589990_41]
- [47] Takahama T, Sakai S. *Soft Computing as Transdisciplinary Science and Technology*. Muroan: Springer-Verlag, 2005. 1019–1029. [doi: 10.1007/3-540-32391-0]
- [48] Wang Y, Cai ZX, Zhou YR, Xiao CX. Constrained optimization evolutionary algorithms. *Ruan Jian Xue Bao/Journal of Software*, 2009,20(1):11–29 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3363.htm> [doi: 10.3724/SP.J.1001.2009.03363]

- [49] Surry PD, Radcliffe NJ. The COMOGA method: Constrained optimization by multi-objective genetic algorithm. *Control and Cybernetics*, 1997,26(3):391–412.
- [50] Venkatraman S, Yen GG. A generic framework for constrained optimization using genetic algorithms. *IEEE Trans. on Evolutionary Computation*, 2005,9(4):424–435. [doi: 10.1109/TEVC.2005.846817]
- [51] Coello CAC. Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization*, 2000,32(3):275–308. [doi: 10.1080/03052150008941301]
- [52] Gong WY, Cai ZH. A multiobjective differential evolution algorithm for constrained optimization. In: *Proc. of the 2008 IEEE Congress on Evolutionary Computation*. Hong Kong: IEEE Press, 2008. 181–188. [doi: 10.1109/CEC.2008.4630796]
- [53] Reynoso-Meza G, Blasco X, Sanchis J, Martínez M. Multiobjective optimization algorithm for solving constrained single objective problems. In: *Proc. of the 2010 IEEE Congress on Evolutionary Computation*. Barcelona: IEEE Service Center, 2010. 3418–3424. [doi: 10.1109/CEC.2010.5586408]
- [54] Singh H, Ray T, Smith W. Performance of infeasibility empowered memetic algorithm for CEC 2010 constrained optimization problems. In: *Proc. of the IEEE Congress on Evolutionary Computation*. Barcelona: IEEE Service Center, 2010. 3770–3777. [doi: 10.1109/CEC.2010.5585946]
- [55] Cai ZX, Wang Y. Combining multiobjective optimization with differential evolution to solve constrained optimization problems. *IEEE Trans. on Evolutionary Computation*, 2012,16(1):117–134. [doi: 10.1109/TEVC.2010.2093582]
- [56] Wang Y, Cai ZX, Guo G, Zhou YR. A dynamic hybrid framework for constrained evolutionary optimization. *IEEE Trans. on Systems, Man, and Cybernetics (B)*, 2012,42(1):203–217. [doi: 10.1109/TSMCB.2011.2161467]
- [57] Li XS, Zhang GS. Biased multiobjective optimization for constrained single-objective evolutionary optimization. In: *Proc. of the 11th World Congress on Intelligent Control and Automation (WCICA)*. Shenyang: IEEE, 2014. 891–896. [doi: 10.1109/WCICA.2014.7052834]
- [58] Gao WF, Yen G, Liu SY. A dual-population differential evolution with coevolution for constrained optimization. *IEEE Trans. on Cybernetics*, 2014,45(5):1108–1121. [doi: 10.1109/TCYB.2014.2345478]
- [59] Wang Y, Cai ZX, Zhou YR, Zeng W. An adaptive trade-off model for constrained evolutionary optimization. *IEEE Trans. on Evolutionary Computation*, 2008,12(1):80–92. [doi: 10.1109/TEVC.2007.902851]
- [60] Wang Y, Cai ZX. Hybrid differential evolution and adaptive trade-off model to solve constrained optimization problems. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2010)*. Barcelona: IEEE Press, 2010. 1–5. [doi: 10.1109/CEC.2010.5586189]
- [61] Elsayed SM, Sarker RA, Essam D. Integrated strategies differential evolution algorithm with a local search for constrained optimization. In: *Proc. of the IEEE Congress of Evolutionary Computation*. New Orleans: IEEE Press, 2011. 2618–2625. [doi: 10.1109/CEC.2011.5949945]
- [62] Jiao LC, Li L, Shang RH, Liu F, Stolkin R. A novel selection evolutionary strategy for constrained optimization. *Information Sciences*, 2013,239:122–141. [doi: 10.1016/j.ins.2013.03.002]
- [63] Wang Y, Liu H, Cai ZX, Zhou YR. An orthogonal design based constrained evolutionary optimization algorithm. *Engineering Optimization*, 2007,39(6):715–736. [doi: 10.1080/03052150701280541]
- [64] Cai ZX, Wang Y. A multiobjective optimization based evolutionary algorithm for constrained optimization. *IEEE Trans. on Evolutionary Computation*, 2006,10(6):658–675. [doi: 10.1109/TEVC.2006.872344]
- [65] Tasgetiren MF, Suganthan PN, Pan Q-K, Mallipeddi R, Sarman S. An ensemble of differential evolution algorithms for constrained function optimization. In: *Proc. of the IEEE Congress on Evolutionary Computation Vol.5*. Barcelona: IEEE Service Center, 2010. 967–975. [doi: 10.1109/CEC.2010.5586396]
- [66] Mani A, Parvarhan C. A novel hybrid constraint-handling technique for evolutionary optimization. In: *Proc. of the 11th Conf. on Congress on Evolutionary Computation (CEC 2009)*. Trondheim: IEEE Press, 2009. 2577–2583. [doi: 10.1109/CEC.2009.4983265]
- [67] Gan M, Peng H, Peng XY, Chen XH, Inoussa G. An adaptive decision maker for constrained evolutionary optimization. *Applied Mathematics and Computation*, 2010,215(12):4172–4184. [doi: 10.1016/j.amc.2009.12.038]

- [68] Deb K, Datta R. A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach. In: Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010). Barcelona: IEEE Press, 2010. 1–8. [doi: 10.1109/CEC.2010.5586543]
- [69] Datta R, Deb K. An adaptive normalization based constrained handling methodology with hybrid bi-objective and penalty function approach. In: Proc. of the IEEE Congress on Evolutionary Computation (CEC 2012). Brisbane: IEEE Press, 2012. 1–8. [doi: 10.1109/CEC.2012.6252955]
- [70] Datta R, Deb K. Individual penalty based constraint handling using a hybrid bi-objective and penalty function approach. In: Proc. of the 2013 IEEE Congress on Evolutionary Computation. Cancun: IEEE, 2013. 2720–2727. [doi: 10.1109/CEC.2013.6557898]

附中文参考文献:

- [48] 王勇,蔡自兴,周育人,肖赤心.约束优化进化算法.软件学报,2009,20(1):11–29. <http://www.jos.org.cn/1000-9825/3363.htm> [doi: 10.3724/SP.J.1001.2009.03363]



李智勇(1971—),男,湖南长沙人,博士,教授,博士生导师,CCF 专业会员,主要研究领域为嵌入式计算与智能系统,智能优化计算.



陈少森(1987—),男,博士生,主要研究领域为智能算法,集群计算.



黄滔(1991—),男,硕士生,主要研究领域为约束优化算法.



李仁发(1956—),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为嵌入式计算系统,CPS,智能系统.