

约束进化算法及其应用研究综述

李 笠 李广鹏 常 亮 古天龙

桂林电子科技大学广西可信软件重点实验室 广西 桂林 541004

摘 要 约束优化问题广泛存在于科学研究和工程实践中,其对应的约束优化进化算法也成为了进化领域的重要研究方向。约束优化进化算法的本质问题是如何有效地利用不可行解和可行解的信息,平衡目标函数和约束条件,使得算法更加高效。首先对约束优化问题进行定义;然后详细分析了目前主流的约束进化算法,同时,基于不同的约束处理机制,将这些机制分为约束和目标分离法、惩罚函数法、多目标优化法、混合法和其他算法,并对这些方法进行了详细的分析和总结;接着指出约束进化算法亟待解决的问题,并明确指出未来需要进一步研究的方向;最后对约束进化算法在工程优化、电子和通信工程、机械设计、环境资源配置、科研领域和管理分配等方面的应用进行了介绍。

关键词 工程实践;约束优化问题;进化算法;约束优化进化算法;约束处理机制

中图法分类号 TP181

Survey of Constrained Evolutionary Algorithms and Their Applications

LI Li, LI Guang-peng, CHANG Liang and GU Tian-long

Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

Abstract Constrained optimization problems exist widely in scientific research and engineering practice, and the corresponding constrained evolutionary algorithms have become an important research direction in the field of evolutionary computation. The essential problem of constrained evolutionary algorithm is how to effectively use the information of infeasible and feasible solutions and balance the objective function and constraints to make the algorithm more efficient. Firstly, this paper defines the problem of constraint optimization. Then it analyzes the current mainstream constraint evolution algorithms in detail. At the same time, based on different constraint handling mechanisms, these mechanisms are divided into constraint and objective separation methods, penalty function methods, multi-objective optimization methods, hybrid methods and so on, and these methods are analyzed and summarized comprehensively. Next, it points out the urgent problems that need to be solved as well as the research direction. Finally, the application of constrained evolutionary algorithm in engineering optimization, electronic and communication engineering, mechanical design, environmental resource allocation, scientific research and management allocation are introduced.

Keywords Engineering practice, Constraint optimization problem, Evolutionary algorithm, Constraint optimization evolutionary algorithm, Constraint handling mechanism

1 引言

在实际科学和工程应用中通常存在限制条件较多且难以求解的问题,这类问题称为约束优化问题(Constrained Optimization Problems, COPs)。因此,对这类广泛存在的问题进行研究具有一定的实际意义。最初,求解约束问题的算法通常是基于梯度信息的,如简约梯度法、投影梯度法和内点惩罚函数法等。这些算法往往只适用于约束条件简单的问题,对可行域不连续和不可导等问题的求解性能则较差。

进化算法是一种模拟自然进化过程的全局优化方法,它采用选择、交叉、变异等操作来完成自然进化的过程,引导种

群逐步趋向最优解。约束优化进化算法(Constrained Optimization Evolutionary Algorithms, COEAs)是一类利用进化算法搜索效率高、通用性强以及健壮性好等特点来求解约束优化问题的方法,吸引了众多学者的兴趣,并且提出了大量的约束处理技术^[1-2]。此外,从2006年至今,IEEE进化计算国际会议(IEEE Congress on Evolutionary Computation, CEC)每年都会举办相关的专题讨论,促进了COEAs的研究和发展。

约束优化进化算法能够充分发挥进化算法的高效搜索性能,并根据约束处理技术合理地处理约束条件,高效地求解约束问题。在研究早期,研究者们提出了大量的约束处理技术,根据不同的约束处理方式,可将其分为5类:惩罚函数法、可

到稿日期:2020-06-24 返修日期:2020-09-26

基金项目:国家自然科学基金项目(62006058, U1811264, U1711263, 61966009);广西自然科学基金项目(2018GXNSFAA138090, 2018GXNSFDA281049, 2017GXNSFAA198283)

This work was supported by the National Natural Science Foundation of China(62006058, U1811264, U1711263, 61966009) and Guangxi Natural Science Foundation(2018GXNSFAA138090, 2018GXNSFDA281049, 2017GXNSFAA198283).

通信作者:李笠(lili@guet.edu.cn)

行性法则、随机排序法、多目标优化法、混合法。惩罚函数法给违反约束条件的个体添加相应的惩罚因子，从而降低不可行个体被选择的概率，其不足之处在于如何选取合适的惩罚参数。可行性法则是由 Deb^[3]提出的，是经典的约束处理方法之一，其规定可行解始终优于不可行解，导致不可行解难以进入进化群体。因此，可行性法则无法应对可行域较小的约束问题。随机排序法根据概率参数 p_f 来平衡目标函数和惩罚函数，然而 p_f 的选取较为困难，往往需要大量实验来进行调整。多目标优化法将约束条件转化成为额外的优化目标，使约束问题变为多目标优化问题，增加了求解的难度和计算代价。混合法是约束优化进化算法的研究热点，其能够利用多种不同约束处理技术的优势来求解约束问题。

纵观国内外，有关约束优化进化领域的综述研究少之又少。国外的综述主要针对 2002 年以前的约束优化进化算法，对近年来的相关算法讨论得较少；国内的综述也是寥寥无几，Wang 等^[4]的讨论主要针对 2009 年之前的研究成果。但是，近年来约束优化算法发展迅速，国内外很多著名专家和学者都对其进行了深入地研究，并取得了众多的研究成果。目前的综述在约束优化进化算法方面的介绍较为简单，仅侧重于对经典算法的介绍，并且缺少对相关应用领域的归纳总结以及对目前蓬勃发展的新的研究成果的分析与整理。任何高效的算法最终都需要落地并解决实际问题。我们认为，对约束优化问题的应用方面的综述有助于开发新的约束处理方法与开展新的应用研究。基于以上讨论，本文对约束优化进化算法及其应用进行了详细的综述。本文首先介绍了约束优化问题及其相关定义，然后对进化算法和约束处理技术进行了详细的介绍和分析，并对该领域的最新研究展开了讨论，最后对约束优化进化算法亟待解决的问题和相关应用展开了综述。

2 约束优化问题及相关定义

在实际应用中，约束问题往往有多个冲突目标，同时还需要满足多个等式和不等式的约束条件。约束多目标优化问题的定义如下：

$$\begin{aligned} \min \quad & F(x) = (f_1(x), \dots, f_m(x))^T \\ \text{s. t.} \quad & g_i(x) \leq 0, i = 1, \dots, q \\ & h_j(x) = 0, j = 1, \dots, p \end{aligned} \tag{1}$$

其中， $F(x) = (f_1(x), \dots, f_m(x))^T$ 为 m 维目标向量， $x = [x_1, \dots, x_n]$ 为决策变量， S 表示 n 维决策空间， $g_i(x)$ 为第 i 个不等式的约束条件， $h_j(x)$ 为第 j 个等式的约束条件。

在约束进化算法中，个体 x 在第 j 个约束条件下的约束违反程度表示为：

$$G_j(x) = \begin{cases} \max\{g_j(x), 0\}, & 1 \leq j \leq q \\ \max\{|h_j(x)| - \sigma, 0\}, & 1 \leq j \leq p \end{cases} \tag{2}$$

在约束算法中，个体 x 的违反程度由其在所有约束条件下的约束违反程度决定，通常表示为：

$$\phi(x) = \sum \|\max\{0, g_i(x)\}\| + \sum \|\max\{0, |h_j(x) - \sigma|\}\| \tag{3}$$

其中， σ 是约束的容忍阈值，通常设置为 0.0001。如果解 x 的 $\phi(x)$ 为 0，则解 x 为可行解；如果 $\phi(x)$ 的值不为 0，则解 x 为

不可行解。满足不等式约束和等式约束的选择区域，称为可行域；反之，称为不可行域。

不同的约束问题的约束条件之间存在差异，可能出现个体的违约度被某个或某些约束条件主导的情况，因此需要采用标准化函数来规范约束条件。在此，我们定义个体 x 的违约度 $\phi(x)$ 为该个体在每个约束条件上的违约度的平均值，具体表达式如下：

$$G_j^{\max} = \max_{i=1, \dots, N} (G_j(x_i)), j \in \{1, \dots, p+q\} \tag{4}$$

$$\phi_{norm}(x) = \frac{1}{m} \sum_{j=1}^m \frac{G_j(x_i)}{G_j^{\max}}, i \in \{1, 2, \dots, N\} \tag{5}$$

其中， $G_j^{\max}$ 为第 j 个目标上的最大违约度， N 为种群规模。

Pareto 支配关系广泛应用于多目标优化问题，同样适用于约束优化，相关的定义如下。

定义 1 (Pareto 支配) 对于两个个体 x 和 y ， $F(x)$ 和 $F(y)$ 为其目标向量， x 支配 y ，当且仅当：

- (1) $\forall i \in \{1, \dots, m\}$ 满足 $f_i(x) \leq f_i(y)$
- (2) $\exists j \in \{1, \dots, m\}$ 满足 $f_j(x) < f_j(y)$

定义 2 (Pareto 最优解) 决策向量 $x \in S$ 称为 S 上的 Pareto 最优解，当且仅当 $\nexists y \in S$ ，使得 $y \succ x$ 。

定义 3 (Pareto 最优解集) Pareto 最优解集为： $p^* = \{x \in S | \nexists y \in S, y \succ x\}$ 。

定义 4 (Pareto 前沿 (Pareto front, PF)) p^* 为最优解集，PF 定义为 $pf^* = \{u = f(x) | x \in p^*\}$ 。

3 约束处理机制

近年来，约束进化算法领域的研究人员提出了大量的约束处理机制。根据不同的约束处理机制，本文将约束优化算法间接分为 4 类(见图 1)：1)约束和目标分离法；2)惩罚函数法；3)多目标优化法；4)混合法。

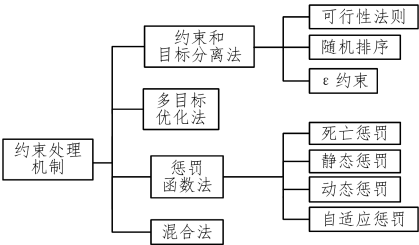


Fig. 1 Classification of constraint handling mechanism

3.1 约束和目标分离法

图 2 给出了约束和目标分离法的执行过程，其本质是将约束条件和目标函数分别用不同的方法来处理。本文将该方法分为 3 种：可行性准则、随机排序法、 ϵ 约束法。

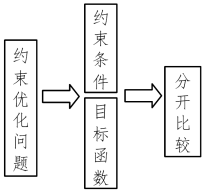


Fig. 2 Implementation process of constraints and objectives separation method

3.1.1 可行性准则

近年来,研究人员提出了大量的约束处理技术,其中 Deb 提出的可行性法则^[3]是最为经典的约束处理机制之一。对两个解 x_1 和 x_2 进行比较,它们的目标向量为 $F(x_1)$ 和 $F(x_2)$, 违约度为 $\phi(x_1)$ 和 $\phi(x_2)$,其约束支配关系如下:

$$\begin{aligned} & (F(x_1), \phi(x_1)) \succ (F(x_2), \phi(x_2)) \\ & \begin{cases} F(x_1) > F(x_2), & \text{if } \phi(x_1), \phi(x_2) = 0 \\ F(x_1) > F(x_2), & \text{if } \phi(x_1) = \phi(x_2) \\ \phi(x_1) < \phi(x_2), & \text{otherwise} \end{cases} \end{aligned} \quad (6)$$

显然,当 x_1 和 x_2 都为可行解时,目标函数值较小的解占优;当 x_1 和 x_2 的违约度相同时,目标函数值较小的解占优;当 x_1 和 x_2 的违约度不同时,违约度较小的解占优。

文献[5]将约束处理准则(Constraint Dominance Principle, CDP)嵌入 NSGA-II 中,得到 NSGA-II-CDP,并将种群划分为可行和不可行两个子种群。在可行子种群中采用 Pareto 关系进行排序,在不可行种群中采用 CDP 进行排序。NSGA-II-CDP 首先选择可行种群中的个体,然后选择不可行种群中的个体,直到子代种群规模为 N (种群规模)。

定义 5(约束条件的相对可行度) F 为可行域, F_i 为满足第 i 个约束条件所对应的空间,可行域 F 和满足第 i 个约束条件所对应的空间 F_i 之间的比值 $\rho_i = |F|/|F_i|$,即为第 i 个约束条件的相对可行度。

定义 6(相对公共满足约束集) x, y 为种群中的两个候选解, G 表示问题的约束集, G_i 和 B_i 分别为个体 x 满足约束条件的集合与违反约束条件的集合, G_j 和 B_j 为个体 y 满足约束集和违反约束集的集合, G_i 和 B_j 的交集 $Q(x \rightarrow y)$,即为 x 相对于 y 的非共同满足约束集合。

定义 7(个体的相对可行度) 设 x 和 y 为种群中的两个不同个体, $Q(x \rightarrow y)$ 指 x 相对于 y 的非共同满足的约束集合, $\rho_k (k \in Q(x \rightarrow y))$ 指第 k 个约束条件的相对可行度。 x 相对于 y 的可行度为:

$$R(x \rightarrow y) = \begin{cases} 0, & Q(x \rightarrow y) = \Phi \\ 1, & Q(x \rightarrow y) = \{1, 2, \dots, m\} \\ \max_{i \in Q(x \rightarrow y)} \rho_i, & \text{otherwise} \end{cases} \quad (7)$$

Cui 等^[6]根据个体之间的优劣性提出了个体相对可行度(Relative Feasibility Degree, RFD)的概念(见式(7)),并将其与差分进化算法相结合,即 RFDDE(Relative Feasibility Degree Based on Differential Evolution)。对两个个体 x 和 y 进行比较时应遵照以下准则。

- (1) x, y 均为可行解时,目标函数值较小的个体占优;
- (2) x 为可行解, y 为不可行解时,可行解 x 占优;
- (3) x, y 均为不可行解时,比较其相对可行度,相对可行度大的解占优;若相对可行度相同,则违约度较小的个体占优。

该算法能够加快搜索过程,快速收敛到可行域并保持良好的分布性。但是 RFDDE 仍然对差分进化控制参数的区间较为敏感,需要根据大量实验来确定。此外,当约束条件的相对可行度相似时,会引起 RFDDE 算法的性能降低。

Cheng 等^[7]提出了一种新颖的算法,并结合了可行性准则,即 RVEA(Reference Vector-guided Evolutionary Algorithm)。该算法利用参考向量将搜索空间分为多个子空间,并根据个体到原点的距离以及个体与参考向量的角度构造角度惩罚距离(Angle-Penalized Distance, APD)。在每个子空间采用 CDP 和 APD 选择最优个体。此外, RVEA 提出了参考向量自适应更新策略和参考向量再生策略,分别用于解决种群分布不均匀和 PF 不规则的问题。角度惩罚距离 APD 的定义如下:

$$APD = d_{t,i,j} = (1 + P(\theta_{t,i,j})) \|f_{t,i}'\| \quad (8)$$

其中, t 为进化代数, j 为子空间的索引, $f_{t,i}'$ 为转换后的目标函数值, $\theta_{t,i,j}$ 为第 t 代时 $f_{t,i}'$ 和参考向量 $V_{t,j}$ 之间的角度, $P(\theta_{t,i,j})$ 为基于角度的惩罚函数, $\gamma_{t,i,j}$ 是参考向量 v_j 与其他参考向量之间最小的角度。

$$P(\theta_{t,i,j}) = M\left(\frac{t}{t_{\max}}\right)^a \frac{\theta_{t,i,j}}{\gamma_{v_{ij}}}, \gamma_{v_{ij}} = \min_{i \in \{1, \dots, N\}, i \neq j} \langle v_{t,i}, v_{t,j} \rangle \quad (9)$$

Fan 等^[8]提出了一种新颖的约束处理机制,基于角度约束处理(Angle-based Constrained Dominance Principle, ACDP)。该方法利用角度信息维持种群的多样性,同时探索不可行域,保证种群的收敛性。两个个体 x_1 和 x_2 之间的角度的定义如下:

$$\begin{aligned} & \text{angle}(x_1, x_2, z^*) = \\ & \arccos\left(\frac{(F(x_1) - z^*)(F(x_2) - z^*)}{\|F(x_1) - z^*\| * \|F(x_2) - z^*\|}\right) \end{aligned} \quad (10)$$

其中, z^* 为理想点, $z^* = \min\{f_i(x) | x \in S\}$ 。

ACDP 根据以下准则来评判个体的优劣:

- (1) 如果两个个体 x_1 和 x_2 均为可行解,则目标函数值较小的个体较优;
- (2) 如果两个个体 x_1 和 x_2 至少有一个不可行解,且 $\text{angle}(x_1, x_2, z^*) \leq \theta$, 则违约度较小的个体较优;
- (3) 如果两个个体 x_1 和 x_2 至少有一个不可行解,且 $\text{angle}(x_1, x_2, z^*) \geq \theta, r < p_f$, 则目标值较小的个体较优,否则不可比较。

其中, θ 为阈值, r 为一个随机数, p_f 为当代可行比。ACDP 利用两个解之间的角度信息,在进化过程中动态地保持种群的多样性,同时利用可行比来保持种群的收敛性和可行性之间的平衡。

可行性准则原理简单、方便执行,但其完全忽略了不可行解的信息,导致算法存在如下问题:

- (1) 过分重视快速趋于可行域,导致搜索方向单一;
- (2) 当存在多个不连续的可行域时,往往会无法穿过不可行域;
- (3) 通常会忽略对可行域边界的探测;
- (4) 忽略了不可行解携带的有用信息,不利于算法快速收敛。

可行性准则在解决不连续的可行域问题时,往往难以穿越不可行域(见图 3),从而导致种群深陷局部最优解,无法取得全局最优解。

对此,研究人员将重心放在设计高效的搜索算法上,以便

通过高效的搜索算法来弥补可行性准则求解约束优化问题时的不足。如文献[9-11]对差分进化算法进行改进作为搜索算法;Dhadwal 等^[12]将改变的粒子群算法作为搜索算法,提高了算法的效率。

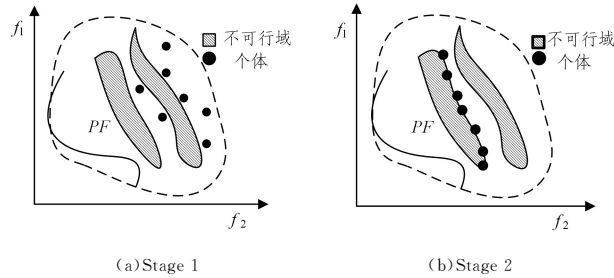


图 3 不连续可行域问题的进程

Fig. 3 Process of discontinuous feasible region problem

3.1.2 随机排序法

Runarsson 等^[13]于 2000 年提出了随机排序法(Stochastic Ranking, SR),该算法采用随机排序来平衡目标函数和惩罚函数。该方法根据概率参数 pf 来决定选择目标函数或个体违约度,进而评价两个个体的优劣(见图 4)。当 $rand < pf$ 时,基于目标函数值来评价个体优劣;当 $rand \geq pf$ 时,基于违约度来评价个体优劣。

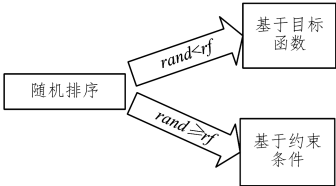


图 4 随机排序选择评价流程

Fig. 4 Random sorting selection process

该方法将违约度和目标函数间的矛盾问题转化为基于概率的随机排序问题,不需要改变惩罚因子。其不足之处在于:概率参数 pf 的值该如何设定。近年来,研究人员也有大量的创新。例如,文献[14]提出了动态改变 pf 值的方法,文献[15]结合蚁群算法来处理约束优化问题。

3.1.3 ϵ 约束法

在 2006 年 CEC 的约束进化算法专题会议上,Takahama 等^[16]提出了一种基于 ϵ 约束的处理方法。其核心思想是,通过 ϵ 的设定值将违约程度小于 ϵ 的个体视为可行解,充分利用了不可行域中目标函数值较好的不可行解的信息,具有更好的收敛性。文献[17]通过以下准则来比较个体的优劣:

$$\begin{aligned} &(F(x_1), \phi(x_1)) \succ_{\epsilon} (F(x_2), \phi(x_2)) \\ &\begin{cases} F(x_1) > F(x_2), & \text{if } \phi(x_1), \phi(x_2) \leq \epsilon \\ F(x_1) > F(x_2), & \text{if } \phi(x_1) = \phi(x_2) \\ \phi(x_1) < \phi(x_2), & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

相比可行性准则, ϵ 约束法利用了不可行解,放宽了对目标函数的比较条件,缓解了对约束条件的贪婪性。如图 5 所示,将不可行解 x 与可行解 y 进行比较,不可行解 x 的违约度 $\phi(x) < \epsilon$,且 $F(x)$ 优于 $F(y)$,这时不可行解 x 占优。

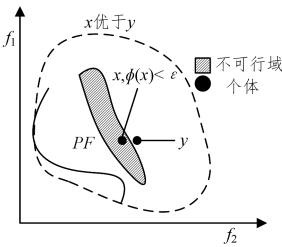


图 5 ϵ 约束中不可行解和可行解的比较

Fig. 5 Comparison of infeasible solution and feasible solution in ϵ constraint

ϵ 值的设定是 ϵ 约束法的关键问题。当 $\epsilon = 0$ 时, ϵ 约束处理方法和可行性准则相同;当 $\epsilon = \infty$ 时, ϵ 约束处理方法就变为了 Pareto 关系。传统的 ϵ 设置方式如下:

$$\begin{aligned} \epsilon(0) &= \phi(x^{\theta}) \\ \epsilon(G) &= \begin{cases} \epsilon(0)(1 - G/T_c)^{c_p}, & 0 \leq G < T_c \\ 0, & G \geq T_c \end{cases} \end{aligned} \tag{12}$$

其中, x^{θ} 是根据初始种群个体按照违约度降序排列后的第 θ 个个体, c_p 是控制 ϵ 下降速度的一个参数。通过实验证明, θ 的值通常为 $0.05 * N$, $c_p \in [2, 10]$, T_c 通常设置在区间 $[0.1 * T_{max}, 0.8 * T_{max}]$, T_{max} 为最大进化代数。 ϵ 约束法具有较好的收敛性,但是在进化过程中, ϵ 值的设置可能会出现不合理的情况。

文献[18]指出,在进化过程中,如果 $\epsilon(G)$ 的值小于所有不可行解的违约度, ϵ 约束法将再次转为 CDP,种群的多样性将难以维持。因此,本文提出了一种基于进化代数和个体的最大最小违约度自适应调整 ϵ 的约束处理方法。该算法加大了在不可行域的搜索压力,使得在其他潜在区域找到可行解的机会更大,同时防止了算法早熟。文献[19]采用了一种自适应差分进化算法(jDEsoco)来解决单目标实值的约束优化问题;文献[20]提出了一种基于 ϵ 等级的约束差分进化算法来求解约束问题。

上文对约束和目标分离法的 3 种不同约束处理技术进行了详细介绍。根据上述 3 种方法的个体比较方式,简述其所具备的优点和存在的缺点,如表 1 所列。

表 1 不同约束处理技术的对比

Table 1 Comparison of different constraint handling technologies

约束处理技术	优点	缺点
可行性准则	执行简单,收敛速度快	搜索方向单一,容易陷入局部最优解
随机排序	采用 pf 引入目标函数信息	pf 的参数调节困难
ϵ 约束法	通过 ϵ 的调节,引入目标函数信息	ϵ 的选择和调节困难

3.2 惩罚函数法

近年来,因惩罚函数的执行过程简单、高效,研究人员将其广泛应用于约束进化算法中以解决约束问题。其核心思想为:基于个体的违约程度构造惩罚项,将目标函数值 $f(x)$ 和惩罚函数值 $p(x)$ 相加,构造惩罚适应度函数 $fitness(x)$ 。本文按照其特性将其分为 4 类(见图 6):死亡惩罚函数法、静态惩罚函数法、动态惩罚函数法和自适应惩罚函数法。

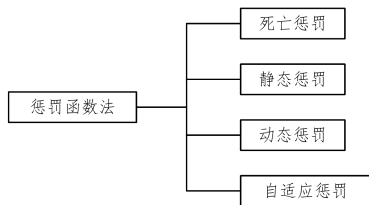


图6 惩罚函数的分类

Fig. 6 Classification of penalty function

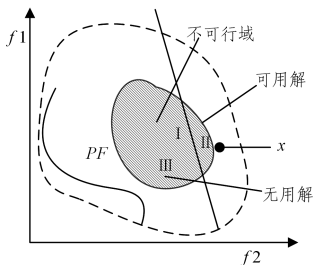
常规的惩罚适应度函数的定义如下:

$$fitness(x) = f(x) + \alpha \sum_{i=1}^m r_i G_i(x), \sum_{i=1}^m r_i = 1 \quad (13)$$

其中, α 为惩罚系数, r_i 是研究人员对每个约束条件定义的约束违反水平数。

为了平衡目标函数值和违约度,合理的惩罚系数至关重要。显然,当 α 值过小时,对不可行解的惩罚不够,不能较好地区分可行解和不可行解,因此不能保证可行解的选择;当 α 值过大时,对不可行解的惩罚过大,不利于不可行解的选择,不能合理地利用不可行解的信息,因此不利于算法收敛。

如图7所示,可行解 x 在二维搜索空间中面临不可行域 I 时,将不可行域大致分为两部分:II 可用解和 III 无解。

图7 可行解 x 将不可行域划分为两部分Fig. 7 Feasible solution x divides the infeasible solution region into two parts

3.2.1 死亡惩罚函数

死亡惩罚函数法^[21]指不可行解的惩罚适应度为 0,拒绝所有不可行解,完全忽略不可行解在种群进化中的有效信息。当初始种群中全部为不可行解时,种群中所有个体的惩罚适应度值都为 0,此时需要重新初始化种群。死亡惩罚函数法不适合于低可行比的约束问题。因此,死亡惩罚函数认为不可行域 I 中无可用解,即可行域 II 的面积为 0,完全忽略不可行解。

3.2.2 静态惩罚函数

静态惩罚函数法指在种群进化过程中惩罚系数为定值。但种群进化过程中往往会出现许多 α 值不合理的情况,因此效果不佳。文献[22]提出了一种分层惩罚参数法,惩罚适应度函数如下:

$$fitness(x) = f(x) + \sum_{i=1}^m R_{k,i} \times G_i(x) \quad (14)$$

其中, $R_{k,i}$ ($k=1, \dots, q, i=1, \dots, p$) 为惩罚系数, q 为研究人员对每个约束条件设置的约束违反水平数。该方法根据约束违反度分为多个层次,不同层次有不同的惩罚系数 $R_{k,i}$,对于违反程度越大的个体,惩罚力度就越大。该方法的实现过程相对简单,不足之处在于产生了较多参数,对于不同的约束问题,都需要反复进行实验,以寻找合适的惩罚系数。此外,文

献[23]给出了一种两阶段免疫算法。静态惩罚函数中惩罚系数往往是定值,因此可用解区域 II 的面积不会发生改变。

3.2.3 动态惩罚函数

文献[24]提出了一种随着进化代数的变化而改变惩罚系数的动态惩罚函数法。其核心思想为:进化初期对不可行解采用较小的惩罚系数,充分利用不可行解的信息,对不可行域进行相应的探索;进化后期对不可行解采用较大的惩罚系数,使得算法在可行域中集中搜索,寻找最优的可行解。惩罚函数适应度的定义如下:

$$fitness(x) = f(x) + (C \times t)^\alpha \sum_{i=1}^m G_i^2(x) \quad (15)$$

其中, α, β 为用户自行设定的常数, t 为进化的代数。与静态惩罚函数和死亡惩罚函数相比,动态惩罚函数能够更好地适应约束问题,在性能方面更加优越,不足之处在于需要反复实验来找到理想的初始惩罚系数。在动态惩罚函数中,区域 II 的面积随着惩罚系数的增大而逐渐减小。

3.2.4 自适应惩罚函数

近年来,很多研究人员在如何构造合理的自适应惩罚函数方面进行了很多研究。Tessema 等提出了一种自适应的惩罚函数^[25],其利用当前种群中的可行比(可行解在种群中所占的比例)自适应地调整惩罚系数,从而控制惩罚的强度。该方法采用双惩罚,旨在挖掘种群中最佳的不可行解,然后逐步向最优解逼近。该方法的惩罚适应度函数如下:

$$fitness(x) = \begin{cases} f_{norm} \\ v_{norm}(x) \\ \sqrt{f_{norm}(x)^2 + v_{norm}(x)^2} + [(1-rf)f_{norm}(x) + rf \times v_{norm}(x)] \end{cases} \quad (16)$$

其中, $f_{norm}(x)$ 为标准化的目标函数, $v_{norm}(x)$ 为标准化的违约度, rf 为当代种群的可行比。

在进化过程中,当种群不包含可行解时,目标函数值被忽略,仅依靠其违约度搜索最佳不可行解;当种群中只包含可行解时,违约度被忽略,依靠目标函数寻找更优的个体;当种群中同时包含可行解和不可行解时,标准化适应值越低的个体越好。该算法充分利用了低违约度和低目标函数值的不可行个体,来寻找可行个体并产生高质量的解。

Melo 等^[26]将自适应惩罚函数和改进的自适应协方差矩阵进化策略(Covariance Matrix Adaptation Evolution Strategy, CMA-ES)相结合,得到了新的约束进化算法。在进化过程中,惩罚系数随种群中不可行解的约束违反信息而变化,其方法如下:

$$fitness(x) = bestfeas + |f(x) - bestfeas| \times |v(x)/max_penalty| \quad (17)$$

其中, $bestfeas$ 是可行解中最优的目标函数值,当种群中没有可行解时, $bestfeas$ 自行设定; $max_penalty$ 指不可行解中最大的约束违反度。由式(17)可知,违反约束程度越大的个体,其惩罚强度也越大;违约度较小的个体可能会具有较好的适应度,但其适应度永远差于种群中最好可行解的适应度。

文献[27]结合粗糙集理论(Rough Set Theory, RST)和遗传算法,提出了粗糙惩罚函数遗传算法(Rough Penalty Genetic Algorithm, RPGA),该算法根据违约度和进化代数自适

应调整惩罚系数。此外,其通过惩罚函数加速了可行空间的搜索和 RST 释放无效约束的惩罚,提高了搜索能力,有效地解决了约束优化问题。

粗糙集理论的基本概念是由感兴趣的对象的信息生成的不可分辨关系。由于缺乏辨别知识,人们无法根据现有的信息识别物体,不可分辨关系将不可分辨物体的颗粒作为基本依据来表达。该算法采用粗糙决策系统,目标值高于平均值的个体为“好”,否则为“差”,具体如下:

$$D(x_j)=\begin{cases} good, & \text{if } f(x_j)<f_{average} \\ bad, & \text{if } f(x_j)\geq f_{average} \end{cases} \quad (18)$$

RPGA 根据违约度和进化时间提出了一种新型的自适应惩罚函数。

$$\phi(x)=f(x)+\sum_{k=1}^m((C\times t)^{\pi(k,t)}\times\max(0,\Phi_k(x))^2) \quad (19)$$

其中, Φ_k 为约束条件 k 的违约度, C 为严重性参数, $\pi(k,t)$ 为惩罚参数,并根据进化代数和粗糙决策系统进行调整,具体如下:

$$\pi(k,t)=\begin{cases} \pi(k,t-1)\times\gamma_k, & \text{if } \mu_k=1 \\ \pi(k,t-1), & \text{if } \mu_k=0 \end{cases}, \pi(k,0)=2 \quad (20)$$

其中, k 为第 k 个约束条件, μ_k 为根据粗糙决策系统分类的结果, $\mu_k=1$ 表示分类为 *good*, $\mu_k=0$ 表示分类为 *bad*, γ_k 为调整参数。

此外,文献[28]提出了新颖的自适应惩罚函数。该算法利用进化中群体的状态信息(种群中的可行比和违约度)来自适应地调整每个约束条件的惩罚系数,同时与稳态遗传算法结合来求解工程问题。稳态情况下的自适应惩罚函数的变体如下。

(1)单调惩罚系数的自适应惩罚函数。惩罚系数取当前值 k^{pre} 和预测值 k^{new} 的较大值。

(2)带阻尼的自适应惩罚函数。惩罚系数由当前值和预测的新值之间的加权平均值决定。

根据上文对 4 种不同惩罚函数的综述,对其优点和缺点进行详细的总结,如表 2 所列。

表 2 4 种不同惩罚函数的对比

Table 2 Comparison of four different penalty functions

方法	优点	缺点
死亡惩罚	操作简单,快速收敛	忽略不可行解信息,容易陷入局部最优
静态惩罚	α 为定值,利用部分不可行解	α 的设置困难,种群进化中 α 会出现不合理现象
动态惩罚	α 随着种群进化动态调整	进化过程中, α 的设置出现不合理现象
自适应惩罚	自适应调整 α ,充分利用不可行解	自适应函数较为复杂

3.3 多目标优化法

由于惩罚函数法的惩罚系数设置得较为复杂,往往需要大量的实验来进行调整,并且容易引起约束和目标函数之间出现不平衡的问题。因此,研究人员提出了一种新的方法,即多目标优化法,其核心思想为将约束优化问题转变成多目标优化问题,如 $(f(x),G_1(x)\cdots G_{p+q}(x))$ 或 $(f(x),G(x))$ 。多目标优化法成功地避免了约束和目标函数的平衡问题,并且能够较好地解决约束优化问题。但是,多目标的求解难度较高,因此双目标的形式较为受欢迎。

Powell 等^[29]将可行解的适应度映射到 $(-\infty,1)$,将不可行解的适应度映射到 $(1,+\infty)$,使得可行解优于不可行解。其个体评估方法如下:

$$fitness(x)=\begin{cases} f(x), & \text{if feasible} \\ 1+rG(x), & \text{otherwise} \end{cases} \quad (21)$$

Fernando 等^[30]提出了一种类类似于多目标优化的 min-max 方法。其个体比较准则如下:

(1)当一个个体为可行解而另一个个体为不可行解时,可行解占优;

(2)当两个个体均为可行解时,目标函数较小的个体占优;

(3)当两个个体均为不可行解时,基于个体的最大违约度进行比较,拥有最小的最大违约度的个体占优。

文献[31]提出了一种三目标进化的新型框架 TiGE,该框架精心设计了收敛性和多样性两个指标,并将其与可行性结合成三目标,共同解决约束优化问题。此外,该框架概念简单,易于实例化,并使用多种平衡方案和排序方法(如非支配排序和随机排序),适用于约束多目标优化问题。平衡三目标的方案如下:

(1)3 个目标同时优化;

(2)首先处理收敛性和多样性之间的平衡,然后处理可行性和多目标优化之间的平衡。

此外,文献[32]提出了一种不可行解存档和替换机制的 CW(Cai and Wang’s algorithm)算法。该算法充分利用不可行解的信息,引导种群快速逼近可行域。

3.4 混合法

目前,大量约束处理技术可以高效地解决约束问题,但是当面临高维度、高复杂度的约束问题时,采用单一的约束处理技术往往达不到理想的效果。因此,许多研究者开始对具有自适应机制和混合机制的约束处理技术进行研究。混合法的核心理念如图 8 所示,将两种及两种以上不同的约束处理机制,或者多种不同进化策略相结合,同时优化和处理约束问题。混合法的关键是如何合理地使用不同约束处理技术的特点,从而高效地处理约束问题。

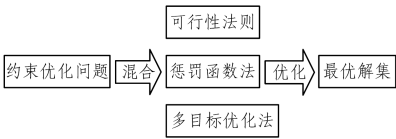


图 8 混合法的原理

Fig. 8 Principle of hybrid method

基于混合法的思想,Wang 等^[33]提出了一种自适应均衡模型(Adaptive Trade-off Model,ATM)。该模型根据进化过程中种群可行解和不可行解的数量将进化过程分为 3 个不同的阶段,并在不同的进化阶段采取不同的处理技术。当进化过程中种群中只有不可行解时,该模型采用 Pareto 支配关系对种群进行排序,并从中选出违约度较小的解进入下一代种群;当种群中既有可行解又有不可行解时,该模型采用自适应惩罚函数将有约束问题转化为无约束问题,根据个体的目标

值和违约度重新定义一种新的个体适应度,再根据个体适应度进行评价,选择较好的个体进入下一代,充分利用不可行解信息;当种群中全部为可行解时,该问题已经变为无约束问题,则该模型基于目标函数进行个体选择。

Tasgetiren 等^[34]在 2010 年的约束优化方向的国际竞赛中提出了 eDE(ensemble of Differential Evolution Algorithms)算法。该算法根据差分进化算法(Different Evolutionary Algorithm, DE)针对相关控制参数的敏感性,采用多种不同的差分进化策略。eDE 在种群进化的不同过程中采用 3 种不同的约束处理技术,分别为可行性准则、自适应惩罚函数和 ϵ 约束处理。可行性准则用于选择种群中的最优个体; ϵ 约束处理用于比较父代和子代的优劣;自适应惩罚函数用于比较通过 VPS(Variable Parameter Search)操作产生的子代个体与其父代的优劣。

Lin 等^[35]提出了一种基于动态混合约束处理机制(Multi-objective Differential Evolution with Dynamic Hybrid Constraint Handling Mechanism, MODE-DCH)的约束进化算法。该算法根据进化过程中种群的可行比将进化过程分为局部搜索和全局搜索。在早期可行比较低的阶段采用局部搜索模型,该算法根据目标值和违约度定义了新的个体适应度,较为优秀的个体被选入下一代。而在后期可行比较大时,该算法采用 ϵ 约束处理方法进行全局搜索。MODE-DCH 的两个不同模块协同工作,平衡了 Pareto 最优前沿的收敛性和分布性。

Li 等^[36]提出了一种两个外部档集的约束进化算法(Constrained-Two Archive Evolutionary algorithm, C-TAEA)。该算法采用 CA(Convergence-oriented Archive)和 DA(Diversity-oriented Archive)两个外部档集,其中 CA 用于保持种群的收敛性,当 CA 中可行解数量大于 N 时,对可行解进行非支配排序,并结合 MODE/D(基于分解的进化算法)框架进行选择;当 CA 中的可行解数量小于等于 N 时,采用多目标优化法进行选择。DA 中采用 MOEA/D 来维持种群的多样性。最后,根据限制性选择杂交策略,分别从 DA 和 CA 中选择父代进行交配,产生子代。

此外,Deb 等^[37-39]将惩罚函数法和多目标优化法相结合以处理约束问题;文献[40]分别用自适应函数法和非支配排序评价个体;文献[41]将由全局搜索和局部搜索组成的混合搜索框架与多目标优化相结合,提出了新颖的算法 HCOEA(Hybrid Constrained Optimization EA)。

3.5 其他算法

除了上述 4 类主要的约束处理机制,还有一种性能优秀的方法,即修复算法。其核心思想是通过修复算子将不可行解修复成可行解。Liepins 等^[42-43]通过一组不同的约束组合优化问题,证明了修复算法具有优秀的性能。

修复算法执行简单、收敛速度快、计算成本低,但是其在搜索过程中可能会引入强烈的偏见,进而损害进化过程本身。此外,修复算法过于依赖问题,每个特定的问题必须设定特定的修复算法,如 Xiao 等使用修复算法^[44-46]将机器人在有障碍物的情况下的不可行路径变为可行路径。

4 亟待解决的问题

虽然近年来研究人员在约束处理机制方面做出了巨大努力,并且取得了有目共睹的成果,但仍有很多问题尚未解决,主要如下。

(1) 约束条件与目标函数之间的不平衡

平衡约束条件和目标函数是约束进化算法的关键问题。在进化过程中,如果过分重视目标函数,则会得到质量较好的解,但可能陷入局部最优解;如果过分重视约束条件,则可能导致搜索方向偏离性能好的区域。

(2) 通用的约束进化算法

面对大量的约束问题,没有任何约束进化算法可以保证其优越的性能。因此,设计一种适应性强、性能优越的约束进化算法是未来的研究重点。

(3) 计算代价昂贵

随着约束优化问题的目标维数的增加,目标函数值和违约度的计算开销将呈现指数级增长。如何降低高维计算代价是一个关键问题。

(4) 混合性

目前,混合算法^[47]吸引了大量研究人员的目光。该方法的搜索效率高、性能优秀,特别是在面对复杂的多峰函数时,仍然能够保持优越的性能。因此,如何设计合理的混合算法是值得研究的问题。

(5) 等式约束条件的处理

目前的约束优化进化算法主要通过动态调整 δ 值来处理等式约束,但是处理离散等式约束和非线性等式约束时效果往往不尽人意。文献[48]采用基于百分比的容忍值调节方法;文献[49]中 δ 随进化代数的增加而逐渐减小。但是,参数 δ 值的设置仍然是值得重视的问题。

(6) 缩减搜索区域

在进化过程中,约束优化进化算法往往从距可行域较远的搜索区域开始探索,对于寻找最优解的帮助较小。因此,可以缩减不必要的搜索空间,从而提高算法的收敛速度。文献[50-51]提出通过自适应转移和缩小搜索区域来求解多目标约束问题。

(7) 理论研究

目前进化算法处理约束优化问题的理论基础还很薄弱。文献[52]对决策空间约束和目标空间约束进行了理论分析;文献[53]对不可行解在进化过程中的作用进行了分析。

(8) 测试函数

在测试函数方面,Deb 等^[54]于 2001 年提出了一整套约束优化测试函数(CTP1-CTP7);Zhang 等^[55]提出了新的测试函数;文献[56-57]改进了 CTP 系列的测试函数。但是,这些测试函数都存在可行域所占比例较大的问题,而实际问题是复杂多样的。因此,设计更能符合实际的多特征测试函数是未来研究的重点。

5 应用

由于实际应用问题以及对应的约束处理算法都在不断发

展之中,各个行业的学者在不断地添砖加瓦,相关的研究不断涌现。本文对几类常见领域的应用问题进行了总结,如表 3 所列。

表 3 COEAs 的应用
Table 3 Application of COEAs

应用内容		COEAs 应用类型及参考文献
工程优化方面	工程设计 工程优化 工程调度 工程计划	A co-evolutionary particle swarm optimization approach(CPSO) ^[58]
		Infeasibility Driven Evolutionary Algorithm(IDEA) ^[59]
		Efficient constraint handling scheme for differential evolutionary algorithm ^[60]
		A self-adaptive penalty approach ^[61]
		A multi-objective constraint-handling method with PSO algorithm ^[62]
		Constraint handling technique for Structural Problems ^[63]
		Co-evolution of control parameters and penalty factors ^[64]
		A new algorithm based on evolutionary computation for job-shop scheduling ^[65]
		A hybrid evolutionary approach to the nurse rostering problem ^[66]
		Feasibility Rules for NLP/MINLP Engineering Problems ^[67]
		A Hybrid Differential Evolution Algorithm ^[68]
		Constrained particle swarm optimizer ^[69]
		GA with different constraint handling techniques ^[70]
电子和通信工程方面	电路优化设计 无线网络设计	Multiobjective optimization in Oil Pipelines Networks ^[71]
		An ϵ -constrained method for constraint handling,an NSGA-II ^[72]
		SR for Robust Layout Synthesis of MEMS Components ^[73]
		Two multiobjective artificial immune systems based on the clonal selection principle ^[74]
		A problem-specific constrained evolutionary algorithm based on decomposition ^[75]
机械设计方面	结构设计 外形设计 机器设计	An MOEA/D ^[76] A multiobjective DE ^[77]
		Distributed genetic algorithm ^[78]
		The normal boundary intersection(NBI) method and an NSGA-II ^[79]
环境资源配置方面	地下水质量监控 土地资源规划 能源配置	New Constraint-Handling Method for Multi-Objective and Multi-Constraint Evolutionary Opt-imization ^[80] , A genetic algorithm with penalty function ^[81]
		Parametrical mechanical design with constraints and preference ^[82]
		SP with genetic algorithm ^[83]
		A COMOGA for gas supply networks ^[84]
科学研究方面	机器人研究 生物计算	A penalty function in Watershed Management Design Problem ^[85]
		Constraint-handling methods for optimal groundwater design ^[86]
		Multiobjective evolutionary optimization of DNA sequences ^[87]
		Automated constraint-based nucleotide sequence selection for DNA computation ^[88]
		A repair algorithm ^[46-48] , An NSGA-II and a SPEA2 ^[89]
		Genetic and hybrid approaches for multiobjective optimization with fuzzy dominance ^[90]
		An MOGA, an NSGA-II, and an MODE with normalized weighting objective functions method ^[91]
		An MOGA ^[92]
		Multi-layer Screening Based Evolution Algorithm for De Novo Protein Structure Prediction ^[93]
管理和分配方面	资源分配 路径规划 背包问题	Bayesian network structure learning algorithm based on differential evolution algorithm ^[94]
		A multiobjective evolutionary particle swarm optimization algorithm (MOEPSO) ^[95]
		An MOEA with two VRPSD-specific heuristics for local exploitation and a route simulation method ^[96] , Mixed method for VRP with Time Windows ^[97]
		Constrained Evolutionary Algorithm in a Compressed-air Station ^[98]
		A repair algorithm for vehicle routing problems ^[99]
		A genetic algorithm with penalty function for VRP ^[100]
		Adaptive Genetic Algorithm for the Vehicle Routing Problem ^[101]
		GA for profitable heterogeneous vehicle routing problem ^[102]
		Effective Local Search Algorithms for Routing and Scheduling Problems ^[103]
		0-1 Knapsack Variant with Time Scheduling ^[104]
		Exact Algorithms and Evolutionary Algorithms for Randomized Time-Varying Knapsack Problem ^[105]
		An efficient evolutionary algorithm for multi vehicle carps ^[106]
		Knapsack problem based on evolutionary algorithm ^[107]

5.1 工程优化方面

工程优化问题通常包含多峰目标函数,涉及高度非线性的不等式或等式约束,因此解决这些问题需要探索非常复杂的搜索空间。文献[60]介绍了一种新的约束处理方法,该方

法利用等式约束集导出的梯度信息来修正不可行解,并结合支配选择来优化多个工程问题。1)反应堆网络设计问题,该问题的目标是找出两个反应器的体积和每个罐中每种产品的浓度,使出口流中产品 B 的浓度最大化;2)换热器网络设计

问题,该问题的目标是最小化所需的全部换热器面积;
3)分离网络综合问题。

文献[67]采用带有 CDP 的混合差分进化算法来求解工程优化领域的非线性规划和混合整数非线性规划基准问题,并将其用于求解双管道原油调和调度问题。详细问题描述如下:双管道原油调和系统由多个加料罐、两条输送管道、一条混合管道、一个流量控制系统和一个原油蒸馏装置组成,有多种原油,每种原油都有一定的储量。优化目标为在 20 个约束条件下合理安排油罐通过两条管道输送石油,并使其按适当的顺序和流量输送,从而为蒸馏装置提供合格、稳定的原料。

5.2 电子和通信工程方面

在电子和通信工程方面,COEA 的主要应用有电路优化设计和无线网络设计。电路和无线网络的设计往往会引发许多极具挑战的问题(如约束处理问题和多目标问题)。文献[75]用修复算法来解决无线传感器网络中的 K 连接部署和功率分配问题,旨在确定传感器位置和发射功率水平,以保证在最大 K 连接性约束下的网络覆盖范围和寿命。通过两种特定的修复启发方式来处理约束条件,将不可行的网络设计转换为可行的设计,同时将其结合到 MOEA/D 中以保持效率。

随着电子技术的不断进步,硅片上晶体管的数量呈指数增长,基于 IP 核的片上系统^[78](System on a Chip, SOC)已成为 VLSI 技术发展的趋势。SOC 系统的主要功能是在目标冲突和众多约束条件的设计空间中找到满足约束条件的最优 IP 配置集。

5.3 机械设计方面

在机械设计方面,COEA 的应用主要包括结构设计、外形设计和机器设计。文献[80]将可行性原则和 NSGA(Non-dominated Sorting Genetic Algorithm)相结合,求解焊接梁设计问题。该问题可描述为:找到一组可行的尺寸参数来承载一定的载重量,同时求得最低的制造总成本(装配成本、焊接人工成本和材料成本等)。

文献[81]采用惩罚函数法和多目标优化法的 COGA 来对桥梁设计、钢结构和输电塔等具有代表性的结构设计问题进行求解,证明 COGA 能够高效地解决机械设计方面的问题。

5.4 环境资源配置方面

环境资源配置主要包括地下水质量检测、土地资源规划和能源分配 3 个方面。文献[108]介绍了水电发电系统的规划调度是一个高维、高约束、动态、非线性的实际问题,该问题在许多国家,特别是以水电为主要能源的巴西,具有重大的经济和环境意义。Marcondes 等^[108]对 111 个发电厂进行了为期 5 年的研究,得到了 13320 个决策变量和 20091 个约束条件,并采用多种不同的 COEA 进行优化。在多数情景中,成本稳步下降,并且在满足约束条件后逐渐收敛。

文献[83]介绍了森林规划问题需要同时满足森林结构、采伐口大小和最小成熟森林板块尺寸等约束条件,并同时优化多个目标的复杂约束优化问题。为此,采用动态惩罚函数处理约束条件和智能搜索策略进行搜索,并将其整合到遗传算法中以解决森林规划问题。

5.5 科学研究方面

科学研究方面主要包括机器人研究和生物计算。文献[44]将修复算法和进化算法相结合,开发出一种适用于移动机器人的自适应导航器,用来规划和导航方向。该算法的主要特点包括:1)能够将不可行路径转换为可行路径;2)可以容纳不同的优化标准;3)能够在距离较近时进行良好的权衡(路径优化),从而高效地处理未知障碍。

COEAs 在生物计算和医学中的应用也较为广泛。Lee 等^[87]将多目标优化法和 NSGA-II 相结合,完成了 DNA 序列优化,取得了较好的结果。该问题有两个约束:1)集合中所有序列的熔化温度(熔化温度为一半以上的双序列分裂为单序列所需的温度)的阈值之和;2)集合中所有的序列被划分为长度为 L 的序列中包含 G 和 C 的个数, C 和 G 为优化目标,分别为两个序列对最大可杂交基对之和,以及集合中所有的序列中存在连续基大于阈值 S 的个数的平方和。

5.6 管理和分配方面

管理和分配方面主要包括两个经典应用:0-1 背包问题和路径规划问题。

0-1 背包问题描述为:有 N 件物品和一个容量为 V 的背包,第 i 件物品的体积是 C_i ,价值为 W_i ,在满足背包容量的约束下,求解将哪些物品装入背包使得价值最大。文献[107]通过改进的差分进化算法高效地求解了背包问题。

车辆路径规划问题(Vehicle Routing Problem, VRP)起源于旅行商问题或弧路径问题。VRP 指一定数量的客户,各自有不同数量的货物需求,配送中心向客户提供货物,由一个车队负责分配货物,规划适当的路线,使得客户的需求得到满足,并能在一定的约束下达到路程最短、成本最小、时间最短等多个目的。文献[100]详细介绍了 COEA 解决 VRP 问题的过程。

VRP(Vehicle Routing Problem)广泛应用于运输、供应链管理、生产计划、电信和电子商务等领域。近年来,人们提出了许多 VRP 问题的变体,其中经典的变体是带时间窗的 VRP(VRP with Time Windows, VRPTW)和多个站点的 MDVRP。Wang 等^[109]提出了一种两阶段的约束优化进化算法(Two-stage Multiobjective Evolutionary Algorithm, TS-MOEA)来求解多个站点的 VRPTW,考虑了车辆数量、总行驶距离、总等待时间、完工时间和总延误时间 5 个目标以及每个客户允许的最大延迟时间、返回时间限制和车辆容量约束等多个约束。TS-MOEA 采用可行性原则来处理约束条件,在第一阶段根据 EC-NSGA-II^[110]快速搜索极值点,促进收敛;在第二阶段采用 MOEA/D 来保持 PF 的分布性。

结束语 约束优化进化算法是一个活跃的、飞速发展的研究领域,是优化算法领域的重要研究课题之一。近年来,约束进化算法取得了较多的研究成果,各种算法层出不穷,但仍然存在大量的问题亟待解决。本文对约束优化进化算法进行了详细的分析,并逐一对不同的约束处理机制进行了分析,同时指出了其中存在的主要不足和亟待解决的问题,最后介绍了约束优化进化算法在工程优化、电子通信工程、机械设计等方面的实际应用。由于本研究涉及的问题太多,不可能面面俱到,希望能够与研究者们一起交流,促进该领域的发展。

参 考 文 献

- [1] MICHALEWICZ Z S M. Evolutionary algorithms for constrained parameter optimization problems[J]. *Evolutionary Computation*, 2014, 4(1): 1-32.
- [2] COELLO C A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms; a survey of the state of the art[J]. *Computer Methods in Applied Mechanics and Engineering*, 2002, 191(11/12): 1245-1287.
- [3] DEB K. An efficient constraint handling method for genetic algorithms[J]. *Computer Methods in Applied Mechanics and Engineering*, 2000, 186(2): 311-338.
- [4] WANG Y, CAI Z X, ZHOU Y R, et al. Constrained optimization evolutionary algorithm [J]. *Acta Software Sinica*, 2009, 20(1): 11-29.
- [5] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [6] CUI C G, LI Y J, WU T J. A relative feasibility degree based approach for constrained optimization problems [J]. *Journal of Zhejiang University-Science C(Computer & Electronics)*, 2010, 11(4): 249-260.
- [7] CHENG R, JIN Y, OLHOFFER M, et al. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 773-791.
- [8] FAN Z, LI W, CAI X, et al. Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems[C] // *Congress on Evolutionary Computation*. 2016: 460-467.
- [9] GORDIAN-RIVERA L, MEZURA-MONTES E. A Combination of Specialized Differential Evolution Variants for Constrained Optimization[C] // *Ibero-American Conference on Artificial Intelligence*. Springer Berlin Heidelberg, 2012.
- [10] MOHAMED A W, SABRY H Z. Constrained optimization based on modified differential evolution algorithm[J]. *Information Sciences*, 2012, 194: 171-208.
- [11] SARKER R A, ELSAYED S M, RAY T. Differential Evolution With Dynamic Parameters Selection for Optimization Problems [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(5): 689-707.
- [12] DHADWAL M K, JUNG S N, KIM C J. Advanced particle swarm assisted genetic algorithm for constrained optimization problems [J]. *Computational Optimization and Applications*, 2014, 58(3): 781-806.
- [13] RUNARSSON T P, YAO X. Stochastic ranking for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3): 284-294.
- [14] ZHANG M, LUO W, WANG X. Differential evolution with dynamic stochastic selection for constrained optimization[J]. *Information Sciences*, 2008, 178(15): 3043-3074.
- [15] GUILLERMO L, COELLO C A C. A boundary search based ACO algorithm coupled with stochastic ranking [C] // *IEEE Congress on Evolutionary Computation*. IEEE, 2007: 165-172.
- [16] TAKAHAMA T, SAKAI S. Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation[C] // *IEEE Congress on Evolutionary Computation*. IEEE, 2010: 1-9.
- [17] TAKAHAMA T, SAKAI S. Efficient constrained optimization by the ϵ constrained differential evolution with rough approximation using kernel regression[C] // *IEEE Congress on Evolutionary Computation*. IEEE, 2012.
- [18] YANG Y, LIU J, TAN S, et al. A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio[J]. *Applied Soft Computing Journal*, 2019, 80: 42-56.
- [19] BREST J, BOSKOVIC B, ZUMER V. An improved self-adaptive differential evolution algorithm in single objective constrained real-parameter optimization [C] // *Evolutionary Computation*. IEEE, 2010: 1-8.
- [20] TAKAHAMA T, SAKAI S. Efficient Constrained Optimization by the ϵ Constrained Rank-Based Differential Evolution [C] // *IEEE Congress on Evolutionary Computation*. IEEE, 2010.
- [21] HOFFMEISTER F, SPRAVE J. Problem-Independent Handling of Constraints by Use of Metric Penalty Functions[J]. *Evolutionary Programming*, 1996: 289-294.
- [22] HOMAIFAR A, QI C X, LAI S H. Constrained Optimization Via Genetic Algorithms[J]. *Simulation*, 1994, 62(4): 242-253.
- [23] HSIEH Y, LEE Y, YOU P. Solving nonlinear constrained optimization problems: An immune evolutionary based two-phase approach[J]. *Applied Mathematical Modelling*, 2015, 39(19): 5759-5768.
- [24] KAZARLIS S, PETRIDIS V. Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase of the Dynamic Penalty Terms[C] // *International Conference on Parallel Problem Solving from Nature*. Heidelberg, Berlin: Springer, 1998.
- [25] TESSEMA B, YEN G G. An Adaptive Penalty Formulation for Constrained Evolutionary Optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2009, 39(3): 565-578.
- [26] DE MELO V V, IACCA G. A modified Covariance Matrix Adaptation Evolution Strategy with adaptive penalty function and restart for constrained optimization[J]. *Expert Systems With Applications*, 2014, 41(16): 7077-7094.
- [27] LIN C. A rough penalty genetic algorithm for constrained optimization[J]. *Information Sciences*, 2013, 241: 119-137.
- [28] AFONSO C L, HELIO J B, HEDER S B. Variants of an adaptive penalty scheme for steady-state genetic algorithms in engineering optimization[J]. *Engineering Computations*, 2015, 32(8): 2182-2215.
- [29] POWELL D, SKOLNICK M M. Using Genetic Algorithms in Engineering Design Optimization with Non-Linear Constraints [C] // *International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc. , 1993: 424-431.
- [30] FERNANDO J, JOSÉ L V. Evolutionary Techniques for Constrained Optimization Problems[C] // *European Congress on Intelligent Techniques & Soft Computing*. 1999.
- [31] ZHOU Y, ZHU M, WANG J, et al. Tri-Goal Evolution Frame-

- work for Constrained Many-Objective Optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(8):3086-3099.
- [32] CAI Z, WANG Y. A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 658-675.
- [33] WANG Y, CAI Z, ZHOU Y, et al. An Adaptive Tradeoff Model for Constrained Evolutionary Optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92.
- [34] TASGETIREN M F, SUGANTHAN P N, PAN Q K, et al. An ensemble of differential evolution algorithms for constrained function optimization[J]. IEEE Congress on Evolutionary Computation, 2010, 5(1): 1-8.
- [35] LIN Y F, DU W, DU W L, et al. Multi-objective differential evolution with dynamic hybrid constraint handling mechanism[J]. Soft Computing, 2019, 23(12): 4341-4355.
- [36] LI K, CHEN R, FU G, et al. Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization [J]. IEEE Transactions on Evolutionary Computation, 2019, 23(2): 303-315.
- [37] DEB K, DATTA R, INSTITUTIONEN F T O S, et al. A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach[J]. Engineering Optimization, 2013, 45(5): 503-527.
- [38] DATTA R, DEB K. An adaptive normalization based constrained handling methodology with hybrid bi-objective and penalty function approach[C]//IEEE Congress on Evolutionary Computation. IEEE, 2012: 1-8.
- [39] DATTA R, DEB K, COSTA M F P, et al. An evolutionary algorithm based pattern search approach for constrained optimization[C]//2013 IEEE Congress on Evolutionary Computation Computation. IEEE, 2013: 1355-1362.
- [40] CAI X, HU Z, FAN Z. A novel memetic algorithm based on invasive weed optimization and differential evolution for constrained optimization[J]. Soft Computing, 2013, 17(10): 1893-1910.
- [41] WANG Y, CAI Z, GUO G, et al. Multiobjective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B(Cybernetics), 2007, 37(3): 560-575.
- [42] LIEPINS G E, VOSE M D. Representational issues in genetic optimization[J]. Journal of Experimental & Theoretical Artificial Intelligence, 1990, 2(2): 101-115.
- [43] HAN B, LEE S J. A genetic algorithm approach to measurement prescription in fault diagnosis[J]. Information Sciences, 1999, 120(1-4): 223-237.
- [44] XIAO J, MICHALEWICZ Z, ZHANG L, et al. Adaptive evolutionary planner/navigator for mobile robots[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 18-28.
- [45] XIAO J, MICHALEWICZ Z, ZHANG L, et al. Evolutionary Planner/Navigator: operator performance and self-tuning[C]//IEEE International Conference on Evolutionary Computation. 1996: 366-371.
- [46] MICHALEWICZ Z, XIAO J. Evaluation of paths in evolutionary planner/navigator[C]//International Workshop on Biologically Inspired Evolutionary Systems. 1995.
- [47] LOZANO M, HERRERA F, KRASNOGOR N, et al. Real-Coded Memetic Algorithms with Crossover Hill-Climbing[J]. Evolutionary Computation, 2004, 12(3): 273-302.
- [48] HO P Y, SHIMIZU K. Evolutionary constrained optimization using an addition of ranking method and a percentage-based tolerance value adjustment scheme[J]. Information Sciences, 2007, 177(14): 2985-3004.
- [49] MEZURA-MONTES E, COELLO C A C. A simple multimembered evolution strategy to solve constrained optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2005, 9(1): 1-17.
- [50] AMIRJANOV A. A changing range genetic algorithm[J]. International Journal for Numerical Methods in Engineering, 2004, 61(15): 2660-2674.
- [51] AMIRJANOV A. The development of a changing range genetic algorithm[J]. Computer Methods in Applied Mechanics and Engineering, 2006, 195(19): 2495-2508.
- [52] LIU Z, WANG Y. Handling Constrained Multiobjective Optimization Problems With Constraints in Both the Decision and Objective Spaces[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(5): 870-884.
- [53] WHILE L, HINGSTON P. Usefulness of infeasible solutions in evolutionary search: An empirical and mathematical study[C]//Congress on evolutionary computation. 2013: 1363-1370.
- [54] DEB K, PRATAP A, MEYARIVAN T. Constrained Test Problems for Multi-objective Evolutionary Optimization[C]//Evolutionary Multi-criterion Optimization, First International Conference. Zurich, Switzerland, 2001: 284-298.
- [55] ZHANG Q F, ZHOU A M, ZHAO S Z, et al. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition [J/OL]. Mechanical engineering, 2018. http://www.al-roomi.org/multimedia/CEC_Database/CEC2009/MultiObjectiveEA/CEC2009_MultiObjectiveEA_TechnicalReport.pdf.
- [56] CHENG P. A Tunable Constrained Test Problems Generator for Multi-objective Optimization[C]//International conference on genetic and evolutionary computing. 2008: 96-100.
- [57] FAN Z, HUANG H, LI W, et al. An opposition-based repair operator for multi-objective evolutionary algorithm in constrained optimization problems[C]//International Conference on Natural Computation. IEEE, 2015: 330-336.
- [58] HE Q, WANG L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems[J]. Engineering Applications of Artificial Intelligence, 2007, 20(1): 89-99.
- [59] SINGH H K, ISAACS A, RAY T, et al. Infeasibility Driven Evolutionary Algorithm (IDEA) for Engineering Design Optimization[C]//AI 2008: Advances in Artificial Intelligence, 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand. 2008.
- [60] KHEAWHOM S. Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering

- optimization problem[J]. *Journal of Industrial and Engineering Chemistry*, 2010, 16(4): 620-628.
- [61] COELLO C A C. Use of a self-adaptive penalty approach for engineering optimization problems[J]. *Computers in Industry*, 2000, 41(2): 113-127.
- [62] LI L D, LI X, YU X, et al. A multi-objective constraint-handling method with PSO algorithm for constrained engineering optimization problems[C]// *World Congress on Computational Intelligence*. 2008: 1528-1535.
- [63] MOTTE D, NORDIN A, BJARNEMO R, et al. Study of the Sequential Constraint-Handling Technique for Evolutionary Optimization With Application to Structural Problems[C]// *Design Automation Conference*. 2011: 521-531.
- [64] FAN Q, YAN X. Differential evolution algorithm with co-evolution of control parameters and penalty factors for constrained optimization problems[J]. *Asia-Pacific Journal of Chemical Engineering*, 2012, 7(2): 227-235.
- [65] ZOU P, RAJORA M, LIANG S Y. A new algorithm based on evolutionary computation for hierarchically coupled constraint optimization; methodology and application to assembly job-shop scheduling[J]. *Journal of Scheduling*, 2018, 21(5): 545-563.
- [66] BAI R, BURKE E K, KENDALL G, et al. A Hybrid Evolutionary Approach to the Nurse Rostering Problem[J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(4): 580-590.
- [67] BAI L, WANG J, JIANG Y, et al. Improved Hybrid Differential Evolution-Estimation of Distribution Algorithm with Feasibility Rules for NLP/MINLP Engineering Optimization Problems[J]. *Chinese Journal of Chemical Engineering*, 2012, 20(6): 1074-1080.
- [68] HU R, QIAN B. A hybrid differential evolutionary algorithm for stochastic finite buffer pipeline scheduling [J]. *Acta automatica Sinica*, 2009, 35(12): 1580-1586.
- [69] CAGNINA L C, ESQUIVEL S C, COELLO C A C. Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer[J]. *Informatica*, 2008, 32(3): 319-326.
- [70] LUCENA R R, BAIOCO J S, LIMA B S L P, et al. Optimal design of submarine pipeline routes by genetic algorithm with different constraint handling techniques[J]. *Advances in Engineering Software*, 2014, 76: 110-124.
- [71] CRUZ J M, ANDRESTORO B, HERRAN A, et al. Multiobjective optimization of the transport in oil pipelines networks[C]// *Emerging Technologies and Factory Automation*. 2003: 566-573.
- [72] SARKER R, RAY T. An improved evolutionary algorithm for solving multi-objective crop planning models[J]. *Computers and Electronics in Agriculture*, 2009, 68(2): 191-199.
- [73] FAN Z, LIU J, SORENSEN T, et al. Improved Differential Evolution Based on Stochastic Ranking for Robust Layout Synthesis of MEMS Components[J]. *IEEE Transactions on Industrial Electronics*, 2009, 56(4): 937-948.
- [74] DAS S, NATARAJAN B, STEVENS D, et al. Multi-objective and constrained optimization for DS-CDMA code design based on the clonal selection principle[J]. *Applied Soft Computing*, 2008, 8(1): 788-797.
- [75] KONSTANTINIDIS A, YANG K. Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition[J]. *Computer Communications*, 2011, 34(1): 83-98.
- [76] PALMERS P, MCCONNAGHY T, STEYAERT M, et al. Massively multi-topology sizing of analog integrated circuits[C]// *European Design and Automation Association*. 2009: 706-711.
- [77] LI X, YIN M. Optimal synthesis of linear antenna array with composite differential evolution algorithm[J]. *Scientia Iranica. Transaction D, Computer Science & Engineering, Electrical*, 2012, 19(6): 1780.
- [78] SAIT S M, FAHEEMUDDIN M, MINHAS M R, et al. Multiobjective VLSI cell placement using distributed genetic algorithm [C]// *Genetic And Evolutionary Computation Conference*. 2005: 1585-1586.
- [79] MASAZADE E, RAJAGOPALAN R, VARSHNEY P K, et al. A Multiobjective Optimization Approach to Obtain Decision Thresholds for Distributed Detection in Wireless Sensor Networks[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2010, 40(2): 444-457.
- [80] OYAMA A, SHIMOYAMA K, FUJII K. New Constraint-Handling Method for Multi-Objective and Multi-Constraint Evolutionary Optimization[J]. *Transactions of the Japan Society for Aeronautical & Space Sciences*, 2007, 50(167): 56-62.
- [81] ERBATUR F, HASANÇEBİ O, TÜTÜNCÜ İ, et al. Optimal design of planar and space structures with genetic algorithms [J]. *Computers and Structures*, 2000, 75(2): 209-224.
- [82] COELHO R F, BERSINI H, BOUILLARD P. Parametrical mechanical design with constraints and preferences; application to a purge valve[J]. *Computer Methods in Applied Mechanics and Engineering*, 2003, 192(39): 4355-4378.
- [83] THOMPSON M P, HAMANN J D, SESSIONS J. Selection and Penalty Strategies for Genetic Algorithms Designed to Solve Spatial Forest Planning Problems[J]. *International Journal of Forestry Research*, 2009: 1-14.
- [84] HILTON A B, CULVER T B. Constraint-handling methods for optimal groundwater remediation design by genetic algorithms [C]// *Systems Man And Cybernetics*. 1998: 3937-3942.
- [85] LAURA J, HARRELL S R R. Evaluation of Alternative Penalty Function Implementations in a Watershed Management Design Problem[C]// *Conference on Genetic & Evolutionary Computation*. 1999: 1551-1558.
- [86] HILTON A B C, CULVER T B. Constraint Handling for Genetic Algorithms in Optimal Remediation Design[J]. *Journal of Water Resources Planning and Management*, 2000, 126(3): 128-137.
- [87] SHIN S, LEE I, KIM D, et al. Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing[J]. *IEEE Transactions on Evolutionary Computation*, 2005, 9(2): 143-158.
- [88] HARTEMINK A J, GIFFORD D K, KHODOR J. Automated constraint-based nucleotide sequence selection for DNA computation[J]. *BioSystems*, 1999, 52(1): 227-235.

- [89] ERBAS C, CERAV-ERBAS S, PIMENTEL A D. Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 358-374.
- [90] KODURU P, DONG Z, DAS S, et al. A Multiobjective Evolutionary-Simplex Hybrid Approach for the Optimization of Differential Equation Models of Gene Networks[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(5): 572-590.
- [91] SARAVANAN R, RAMABALAN S, EBENEZER N G R, et al. Evolutionary multi criteria design optimization of robot grippers [J]. Applied Soft Computing Journal, 2009, 9(1): 159-172.
- [92] CASTILLO O, TRUJILLO L, MELIN P. Multiple Objective Genetic Algorithms for Path-planning Optimization in Autonomous Mobile Robots [J]. Soft Computing, 2007, 11(3): 269-279.
- [93] LI Z W, HAO X H, ZHANG G J. Evolutionary algorithm of multi-level individual screening for protein structure prediction from scratch [J]. Computer Science, 2019, 46(S1): 80-84.
- [94] LIU B, FAN R X, LIU H R, et al. Learning algorithm of Bayesian network structure based on hybrid Thaliacea differential evolution algorithm [J]. Journal of Communications, 2019, 40(7): 151-161.
- [95] LIU D S, TAN K C, HUANG S Y, et al. On solving multiobjective bin packing problems using evolutionary particle swarm optimization[J]. European Journal of Operational Research, 2008, 190(2): 357-382.
- [96] TAN K C, CHEONG C Y, GOH C K. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation[J]. European Journal of Operational Research, 2007, 177(2): 813-839.
- [97] OMBUKI B, ROSS B J, HANSHAR F. Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows [J]. Applied Intelligence, 2006, 24(1): 17-30.
- [98] LIU J, YANG Y, TAN S, et al. Application of Constrained Multi-objective Evolutionary Algorithm in a Compressed-air Station Scheduling Problem[C]// Chinese Control Conference. 2019: 2023-2028.
- [99] LAMONT G B, SLEAR J N, MELENDEZ K. UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary Algorithms[C]// IEEE Symposium on Computational Intelligence in Multicriteria Decision Making. IEEE, 2007: 10-20.
- [100] BAKER B M, AYECHHEW M A. A genetic algorithm for the vehicle routing problem[J]. Computers and Operations Research, 2003, 30(5): 787-800.
- [101] ZHU K Q. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows[C]// IEEE International Conference on Tools with Artificial Intelligence. 2003: 176-183.
- [102] BANIAMERIAN A, BASHIRI M, TAVAKKOLI-MOGHADDAM R. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking[J]. Applied Soft Computing Journal, 2019, 75: 441-460.
- [103] IBARAKI T, IMAHORI S, KUBO M, et al. Effective Local Search Algorithms for Routing and Scheduling Problems with General Time-Window Constraints[J]. Transportation Science, 2005, 39(2): 206-232.
- [104] WANG Z L, XIE T, HE K, et al. 0-1 knapsack scheduling problem considering time factor[J]. Computer science, 2018, 45(4): 53-59.
- [105] HE Y C, WANG X Z, LI W B, et al. Exact algorithm and evolutionary algorithm for solving stochastic time-varying knapsack problem [J]. Journal of software, 2017, 28(2): 185-202.
- [106] ZHU Z Y, YANG Y, DENG X, et al. An efficient evolutionary algorithm for solving multi vehicle carps [J]. Computer engineering and application, 2008(8): 212-216.
- [107] ALI I M, ESSAM D, KASMARIK K, et al. An Efficient Differential Evolution Algorithm for Solving 0-1 Knapsack Problems [C]// Congress on Evolutionary Computation. 2018: 1-8.
- [108] MARCONDES E J, ZANETTE B N, PERRONI P F, et al. Evolutionary Algorithms with Constraint Handling for the Hydroelectric Dispatch Planning[C]// 2019 8th Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2019: 527-532.
- [109] WANG J H, WENG T, ZHANG Q. A Two-Stage Multiobjective Evolutionary Algorithm for Multiobjective Multidepot Vehicle Routing Problem With Time Windows[J]. IEEE Transactions on Cybernetics, 2019, 49(7): 2467-2478.
- [110] DEB K, CHAUDHURI S, MIETTINEN K. Towards estimating nadir objective vector using evolutionary approaches[C]// Genetic and Evolutionary Computation Conference (GECCO 2006). Seattle, Washington, USA, 2006.



LI Li, born in 1986, Ph.D, M.S supervisor, is a member of China Computer Federation. His main research interests include multi-objective optimization methods and their applications.