进化多目标组合优化的自适应复合范数分解

摘要

基于分解的多目标进化算法（MOEA/D）将多目标问题分解为一系列单目标子问题进行协同优化。加权和（WS）方法和切比雪夫（TCH）方法是两种最流行的标量化方法。它们在解决多目标组合优化问题（MOCOP）方面各有利弊，WS 方法允许 MOEA/D 比 TCH 方法收敛得更快；基于 WS 的 MOEA/D 无法获得无支持的非支配向量，而基于 TCH 的 MOEA/D 可以克服这一缺点。本文复现的是一种使用 WS 方法将种群推向 Pareto 前沿，然后切换到 TCH 方法以维持更多样化的种群的算法。为了更好地平衡收敛性和多样性，这个算法使用了一种新的标量化策略，称为自适应复合范数（ACN）。ACN 策略可以将 WS 和 TCH 结合起来并动态加权。

关键词：多目标优化；分解；组合优化

1 引言

多目标组合优化问题（MOCOP）可以写为：

$$\begin{aligned} &\text{minimize} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\intercal, \\ &\text{subject to} \quad \mathbf{x} \in X, \end{aligned} \tag{1}$$

其中，$\mathbf{x} = (x_1, \ldots, x_n)^\intercal$ 是决策向量（也称为解），$X \subset \mathbb{Z}^n$ 表示决策空间。$\mathbf{f} : \mathbb{Z}^n \to \mathbb{Z}^m$ 由 m 目标函数组成，$\mathbf{f}(\mathbf{x})$ 是 x 对应的目标向量。

MOCOP 的 PF 可以分为两部分：支持的和不支持的非支配向量 [13]。想要获得完整的 PF 是非常困难的。通常，采用多目标进化算法来寻找一组具有良好收敛性和多样性的解。收敛性反映了获得的目标向量集对 PF 的近似程度，而多样性则衡量了目标向量在 PF 上的分布。基于分解的多目标进化算法（MOEA/D）[19] 将多目标优化问题（MOP）转化为许多单目标子问题进行协作优化。子问题函数由标量化方法定义，MOEA/D 的性能关键取决于 [3,4,6,7,20]。许多研究研究了标量化技术来提高算法性能，例如设置适当的参数、自适应调整参数、结合多种标量化方法以及开发新的标量化方法。

加权和（WS）方法和切比雪夫（TCH）方法是流行的标量化方法。现有关于 WS 和 TCH 方法的研究可以总结如下，可以发现它们是互补的：

对于任何 Pareto 最优解，都存在一个基于 TCH 的子问题，其最优解就是 Pareto 最优解 [8]，而 WS 方法不具有这样的性质。也就是说，MOEA/D-WS 可能会错过不受支持的有效解决方案 [17]。基于 WS 的子问题比基于 TCH 的子问题收敛得更快 [2,12]。人们经常解释说，基于 WS 的子问题的开角比基于 TCH 的子问题的开角更大 [1,5]。

在处理 MOCOP 时，WS 方法被广泛使用。这意味着这些算法完全放弃了不支持的非支配向量，这可能会导致获得的非支配向量分布较差，从而无法很好地逼近 PF。在复现的论文中研究了如何将这两种方法结合起来进行 MOEA/D。并且实验结果表明，MOEA/D-WS 优于 MOEA/D-TCH，这表明更快的收敛可以在处理 MOCOP 时带来更好的最终性能。此外，结果表明具有 “早期 WS 和晚期 TCH” 的 MOEA/D 表现优于 MOEA/D-WS。相比之下，具有 “早期 TCH 和晚期 WS” 的 MOEA/D 与这两种算法相比效果较差。这些结果表明，在多样性之前优先考虑收敛是解决 MOCOP 的明智策略。换句话说，可以使用 WS 方法快速逼近 PF，然后使用 TCH 方法来增加种群多样性。MOEA/D-ACN 是基于复合范数（CN）方法，将 WS 方法和 TCH 方法结合在一起，并赋予其权重。因此复现的这篇论文提出了一种自适应调整权值的方案，以更好地平衡收敛性和分集性。

2 相关工作

2.1 MOEA/D 框架

MOEA/D 采用 N 均匀分布权重向量 $\{\mathbf{w}^j\}_{j=1}^N$ 来生成 N 子问题。MOEA/D 计算每两个子问题的权重向量之间的欧几里德距离，并使用这些距离来定义每个子问题的邻域。MOEA/D 维护一个具有 N 解 $\mathbf{x}^1, \ldots, \mathbf{x}^N$ 的群体，其中 $\mathbf{x}^j$ 与 $j = 1, \ldots, N$ 的第 j 个子问题相关。$\mathbf{x}^j$ 的邻域（表示为 $\mathbf{B}^j$）由第 j 个子问题的 T 最近邻的解组成。在每次迭代中，MOEA/D 对每个 $j \in \{1, \ldots, N\}$ 的第 j 个子问题进行如下操作：

1. 对从 B^j 中随机选择的解进行复制算子，生成新的解 x^{new}。
2. 对于 B^j 的每个解 x^k ，如果 x^{new} 比它更好，则将 x^k 替换为 x^{new}。

子问题函数由标量化方法定义。不同的标量化方法有不同的特点。标量化方法对 MOEA/D 的性能有显着影响。

2.2 标量化方法

标量化方法将多目标优化问题（MOP）转换为单目标优化问题。最常采用的两种标量化方法是 WS 方法和 TCH 方法。WS 方法定义单目标优化子问题如下：

$$\text{minimize } g^{ws}(x|w^j) = \sum_{i=1}^{m} w_i^j f_i(x), \tag{2}$$

其中 $\mathbf{w}^j = (w_1^j, \ldots, w_m^j)^T$ 是一个权重向量，满足对所有的 $i = 1, \ldots, m$ 都有 $w_i^j > 0$ 且 $\sum_{i=1}^m w_i^j = 1$。可以通过绘制其等高线和最优目标向量来说明基于 WS 的子问题。位于同一条等高线上的目标向量具有相同的子问题函数值。换句话说，等高线代表了子问题的一组同样好的解。优化基于 WS 的子问题只能得到支持的非支配向量，并且会错过不支持的向量 [9]。

TCH 方法定义的单目标优化子问题可以表示为:

$$\text{minimize } g^{tch}(x|w^j, z^*) = \max_{1 \le i \le m} \left\{ w_i^j |f_i(x) - z_i^*| \right\}, \tag{3}$$

其中 z^* 是参考向量，它通常被设置为理想向量、乌托邦向量或最低点向量 [15, 16]。z^* 也可以通过用户偏好来定义 [10]。定义 $\lambda^j = (1/w_1^j, \ldots, 1/w_m^j)^T$ 作为基于 TCH 的子问题的方向向量。方向向量和 PF 之间的交集是子问题的最优目标向量 [11]。对于任意非支配向量，其对应的 Pareto 最优解必定是式 (3) 的最优解。

WS 方法和 TCH 方法都是 L_p 标量化的特例。L_p 标量化将单目标优化子问题定义为：

$$g^{lp}(\mathbf{x}|\mathbf{w}^j, \mathbf{z}^*, p) = \left(\sum_{i=1}^{m} \left(w_i^j \, |f_i(\mathbf{x}) - z_i^*| \right)^p \right)^{\frac{1}{p}}. \tag{4}$$

L_1 标量化方法与 WS 方法相同，因为去掉了绝对值符号。这是因为 z^* 的设定不会影响任何基于 L_1 的子问题的最优解。当 $p \to \infty$ 时，L_p 标量化方法等同于 TCH 方法。

3 本文方法

为了更好地结合 WS 方法和 TCH 方法，本文提出了一种精心设计的算法，称为 MOEA/D-ACN。总体思路是前期使用 WS 方法，并根据积累的信息自适应转向 TCH 方法

3.1 本文 CN 方法概述

CN 方法 [18] 结合了 WS 方法（即 L_1 标量化）和 TCH 方法（即 L_∞ 标量化），将单目标子问题定义为：

$$g^{cn}(x|\mathbf{w}, \mathbf{z}^*, \rho) = \rho \sum_{i=1}^{m} w_i |f_i(x) - z_i^*| + (1-\rho) \max_{1 \le i \le m} \{ w_i |f_i(x) - z_i^*| \}, \tag{5}$$

其中 $\rho \in [0, 1]$ 是一个控制参数。当 $\rho = 1$ 时，方程 (5) 是一个基于 WS 的子问题；当 $\rho = 0$ 时，方程 (5) 是一个基于 TCH 的子问题。任何基于 CN 的子问题（$\rho \in (0, 1]$）的最优解绝对是一个帕累托最优解 [18]。CN 方法与增强 Tchebycheff 方法类似，但具有不同的属性。

3.2 MOEA/D-ACN 框架

算法1给出了 MOEA/D-ACN 的伪代码。MOEA/D-ACN 算法初始化步骤与 MOEA/D 类似，需要 N 权重向量、N 解、N 邻域和参考向量。在 MOEA/D-ACN 中，参考向量 z_i^* 设置为算法迄今为止获得的最小目标值。此外，还维护了一个外部档案 A ，并从初始群体 P 中的非支配解开始。P 和 A 不断迭代改进。当满足停止条件时，输出 A 作为最终的解集。

MOEA/D-ACN 的主循环如图1所示。每次迭代都以 ACN 策略开始。之后，进行 MOEA/D 的繁殖和替换。对于每个子问题，使用其相邻子问题的相关解决方案生成一个新的解决方案。然后，使用这个新解决方案来更新 z^* 及其邻居。迭代的最后是更新 A。A 是一个有界存档，最多保留 N 个不同的解决方案。采用文献 [21] 中的截断方法来移除包括重复解决方案在内的冗余解决方案。

ACN 策略是 MOEA/D-ACN 的一个关键部分，它为每个子问题确定一个 ρ 值。将第 j 个子问题的 ρ 值表示为 ρ_j。一般的想法是，每 L 次迭代，ρ_j 会根据 $(k-2)\Delta\rho$ $(k \in \{1, 2, 3\})$ 进行变化，其中 $\Delta\rho$ 和 L 是预定义的值，k 代表 ρ_j 的转变状态（即增加、不变或减少）。k 是通过以下两步程序确定的。

- 测试：在每次迭代中，随机选择一个子问题来测试三种转换状态的有效性。
- 验证：每 L 次迭代，使用从前 L 次测试中收集的证据来确定最有效的转换状态。

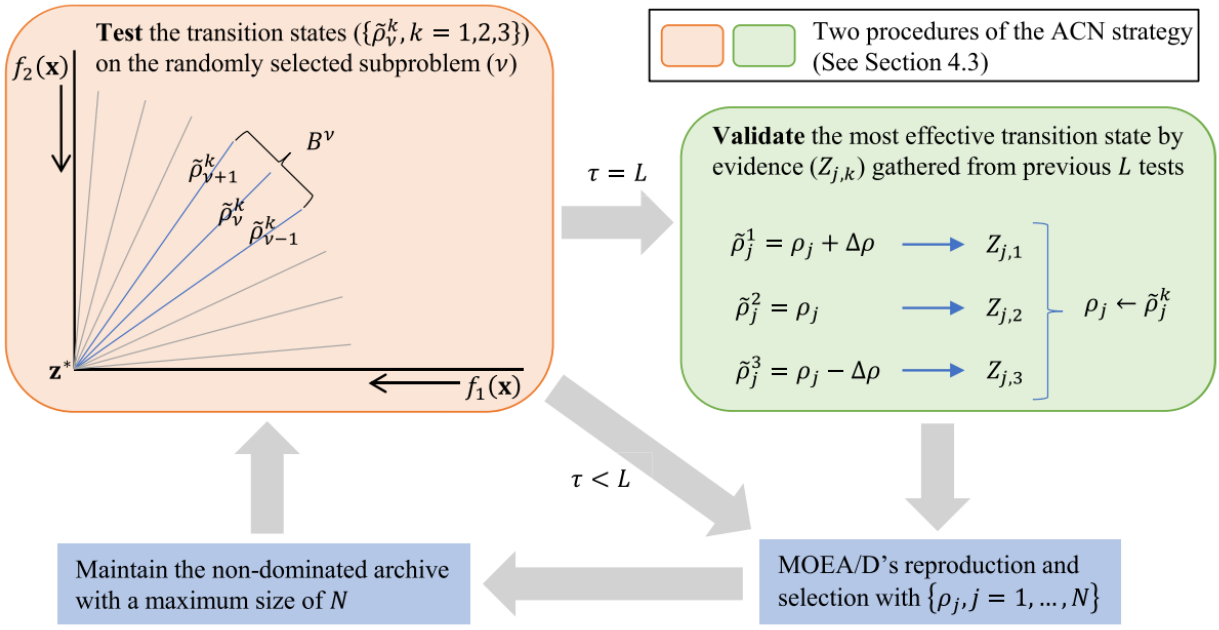

图 1. MOEA/D-ACN 流程图

3.3 ACN 策略

3.3.1 测试

测试的伪代码如算法2所示。ρ_j 的三种不同状态，表示为:

$$\hat{\rho}_j^k = \min\left\{\max\left\{\rho_j + (k-2)\Delta\rho, 0\right\}, 1\right\}, \quad k = 1,2,3, \tag{6}$$

在每一代中，随机选择一个子问题，其索引表示为 v 来评估它们。请注意，对于具有相同权重向量但不同 ρ 值的基于 CN 的子问题，最佳解决方案可能不同。首先，通过从总体和档案中匹配第 v 子问题邻域的最佳解决方案来构造三个过渡状态的三个测试总体。第 k 个状态的测试种群表示为 $\{x^{l,k}, l \in B^v\}$，而第 k 个状态中的第 l 个子问题与 $x^{l,k}$ 关联。然后生成新的解决方案。亲本的选择是预定义的：$x^{v,k}$ 总是被用作亲本之一，另一个亲本则轮流从 $x^{v,k}$ 的邻居中选取。在三个测试种群中，第二个测试种群首先产生新的解决方案 x^{new}。如果第一个（或第三个）状态的亲本与第二个状态的亲本完全相同，则第一个（或第三个）状态的亲本将 x^{new} 作为它们的后代。否则，亲本将产生一个新的后代。这是为了节省计算资源。总共，消耗的函数评估不超过 $3T$ 次。第 k 个状态的新解决方案存储在 O^k 中。此外，所有新的解决方案都存储在 O 中，以便更新 P 和 A。

此后，矩阵 Z 通过 $\tilde{O}^k$ $(k = 1,2,3)$ 更新，它被赋予了指示第 k 状态对第 j 个子问题有效性的任务。$Z_{j,k}$ 是第 j 个子问题在第 k 转换状态下的累积改进次数。例如，如果在 $\tilde{O}^k$ 中的一个新解决方案对于第 k 状态的所有相邻子问题都是更好的解决方案，那么对于所有 $l \in B^v$，$Z_{l,k} \leftarrow Z_{l,k} + 1$；如果这个新解决方案对于任何相邻子问题都不是更好的解决方案，那么对于所有 $l \in B^v$，$Z_{l,k}$ 将保持不变。

Algorithm 1 MOEA/D-ACN

Input: an MOCOP, stopping criteria, N (the population size), T (the neighborhood size), $\Delta\rho, L$

Output: A (archive)

Generate N uniformly distributed weight vectors $\{w^1, \ldots, w^N\}$
Determine the neighborhood B^j of w^j by finding the indexes of T closest weight vectors to w^j
Generate an initial population $P = \{x^1, \ldots, x^N\}$
$z_i^* \leftarrow \min\{f_i(x)|x \in P\}$ for $i = 1 \ldots, m$
$A \leftarrow$ non-dominated solutions of P
$\rho_j \leftarrow 1$ for $j = 1, \ldots, N$
$Z_{j,k} \leftarrow 0$ for $j = 1, \ldots, N, k = 1, 2, 3$
$\tau \leftarrow 0$
while the stopping criteria are not met **do**
    $O \leftarrow \emptyset$
    Test (Algorithm 2)
    Validation (Algorithm 3)
    for $j \in \{1, \ldots, N\}$ **do**
        Apply operators on x^j and randomly selected neighbors of j-th subproblem to generate a solution x^{new}
        $O \leftarrow O \cup x^{\text{new}}$
        $z_i^* \leftarrow \min\{z_i^*, f_i(x^{\text{new}})\}$ for $i = 1 \ldots, m$
        for all $k \in B^j$ **do**
            if $g^{cn}(x^{\text{new}}|w^k, z^*, \rho) \leq g^{cn}(x^k|w^k, z^*, \rho)$ **then**
                $x^k \leftarrow x^{\text{new}}$
            end if
        end for
    end for
    $A \leftarrow$ non-dominated solutions of $A \cup O$
    if $|A| > N$ **then**
        $A \leftarrow N$ solutions of A selected by the truncation method
    end if
end while

Algorithm 2 Test

Randomly select a subproblem and denote its index as v;
for $k \in \{1, 2, 3\}$ **do**
    $\tilde{x}^{l,k} \leftarrow \arg\min_{x \in P \cup A} g^{cn}(x|w^l, z^*, \hat{\rho}_l^k)$ for all $l \in B^v$;
end for
for $k = 1, 2, 3$ **do**
    $\tilde{O}^k \leftarrow \emptyset$;
end for
for $l \in B^v$ **do**
    $x^{\text{new}} \leftarrow$ Apply operators on $\tilde{x}^{v,2}$ and $\tilde{x}^{l,2}$ to generate a new solution;
    $\tilde{O}^2 \leftarrow \tilde{O}^2 \cup x^{\text{new}}$;
end for
for $k \in \{1, 3\}$ **do**
    if $\{\tilde{x}^{v,2}, \tilde{x}^{l,2}\} = \{\tilde{x}^{v,k}, \tilde{x}^{l,k}\}$ **then**
        $\tilde{O}^k \leftarrow \tilde{O}^k \cup x^{\text{new}}$;
    else
        $x^{\text{new}} \leftarrow$ Apply operators on $\tilde{x}^{v,k}$ and $\tilde{x}^{l,k}$ to generate another new solution;
        $\tilde{O}^k \leftarrow \tilde{O}^k \cup x^{\text{new}}$;
    end if
end for
$O \leftarrow \tilde{O}^1 \cup \tilde{O}^2 \cup \tilde{O}^3$;
$z^* \leftarrow \min\{z^*, \{f_i(x)|x \in O\}\}$ for $i = 1 \ldots, m$;
for $k \in \{1, 2, 3\}$ **do**
    for all $x^{\text{new}} \in \tilde{O}^k$ **do**
        for all $l \in B^v$ **do**
            if $g^{cn}(x^{\text{new}}|w^l, z^*, \rho_l^k) < g^{cn}(\tilde{x}^{l,k}|w^l, z^*, \rho_l^k)$ **then**
                $Z_{l,k} \leftarrow Z_{l,k} + 1$;
            end if
        end for
    end for
end for
for all $x^{\text{new}} \in O$ **do**
    for all $l \in B^v$ **do**
        if $g^{cn}(x^{\text{new}}|w^l, z^*, \rho_l) \leq g^{cn}(x^l|w^l, z^*, \rho_l)$ **then**
            $x^l \leftarrow x^{\text{new}}$;
        end if
    end for
end for

3.3.2 验证

验证的伪代码显示在算法3中。τ 是一个迭代计数跟踪器。"$\tau = L$" 意味着自从最后一次状态转换以来已经过去了 L 代，然后应该根据 Z 进行新的状态转换。在实践中，由于测试数量有限，$Z_{j,k}$ 并不足以代表有效性。因此，使用第 j 个子问题的 $\lceil \frac{N}{2} \rceil$ 个邻近子问题来计算一个更可靠的值来指示有效性。

集合 $\{\bar{Z}_{j,k}, k = 1, 2, 3\}$ 中的最大元素被认为是第 j 个子问题的最有效的转换状态。因此，相应的 k 被选为第 j 个子问题的下一个状态。此外，为了打破平局，让第三状态具有最高优先级，其次是第二状态，然后是第一状态。例如，当 $\bar{Z}_{j,1} = \bar{Z}_{j,2} = \bar{Z}_{j,3}$ 时，ρ_j 减小；当 $\bar{Z}_{j,1} = \bar{Z}_{j,2} > \bar{Z}_{j,3}$ 时，ρ_j 保持不变。具体来说，通过添加"0.1k" 来实现这一点。由于 $Z_{j,k}$ 必须大于 1，所以在没有平局的情况下"0.1k" 不会影响结果。这种打破平局的机制确保了 $\rho_j = 0$ 最终会发生。之后，如果 ρ_j 增加或减少，则将执行第 j 个子问题的匹配过程。z 在验证过程结束时重置。

Algorithm 3 Validation

$\tau \leftarrow \tau + 1$;
if $\tau = L$ **then**
  $\tau \leftarrow 0$;
  for $j \in \{1, \ldots, N\}$ **do**
    $\bar{Z}_{j,k} \leftarrow \sum_{j \in C^j} Z_{j,k}$ where C^j is the indexes of $\lceil \frac{N}{m+1} \rceil$ closest weight vectors to w^j;
    $l \leftarrow \arg\max_{k \in \{1,2,3\}} (\bar{Z}_{j,k} + 0.1k)$;
    $\rho_j \leftarrow \rho_j^l$;
    if $l \in \{1, 3\}$ **then**
      $x^j \leftarrow \arg\min_{x \in P \cup A} g^{cn}(x|w^j, z^*, \rho_j)$;
    end if
  end for
  $Z_{j,k} \leftarrow 0$ for $j = 1, \ldots, N, k = 1, 2, 3$;
end if

4 复现细节

4.1 与已有开源代码对比

本篇论文是开源代码的，源代码可以在 https://github.com/EricZheng1024/MOEA-D-ACN 找到。本次复现工作参考了源代码。

4.2 实验环境搭建

实验平台：MATLAB R2024b 和 PlatEMO（https://github.com/BIMK/PlatEMO）。PlatEMO [14] 是一个进化多目标优化平台。原文开源代码 MOEA/D-ACN 是在 PlatEMO 平台（版本 3.x）上实现的。复现代码是在 PlatEMO 平台（版本 4.9）中实现的。

4.3 界面分析与使用说明

PlatEMO 平台使用方法可参考 https://github.com/BIMK/PlatEMO 中的中文使用说明。

4.4 创新点

更改生成子代的方式，将源代码中的遗传算法的交叉和变异算子改为使用 PlatEMO 中的 OperatorGAhalf 遗传算法的交叉和变异。该遗传算法的优点在于灵活性强、适应性高，能够处理多种变量类型（实数、整数、标签、二进制、排列），并针对不同类型采用优化的交叉与变异操作。此外，通过只评估一半子代的策略，有效降低了计算成本，提高了运行效率。其模块化设计方便扩展，适用于多目标优化和混合编码问题，特别适合在计算资源有限的复杂优化场景中应用，兼顾解的探索性和开发性。

5 实验结果分析

图2展示了两个算法在二个目标上 600 个决策变量和 1000 个决策变量多目标背包问题上的超体积值。图3展示了两个算法在二个目标上 100 个决策变量和 200 个决策变量多目标旅行商问题上的超体积值。图4展示了两个算法在二个目标上 600 个决策变量和 1000 个决策变量多目标背包问题上平均豪斯多夫距离（Δp）的值。图5展示了两个算法在二个目标上 100 个决策变量和 200 个决策变量多目标旅行商问题上平均豪斯多夫距离（Δp）的值。HV 的值是越大越好，Δp 的值是越小越好。通过结果综合分析可以得出，简单更换遗传算法的方式并没有获得有效的改进。

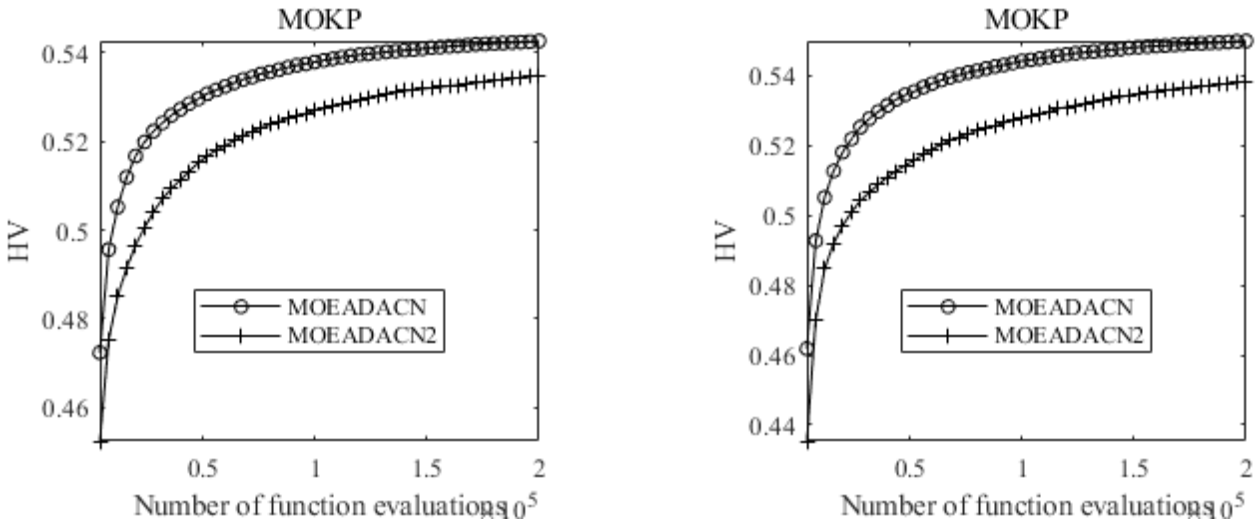

图 2. 多目标背包问题（HV）

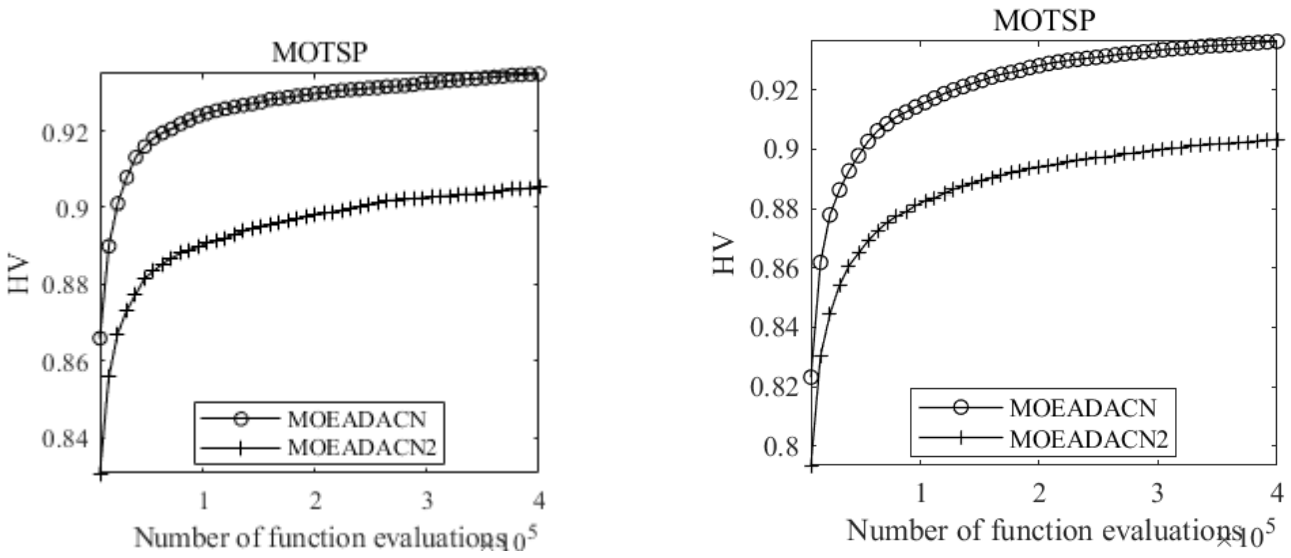

图 3. 多目标旅行商问题（HV）

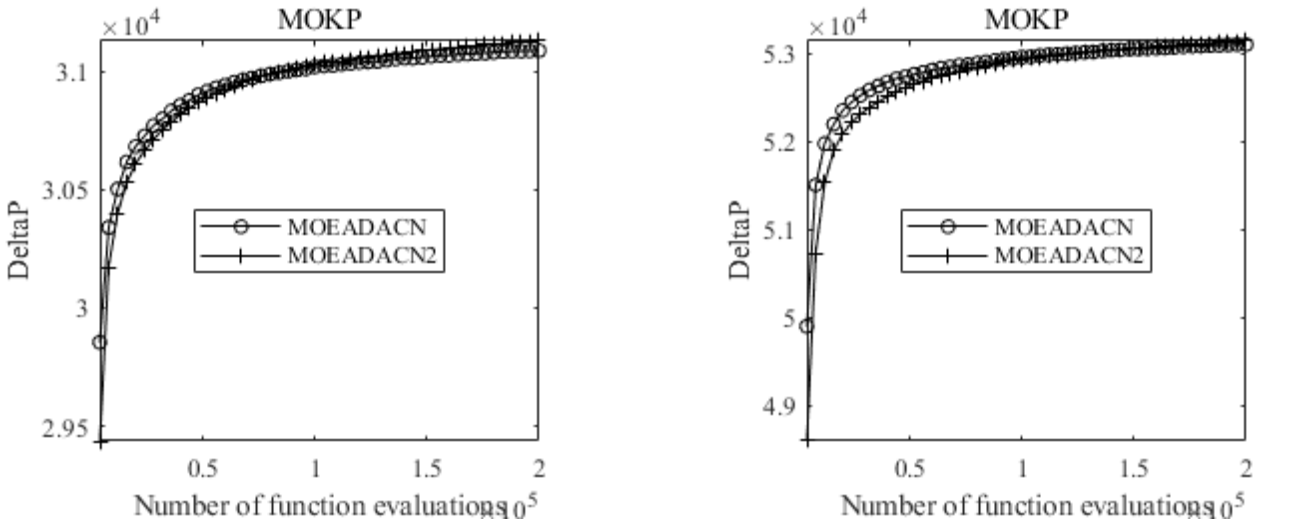

图 4. 多目标背包问题（Δp）

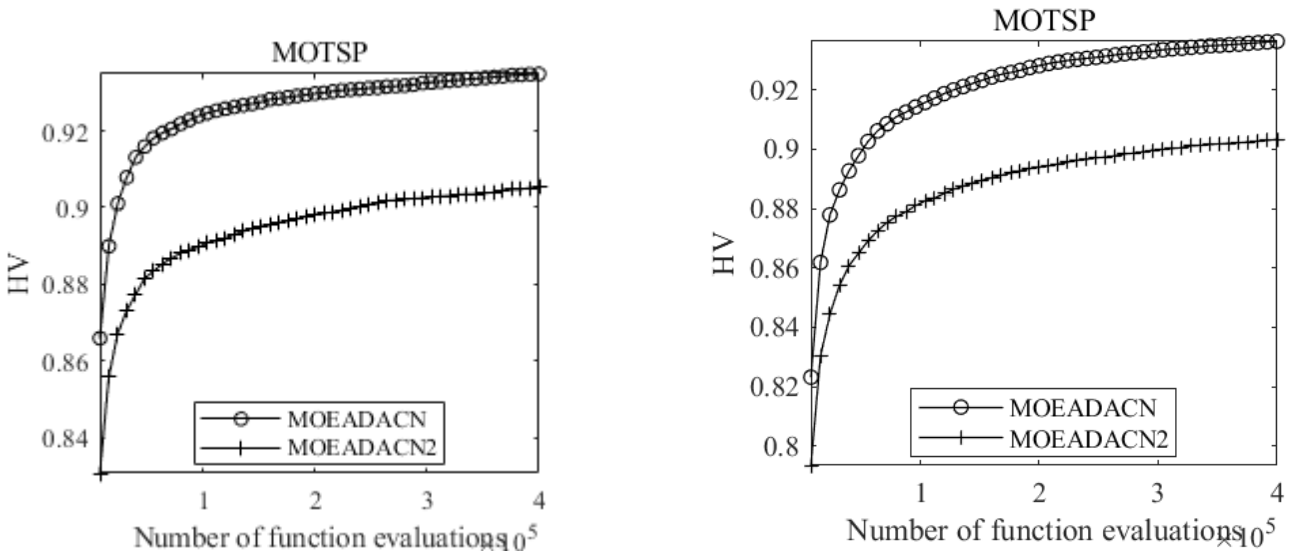

图 5. 多目标旅行商问题（Δp）

6 总结与展望

本次复现的论文提出了一种有效的标量化策略，称为基于分解的 MOEA 的 ACN 策略。首先启动 WS 方法，然后过渡到 TCH 方法，在收敛性和多样性之间取得更好的平衡。ACN 策略利用 CN 方法，自适应地调整每个子问题的 ρ 值，可以更好地解决这个问题。

未来可以在一些其他的工作上实施 ACN 策略，可以研究无参数、更高效、更通用的标量化策略。

参考文献

[1] Ioannis Giagkiozis and Peter J Fleming. Methods for multi-objective optimization: An analysis. *Information Sciences*, 293:338–350, 2015.

[2] Michael Pilegaard Hansen. Use of substitute scalarizing functions to guide a local search based heuristic: The case of motsp. *Journal of heuristics*, 6(3):419–431, 2000.

[3] Linjun He, Auraham Camacho, Yang Nan, Anupam Trivedi, Hisao Ishibuchi, and Dipti Srinivasan. Effects of corner weight vectors on the performance of decomposition-based multiobjective algorithms. *Swarm and Evolutionary Computation*, 79:101305, 2023.

[4] Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A study on the specification of a scalarizing function in moea/d for many-objective knapsack problems. In *Learning and Intelligent Optimization: 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers 7*, pages 231–246. Springer, 2013.

[5] Hisao Ishibuchi, Yuji Sakane, and Yusuke Nojima. Use of multiple grids with different scalarizing functions in moea/d. In *SCIS & ISIS SCIS & ISIS 2010*, pages 898–903. Japan Society for Fuzzy Theory and Intelligent Informatics, 2010.

[6] Genghui Li, Zhenkun Wang, Qingfu Zhang, and Jianyong Sun. Offline and online objective reduction via gaussian mixture model clustering. *IEEE Transactions on Evolutionary Computation*, 27(2):341–354, 2022.

[7] Genghui Li and Qingfu Zhang. Multiple penalties and multiple local surrogates for expensive constrained optimization. *IEEE Transactions on Evolutionary Computation*, 25(4):769–778, 2021.

[8] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[9] Özgür Özpeynirci and Murat Köksalan. An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315, 2010.

[10] Yutao Qi, Xiaodong Li, Jusheng Yu, and Qiguang Miao. User-preference based decomposition in moea/d without using an ideal point. *Swarm and evolutionary computation*, 44:597–611, 2019.

[11] Yutao Qi, Xiaoliang Ma, Fang Liu, Licheng Jiao, Jianyong Sun, and Jianshe Wu. Moea/d with adaptive weight adjustment. *Evolutionary computation*, 22(2):231–264, 2014.

[12] Celso C Ribeiro, Pierre Hansen, Pedro Castro Borges, and Michael Pilegaard Hansen. A study of global convexity for a multiple objective travelling salesman problem. *Essays and surveys in metaheuristics*, pages 129–150, 2002.

[13] Tetsuzo Tanino, Tamaki Tanaka, Masahiro Inuiguchi, Matthias Ehrgott, and Xavier Gandibleux. Multiple objective combinatorial optimization—a tutorial. *Multi-objective Programming and Goal Programming: Theory and Applications*, pages 3–18, 2003.

[14] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. Platemo: A matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12(4):73–87, 2017.

[15] Zhenkun Wang, Yew-Soon Ong, Jianyong Sun, Abhishek Gupta, and Qingfu Zhang. A generator for multiobjective test problems with difficult-to-approximate pareto front boundaries. *IEEE Transactions on Evolutionary Computation*, 23(4):556–571, 2018.

[16] Zhenkun Wang, Qingfu Zhang, Hui Li, Hisao Ishibuchi, and Licheng Jiao. On the use of two reference points in decomposition based multiobjective evolutionary algorithms. *Swarm and evolutionary computation*, 34:89–102, 2017.

[17] Zhenkun Wang, Hui-Ling Zhen, Jingda Deng, Qingfu Zhang, Xijun Li, Mingxuan Yuan, and Jia Zeng. Multiobjective optimization-aided decision-making system for large-scale manufacturing planning. *IEEE Transactions on Cybernetics*, 52(8):8326–8339, 2021.

[18] Andrzej P Wierzbicki. A methodological approach to comparing parametric characterizations of efficient solutions. In *Large-Scale Modelling and Interactive Decision Analysis: Proceedings of a Workshop sponsored by IIASA (International Institute for Applied Systems Analysis) and the Institute for Informatics of the Academy of Sciences of the GDR Held at the Wartburg Castle, Eisenach, GDR, November 18–21, 1985*, pages 27–45. Springer, 1986.

[19] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

[20] Chunliang Zhao, Yuren Zhou, and Yuanyuan Hao. Decomposition-based evolutionary algorithm with dual adjustments for many-objective optimization problems. *Swarm and Evolutionary Computation*, 75:101168, 2022.

[21] E ZITZLER, M LAUMANNS, and L THIELE. Improving the strength pare—to evolutionary algorithm. *Evolutionary Computation*, 5(2):121, 2001.