

基于自适应迁移知识来源的约束多因子进化算法

崔春玲

广东工业大学数学与统计学院, 广东 广州

收稿日期: 2024年1月10日; 录用日期: 2024年1月31日; 发布日期: 2024年4月10日

摘要

在约束多任务优化中, 如何在任务之间迁移知识是一个复杂的问题。现有的方法是从源任务的种群中随机选择个体进行迁移, 忽略了源任务中一些目标函数值好的个体对目标任务的积极影响。本文针对约束多任务优化问题, 提出基于自适应迁移知识来源的约束多因子进化算法。首先, 利用一个归档集来存储每个任务中目标函数值最好的个体, 用于产生高质量子代。然后, 提出一个自适应迁移知识来源策略, 通过一个自适应概率来确定被迁移的个体是来自于种群还是归档集。实验结果表明, 与其他现有的约束多任务进化算法相比, 所提算法可以产生更好的或至少相当的性能。

关键词

约束多任务, 多因子进化算法, 知识迁移, 归档集

A Constrained Multifactorial Evolutionary Algorithm Based on Adaptive Transferred Knowledge Sources

Chunling Cui

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: Jan. 10th, 2024; accepted: Jan. 31st, 2024; published: Apr. 10th, 2024

Abstract

In constrained multitask optimization, transferring knowledge between tasks is a complex problem. Existing methods randomly select individuals from the source task's population for transfer,

disregarding the positive impact of individuals with good objective function value on the target task. This paper proposes a constrained multifactorial evolutionary algorithm based on adaptive transferred knowledge sources. Firstly, an archive is used to store individuals with the best objective function value in each task to generate high-quality offspring. Then, an adaptive transfer knowledge source strategy determines whether the transferred individuals are from the population or the archive through adaptive probability. The experimental results demonstrate that the proposed algorithm can achieve better or at least comparable performance.

Keywords

Constrained Multitasking, Multifactorial Evolutionary Algorithm, Knowledge Transfer, Archive

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

多任务优化(MTO) [1]旨在利用任务间的知识迁移,同时求解多个优化任务。在这些优化任务中,任务之间是存在相关性的。通过利用任务之间的知识迁移,可以帮助整个MTO问题获得更好的性能[2][3][4]。近几年来,MTO在无约束多任务优化领域取得了显著的进展,并成功应用于求解一些现实问题,如车辆路径问题[5]、铁路旅客舒适度评价问题[6]、光伏模型的参数提取[7]。然而,在现实生活中,许多优化问题都包含特定的约束。以多辆车的路径规划为例,这些路径规划问题存在许多的约束条件(如货物需求量、交货时间、车辆容量限制等)。因此如何高效求解带约束的MTO问题具有重要的研究意义。

目前关于约束多任务优化(CMTO) [8]的研究比较少。在现有的CMTO研究中,研究人员在[8]中设计了一组CMTO基准问题,并将可行性优先约束处理技术[9]结合到多任务进化算法中,用于求解CMTO问题。后来,Xing等人提出了一个基于自适应归档集的约束多因子进化算法(A-CMFEA) [10]。A-CMFEA使用归档集策略和变异策略来防止种群陷入局部最优,使用一个自适应的随机配对概率策略来调整知识迁移的频率。

相比于直接剔除违反约束的个体[8],利用归档集保留部分目标函数值好的个体能够帮助种群跳出局部最优,从而收敛到全局最优。然而,在先前的基于归档集的约束多任务优化算法中,归档集中的个体只是简单地参与环境选择,一些目标函数值好的个体并没有得到充分的利用。另外,在当前的约束多任务优化中,被迁移的个体主要是从种群中获取的,然而当目标任务的约束最优解与源任务的无约束最优解更靠近时,从归档集中选择目标值小的个体进行迁移更有利于加快目标任务的进化。

因此,为了充分利用种群中目标函数值好的个体中的有用信息,本文提出了一个基于自适应迁移知识来源的约束多因子进化算法。在所提算法中,将可行性优先约束处理技术结合到AT-MFEA中,并增加了归档集策略和自适应迁移知识来源策略。本文的主要贡献如下:

- 1) 提出了一个新的归档集策略用于保留部分目标函数值最好的个体,以充分利用目标函数的有用信息以产生更多高质量子代,从而帮助种群跳出局部最优。

- 2) 提出自适应迁移知识来源策略。通过计算从归档集中获取的迁移个体和从种群中获取的迁移个体的成功率,比较两者的大小,自适应确定被迁移个体的种群来源。

3) 广泛的实验验证了所提算法的有效性。

本文的其余部分如下。第二节介绍了所提出的算法。第三节介绍了实验和结果，并对其进行了分析。第四节为本文进行了总结。

2. 所提算法

2.1. 相关知识

2.1.1. 约束多任务优化

CMTO 是指在进化过程中同时优化两个或两个以上的约束任务，并为每个任务找到最优解[8]。具体地，对于一个有 K 个任务的 CMTO 问题，令 $X_i (i=1,2,\dots,K)$ 为第 i 个任务的搜索空间， $f_i : X_i \rightarrow \mathbb{R}$ 为目标函数，并假设所有任务都是最小化问题。每个任务有 m 个约束， $g : X_i \rightarrow \mathbb{R}$ 表示不等式约束， $h : X_i \rightarrow \mathbb{R}$ 表示等式约束，CMTO 的目标是找到一组可行解集 $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_K^*\}$ ，使其满足：

$$\begin{aligned} \mathbf{x}_i^* &= \arg \min f_i(\mathbf{x}) \quad i=1,2,\dots,K \\ g_{i,j}(\mathbf{x}) &\leq 0 \quad j=1,2,\dots,p \\ h_{i,k}(\mathbf{x}) &= 0 \quad k=p+1,\dots,m \end{aligned} \tag{1}$$

一般地，等式约束可以通过一个趋向于 0 的正数 ε 松弛为一个不等式约束：

$$|h_{i,k}(\mathbf{x})| - \varepsilon \leq 0 \tag{2}$$

如果一个解 \mathbf{x}_i^* 满足 $g_{i,j}(\mathbf{x}_i^*) \leq 0$ 且 $|h_{i,k}(\mathbf{x}_i^*)| - \varepsilon \leq 0$ ，则称 \mathbf{x}_i^* 为可行解。另外，为了在多因子优化环境比较不同个体的质量，每个个体都具备一组属性[1]，其定义如下：

定义 1 (因子代价)：个体 p_i 在任务 T_j 上的因子代价定义为 $\psi_j^i = \lambda \cdot \delta_j^i + f_j^i$ ，其中 $j \in \{1,2,\dots,K\}$ ， λ 为惩罚系数， f_j^i 和 δ_j^i 分别为个体 p_i 在任务 T_j 上的目标函数和总体约束违反程度。如果个体 p_i 在任务 T_j 上是可行的，则其约束违反程度为 0， $\psi_j^i = f_j^i$ 。

定义 2 (因子排序)：个体 p_i 在任务 T_j 上的因子排序 r_j^i 定义为个体在种群中的索引，而这个索引是由种群中所有个体的因子代价 ψ_j^i 从小到大的排序来确定。

定义 3 (适应值标量)：个体 p_i 的适应值标量定义为 $\varphi_i = 1 / \min\{r_j^i\}$ ， $j \in \{1,2,\dots,k\}$ 。

定义 4 (技能因子)：个体 p_i 的技能因子定义为 $\tau_i = \arg \min_j \{r_j^i\}$ ， $j \in \{1,2,\dots,k\}$ 。

定义 5 (多因子最优)：当且仅当 p^* 是所有 K 个组成任务的全局最优时，它被称为多因子最优。

2.1.2. 仿射增强多因子进化算法(AT-MFEA)

为了解异构 CMTO 问题，即不同任务的决策空间或适应度空间不相似的 CMTO 问题，Xue 等人在 [11] 中提出了仿射增强多因子进化算法(AT-MFEA)。在 AT-MFEA 中，作者设计了一个比较容易计算的秩损失函数，用于减少源任务和目标任务之间由于排序的场景不匹配导致的问题。该秩损失函数计算公式如下：

$$\phi^* = \min_{\phi \in \Phi} \int_x \left\| \mathcal{R}[p^s(x)] - \mathcal{R}[p^t(\phi(x))] \right\|^2 \tag{3}$$

其中， ϕ^* 表示从源任务到目标任务的映射解， Φ 表示完全映射空间， \mathcal{R} 是排序操作， $p^s(x)$ 和 $p^t(\phi(x))$ 分别表示源任务和目标任务的高斯渐进分布模型。

在 AT-MFEA 中，仿射变换可以表示为：

$$\tilde{x} = \phi_a(x; \theta) = xA + b \quad (4)$$

其中 $\tilde{x}_{1 \times n}$ 是迁移解, $\phi_a(\cdot)$ 是仿射变换操作, $x_{1 \times n}$ 是源任务中的解, $\theta = [A, b]$ 是仿射变换的参数集, $A_{n \times n}$ 和 $b_{1 \times n}$ 分别是收缩变换和平移变换。在以上的变换中, 通过多元高斯分布的数学推导可得到矩阵 A 和 b , 并通过(4)得到迁移解。

算法 1 给出 AT-MFEA 基于映射的任务间交叉的具体操作。首先输入来自不同任务的两个个体 x_i 和 x_j 及其相应的进化路径 E^i 和 E^j 。其中, 被评估的个体从初始种群到当前种群被称为其进化路径。然后, 根据进化路径, 生成渐进表达式 $[p^i(x), p^j(x)]$ 。然后, 通过源实例与目标实例的仿射变换, 可得到参数矩阵 $[A_{ij}, b_{ij}]$ 。通过式(4), 可以得到的转换解 x'_i 和 x'_j 。最后 x_i 和 x_j 与转化后的解杂交产生子代。

算法 1. 基于映射的任务间交叉

输入: 来自不同任务的两个个体 x_i 和 x_j , 以及相应的进化路径 E^i 和 E^j

输出: 两个子代 x_a 和 x_b

1. $[p^i(x), p^j(x)] \leftarrow$ 基于 E^i 和 E^j 的渐进表示
2. $[A_{ij}, b_{ij}] \leftarrow$ 由源 $p^i(x)$ 和目标 $p^j(x)$ 形成的仿射变换
3. $[A_{ji}, b_{ji}] \leftarrow$ 由源 $p^j(x)$ 和目标 $p^i(x)$ 形成的仿射变换
4. $x'_i \leftarrow x_i A_{ij} + b_{ij}$
5. $x'_j \leftarrow x_j A_{ji} + b_{ji}$
6. $x_a \leftarrow x'_i$ 与 x_j 交叉
7. $x_b \leftarrow x'_j$ 与 x_i 交叉

2.2. 归档集策略

在本文中归档集用于保存每个任务的前 $arc_maxsize$ 个目标函数值最好的个体。用于参与子代的产生和迁移知识的选择, 其更新过程由算法 2 给出。

首先, 由归档集(archive)和子代种群(offspring)组成混合种群。然后在混合种群中根据技能因子, 找到每个任务对应的子种群, 将这些子种群中的个体的目标值进行排序, 找到前 $arc_maxsize$ 个目标函数值最小的个体放入到归档集中, 这里的 $arc_maxsize$ 表示每个任务的归档集的种群规模。

算法 2. 归档集策略

输入: archive, offspring, $arc_maxsize$, tasks

输出: archive

1. For $i=1:\text{length}(\text{tasks})$
2. 对技能因子为 i 的个体的目标函数值按升序排序
3. $archive_i \leftarrow$ 前 $arc_maxsize$ 个具有最小目标值的个体
4. end

2.3. 自适应迁移知识来源策略

由于在没有先验知识的情况下很难知道源任务最优解和目标任务最优解之间的关系, 一个简单有效

的方式是通过比较两种迁移方式的成功率。具体地，一开始，先以初始概率 atp_0 迁移归档集中的个体，然后在进化过程中，通过比较从归档集中迁移的个体的成功率和从种群中迁移的个体的成功概率，如果前者的概率大于后者，则增加其迁移概率，否则减少其迁移概率。算法 3 给出更新归档集个体的迁移概率的伪代码。

首先，输入 6 个参数，其中 kt_num 和 akt_num 分别表示来自种群和归档集的迁移个体的数量， skt 和 $sakt$ ，分别表示来自种群的迁移个体成功的数量和来自归档集的迁移个体成功的数量，其初始值均为 0。在本文中，将在种群中存活的迁移个体称为成功个体。 $atp(gen-1)$ 为上一代的 atp ， c 为用于调整 atp 的常数。行 1~2 计算个体的成功率，行 3~7 更新 atp 。

算法 3. 更新 atp

输入: kt_num , skt , akt_num , $sakt$, $atp(gen-1)$, c
 输出: $atp(gen)$

1. $sr_kt = skt / kt_num$
2. $sar_kt = sakt / akt_num$
3. if $sar_kt > sr_kt$
4. $atp(gen) = \min(atp(gen-1) + c, 1)$
5. else
6. $atp(gen) = \max(atp(gen-1) - c, 0)$
7. end

2.4. 所提算法框架

本文将所提的归档集策略和自适应迁移知识来源策略嵌入到基于可行性优先约束处理技术的 AT-MFEA 中，其细节由算法 4 给出。

一开始，我们初始化种群并为每个个体分配技能因子，接着我们初始化归档集和所有参数，然后判断是否满足终止条件，如果不满足则进入主循环，否则退出主循环。

在主循环中，首先，我们先从种群和归档集的混合种群中选择两个个体 x_i 和 x_j ，如果它们的技能因子相同，则执行任务内的 SBX 交叉操作产生两个子代 c_1 和 c_2 ，并分配技能因子。如果它们的技能因子不同且随机数 $rand1$ 小于迁移概率 rmp ，则执行任务间知识迁移操作。当执行任务间知识迁移操作时，随机生成一个随机数 $rand$ ，若 $rand$ 小于 $atp(\tau_i)$ ，则从源任务的归档集中选择个体与当前个体进行任务间映射交叉产生子代，此时 $akt_num(\tau_i)+1$ 。否则，则从源任务的种群中选择个体与当前个体进行任务间映射交叉产生子代，此时 $kt_num(\tau_i)+1$ 。并令子代继承当前个体的技能因子。如果 x_i 和 x_j 的技能因子不同且 $rand1 \geq rmp$ 时，我们从混合种群中随机选择两个分别与 x_i 和 x_j 有相同技能因子的 x_1 和 x_2 ，通过 SBX 交叉操作产生子代 c_1 和 c_2 。接着，我们合并当前种群、子代和归档集，利用可行优先约束处理技术更新种群 $P(t+1)$ ，利用算法 2 更新归档集 $archive(t+1)$ 。然后，在 31~33 行，分别计算每个任务对应的子种群中来自源任务归档集的迁移个体的数量 $sakt(\tau_i)$ 和来自源任务种群的迁移个体的数量 $skt(\tau_i)$ ，最后，根据算法 3 更新 atp 。

3. 实验设计和分析

3.1. 参数设置

对比实验的参数设置为：

- 1) 目标函数维度：50

算法 4. 所提算法框架

输入: N (种群规模), $arc_maxsize$ (归档集规模), rmp (知识迁移概率)

输出: $\mathbf{x}_{1:K}^*$

1. 初始化种群 $P(0)$ 并评估
 2. 给每个个体分配一个技能因子 τ
 3. 初始化归档集 $archive(0)$
 4. 初始化所有参数: $atp = 0.3$, $kt_num = 0$, $skt = 0$, $akt_num = 0$, $sakt = 0$, 令 $t = 0$
 5. While 终止条件不满足时 do
 6. 从 $[P(t) \cup archive(t)]$ 中随机选择两个个体 x_i 和 x_j
 7. 产生一个随机数 $rand1 \in [0, 1]$
 8. if $\tau_i == \tau_j$ or $rand1 < rmp$ then
 9. if $\tau_i == \tau_j$ then
 10. x_i 和 x_j 执行 SBX 交叉操作产生子代 c_1 和 c_2
 11. c_1 和 c_2 分别继承 x_i 和 x_j 的技能因子
 12. else
 13. if $rand < atp(\tau_i)$ then
 14. 从 $archive(t)$ 中随机选择一个技能因子为 τ_j 的个体与个体 x_i , 利用算法 1 产生子代 c_1
 15. c_1 继承其父代的技能因子 τ_i
 16. $akt_num(\tau_i) = akt_num(\tau_i) + 1$
 17. else
 18. 从 $P(t)$ 中随机选择一个技能因子为 τ_j 的个体与个体 x_i , 利用算法 1 产生子代 c_1
 19. c_1 继承其父代的技能因子 τ_i
 20. $kt_num(\tau_i) = kt_num(\tau_i) + 1$
 21. end if
 22. 重复 13-21 行产生子代 c_2
 23. end if
 24. else
 25. 从 $[P(t) \cup archive(t)]$ 中随机选择一个技能因子为 τ_i 的个体与个体 x_i SBX 交叉产生子代 c_1
 26. 从 $[P(t) \cup archive(t)]$ 中随机选择一个技能因子为 τ_j 的个体与个体 x_j SBX 交叉产生子代 c_2
 27. c_1 和 c_2 分别继承 x_i 和 x_j 的技能因子
 28. end if
 29. $P(t) = P(t) \cup [c_1, c_2]$
 30. $archive(t) = archive(t) \cup [c_1, c_2]$
 31. 更新种群 $P(t+1)$ 和归档集 $archive(t+1)$
 32. 计算每个任务子种群 pop^{τ_i} 中来自源任务归档集的迁移个体的数量 $sakt(\tau_i)$
 33. 计算每个任务子种群 pop^{τ_i} 中来自源任务种群的迁移个体的数量 $skt(\tau_i)$
 34. $t = t + 1$
 35. 利用算法 3 更新 atp
 36. end while
-

- 2) 种群规模: $100 * K$, K 为测试问题的任务数量
- 3) 最大函数评估次数: 100,000
- 4) 独立运行次数: 30
- 5) 所提算法的其他参数设计
 - a) SBX 交叉算子: $p_c = 1, \eta_c = 2$
 - b) 多项式变异: $p_m = 1/D_{dim}, \eta_m = 5$
 - c) 迁移概率 rmp : $rmp = 0.3$, 初始 $atp0$: $atp0 = 0.3$, 常数 c : $c = 0.01$
 - d) 归档集种群规模 $arc_maxsize$: $arc_maxsize = 0.1 * 100 * K$
- 6) 对比算法的其他参数设置与其原文献相同。

3.2. 基准函数和对比算法

本文选用 CMT [8]系列作为约束多任务优化问题的基准测试函数。另外, 本文将可行性优先约束处理技术嵌入到 MFDE [12]、MFEA [1]、MFEA-II [13]、MFMP [14]、MTEA-AD [15]、AT-MFEA [11]中构建了 6 个约束多任务优化算法进行性能比较。分别记为 PF-MFDE、PF-MFEA、PF-MFEA-II、PF-MFMP、PF-MTEA-AD、PF-AT-MFEA。

3.3. 所提算法在基准函数上的表现

本节将所提算法与 6 个对比算法在基准测试问题上进行性能比较, 表 1 给出了各算法在测试问题上

Table 1. Average objective function values from 30 independent runs of 7 algorithms on the test instances
表 1. 7 个算法在测试实例上独立运行 30 次的平均目标函数值

Problem	Task	PF-MFDE	PF-MFEA	PF-MFEA-II	PF-MFMP	PF-MTEA-AD	PF-AT-MFEA	Ours
CMT1	T1	3.84e-01-	8.22e-01-	4.88e-02-	4.40e-01-	2.26e-03-	2.25e-03-	7.45e-04
	T2	3.56e+02-	2.00e+02-	8.59e+01-	2.21e+02-	4.31e+00-	4.39e+00-	4.37e-01
CMT2	T1	3.01e+00-	4.17e+00-	3.72e-01-	1.04e+01-	6.97e-02-	7.94e-02-	1.80e-02
	T2	3.57e+02-	2.25e+02-	2.97e+01-	8.59e+02-	2.11e+00-	2.56e+00-	1.79e-01
CMT3	T1	2.03e+00-	5.18e+00-	4.78e-02-	1.30e+01-	2.16e-02-	3.69e-02-	8.99e-03
	T2	2.89e+01-	1.78e+03-	3.56e+01-	1.33e+04-	1.68e-02-	4.89e-02-	3.76e-03
CMT4	T1	4.45e+02-	1.89e+02-	2.15e+02-	1.56e+02-	1.10e+02-	1.45e+02-	5.91e+01
	T2	4.42e+02-	4.31e+02-	4.21e+02-	4.14e+02=	4.15e+02=	4.17e+02=	4.15e+02
CMT5	T1	2.83e+00-	3.87e+00-	1.77e+01-	1.25e+01-	5.74e-02-	1.16e-01-	2.57e-02
	T2	7.66e+02-	3.90e+03-	1.99e+02-	1.92e+06-	4.89e+01-	5.09e+01-	4.81e+01
CMT6	T1	2.89e+00-	4.20e+00-	7.35e-02-	1.22e+01-	1.17e-01-	6.92e-02-	1.69e-02
	T2	1.14e+01-	7.54e+00-	3.16e-01-	5.29e+01-	4.46e-01-	5.13e-01-	1.74e-01
CMT7	T1	1.04e+04-	1.39e+05-	1.61e+03-	2.33e+09-	2.19e+03-	1.78e+03-	1.30e+03
	T2	4.40e+02-	2.85e+02-	1.26e+02-	2.35e+04-	1.52e+02-	1.68e+02-	6.93e+01
CMT8	T1	6.17e+00-	6.52e+00-	6.04e+00+	9.37e+00-	6.05e+00=	6.03e+00+	6.05e+00
	T2	6.50e+01-	5.28e+01-	4.90e+01+	7.16e+01-	5.02e+01+	5.01e+01+	5.13e+01
CMT9	T1	NaN-	4.93e+04+	NaN-	3.40e+04+	4.31e+04+	NaN-	1.08e+05
	T2	1.92e+04-	1.69e+04-	1.66e+04+	2.00e+04-	1.66e+04+	1.66e+04+	1.67e+04
+/-/=		0/18/0	1/17/0	3/15/0	1/16/1	3/13/2	3/14/1	Base

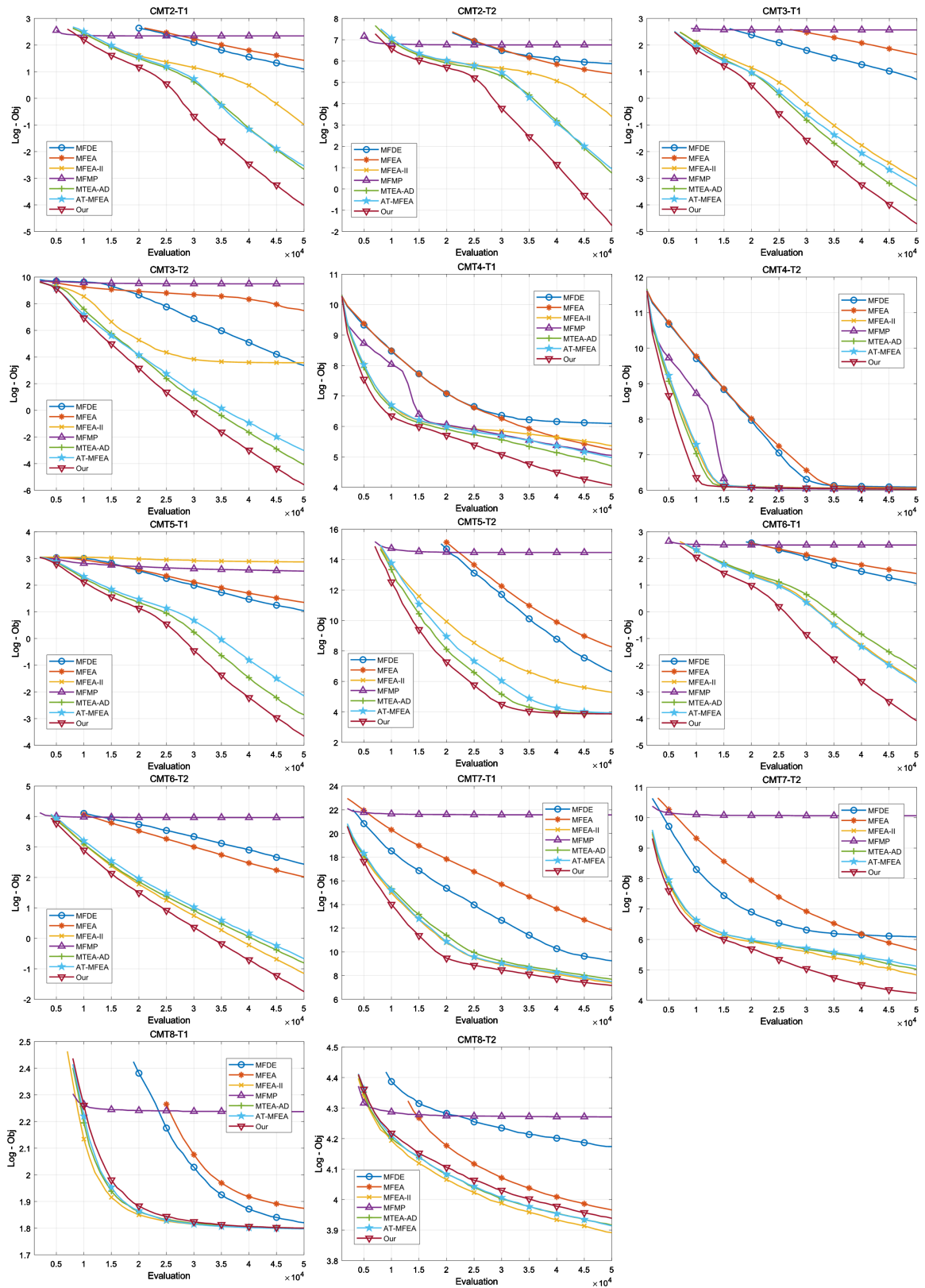


Figure 1. The average objective function value convergence trends of 7 algorithms over 30 independent runs on test instances
 图 1. 7 个算法在测试实例上独立运行 30 次的平均目标函数收敛趋势

独立运行 30 次获得的平均目标值。所有算法均在 MATLAB R2021a 上运行。在表 1 中用黑色粗体突出最好的结果。为了验证算法的性能,我们利用 Wilcoxon 秩和检验在 0.05 的显著性水平下分析算法的显著性。记号 “+”, “-” 和 “=” 分别表示比较算法优于、差于和近似于所提算法。需要注意的是,当目标函数值为 “NaN” 时,表明该算法在 30 次运行中都没有找到当前任务的可行解。

从表 1 可以看出,所提算法在 9 个问题 18 个任务中,有 14 个任务优于 PF-AT-MFEA,这说明了本文所提的归档集策略和自适应迁移知识来源策略能够有效地提高算法的性能。可行性优先约束处理技术虽然能快速找到可行域,但对于可行区域小且不连续的任务,该方法容易导致算法早熟。而归档集保留部分目标函数值好的个体并参与子代的产生,有利于产生更多目标函数值更好的高质量子代,从而帮助种群穿越不可行区域到达离全局最优解更近的可行区域。另外,当目标任务的约束最优解与源任务的无约束最优解更靠近时,从源任务的归档集中选择迁移个体能够加快种群收敛。反之,当目标任务的约束最优解与源任务的约束最优解更靠近时,从源任务的种群中选择迁移知识能够加快种群收敛。而自适应迁移知识来源策略能够自适应地调整这两种迁移方式的频率。

表 1 最后一行的显著性分析结果表明,所提算法在 9 个问题 18 个任务中,分别有 18、17、15、16、13 个任务优于 F-MFDE、PF-MFEA、PF-MFEA-II、PF-MFMP、PF-MTEA-AD。所提算法保留了 PF-AT-MFEA 的优势,同时增加的两个策略提高了算法的有效性。

另外,为了更加直观地比较这些算法的性能,我们在图 1 给出 7 个算法在 CMT 测试问题上独立运行 30 次后的平均目标趋势图。由于篇幅的限制,这里给出了 CMT2-8 的趋势图。从图 1 可以看出,在绝大多数实验中,所提算法都获得了更快地收敛速度,这表明与其他约束多任务算法相比,所提算法有更好的收敛性。即使在进化的早期,所提算法也展现出了更显著的收敛性,这进一步验证了所提算法的性能优于其他算法。

4. 总结

考虑到基于可行优先的约束多任务算法容易使种群陷入局部最优,本文提出了一个新的归档集策略用于保存部分目标值好的个体,并让这些个体参与子代的产生。另外,考虑到目标任务的约束最优解与源任务的无约束最优解和约束最优解之间的位置关系,本文提出了自适应迁移知识来源策略,用于确定迁移知识是来自于源任务的归档集还是种群。本文将所提的两个策略应用于一个基于可行性优先约束处理技术的约束多任务算法上。数据实验结果表明所提算法获得了更好的性能,能够有效地求解约束多任务优化问题。

参考文献

- [1] Gupta, A., Ong, Y. and Feng, L. (2016) Multifactorial Evolution: Toward Evolutionary Multitasking. *IEEE Transactions on Evolutionary Computation*, **20**, 343-357. <https://doi.org/10.1109/TEVC.2015.2458037>
- [2] Feng, L., Huang, Y., Zhou, L., Zhong, J., Gupta, A., Tang, K. and Tan, K.C. (2020) Explicit Evolutionary Multitasking for Combinatorial Optimization: A Case Study on Capacitated Vehicle Routing Problem. *IEEE Transactions on Cybernetics*, **51**, 3143-3156. <https://doi.org/10.1109/TCYB.2019.2962865>
- [3] Wu, D. and Tan, X. (2018) Multitasking Genetic Algorithm (MTGA) for Fuzzy System Optimization. *IEEE Transactions on Fuzzy Systems*, **28**, 1050-1061. <https://doi.org/10.1109/TFUZZ.2020.2968863>
- [4] Wang, Z. and Wang, X. (2019) Multiobjective Multifactorial Operation Optimization for Continuous Annealing Production Process. *Industrial & Engineering Chemistry Research*, **58**, 19166-19178. <https://doi.org/10.1021/acs.iecr.9b03399>
- [5] Wu, Y., Cai, Y. and Fang, C. (2023) Evolutionary Multitasking for Bidirectional Adaptive Codec: A Case Study on Vehicle Routing Problem with Time Windows. *Applied Soft Computing*, **145**, Article ID: 110605. <https://doi.org/10.1016/j.asoc.2023.110605>

-
- [6] Cheng, B., Fu, H., Li, T., Zhang, H., Huang, J., Peng, Y., Chen, H. and Fan, C. (2023) Evolutionary Computation-Based Multitask Learning Network for Railway Passenger Comfort Evaluation from EEG Signals. *Applied Soft Computing*, **136**, Article ID: 110079. <https://doi.org/10.1016/j.asoc.2023.110079>
- [7] Li, Y., Gong, W. and Li, S. (2022) Multitasking Optimization via an Adaptive Solver Multitasking Evolutionary Framework. *Information Sciences*, **630**, 688-712. <https://doi.org/10.1016/j.ins.2022.10.099>
- [8] Li, Y., Gong, W. and Li, S. (2022) Evolutionary Constrained Multi-Task Optimization: Benchmark Problems and Preliminary Results. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 9-13 July 2022, Boston, MA, 443-446. <https://doi.org/10.1145/3520304.3528890>
- [9] Deb, K. (2000) An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, **186**, 311-338. [https://doi.org/10.1016/S0045-7825\(99\)00389-8](https://doi.org/10.1016/S0045-7825(99)00389-8)
- [10] Xing, C., Gong, W. and Li, S. (2023) Adaptive Archive-Based Multifactorial Evolutionary Algorithm for Constrained Multitasking Optimization. *Applied Soft Computing*, **143**, Article ID: 110385. <https://doi.org/10.1016/j.asoc.2023.110385>
- [11] Xue, X., Zhang, K., Tan, K.C., Feng, L., Wang, J., Chen, G., Zhao, X., Zhang, L. and Yao, J. (2020) Affine Transformation-Enhanced Multifactorial Optimization for Heterogeneous Problems. *IEEE Transactions on Cybernetics*, **52**, 6217-6231. <https://doi.org/10.1109/TCYB.2020.3036393>
- [12] Feng, L., Zhou, W., Zhou, L., Jiang, S., Zhong, J., Da, B., Zhu, Z. and Wang, Y. (2017) An Empirical Study of Multifactorial PSO and Multifactorial DE. *Proceedings of 2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, Spain, 5-8 June 2017, 921-928. <https://doi.org/10.1109/CEC.2017.7969407>
- [13] Bali, K.K., Ong, Y., Gupta, A. and Tan, P.S. (2020) Multifactorial Evolutionary Algorithm with Online Transfer Parameter Estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation*, **24**, 69-83. <https://doi.org/10.1109/TEVC.2019.2906927>
- [14] Li, G., Lin, Q. and Gao, W. (2020) Multifactorial Optimization via Explicit Multipopulation Evolutionary Framework. *Information Sciences*, **512**, 1555-1570. <https://doi.org/10.1016/j.ins.2019.10.066>
- [15] Wang, B., Li, H., Zhang, Q. and Wang, Y. (2018) Decomposition-Based Multiobjective Optimization for Constrained Evolutionary Optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **51**, 574-587. <https://doi.org/10.1109/TSMC.2018.2876335>