ATM-R：一种用于约束多目标进化优化的基于参考点的自适应权衡模型

王炳川、秦云川、孟宪兵、刘志忠

arXiv:2301.03317v1 [cs.NE] 9 Jan 2023

摘要—**约束多目标进化优化的目标是获得一组收敛性良好且分布均匀的可行解。为实现这一目标，需要在可行性、多样性和收敛性之间进行权衡。然而，由于这三项因素的重要性会随进化阶段而变化，因此仅使用一种权衡模型同时平衡这三者并非易事。作为替代方案，我们在不同阶段采用不同的权衡模型，并提出了一种名为 ATM-R 的新算法。在不可行阶段，ATM-R 将多样性与可行性之间的权衡纳入考量，旨在引导种群沿着多种搜索方向向可行区域移动。在半可行阶段，ATM-R 促使“可行性与多样性之间的权衡”向“多样性与收敛性之间的权衡”转变，从而有助于发现足够多的可行区域，并随后加快对可行帕累托最优解的搜索。在可行阶段，算法考虑多样性与收敛性之间的权衡，以获得一组收敛性良好且分布均匀的可行解。值得注意的是，ATM-R 利用参考点的优势来实现这些权衡模型。此外，ATM-R 还设计了一种多阶段交配选择策略，以生成适用于不同进化阶段的有潜力解。在大量基准测试函数上开展的系统实验表明，与五种最先进的约束多目标进化优化算法相比，ATM-R 有效且具有竞争力。**

索引词——**约束多目标进化优化、自适应权衡模型、参考点、多阶段交配选择**

I. 引言

MANY 科学或工程问题涉及在约束条件下优化冲突目标，可将其表述为约束多目标优化问题（CMOPs） [1]:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^{\mathrm{T}} \in \mathbb{R}^m \\ \text{s.t.} \quad & g_j(\mathbf{x}) < 0,\ j = 1, \cdots, n_g \\ & h_j(\mathbf{x}) = 0,\ j = n_g + 1, \cdots, n_g + n_h \\ & \underline{x}_j \le x_j \le \overline{x}_j,\ j = 1, \cdots, D \end{aligned}, \quad (1)$$

其中，$\mathbf{F}(\mathbf{x})$表示由 m个冲突目标组成的目标向量（即 $f_i(\mathbf{x})$ $i = 1, \cdots, m$）；$\mathbf{x} = (x_1, \cdots, x_D)^{\mathrm{T}}$是一个 D维决策向量/解；$\underline{x}_j$ 和 $\overline{x}_j$ 分别是 x_j的下界和上界；$\mathbb{S} = \prod_{j=1}^{D}[\underline{x}_j, \overline{x}_j]$ 表示决策空间； $g_j(\mathbf{x})$和$h_j(\mathbf{x})$分别表示第 j个不等式约束和第（$j - n_g$）个等式约束； n_g 和 n_h 分别是不等式约束和等式约束的数量。

求解约束多目标优化问题时，我们通常用约束违反程度来量化约束违反：

$$G(\mathbf{x}) = \sum_{j=1}^{n_g+n_h} G_j(\mathbf{x}). \quad (2)$$

G_j（$\mathbf{x}$）表示第 j 个约束 [2] 的约束违反程度：

$$G_j(\mathbf{x}) = \begin{cases} \max(0, g_j(\mathbf{x})), & 1 \le j \le n_g \\ \max(0, |h_j(\mathbf{x})| - \delta), & n_g + 1 \le j \le n_g + n_h \end{cases} \quad (3)$$

其中， δ 是一个较小的正值，用于在一定程度上放宽等式约束。当且仅当 $G(\mathbf{x}) = 0$时，解 $\mathbf{x}$ 称为可行解。所有可行解构成可行域：$\Omega = \{\mathbf{x} \in \mathbb{R}^D | G(\mathbf{x}) = 0\}$。对于两个解 $\mathbf{x}_u, \mathbf{x}_v \in \Omega$，当且仅当 $\forall j \in \{1, \cdots, m\}, f_j(\mathbf{x}_u) \le f_j(\mathbf{x}_v) \bigwedge \exists j \in \{1, \cdots, m\}, f_j(\mathbf{x}_u) < f_j(\mathbf{x}_v)$ 时，称 $\mathbf{x}_u$ 帕累托支配 $\mathbf{x}_v$，记作 $\mathbf{x}_u \prec \mathbf{x}_v$。当且仅当$\neg\exists \mathbf{x}_v \in \Omega, \mathbf{x}_v \prec \mathbf{x}_p$时，解 $\mathbf{x}_p \in \Omega$ 被视为帕累托最优解。所有帕累托最优解构成约束帕累托集，其在目标空间中的映像称为约束帕累托前沿（CPF）。约束多目标进化优化的目标是寻求一组收敛性良好且分布均匀的可行解，以逼近约束帕累托前沿。

为实现这一目标，约束多目标优化领域已形成共识：应在可行性、多样性和收敛性之间取得良好权衡 [3]。值得注意的是，这三个要素的重要性会因进化阶段不同而有所变化。以可行性为例，在不可行阶段，可行性信息对于定位可行区域不可或缺，因此可行性非常重要，这对约束多目标优化至关重要。然而，在可行阶段，由于所有解都已变为可行，可行性便无关紧要。我们只需考虑多样性与收敛性之间的权衡。由于

B.-C. Wang 任职于中国长沙 410083 中南大学自动化学院（电子邮箱：bingcwang@csu.edu.cn）。

Y. Qin 和 Z.-Z. Liu 任职于中国长沙 410082 湖南大学信息科学与电子工程学院（电子邮箱：liuzz@hnu.edu.cn；qinyunchuan@hnu.edu.cn）。

X.-B. Meng任职于中国广州510006华南理工大学计算机科学与工程学院（电子邮箱：axbmeng@scut.edu.cn）。

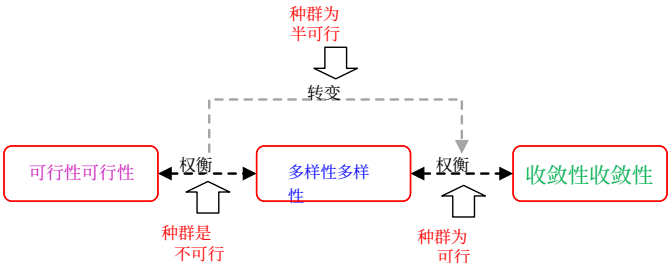

图 1. 在可行性、多样性和收敛性之间实现权衡的任务分解。

它们的重要性各不相同，要用单一的权衡模型同时平衡这三个要素并非易事。

作为替代方案，我们在不同进化阶段采用不同的权衡模型，提出一种带参考点的自适应权衡模型（ATM-R）来处理约束多目标优化问题。图 1 展示了 ATM-R 中考虑的权衡：

- 在不可行阶段实现可行性与多样性之间的权衡：当种群中的个体全部不可行时，首要目标是尽可能找到更多可行区域，因为帕累托最优解可能分散在不同的可行区域中。为此，应在可行性和多样性之间实现权衡，使种群从不同的搜索方向朝可行区域移动。

- 在半可行阶段，推动从“可行性与多样性之间的权衡”转向“多样性与收敛性之间的权衡”：当种群处于半可行状态时（即种群中既有不可行解，也有可行解），需要考虑两种情况。在早期阶段，仅发现了少数可行区域。在这种情况下，仍应优先权衡可行性与多样性，以寻找更多有潜力的可行区域。一旦找到足够多的可行区域，在后期阶段，就应着重推动种群快速趋近约束帕累托前沿（CPF），同时使其均匀分布在该前沿上。因此，应重点考虑收敛性与多样性之间的权衡。总之，在这一阶段，我们应从“可行性与多样性之间的权衡”转向“多样性与收敛性之间的权衡” [3]。

- 在可行阶段实现多样性与收敛性之间的权衡：当种群完全可行时，最终任务是推动可行解快速逼近约束帕累托前沿，同时保持良好的多样性。显然，应当实现多样性与收敛性之间的权衡 [4]。

总之，约束多目标进化算法的核心在于如何实现上述权衡。可行阶段中的权衡已在进化多目标优化领域得到充分研究。为方便起见，ATM-R 直接采用一种现成的无约束多目标进化算法（MOEA）来实现这一权衡。至于其他两个阶段中的权衡，相关研究仍然相对匮乏。尤其是关于权衡半可行阶段中的权衡，鲜有研究关注这一问题。事实上，要实现这两个阶段中的权衡，一个重要的问题是如何处理不可行解。约束多目标进化优化领域的以往经验表明，不可行解不仅有助于维持多样性，还能加快收敛。在 ATM-R 中，利用参考点的优势，选择适用于不同进化阶段的不同类型的不可行解。总之，本文的主要贡献如下：

- 我们不采用单一的权衡模型，而是在不同进化阶段采用不同的权衡模型，由此提出一种新型约束多目标优化算法（CMOEA），称为 ATM-R。尽管算法在进化过程中不可避免地会经历三个阶段，但鲜有研究尝试针对不同阶段开发不同的权衡模型，以实现更明确的适应性。

- 借助参考点的优势，我们提出了一种新的视角，用于选择适合不同进化阶段的有前景的不可行解。据我们所知，沿这一方向开展的相关研究仍然很少。

- 本文提出了一种多阶段交配选择策略，可针对不同进化阶段自适应地选择合适的交配亲本。

- 已在三组测试套件上开展了系统实验，其中包括 36 个基准约束多目标优化问题，以验证 ATM-R 的有效性。与五种先进的约束多目标进化算法进行比较后发现，在大多数测试问题上，ATM-R 的性能显著优于或与竞争算法相当。此外，实验还验证了 ATM-R 中若干重要算法组件的优势。

本文其余部分安排如下。第二节简要回顾相关的约束多目标进化算法。第三节介绍 ATM-R 的详细内容。第四节将 ATM-R 与五种具有代表性的约束多目标进化算法进行比较。第五节深入分析 ATM-R 的若干方面。第六节给出总结与未来工作。

II. 相关工作

约束多目标优化已成为进化计算领域的研究热点，并涌现出大量约束多目标进化算法。根据是否利用不可行解，这些算法可分为两类：可行性驱动型约束多目标进化算法和不可行性辅助型约束多目标进化算法。

A. 可行性驱动的约束多目标进化算法

可行性驱动型约束多目标进化算法由可行性信息驱动，其中可行解被认为优于不可行解。一些可行性驱动的约束多目标进化算法使用约束支配原则（CDP）比较两个解 [5]。在 CDP 中，如果满足以下条件之一，则称解$\mathbf{x}_u$ 优于另一个解

$\mathbf{x}_v$:

- 和 $\mathbf{x}_u$ 和 $\mathbf{x}_v$ 都不可行，且 $G(\mathbf{x}_u) < G(\mathbf{x}_v)$;
- $\mathbf{x}_u$ 可行，而$\mathbf{x}_v$ 不可行;
- $\mathbf{x}_u$ 和 $\mathbf{x}_v$ 都可行，且 $\mathbf{x}_u \prec \mathbf{x}_v$。

由于偏好可行解，约束支配原则可以促使种群迅速向可行区域演化。它已广泛应用于不同类型的多目标进化算法中 [6],[7] ，并用于一系列工程优化问题 [8], [9]。Liu 等 [6] 将基于角度的选择策略、基于移位的密度估计策略和约束支配原则结合起来，用于约束多目标优化。Jain 和 Deb [7] 提出了一种基于参考点的非支配排序方法，并将其与约束支配原则结合，用于约束多目标优化。Jan 和 Khanum [10] 将约束支配原则嵌入 MOEA/D 框架中，并将其性能与随机排序 [11]进行了比较。在评估约束多目标进化算法的性能时，基于约束支配原则的约束多目标进化算法通常用作基准算法 [12]–[14]。

可行性规则广泛用于约束单目标优化，现已扩展用于求解约束多目标优化问题。Liu 等人 [15] 将可行性规则与基于指标的多目标进化算法相结合，并将其性能与其他类型的约束多目标进化算法进行了比较。Fan等人 [16] 对用于约束多目标优化的 MOEA/D 开展了比较研究。研究将包括可行性规则在内的不同约束处理技术嵌入 MOEA/D 框架中。

有些约束多目标进化算法在种群中没有可行解时会着重处理约束。Woldesenbet 和 Yen [17]提出了一种用于求解约束多目标优化问题的自适应惩罚方法，该方法结合了自适应惩罚函数和距离度量来处理约束。事实上，当种群中的解全部不可行时，该自适应惩罚方法会仅依据约束而不考虑目标来比较两个解。Liu 和 Wang [18]提出了一种用于求解约束多目标优化问题的两阶段约束多目标进化算法。当种群中的解全部不可行时，该算法将所有目标结合起来，并使用可行性规则处理约束。凭借其搜索算法的优越能力，两阶段约束多目标进化算法能够处理决策空间中的复杂约束。Jimenez 等人 [19] 设计了一种用于约束多目标优化的约束多目标进化算法，其中采用极小极大形式处理约束。此外，在将子代插入新种群时，使用可行性规则比较两个解。Miyakawa 等人[20] 开发了一种用于求解约束多目标优化问题的两阶段非支配排序方法。该方法依据约束进行非支配排序，将种群划分为若干前沿；随后，再依据目标对所得前沿进行非支配排序。在这种方式下，环境选择中约束优先于目标。

B. 不可行性辅助型约束多目标进化算法

不可行性辅助约束多目标进化算法利用不可行解来求解约束多目标优化。目前大多数先进的约束多目标进化算法都属于这一类。

一些约束多目标进化算法通过采用同时考虑约束和目标的比较准则，隐式地利用不可行解。Ma 和 Wang [3] 提出了一种基于偏移的惩罚函数，根据附近可行解所提供的信息对不可行解进行惩罚。Jiao等人 [21] 提出了一种修正目标函数法。当种群中的解全部不可行时，修正后的目标函数等价于一种距离度量，其中约束和目标被视为同等重要。Fan等人 [**?**] 提出了一种用于约束多目标优化的基于角度的约束支配原则。给定一个可行解和一个不可行解，若这两个解之间的夹角小于预定义阈值，则二者互不支配。因此，一些不可行解可以进入新种群，取代一些可行解。Young [22] 提出了一种融合排序度量来选择解。通过将个体在目标空间中的排序与其在约束空间中的排序相结合，不可行解可能优于可行解。同样，Ma 等人 [13] 设计了一种具有两种排序的新适应度函数，其中一种排序值依据约束支配原则获得，另一种则基于不考虑约束的帕累托支配计算。该约束方法可通过调整阈值 ε [2]来利用不可行性信息，因此已被广泛用于求解约束多目标优化问题 [23]。Zapotecas-Mart´ınez 和 Ponsich [24]将 MOEA/D 与该约束方法相结合来求解约束多目标优化问题，其中 ε 值根据约束违反程度设定。Fan 等人 [25] 通过动态设置 ε 值改进了该约束方法。Zhou 等人 [26] 扩展了该约束方法以求解约束多目标优化问题。当不可行解的约束违反程度大于 ε 值时，将谨慎地维持其多样性。原本用于约束单目标优化且广受欢迎的随机排序，也已被扩展用于求解约束多目标优化问题 [15], [27]。

一些约束多目标进化算法通过存档或协同进化，明确利用不可行解的优势。Ray 等 [28]提出了一种不可行性驱动进化算法，其中保留了少量靠近约束边界的不可行解。Li等 [29] 为约束多目标优化设计了一种双档案进化算法。一个档案集用于促进收敛性，另一个则用于保持多样性。多样性档案集的演化不考虑约束；因此，能够充分利用目标函数值较好的不可行解。Liu 等 [4]尝试通过双向协同进化求解约束多目标优化问题。约束支配原则用于推动主种群从搜索空间的可行一侧朝约束帕累托前沿演化。此外，依次采用非支配排序过程和基于角度的选择方案，促使种群在不可行区域内朝约束帕累托前沿演化。Tian 等 [30] 为约束多目标优化开发了一种协同进化框架。类似地，一个种群通过约束支配原则更新，另一个种群则通过无约束多目标进化算法更新。此外，从这两个种群中选出的精英个体会用于生成子代。Ishibuchi 等 [31]为约束多目标优化设计了一种 MOEA/D 双网格模型。该模型维护两个种群，并将不可行

目标函数值较好的解优先保留在次级种群中。Zhu 等 [32] 在 MOEA/D 中采用两种类型的权重向量来求解约束多目标优化问题。与收敛权重向量关联的解基于聚合函数进行更新，而与多样性权重向量关联的解则根据聚合函数和约束违反程度进行更新。Peng 等 [14] 为约束多目标优化采用了两种权重向量。具体而言，将约束违反程度视为另一个目标。随后，利用一组可行权重向量和一组不可行权重向量更新种群。此外，动态调整不可行权重向量集，以维持一定数量的目标函数值较好且约束违反程度较小的不可行解。

一些约束多目标进化算法将进化过程划分为多个阶段，并在其中一个阶段侧重于目标。Yang 等 [33] 将进化过程划分为约束搜索模式和无约束搜索模式。这两种搜索模式通过动态约束处理机制执行。Fan 等 [12] 提出了一种用于求解约束多目标优化问题的推拉搜索（PPS）框架，其中进化过程分为两个阶段：推阶段和拉阶段。在推阶段，种群由无约束多目标进化算法更新。在拉阶段，设计了一种改进的 ε 约束处理方法来应对复杂约束。自提出以来，PPS 框架已应用于多个领域 [34]，[35]。Yu等 [36] 提出了一种由动态选择偏好辅助的约束多目标差分进化算法。随着优化的推进，对解的选择偏好会从不可行性转向可行性。Tian 等 [37]提出了一种两阶段约束多目标进化算法，以平衡目标优化与约束满足。这两个阶段根据种群中的可行解比例动态执行。最近，Ming 等 [38]提出了一种用于约束多目标优化的简单两阶段进化算法。该算法在第一阶段侧重于逼近无约束帕累托前沿，并将可行解存入档案集。在第二阶段，该方法寻求逼近约束帕累托前沿，而已存档的可行解则

被用作初始种群。Peng等 [39] 针对具有欺骗性约束的约束多目标优化提出了一种两阶段进化算法。在第一阶段，分别采用两个子种群来探索可行区域和整个空间。第二阶段旨在逼近约束帕累托前沿。此外，还设计了一种不可行性利用策略，以利用不可行解所提供的有用信息。

III. 所提方法

ATM-R 的总体流程图如图 2 所示。顾名思义，ATM-R 利用参考点，在不同的进化阶段——即不可行阶段、半可行阶段和可行阶段——自适应地实现不同的权衡。这三个不同阶段中更新机制的细节分别在第三节 A 部分、

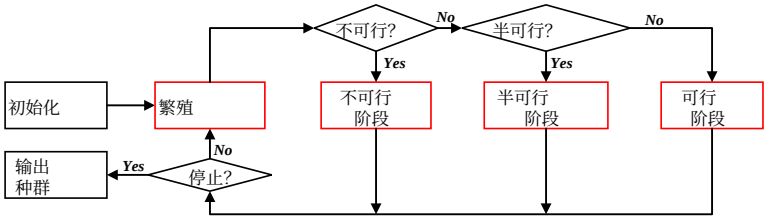

图2. ATM-R流程图。

算法1： 不可行阶段的更新机制

输入： 种群 $\mathbb{P}$，子代种群 $\mathbb{O}$
输出： 新种群 $\mathbb{P}$

$\mathbb{Q} \leftarrow \mathbb{P} \cup \mathbb{O}$;
根据以下内容将 $\mathbb{Q}$ 划分为 k 个前沿 $\hat{\mathbf{F}}(\mathbf{x})$: $\mathbb{F}_1, \cdots, \mathbb{F}_k$;
$\mathbb{P} \leftarrow \varnothing$;
对 $l = 1 : k$ **执行**
  如果 $|\mathbb{P}| + |\mathbb{F}_l| \geq N$ **则**
    中断;
  $\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{F}_l$;
如果 $|\mathbb{P}| + |\mathbb{F}_l| > N$ **则**
  对 n 均匀分布的参考点进行采样并生成相应的权重向量: $\mathbf{w}1, \cdots, \mathbf{w}n$;
  将 $\mathbb{F}_l$ 中的每个解分配给一个权重向量根据式（5）–（7）;
  当 $|\mathbb{P}| + |\mathbb{F}_l| > N$ **时**
    选择与其关联的权重向量解的最大数量: $\mathbf{w}_c$;
    在分配给$\mathbf{w}_c$的解中，选择 $G(\mathbf{x})$值最大的解: $\mathbf{x}_w$;
    $\mathbb{F}_l \leftarrow \mathbb{F}_l \backslash \mathbf{x}_w$;
$\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{F}_l$;

第三节 B 部分和第三节 C 部分中描述。除了环境选择过程之外，约束多目标进化算法的另一个关键要素是交配选择过程。在 ATM-R 中，我们提出了一种多阶段交配选择策略，以生成有助于实现不同权衡的潜在优良解。该策略的细节见第三节 D 部分。最后，ATM-R 的框架及相关讨论分别见第三节 E 部分和第三节 F 部分。

A. 不可行阶段的更新机制

在此阶段，ATM-R 旨在平衡可行性与多样性。换言之，它促使种群从不同的搜索方向朝可行性推进，从而尽可能找到更多可行区域。**算法1** 展示了 ATM-R 如何实现这一权衡。总体而言，这一过程包含两个关键要素。

1）在变换目标空间中进行非支配排序：遵循 [40], 中的思路，我们将 G（$\mathbf{x}$）视为附加目标函数，并将（1）变换为

无约束多目标优化问题：

$$\min \hat{\mathbf{F}}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}), G(\mathbf{x}))^{\mathrm{T}} \in \mathbb{R}^{m+1}. \quad (4)$$

显然，这种变换不会引入任何额外参数。此外，（4）同时考虑了目标函数和约束，这有助于保持种群多样性，并增强推动解趋向可行区域的驱动力。基于 $\hat{}\mathbf{F}$（$\mathbf{x}$），通过在变换目标空间中执行非支配排序过程，将种群划分为若干前沿，记为 $\mathbb{F}_1$，$\cdots$，$\mathbb{F}_k$。随后，依次从每个前沿中选取解，直到达到 $\sum_{i=1}^{l-1}|\mathbb{F}_i| < N \leq \sum_{i=1}^{l}|\mathbb{F}_i|$，其中 N 表示最终解集的大小。

2）基于规则参考点的选择：如果 $\sum_{i=1}^{l}|\mathbb{F}_i|$ 大于 N，则应从最后一个期望前沿 $\mathbb{F}_l$ 中进一步选择 ($n = N - \sum_{i=1}^{l-1}|\mathbb{F}_i|$) 个解。为完成此任务，本研究利用均匀分布的参考点，设计了一种基于常规参考点的选择方案。该方案实现起来相当简单。

- 首先，在目标空间中采样一组规则（即均匀分布的）参考点，以生成权重向量，记为 $\{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$，具体思路参见 [41]。
- 随后，将 $\mathbb{F}_l$ 中的一个解（记为$\mathbf{x}$）分配给与其归一化目标向量夹角最小的权重向量：

$$I = \underset{j\in\{1,\cdots,n\}}{\arg\min}\ \theta_j, \quad (5)$$

$$\theta_j = \arccos\left|\frac{\mathbf{F}'(\mathbf{x})^{\mathrm{T}}\mathbf{w}_j}{\|\mathbf{F}'(\mathbf{x})\|\cdot\|\mathbf{w}_j\|}\right|,\ j = 1, \cdots, n, \quad (6)$$

$$f_j'(\mathbf{x}) = \frac{f_j(\mathbf{x}) - z_j^{\min}}{z_j^{\max} - z_j^{\min}},\ j = 1, \cdots, m, \quad (7)$$

其中，I 表示解 $\mathbf{x}$ 被分配到的权重向量；θ_j 表示 $\mathbf{w}_j$ 与归一化目标向量 $\mathbf{F}'(\mathbf{x}) = (f_1'(\mathbf{x}) \cdots, f_m'(\mathbf{x}))^{\mathrm{T}}$之间的夹角；$\|\cdot\|$ 表示计算向量二范数的函数；$\mathbf{z}^{\max} = (z_1^{\max}, \cdots, z_m^{\max})^{\mathrm{T}}$ 和 $\mathbf{z}^{\min} = (z_1^{\min}, \cdots, z_m^{\min})^{\mathrm{T}}$ 分别表示估计的最差点和理想点。

- 随后，采用“多样性优先，可行性其次”的策略，逐一删除（$|\mathbb{F}_l| - n$）劣解。具体而言，首先找出关联解数量最多的权重向量[1]。直观来说，由于这些解关联同一个权重向量，它们的搜索方向会非常相似。为了保持搜索方向的多样性，有必要删除其中一个解。删除时会考虑这些解的可行性信息。丢弃 G（$\mathbf{x}$）值最大的解。这两

这些步骤将持续进行，直到删除（$|\mathbb{F}_l| - n$）个解。

图3给出了一个简单示例，以便更好地理解基于常规参考点的选择方案。我们考虑一个包含两个目标的约束多目标优化问题。假设有七个

[1]注意，若出现并列，则随机打破平局

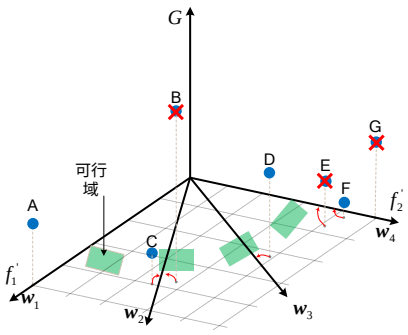

图3. 不可行阶段的更新机制。

解位于种群中，并且它们在变换目标空间中处于同一前沿。根据G(**x**)的值，这些个体按升序排列为**F**、**C**、**E**、**A**、**G**、**D**和**B** 。任务是选择四个解作为下一代。

1) 首先，均匀采样四个参考点，以生成四个权重向量，记为 {w1, $\cdots$, w4}。2) 接下来，将种群中的每个解分配给一个权重向量：$\mathrm{w}_1 \leftrightarrow$ {A}, $\mathrm{w}_2 \leftrightarrow$ {B, C}, $\mathrm{w}_3 \leftrightarrow$ {D}, and $\mathrm{w}_4 \leftrightarrow$ {E, F, G}。3) 随后，逐个删除三个解。首先删除 G，因为 w_4 匹配的解数量最多，并且与 E 和 F 相比，G 的 G(x) 值最大。根据这一原则，B 和 E 也将被删除。4) 最后，这些解（即 A、C、D 和 F）将进入下一代。

备注*1*：与 ATM-R 一样，ATMES[2] [40] 和 IDEA [28] 均在变换目标空间中采用非支配排序。它们的主要区别在于如何区分同一前沿中的解。具体而言，在 ATMES 中，仅根据 G(**x**) 选择解。 G(**x**) 值较小的解会被优先选择。这样一来，ATMES 会过度强调约束。这会导致搜索多样性方面的性能下降，而搜索多样性对于尽可能发现更多有前景的可行区域至关重要。相反，在 IDEA 中，仅考虑变换目标空间中的多样性来更新最后一个期望前沿 $\mathbb{F}_l$。遗憾的是，这种方式对可行区域的驱动力有限，进而导致收敛速度相对较低。与这两种方法不同，ATM-R 在更新 $\mathbb{F}_l$ 时同时考虑多样性和可行性，由此形成了“多样性优先、可行性其次”的策略。如图3所示，ATM-R 能够在多样性和可行性之间取得良好平衡，从而促使种群沿着多样化的搜索方向朝可行区域推进。

B. 半可行阶段的更新机制

ATM-R旨在推动半可行阶段从“可行性与多样性之间的权衡”转变为“多样性与

[2]尽管 ATMES 最初是为约束单目标优化而设计的，但它可以直接用于求解约束多目标优化问题。

算法2： 半可行阶段的更新机制

输入： 种群 $\mathbb{P}$、子代种群 $\mathbb{O}$, FEs, $MaxFEs$

输出： 新种群 $\mathbb{P}$

$\mathbb{Q} \leftarrow \mathbb{P} \cup \mathbb{O}$, $\mathbb{P} \leftarrow \varnothing$;
$\mathbb{Q}_f \leftarrow \{\mathbf{x} \in \mathbb{Q} | G(\mathbf{x}) = 0\}$, $\mathbb{Q}_{if} \leftarrow \{\mathbf{x} \in \mathbb{Q} | G(\mathbf{x}) > 0\}$;
如果 $|\mathbb{Q}_f| > N$ **则**
    $\mathbb{Q}_f \leftarrow N$ 个可行解从 $\mathbb{Q}_f$ 中选出，通过一种无约束多目标进化算法;
$\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{Q}_f$;
如果 $|\mathbb{Q}_{if}| > N$ **则**
    if $\frac{FEs}{MaxFEs} < 0.5$ 或 $|\mathbb{Q}_f| < N$ **则**
        $\mathbb{Q}_{if} \leftarrow N$ 从中选出的不可行解 $\mathbb{Q}_{if}$ 通过使用**算法1**;
    else
        使用以下方式生成 $|\mathbb{Q}_f|$ 权重向量 $\mathbb{Q}_f$ 中的解生成 $\mathbb{Q}_f$ 个权重向量，根据（8）–（9）;
        将 $|\mathbb{Q}_{if}|$ 中的每个解分配给一个权重向量，根据（5）–（7）;
        当 $|\mathbb{Q}_{if}| > N$ **时，执行**
            选择与以下对象关联的权重向量解的最大数量：$\mathbf{w}_c$;
            在分配给$\mathbf{w}_c$的解中，选择距离可行解最远的那个用于生成$\mathbf{w}_c$：$\mathbf{x}_w$;
            $\mathbb{Q}_{if} \leftarrow \mathbb{Q}_{if} \backslash \mathbf{x}_w$;
$\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{Q}_{if}$;

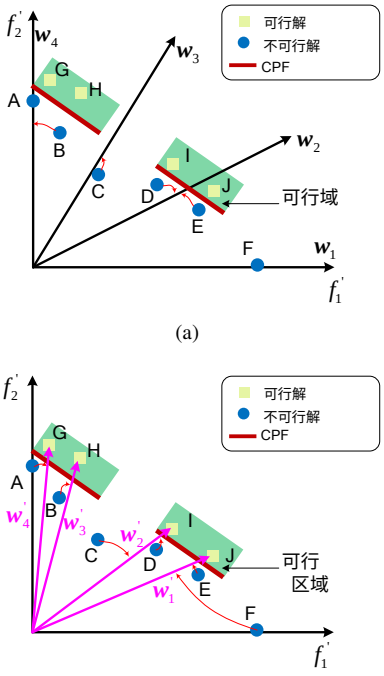

图4. 基于规则参考点的选择与自适应参考点选择中权重向量的差异示意图：(a) 基于规则参考点的选择中的权重向量；(b) 自适应参考点选择中的权重向量。

收敛性之间的权衡”（即种群同时包含不可行解和可行解）。这种转变有两个原因。在半可行阶段的早期，ATM-R必须尽可能多地找到可行区域。为此，它必须着重权衡可行性与多样性。在找到足够多的可行区域后，在后期，ATM-R应迅速引导种群趋向约束帕累托前沿（CPF），同时使其沿约束帕累托前沿均匀分布。因此，应优先考虑收敛性与多样性之间的权衡。**算法2** 展示了ATM-R如何在半可行阶段更新解。

从**算法2**可以看出，ATM-R 分别更新可行解和不可行解。为更新可行解，若可行种群的规模大于N，则使用无约束多目标进化算法对可行种群 $\mathbb{Q}_f$ 进行截断；否则，保留所有可行解。为更新不可行解，ATM-R 考虑了两种情况。在早期阶段，其目标是在可行性与多样性之间实现权衡，这与不可行阶段相同。因此，可直接在该阶段应用不可行阶段所用的更新机制（即**算法1**）。而在后期阶段，ATM-R 将重点转向多样性与收敛性之间的权衡。要实现这一权衡，一项重要任务是如何保留能够同时促进多样性和收敛性的不可行解。ATM-R 设计了以下两个步骤来完成这项任务。

1）非支配不可行解的发现：与当前种群中的可行解相比，非支配不可行解的目标函数值通常更小。利用它们的优势来促进收敛性是很自然的。为了区分这些不可行解，我们首先采用非支配排序过程，根据式（4）中的$\hat{}$**F(x)**，将合并种群（即**算法2**中的 $\mathbb{Q}$ ）划分为若干前沿。随后，选出第一前沿中的不可行解。如果这些非支配不可行解（记为 M）的数量小于 N，则全部保留；否则，将通过以下自适应参考点选择进一步区分。

2）自适应参考点选择：在这里，不再使用常规参考点辅助选择。原因是约束帕累托前沿可能是不连通的（见图 4），而利用均匀分布的参考点生成的某些权重向量（即图 4(a) 中的 $\mathbf{w}_1$ 和$\mathbf{w}_3$ ）无法指向约束帕累托前沿的任何部分。因此，利用这类权重向量保留的解（即**C** 和 **F**）远离约束帕累托前沿，几乎无法提升收敛速度，这是不理想的。取而代之的是，我们使用自适应参考点进行解选择。为方便起见，在本研究中，我们将可行解视为自适应参考点，因为它们能够为定位约束帕累托前沿提供重要线索（见图 4(b)）。基于这些参考点，可以方便地获得一组自适应权重向量。具体而言，对于第 i 个

可行解$\mathbf{x}_i$，其对应的权重向量（记为 $\mathbf{w}_i^{'}=(w_{i,1}^{'},\cdots,w_{i,m}^{'})^{\mathrm{T}}$）生成如下：

$$w_{i,j}^{'}=\frac{f_j^{'}(\mathbf{x})}{\sum_{j=1}^{m}f_j^{'}(\mathbf{x})},\ j=1,\cdots,m, \tag{8}$$

$$f_j^{'}(\mathbf{x})=\frac{f_j(\mathbf{x})-z_j^{\min}}{z_j^{\max}-z_j^{\min}},\ j=1,\cdots,m, \tag{9}$$

其中，$(f_1^{'}\mathbf{x})\ \cdots,f_m^{'}(\mathbf{x}))^{\mathrm{T}}$是归一化目标向量，$(z_1^{\max},\cdots,z_m^{\max})^{\mathrm{T}}$和 $(z_1^{\min},\cdots,z_m^{\min})^{\mathrm{T}}$分别表示估计的最差点和理想点。图 4 展示了使用常规参考点和自适应参考点生成的权重向量之间的差异。显然，使用自适应参考点获得的权重向量更符合约束帕累托前沿的特征。

准备好自适应权重向量后，基于自适应参考点的选择方案中的后续步骤相当简单。首先，按照基于常规参考点的选择方案中的方法（见公式 (5)–(7)），将每个非支配不可行解分配给一个权重向量。随后，以两步方式逐个删除 (M-N) 个不可行解。第一步是找出关联解数量最多的权重向量。第二步是在分配给该权重向量的解中，删除目标空间里距离与该权重向量对应的可行解最远的解。通常，第一步与许多基于分解的方法类似，有助于保持种群多样性。第二步则有助于保留那些接近可行解的不可行解，从而从搜索空间的不可行区域一侧提供推动力，使搜索趋向约束帕累托前沿。直观而言，这种方法可以加快收敛。

备注 *2*：在半可行阶段，ATM-R 中的种群规模大于或等于 N。原因是，较大的种群能够增强种群多样性，这对于“可行性与多样性之间的权衡”以及“多样性与收敛性之间的权衡”都至关重要。至于如何判断算法是否已进入半可行阶段的后期，我们考虑了两个必须同时满足的简单条件。第一个条件是，$\frac{FEs}{MaxFEs}$应大于 0.5。请注意，FEs和 $MaxFEs$分别表示函数评估次数和最大函数评估次数。第二个条件取决于可行解的数量，该数量应等于 N。第一个条件意味着算法已投入足够的搜索工作来寻找可行区域，而设置第二个条件是为了确保参考点的数量充足。在半不可行阶段的后期，如果未发现非支配不可行解，算法将进入可行阶段。

C. 可行阶段的更新机制

在此阶段，所有解均为可行解。在这种情况下，只需考虑多样性与收敛性之间的权衡，从而促使可行解快速趋近约束帕累托前沿，同时保持良好的多样性。显然，目前有效的无约束多目标进化算法可以用于实现这种平衡。因此，在 ATM-R 中，这一阶段直接采用现成的无约束多目标进化算法。

算法3： 多阶段交配选择策略

输入： 种群 $\mathbb{P}$, N
输出： 交配种群 $\mathbb{C}$

```
ℂ ← ∅;
对于 i = 1 : N，执行
    随机选择两个不同的解，分别记为 x_a 和x_b，从 ℙ中选取;
    x ℙ 完全不可行则
        如果 rand < 0.5 那么
            x_m ← 根据约束程度，x_a 和x_b中较优的一个
                  根据约束程度
                  违反;
        else
            x_m ← 在x_a 和x_b之间较优的一个
                  基于多样性;
    否则如果 ℙ 是可行的则
        如果 x_a ≺x_b 则
            x_m ← x_a;
        否则，如果 x_b ≺x_a ，则
            x_m ←x_b;
        else
            x_m ← 在 x_a 和 x_b之间较优的一个
                  基于多样性;
    否则，如果 ℙ 是半可行则
        如果 i < N/2，则
            x_m ← 通过x_a 和x_b ，选出较优者
                  采用不可行阶段的方法;
        else
            x_m ← 通过x_a 和x_b，选出较优者
                  采用可行阶段的方法;
    ℂ ← ℂ ∪x_m;
```

D. 多阶段交配选择策略

除了环境选择中的多阶段策略外，ATM-R还采用了多阶段交配选择策略。该策略为不同的进化阶段选择合适的交配亲本。该多阶段交配选择策略的具体细节如**算法3**所示。类似地，该策略也考虑了三个不同的阶段。

- 在不可行阶段，应同时关注种群多样性和可行性。因此，在锦标赛选择中，以相同的概率（即0.5）根据多样性和约束违反程度对解进行比较。请注意，多样性采用与 [30]中相同的方式进行量化。

- 在可行阶段，应同时考虑种群多样性和收敛性。借鉴 NSGA-II [5], 中锦标赛选择的思路，根据帕累托支配关系对解进行比较。

算法4： ATM-R

输入： 一个约束多目标优化问题，N、$MaxFEs$

输出： 最终种群 $\mathbb{P}$

$\mathbb{P} \leftarrow$ 从决策空间初始化的种群；
$FEs \leftarrow N$；
当 $FEs < MaxFEs$ **时**
  $\mathbb{C} \leftarrow$ 使用以下方法从 $\mathbb{P}$ 中选出的交配种群 **算法3**；
  $\mathbb{O} \leftarrow$ 通过以下方式生成的子代种群 对 $\mathbb{C}$执行遗传算子；
  $FEs \leftarrow FEs + N$；
  $\mathbb{Q} \leftarrow \mathbb{P} \cup \mathbb{O}$；
  如果 $\mathbb{Q}$ 完全不可行**则**
    $\mathbb{P} \leftarrow$ 使用以下方法从 $\mathbb{Q}$ 中选出的解 **算法1**；
  否则如果 $\mathbb{Q}$半可行**则**
    $\mathbb{P} \leftarrow$ 使用以下方法从 $\mathbb{Q}$ 中选出的解 **算法2** ；
  否则，如果可行，**则**
    $\mathbb{P} \leftarrow$ 通过无约束多目标进化算法从 $\mathbb{Q}$ 中选择的解 无约束多目标进化算法 ；

此外，如果两个解互不支配，则根据多样性进行比较。

- 半可行阶段需要弥合可行阶段与不可行阶段之间的差距。因此，在此阶段，交配种群的前一半采用不可行阶段的方法进行选择，另一半则采用可行阶段的方法进行选择。

E. ATM-R

总之，ATM-R的具体内容如**算法4**所示。首先，在决策空间中均匀采样一个包含 N 个解的种群（第1–2行）。随后，利用该种群搜索约束帕累托前沿，直至达到最大函数评估次数（第3–15行）。在搜索过程中，首先使用**算法3**中的多阶段交配选择策略选择 N 个交配亲本以生成子代（第4行）。接着，通过模拟二进制交叉（SBX） [42] 和多项式变异（PM） [43] 产生N 子代（第5–6行）。之后，根据种群可行性选择有前景的解（第7–14行），其中**算法1**和**算法2**分别用于不可行阶段和半可行阶段。注意，如果 $FEs \geq MaxFEs$，则输出最终种群 $\mathbb{P}$ 。

F. 讨论

本质上，ATM-R 是一种多阶段约束多目标进化算法。ATM-R 旨在不可行阶段实现多样性与可行性之间的权衡，在半可行阶段推动从“可行性与多样性之间的权衡”向“多样性与收敛性之间的权衡”转变，并在可行阶段实现多样性与收敛性之间的权衡。据我们所知，ATM-R 是首个在不同进化阶段同时考虑这些权衡的算法。此外，ATM-R 的一个特点是，它利用两类参考点，筛选出适用于不同进化阶段的有前景的不可行解。据我们所知，这一方向的相关研究几乎空白。根据我们的分析，显然，ATM-R 是一种用于约束多目标优化的全新约束多目标进化算法。

ATM-R 的计算时间复杂度主要由非支配排序和无约束多目标进化算法决定。假设 ATM-R 采用快速非支配排序和 NS- GAII [5] 。在不可行阶段的最坏情况下，变换目标空间中没有解能够支配另一个解。该非支配排序的时间复杂度为 $O((m+1)\cdot N^2)$。将每个解分配给一个权重向量的时间复杂度为 $O(m\cdot N^2)$。选择 N 个解的时间复杂度为 $O(N^2)$。因此，不可行阶段的时间复杂度为 $O((m+1)\cdot N^2) + O(m\cdot N^2) + O(N^2) = O((m+1)\cdot N^2)$。在半可行阶段，选择可行解的最坏情况时间复杂度为 $O(m\cdot N^2)$。在半可行阶段的早期，其最坏情况时间复杂度与不可行阶段相同： $O((m+1)\cdot N^2)$。在后期的最坏情况下，没有不可行解能够支配另一个解。这与不可行阶段相同。因此，其时间复杂度为 $O((m+1)\cdot N^2)$。半可行阶段的时间复杂度为 $O(m\cdot N^2)+O((m+1)\cdot N^2)+O((m+1)\cdot N^2) = O((m+1)\cdot N^2)$。在可行阶段，时间复杂度与 NSGA-II 相同： $O(m\cdot N^2)$。总之，ATM-R 的计算时间复杂度为$O((m+1)\cdot N^2) + O((m+1)\cdot N^2) + O(m\cdot N^2) = O(m\cdot N^2)$，这是可以接受的。

IV. 性能比较

本节基于大量基准测试函数评估 ATM-R 的性能。具体而言，我们使用 ATM-R 求解了三个测试套件，并将其性能与五种具有代表性的约束多目标进化算法进行比较。需要指出的是，所有实验均通过 PlatEMO 工具箱 [44] 实现。

A. 实验设置

1) 测试函数： 本研究采用了由 36 个基准测试函数组成的三个测试套件（例如，MW [45], CTP [46], 和 LIRC-MOP [25]）。这些测试函数具有各种颇具挑战性的特征，因此能够充分评估约束多目标进化算法的性能。大多数先进的约束多目标进化算法都采用这些测试函数进行实证研究。需要指出的是，MW 和 LIRCMOP 中决策变量的数量分别设为 15 和 10。有关这些测试函数的详细信息，请参见 [25],[45], [46]。

2）对比算法：为比较性能，本文考虑了五种具有代表性的约束多目标进化算法：NSGAII-CDP [5],PPS [12],约束双档案进化算法（CTAEA） [29],协同进化约束多目标

表 I NSGAII-CDP、PPS、CTAEA、CCMO、ToP 和 ATM-R 在三组基准测试函数上的 IGD 值。

测试函数	NSGAII-CDP 平均IGD（标准差）	PPS 平均IGD（标准差）	CTAEA 平均IGD（标准差）	CCMO 平均IGD（标准差）	ToP 平均IGD（标准差）	ATM-R 平均IGD（标准差）
MW1	4.0545e-2 (1.02e-1) -	2.3190e-2 (4.01e-2) -	2.1884e-3（9.96e-4）-	1.8990e-3 (1.42e-3) +	非数 (非数) -	2.1748e-3（1.70e-3）
MW2	2.3926e-2 (7.65e-3) -	4.2401e-2 (3.25e-2) -	1.7953e-2 (6.74e-3) ≈	2.1515e-2（8.20e-3）-	2.3108e-1 (1.89e-1) -	1.9130e-2 (9.76e-3)
MW3	7.4318e-2 (2.31e-1) -	7.5935e-3 (9.94e-4) -	5.4804e-3 (4.86e-4) ≈	5.2178e-3 (4.41e-4) ≈	5.9698e-1（2.78e-1）-	5.3646e-3 (4.08e-4)
MW4	5.5780e-2 (2.97e-3) -	5.3955e-2 (1.73e-3) -	4.6413e-2 (4.99e-4) -	4.1285e-2 (3.48e-4) ≈	非数 (非数) -	4.1255e-2 (3.45e-4)
MW5	4.2761e-1 (3.35e-1) -	1.4507e-1 (1.97e-1) -	1.5758e-2 (3.38e-3) -	4.6474e-3 (7.30e-3) -	非数 (非数) -	4.0638e-3 (1.06e-2)
MW6	8.0099e-2 (1.51e-1) -	1.0037e-1 (1.61e-1) -	1.1188e-2 (6.68e-3) ≈	5.2473e-2 (1.26e-1) -	1.0872e+0 (1.81e-1) -	1.5369e-2 (8.69e-3)
MW7	1.0205e-1 (1.93e-1) -	2.5520e-2 (1.88e-2) -	7.2156e-3 (5.22e-4) -	4.8994e-3 (4.80e-4) +	4.7226e-1 (2.39e-1) -	5.2004e-3（4.73e-4）
MW8	6.1793e-2 (8.78e-3) -	7.4112e-2 (2.69e-2) -	5.5531e-2 (2.47e-3) -	4.9189e-2 (1.58e-2) ≈	9.5949e-1 (2.01e-1) -	4.6368e-2（5.74e-3）
MW9	2.1737e-1 (3.10e-1) -	7.4032e-2 (1.81e-1) -	8.8691e-3 (9.23e-4) ≈	5.1927e-2 (1.79e-1) -	非数 (非数) -	9.9563e-3 (2.88e-3)
MW10	2.3341e-1 (2.36e-1) -	1.3321e-1 (1.48e-1) -	1.7599e-2 (1.22e-2) ≈	4.2867e-2 (2.54e-2) -	非数 (非数) -	2.7242e-2 (2.33e-2)
MW11	4.7335e-1 (3.24e-1) -	1.3565e-2 (2.15e-2) -	1.6564e-2 (2.80e-3) -	6.3416e-3 (5.30e-4) ≈	9.2141e-1 (1.37e-1) -	6.1791e-3 (2.34e-4)
MW12	8.2766e-2 (2.23e-1) -	2.9552e-2 (1.20e-1) +	8.0645e-3 (6.84e-4) +	3.0553e-2 (1.40e-1) ≈	非数 (非数) -	7.8769e-2 (2.24e-1)
MW13	2.0642e-1 (2.77e-1) -	1.3735e-1（6.32e-2）-	3.8211e-2 (2.66e-2) ≈	8.2172e-2（4.41e-2）-	8.3328e-1 (5.33e-1) -	5.2527e-2 (3.23e-2)
MW14	1.2974e-1 (1.27e-2) -	2.5313e-1（9.28e-2）-	1.1279e-1 (6.91e-3) +	9.8349e-2 (2.41e-3) +	4.9059e-1 (5.81e-1) -	1.1492e-1 (4.72e-2)
CTP1	8.1699e-2 (6.62e-2) -	1.9234e-2 (1.80e-2) -	1.8672e-2 (3.64e-2) -	4.4317e-3 (1.05e-3) -	3.9400e-3 (1.48e-4) -	3.2367e-3 (7.38e-5)
CTP2	2.4408e-3 (1.89e-3) -	3.7003e-3 (6.96e-4) -	4.6860e-2（1.25e-2）-	1.6836e-3（1.65e-4）-	4.4453e-3 (1.08e-3) -	1.4735e-3 (5.94e-5)
CTP3	6.2833e-2 (9.95e-2) -	3.1094e-2 (4.07e-3) -	5.8093e-2 (5.79e-3) -	2.2180e-2 (2.24e-3) -	3.2847e-2 (6.99e-3) -	1.0066e-2 (1.60e-3)
CTP4	2.4494e-1 (1.29e-1) -	1.4930e-1（1.84e-2）-	1.5350e-1 (1.92e-2) -	1.3538e-1 (2.18e-2) -	1.8414e-1 (3.29e-2) -	7.9556e-2 (1.13e-2)
CTP5	7.2574e-3 (2.92e-3) -	1.8168e-2 (6.11e-3) -	1.8209e-2（4.61e-3）-	7.6639e-3 (1.76e-3) -	1.2167e-2 (3.45e-3) -	3.3142e-3 (4.13e-4)
CTP6	1.1404e-2 (4.04e-4) -	1.3061e-2 (7.81e-4) -	3.8535e-2 (5.23e-3) -	1.0141e-2 (3.56e-4) -	1.5214e-2（2.78e-3）-	9.7103e-3 (3.13e-4)
CTP7	1.6882e-3（1.39e-3）-	1.6825e-3 (7.14e-5) -	1.6364e-3 (1.31e-4) -	1.1669e-3 (4.52e-5) ≈	1.5176e-3 (5.54e-5) -	1.1599e-3 (4.62e-5)
CTP8	1.2019e-1 (1.45e-1) -	1.1932e-2（5.26e-3）-	3.4505e-2（4.79e-3）-	5.5516e-3 (6.49e-4) -	8.0925e-2（1.38e-1）-	4.7357e-3（2.32e-4）
LIRCMOP1	2.6010e-1（8.10e-2）-	1.1024e-1（3.40e-2）-	3.7900e-1 (1.66e-1) -	2.0503e-1 (6.82e-2) -	1.2547e-1 (1.36e-1) -	3.5295e-2 (1.26e-2)
LIRCMOP2	1.9890e-1 (7.22e-2) -	7.3024e-2 (2.79e-2) -	1.2324e-1 (6.12e-2) -	1.1419e-1 (3.19e-2) -	6.8227e-2（5.38e-2）-	3.1146e-2 (9.36e-3)
LIRCMOP3	2.4894e-1 (8.38e-2) -	1.7697e-1（5.80e-2）-	3.4751e-1 (1.14e-1) -	2.0960e-1 (7.98e-2) -	3.6351e-1 (5.72e-2) -	2.3380e-2 (1.06e-2)
LIRCMOP4	2.3080e-1 (6.23e-2) -	1.4996e-1（5.59e-2）-	2.7661e-1 (1.38e-1) -	1.9069e-1（7.18e-2）-	3.2442e-1 (5.76e-2) -	2.3928e-2（1.11e-2）
LIRCMOP5	7.3176e-1 (4.81e-1) -	8.4362e-2 (2.44e-2) -	1.3918e-1 (4.41e-2) -	1.4046e-2 (8.21e-3) ≈	1.2091e-1 (3.48e-1) -	1.3635e-2 (6.48e-3)
LIRCMOP6	5.7447e-1 (4.57e-1) -	9.5258e-2 (6.31e-2) -	1.3633e-1 (1.13e-1) -	1.1357e-2 (7.75e-3) ≈	6.4593e-3 (3.45e-4) +	5.7790e-2 (1.51e-1)
LIRCMOP7	1.7441e-2 (1.32e-2) ≈	5.6488e-2 (5.84e-2) -	2.5246e-2 (9.12e-3) -	1.1404e-2 (6.38e-3) ≈	8.6357e-3 (2.52e-4) +	1.2400e-2 (5.03e-3)
LIRCMOP8	3.6946e-2 (4.51e-2) -	6.6479e-2 (7.05e-2) -	3.6096e-2 (6.64e-2) -	9.1531e-3 (5.01e-3) ≈	8.6820e-3 (4.53e-4) +	9.4267e-3 (3.85e-3)
LIRCMOP9	5.3564e-1 (1.24e-1) -	1.4063e-1 (8.94e-2) ≈	1.1622e-1 (5.42e-2) ≈	3.4398e-2 (3.91e-2) +	2.4115e-1 (1.73e-1) -	1.1216e-1 (7.48e-2)
LIRCMOP10	3.6496e-1 (9.66e-2) -	8.2848e-3 (1.47e-2) -	6.0919e-2 (6.50e-2) -	5.4399e-3 (3.36e-4) +	5.4878e-3 (2.21e-4) +	6.9018e-3 (6.27e-4)
LIRCMOP11	2.4114e-1（1.80e-1）-	8.1119e-3 (7.48e-3) -	1.3778e-1 (3.83e-2) -	2.4538e-3 (8.89e-5) +	1.2447e-1 (6.37e-2) -	5.3691e-3 (1.45e-2)
LIRCMOP12	1.5180e-1（8.66e-2）-	1.5216e-2 (2.43e-2) ≈	3.1152e-2（1.68e-2）-	4.6113e-3 (2.58e-3) ≈	2.9104e-2 (5.29e-2) ≈	7.8014e-3 (7.28e-3)
LIRCMOP13	2.3757e-1 (3.69e-1) -	1.1968e-1 (3.45e-3) -	1.0834e-1 (3.97e-4) -	9.3972e-2 (1.13e-3) -	1.2450e-1 (3.78e-3) -	9.3120e-2 (9.31e-4)
LIRCMOP14	2.0248e-1 (2.93e-1) -	1.1859e-1 (3.97e-3) -	1.1126e-1 (7.98e-4) -	9.5773e-2 (7.40e-4) -	1.1883e-1 (4.04e-3) -	9.4848e-2 (7.79e-4)
+/-/≈	0/35/1	1/33/2	2/27/7	6/19/11	4/25/1	

优化（CCMO） [30], 和两阶段进化算法（ToP） [18]。NSGAII-CDP 是一种经典的约束多目标进化算法，通常用作基准算法；其他四种约束多目标进化算法则是近期提出的最先进算法。NSGAII-CDP 是可行性驱动型约束多目标进化算法，其他算法则是不可行性辅助型约束多目标进化算法。在这四种不可行性辅助型约束多目标进化算法中，CTAEA 和 CCMO 属于多种群方法，通过档案集或额外种群显式利用不可行解。PPS 和 ToP 属于多阶段方法，它们将进化过程划分为若干阶段，并在某些阶段侧重于目标。需要指出的是，ATM-R 也是一种多阶段方法。

3）性能指标：采用两种常用的性能指标来评估约束多目标进化算法的性能：反向世代距离（IGD）和超体积（HV）。IGD 和 HV 均可衡量解集的收敛性和覆盖度。关于这两种指标的更多详情，请参阅 [47]。

4）参数设置：实验中涉及的参数如下：

- 所有参与比较的约束多目标进化算法的最终解集大小：$N = 100$；
- $MaxFEs$：$MaxFEs = 60,000$，适用于 MW 和 CTP测试套件；$MaxFEs = 300,000$，适用于 LIRCMOP测试套件；

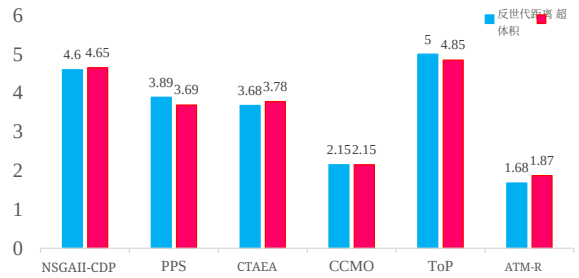

图 5. 六种约束多目标进化算法在 36 个测试函数上的反世代距离/超体积平均排名。排名值越低，性能越好。

- 独立运行次数：30。

除 ToP 外，所有约束多目标进化算法均使用 SBX 和多项式变异作为遗传算子。SBX 和多项式变异的参数如下：

- SBX 的交叉概率：1；
- PM 的变异概率：$1/D$；
- SBX 和多项式变异的分布指数：20。

此外，五种对比约束多目标进化算法的算法特定参数取自其原始论文。

表 II NSGAII-CDP、PPS、CTAEA、CCMO、ToP 和 ATM-R 在三组基准测试函数上的超体积值。

测试函数	NSGAII-CDP 平均超体积（标准差）	PPS 平均超体积（标准差）	CTAEA 平均超体积（标准差）	CCMO 平均超体积（标准差）	ToP 平均超体积（标准差）	ATM-R 平均超体积（标准差）
MW1	4.5445e-1（8.09e-2） -	4.6529e-1 (3.63e-2) -	4.8849e-1 (2.03e-3) -	4.8927e-1 (3.04e-3) +	非数 (非数) -	4.8853e-1 (3.60e-3)
MW2	5.4798e-1（1.15e-2） -	5.2241e-1 (4.44e-2) -	5.5765e-1 (1.14e-2) ≈	5.5199e-1 (1.30e-2) -	3.2482e-1（1.46e-1） -	5.5635e-1 (1.54e-2)
MW3	5.0168e-1（1.37e-1） -	5.4398e-1 (4.88e-4) +	5.4413e-1 (6.14e-4) +	5.4368e-1 (7.81e-4) +	1.2745e-1 (1.27e-1) -	5.4292e-1 (7.86e-4)
MW4	8.2309e-1 (5.63e-3) -	8.2478e-1 (2.49e-3) -	8.3814e-1 (4.04e-4) -	8.4116e-1 (4.35e-4) +	非数 (非数) -	8.4001e-1 (7.93e-4)
MW5	1.7725e-1 (9.80e-2) -	2.5212e-1（6.86e-2） -	3.1449e-1 (2.61e-3) -	3.2205e-1 (5.38e-3) -	非数 (非数) -	3.2214e-1 (6.45e-3)
MW6	2.8267e-1 (4.89e-2) -	2.5928e-1 (6.08e-2) -	3.1251e-1 (9.93e-3) ≈	2.9009e-1 (5.16e-2) -	1.2194e-2 (2.75e-2) -	3.0911e-1 (1.20e-2)
MW7	3.7706e-1 (6.78e-2) ≈	4.0647e-1 (2.09e-3) -	4.0868e-1 (1.03e-3) -	4.1205e-1 (5.95e-4) +	1.9015e-1 (7.70e-2) -	4.1019e-1 (9.75e-4)
MW8	4.9733e-1 (2.20e-2) -	4.7275e-1 (5.64e-2) -	5.2198e-1 (1.16e-2) -	5.2798e-1 (3.48e-2) -	4.6501e-2 (7.81e-2) -	5.3338e-1 (1.72e-2)
MW9	2.6792e-1 (1.71e-1) ≈	3.4455e-1 (1.00e-1) -	3.9100e-1 (2.43e-3) +	3.7160e-1（1.01e-1） -	非数 (非数) -	3.8287e-1 (4.60e-3)
MW10	3.1175e-1 (1.18e-1) -	3.5982e-1 (7.57e-2) -	4.3564e-1 (1.30e-2) ≈	4.1378e-1 (1.88e-2) -	非数 (非数) -	4.2764e-1 (1.94e-2)
MW11	3.2816e-1 (8.07e-2) -	4.4157e-1 (9.48e-3) -	4.4127e-1 (1.39e-3) -	4.4609e-1 (2.03e-3) -	2.2321e-1 (4.19e-2) -	4.4746e-1 (2.05e-4)
MW12	5.4172e-1 (1.81e-1) -	5.8181e-1 (1.06e-1) +	6.0052e-1 (7.80e-4) +	5.8415e-1 (1.10e-1) ≈	非数 (非数) -	5.4377e-1 (1.82e-1)
MW13	4.0153e-1 (5.63e-2) -	4.1137e-1 (4.30e-2) -	4.6130e-1 (1.23e-2) ≈	4.3974e-1 (2.53e-2) -	2.3054e-1 (1.15e-1) -	4.5371e-1 (1.66e-2)
MW14	4.5123e-1 (5.66e-3) -	4.2008e-1 (2.54e-2) -	4.6575e-1 (3.90e-3) ≈	4.7246e-1 (1.53e-3) +	3.4138e-1（1.53e-1） -	4.6217e-1 (1.49e-2)
CTP1	3.5920e-1 (1.97e-2) -	3.7510e-1 (5.31e-3) -	3.7588e-1 (1.03e-2) -	3.8065e-1 (3.93e-4) -	3.8036e-1 (1.15e-4) -	3.8106e-1 (1.09e-4)
CTP2	4.3083e-1（1.66e-3） -	4.2928e-1 (7.86e-4) -	3.9367e-1 (8.22e-3) -	4.3073e-1 (2.94e-4) -	4.2689e-1（1.40e-3） -	4.3128e-1 (2.45e-4)
CTP3	3.7267e-1 (5.45e-2) -	3.8376e-1 (4.03e-3) -	3.5588e-1 (7.25e-3) -	3.9219e-1 (2.18e-3) -	3.8119e-1 (6.97e-3) -	4.0507e-1 (1.75e-3)
CTP4	2.2838e-1 (6.88e-2) -	2.5406e-1 (1.86e-2) -	2.4611e-1 (2.00e-2) -	2.7265e-1（2.41e-2） -	2.2246e-1 (2.88e-2) -	3.3430e-1 (1.20e-2)
CTP5	3.9329e-1（2.59e-2） -	3.8822e-1 (4.53e-3) -	3.5629e-1 (9.01e-3) -	3.9643e-1（2.52e-3） -	3.8643e-1 (5.29e-3) -	4.0665e-1（1.86e-3）
CTP6	4.6359e-1 (3.75e-4) -	4.6198e-1 (6.48e-4) -	4.4896e-1 (2.68e-3) -	4.6381e-1 (3.03e-4) -	4.6034e-1 (1.83e-3) -	4.6468e-1 (2.22e-4)
CTP7	5.6701e-1 (2.23e-3) -	5.6676e-1 (9.22e-4) -	5.6637e-1 (4.11e-4) -	5.6745e-1 (1.69e-4) ≈	5.6692e-1 (1.86e-4) -	5.6721e-1 (1.62e-3)
CTP8	3.4937e-1 (2.45e-2) -	3.6598e-1 (2.71e-3) -	3.5213e-1 (3.79e-3) -	3.6932e-1 (8.68e-4) -	3.5503e-1 (2.42e-2) -	3.7069e-1 (4.25e-4)
LIRCMOP1	1.3114e-1 (2.17e-2) -	1.9042e-1 (1.06e-2) -	1.0593e-1 (3.85e-2) -	1.4954e-1 (1.82e-2) -	1.8833e-1（4.61e-2） -	2.2304e-1 (6.36e-3)
LIRCMOP2	2.5580e-1 (2.95e-2) -	3.2332e-1 (1.35e-2) -	2.9229e-1 (3.67e-2) -	2.9325e-1（2.05e-2） -	3.2282e-1 (2.82e-2) -	3.4702e-1 (3.44e-3)
LIRCMOP3	1.1697e-1 (2.61e-2) -	1.4007e-1 (1.86e-2) -	9.9083e-2 (2.08e-2) -	1.2942e-1 (2.47e-2) -	9.1646e-2 (1.42e-2) -	1.9947e-1 (4.24e-3)
LIRCMOP4	2.1773e-1 (2.72e-2) -	2.4241e-1 (3.29e-2) -	1.8974e-1 (4.86e-2) -	2.3242e-1 (3.14e-2) -	1.8379e-1 (2.31e-2) -	3.0693e-1 (3.76e-3)
LIRCMOP5	8.9983e-2 (1.05e-1) -	2.4475e-1（1.22e-2） -	2.4215e-1 (1.32e-2) -	2.8700e-1 (5.09e-3) ≈	2.6214e-1 (8.89e-2) -	2.8657e-1 (5.90e-3)
LIRCMOP6	8.6641e-2（5.68e-2） -	1.7269e-1 (1.24e-2) -	1.4582e-1 (3.86e-2) -	1.9402e-1 (3.37e-3) ≈	1.9677e-1 (1.59e-4) +	1.8377e-1 (3.56e-2)
LIRCMOP7	2.8752e-1 (6.87e-3) ≈	2.6957e-1 (2.12e-2) -	2.8582e-1 (3.18e-3) -	2.9114e-1 (4.01e-3) ≈	2.9389e-1 (1.55e-4) +	2.8957e-1 (4.05e-3)
LIRCMOP8	2.8236e-1 (1.47e-2) -	2.6950e-1 (1.80e-2) -	2.8424e-1 (1.40e-2) -	2.9321e-1 (3.40e-3) ≈	2.9387e-1 (2.04e-4) ≈	2.9235e-1 (3.25e-3)
LIRCMOP9	3.5772e-1 (8.15e-2) -	5.2821e-1 (2.63e-2) ≈	4.9955e-1 (2.81e-2) -	5.5712e-1 (8.54e-3) +	4.9582e-1 (5.19e-2) -	5.3708e-1 (2.00e-2)
LIRCMOP10	5.1193e-1 (6.34e-2) -	7.0668e-1 (5.59e-3) +	6.7264e-1 (2.74e-2) -	7.0659e-1 (3.77e-4) +	7.0755e-1 (1.24e-4) +	7.0630e-1 (4.01e-4)
LIRCMOP11	5.3737e-1 (1.32e-1) -	6.9062e-1 (4.88e-3) -	6.4145e-1 (1.47e-2) -	6.9392e-1 (7.61e-5) ≈	6.1686e-1 (4.29e-2) -	6.9393e-1 (5.71e-5)
LIRCMOP12	5.5161e-1 (4.68e-2) -	6.1582e-1 (9.12e-3) ≈	6.0522e-1 (7.22e-3) -	6.1952e-1 (1.29e-3) ≈	6.0839e-1 (2.46e-2) -	6.1811e-1（3.04e-3）
LIRCMOP13	4.7950e-1 (1.63e-1) -	5.3426e-1 (4.14e-3) -	5.4704e-1 (3.37e-4) -	5.5421e-1 (1.49e-3) -	5.1626e-1 (3.56e-3) -	5.5578e-1 (1.23e-3)
LIRCMOP14	4.9121e-1 (1.36e-1) -	5.3744e-1 (4.86e-3) -	5.4656e-1 (7.33e-4) -	5.5357e-1 (1.22e-3) -	5.2944e-1 (4.28e-3) -	5.5604e-1 (1.16e-3)
+/-/≈	0/33/3	3/31/2	3/28/5	7/20/9	3/32/1	

B. 比较结果

首先，我们比较了 ATM-R 与其他五种约束多目标进化算法的性能。表 I 汇总了 36 个测试函数在 30 次独立运行中的平均IGD值和标准差。HV值方面的结果列于表 II。每张表中的“std”表示 30 次独立运行所得 IGD/HV 值的标准差。“NaN”表示某种约束多目标进化算法在 30 次独立运行中均未能为某个测试函数找到可行解。对于给定的测试函数，我们采用显著性水平为 0.05 的弗里德曼检验和邦费罗尼校正，将 ATM-R 与每种竞争算法进行比较。为方便起见，我们用 “+” 、“-”和“≈”分别表示某种竞争算法优于、劣于或与 ATM-R 相当。此外，对于每个测试函数，六种约束多目标进化算法中结果最佳者以灰色突出显示。为直观展示结果，我们绘制了六种约束多目标进化算法在三个具有代表性的约束多目标优化问题上某次典型运行所得的收敛前沿，如图 6–8 所示。典型运行是指所有运行中产生中位数 IGD 值的那次运行。

1）总体性能：总体而言，如表 I 和表 II 所示，就 IGD 和 HV 值而言，ATM-R 在大多数测试函数上取得了最佳结果。此外，在大多数测试函数上，其性能显著优于另外五种竞争算法。我们采用多问题 Friedman 检验 [?]同时比较这六种约束多目标进化算法。如图 5 所示，ATM-R 在六种约束多目标进化算法中的排名值最低。此外，图 6–8 的结果表明，ATM-R 能够获得一组收敛性良好且分布均匀的解。下文将详细讨论不同的测试套件。

2）*MW*测试套件上的性能：就反世代距离（IGD）值而言，ATM-R在14、13、6、6和14个测试函数上的表现分别优于NSGAII-CDP、PPS、CTAEA、CCMO和ToP。反之，这些对比约束多目标进化算法分别仅在0、1、2、3和0个测试函数上的表现优于ATM-R。ATM-R在4个测试函数上取得了最佳结果，在这些函数上它的表现优于其他5种竞品。CCMO在4个测试函数上取得了最佳结果，其中1个函数上的表现与ATM-R相近。尽管CTAEA在6个测试函数上取得了最佳结果，但其中5个函数上的表现与ATM-R相近。

就超体积值而言，ATM-R 在 12、12、6、8 和 14 个测试函数上的表现分别优于 NSGAII-CDP、PPS、CTAEA、CCMO 和 ToP。相反，这些对比约束多目标进化算法在 0、2、3、5 和 0 个测试函数上的结果分别优于 ATM-R。ATM-R 在 3 个测试函数上取得了最佳结果，并且在这些函数上的表现优于其他 5 个对手。CCMO

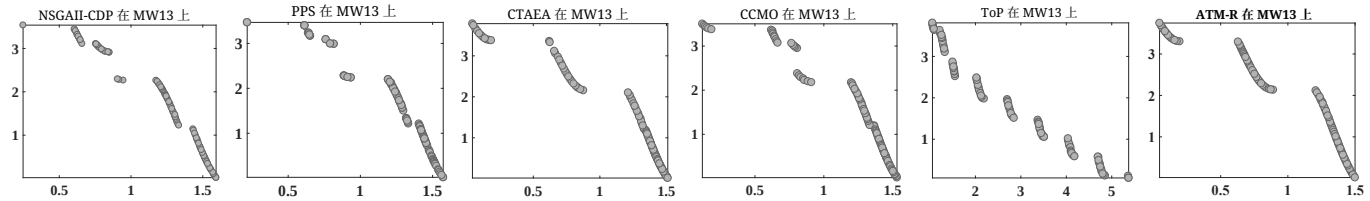

图6. 约束帕累托前沿　　NSGAII-CDP、PPS、CTAEA、CCMO、ToP和ATM-R在MW13上30次运行所得的中位数。

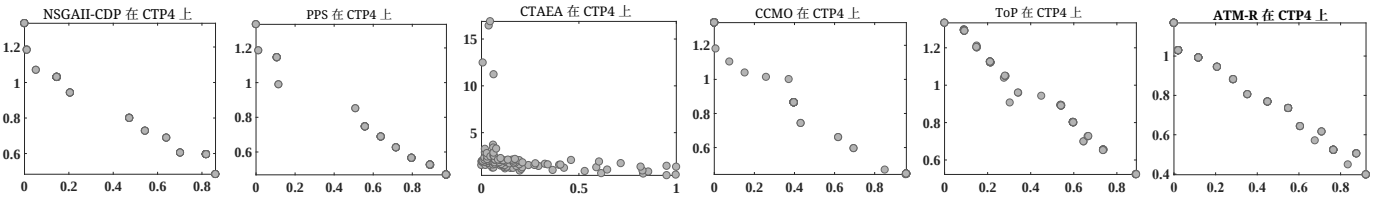

图7. NSGAII-CDP、PPS、CTAEA、CCMO、ToP和ATM-R在CTP4上30次运行所得的中位数对应的约束帕累托前沿。

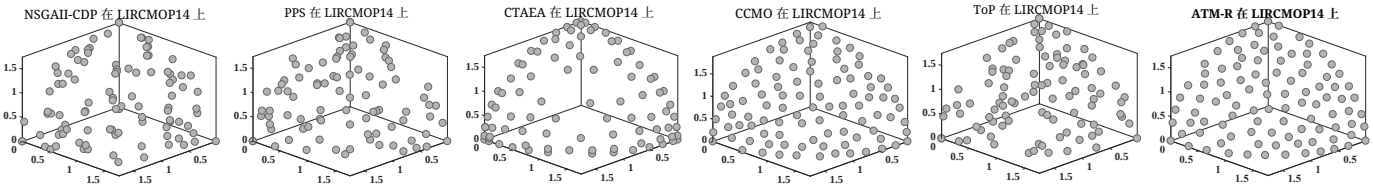

图8. 约束帕累托前沿及其　　30次运行所得的中位数，由NSGA-II　　-CDP、PPS、CTAEA、CCMO、ToP和ATM　　-LIRCMOP1上的R　4.

在 4 个测试函数上取得了最佳结果。尽管 CTAEA 在 7 个测试函数上取得了最佳结果，但在其中 4 个测试函数上的表现与 ATM-R 相近。

此外，如图6所示，ATM-R获得了一组收敛性良好、分布均匀且接近约束帕累托前沿的可行解。然而，ToP未能收敛到约束帕累托前沿。NSGAII-CDP、PPS、CTAEA和CCMO均遗漏了约束帕累托前沿的部分区域。结果表明，在MW测试套件上，ATM-R的性能优于其他五种竞争算法。

3）在*CTP*测试套件上的性能：对于CTP测试套件，在大多数测试函数上，ATM-R在反世代距离和超体积指标方面的表现均优于其他五种竞争算法。此外，在大多数测试函数上，它都取得了最佳的反世代距离/超体积指标值。

对于CTP1，约束帕累托前沿的部分区域来自无约束帕累托前沿。对于CTP6，目标空间中朝向帕累托最优区域分布着宽度各异的不可行孔洞。对于CTP2-CTP5、CTP7和CTP8，约束帕累托前沿被划分为若干互不连通的片段。要有效解决这些问题，必须谨慎利用不可行性信息。因此，仅在不可行阶段考虑约束的NSGAII-CDP和ToP，其表现均不如ATM-R。如 [4], 所述，由于约束帕累托前沿具有复杂特征（即互不连通且离散），使用参考点或向量的约束多目标进化算法性能会较差。因此，CTAEA的表现不如ATM-R。PPS在早期阶段侧重于目标，而CCMO则采用特定种群来利用不可行性信息。与MW测试套件相比，CTP测试套件中的测试函数具有更高的可行率。因此，过多的不可行性信息会损害约束多目标进化算法的性能。这或许就是PPS和CCMO的表现不如ATM-R的原因。

此外，如图7所示，与其他五种竞争算法相比，ATM-R能够更快地收敛到CTP4的约束帕累托前沿。此外，它能够覆盖约束帕累托前沿中比其他五种竞争算法更多的区域。结果表明，在CTP测试套件上，ATM-R的性能优于其他五种竞争算法。

4）*LIRCMOP*测试套件上的性能：对于LIRC-MOP测试套件，就反世代距离和超体积值而言，ATM-R在一半的测试函数上取得了最佳结果。与CTP测试套件类似，LIRCMOP测试套件中的测试函数在目标空间中存在不可行孔洞，并且一些测试函数的收敛前沿是不连通的。为有效解决这些测试函数，应谨慎使用不可行性信息。由于在很大程度上忽略了不可行性信息，NSGAII-CDP和ToP在大多数测试函数上的表现不如ATM-R。就反世代距离和超体积值而言，ToP分别在四个和三个测试函数上优于ATM-R。这归因于ToP所采用的强大遗传算子（即差分进化）。

在不可行性辅助型约束多目标进化算法中，PPS 在早期阶段会引导种群趋向无约束帕累托前沿。在 CTAEA 和 CCMO 中，则采用额外的种群来逼近无约束帕累托前沿。因此，这三种约束多目标进化算法无法解决约束帕累托前沿远离无约束帕累托前沿的约束多目标优化问题（即 LIRCMOP1-LIRCMOP4）。

对于 LIRCMOP5 和 LIRCMOP6，其收敛前沿与无约束帕累托前沿相同。对于 LIRCMOP7 和 LIRCMOP8，其收敛前沿接近无约束帕累托前沿。对于这四个测试函数，不可行性辅助型约束多目标进化算法能够轻松逼近约束帕累托前沿，因此均匀性是影响其性能的关键因素。在这些测试函数上，ATM-R 和 CCMO 的表现优于 PPS 和 CTAEA，因为它们能够更有效地保持多样性。对于 LIRCMOP9-LIRCMOP12，约束帕累托前沿被分成若干互不连通的片段。要有效求解这些测试函数，必须谨慎维持多样性。由于两个种群之间的协作较弱，CCMO 在进化过程中能够有效保持多样性。因此，在这四个测试函数上的表现优于 ATM-R。对于两个三目标测试函数（即 LIRCMOP13-LIRCMOP14），无论是反世代距离还是超体积值，ATM-R 的表现都优于其他五种竞争算法。这表明，对于三目标约束多目标优化问题，ATM-R 能够更好地权衡可行性、多样性和收敛性。

此外，如图8所示，所有六种约束多目标进化算法均能成功收敛到约束帕累托前沿。在均匀性方面，ATM-R 的表现优于其他五种算法，因为对于三目标约束多目标优化问题，它能够在可行性、多样性和收敛性之间实现更好的权衡。结果表明，ATM-R 在 LIRCMOP测试套件上的表现优于其他五种算法。

总之，在具有各种挑战性特征的36个测试函数上进行的大量实验表明，ATM-R 能够成功解决复杂的约束多目标优化问题。

五、进一步分析

A. ATM-R 相较于 *ATM* 的优势

如备注*1*所述，直接扩展 ATM [40] 以解决约束多目标优化问题并不可行。本小节通过实验展示 ATM-R 相较于 ATM 的优势。表III总结了两者在36个测试函数上的反世代距离和超体积比较结果，其中“+”、“-”和“≈”分别表示在反世代距离/超体积方面，ATM 的表现优于、逊于或与 ATM-R 相当。如表III所示，在这三个测试套件上，ATM-R 的反世代距离和超体积表现均优于 ATM。具体而言，在反世代距离方面，ATM-R 在 MW、CTP 和 LIRCMOP测试套件中分别有9、5和12个测试函数优于 ATM。相比之下，在这些测试套件中，ATM 优于 ATM-R 的测试函数不超过3个。在超体积方面，ATM-R 分别在8、6和12个测试函数上优于 ATM。反之，ATM 在这些测试套件中优于 ATM-R 的测试函数不超过4个。总之，在这些具有不同特征的测试函数上进行的实验结果表明，ATM-R 优于 ATM。

表III ATM 与 ATM-R 在 36 个测试函数上的结果。

测试函数	IGD +/-/≈	HV +/-/≈
MW1-MW14	2/9/3	4/8/2
CTP1-CTP8	3/5/0	2/6/0
LIRCMOP1-LIRCMOP14	1/12/1	1/12/1

B. 不可行阶段的有效性

为验证不可行阶段更新机制的有效性，我们采用该阶段的不同更新机制实现了三个变体（分别记为 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv）。具体而言，在 ATM-RICDP 中，使用约束支配原则进行解选择。在 ATM-RIobj 中，不考虑约束，而是依据帕累托支配选择解。在 ATM-RIdiv 中，对多样性进行量化，并据此选择有潜力的解。通过比较 ATM-R 与 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv，可以验证不可行阶段更新机制的有效性。上述 36 个测试函数的比较结果汇总于表 IV，其中“+”、“-”和“≈”分别表示竞争算法在 IGD/HV 值方面优于、劣于和表现与 ATM-R 相近。

表 IV ATM-RICDP 与 ATM-R、ATM-RIOBJ 与 ATM-R 以及 ATM-RIDIV 与 ATM-R 在 36 个测试函数上的结果。

算法	IGD +/-/≈	HV +/-/≈
ATM-RICDP 对比 ATM-R	2/19/15	2/23/11
ATM-RIobj 对比 ATM-R	5/11/20	7/11/18
ATM-RIdiv 与 ATM-R	8/14/14	10/14/12

如表 IV 所示，在 IGD 和 HV 值方面，ATM-R 的表现均优于 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv。具体而言，就 IGD 值而言，ATM-R 在 19、11 和 14 个测试函数上的表现分别优于 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv。相反，ATM-RICDP、ATM-RIobj 和 ATM-RIdiv 分别在 2、5 和 8 个测试函数上的表现优于 ATM-R。就 HV 值而言，ATM-R 在 23、11 和 14 个测试函数上的表现分别优于 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv。相比之下，ATM-RICDP、ATM-RIobj 和 ATM-RIdiv 分别在 2、7 和 4 个测试函数上的表现优于 ATM-R。实验结果表明，不可行阶段的更新机制对 ATM-R 至关重要。

C. 半可行阶段的有效性

为验证半可行阶段更新机制的有效性，我们在该阶段采用不同的更新机制实现了三个变体（即 ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv）。具体而言，在 ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv 中，分别使用 CDP 、帕累托支配和多样性进行解选择。通过将 ATM-R 与 ATM-RICDP、ATM-RIobj 和 ATM-RIdiv 分别进行比较，可以验证半可行阶段更新机制的有效性。具体而言，上文提到的 36 个测试函数上的比较结果汇总于表 V，其中

表 V ATM-RSCDP 与 ATM-R、ATM-RSOBJ 与 ATM-R，以及 ATM-RSDIV 与 ATM-R 在 36 个测试函数上的结果。

算法	IGD +/-/≈	HV +/-/≈
ATM-RSCDP 与 ATM-R	3/24/9	5/23/8
ATM-RSobj 与 ATM-R	3/32/1	2/31/3
ATM-RSdiv 与 ATM-R	0/36/0	0/36/0

表 VI ATM-RMCDP 与 ATM-R、ATM-RMOBJ 与 ATM-R 以及 ATM-RMDIV 与 ATM-R 在 36 个测试函数上的结果。

算法	IGD +/-/≈	HV +/-/≈
ATM-RMCDP 与 ATM-R	5/19/12	2011年3月22日
ATM-RMobj 与 ATM-R 对比	2019年7月10日	2016年7月13日
ATM-RMdiv 与 ATM-R 对比	2022年	2024年
	5月9日	4月8日

“+”、“-”和“≈”分别表示竞争算法在 IGD/HV 值方面优于、劣于和表现与 ATM-R 相近。

如表 V 所示，在 IGD 和 HV 值方面，ATM-R 的表现均优于 ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv。就 IGD 值而言，ATM-R 在 24、32 和 36 个测试函数上的表现分别优于 ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv。相比之下，ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv 各自优于 ATM-R 的测试函数均不超过 3 个。就 HV 值而言，ATM-R 在 23、31 和 36 个测试函数上的表现分别优于 ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv。反之，ATM-RSCDP、ATM-RSobj 和 ATM-RSdiv 各自优于 ATM-R 的测试函数均不超过 5 个。实验结果表明，半可行阶段的更新机制对 ATM-R 至关重要。

D. 多阶段交配选择策略的有效性

为验证多阶段交配选择策略的有效性，我们采用不同的选择方法来选择交配解，并实现了三种变体（即 ATM-RMCDP、ATM-RMobj 和 ATM-RMdiv）。具体而言，在 ATM-RMCDP、ATM-RMobj 和 ATM-RMdiv 中，分别使用 CDP、帕累托支配和多样性进行解选择。通过将 ATM-R 与 ATM-RMCDP 、 ATM-RMobj 和 ATM-RMdiv 分别进行比较，可以展示多阶段交配选择策略的有效性。具体来说，上文提到的 36 个测试函数的比较结果汇总于表 VI，其中“+”、“-”和“≈”分别表示竞争算法在 IGD/HV 值方面优于、劣于或与 ATM-R 表现相当。

如表 VI 所示，在 IGD 和 HV 值方面，ATM-R 的表现均优于 ATM-RMCDP、ATM-RMOBJ 和 ATM-RMDIV。就 IGD 值而言，ATM-R 在 19、10 和 9 个测试函数上的表现分别优于 ATM-RMCDP、ATM-RMOBJ 和 ATM-RMDIV。相反，ATM-RMCDP、ATM-RMOBJ 和 ATM-RMDIV 分别在 5、7 和 5 个测试函数上的表现优于 ATM-R。就 HV 值而言，ATM-R 在 22、13 和 8 个测试函数上的表现分别优于 ATM-RMCDP、ATM-RMOBJ 和 ATM-RMDIV。反之，ATM-RMCDP、ATM-RMOBJ 和 ATM-RMDIV 分别在 3、7 和 4 个测试函数上的表现优于 ATM-R。实验结果表明，多阶段选择策略对 ATM-R 至关重要。

VI. 结论

本文深入分析了约束多目标优化中的关键任务，并首次将其明确分解为三个子任务。为在不同进化阶段完成这三个子任务，本文设计了一种基于参考点的自适应权衡模型（ATM-R）。具体而言，ATM-R 利用不可行解，在这三个子任务中实现不同的权衡。在不可行阶段，ATM-R 区分并利用多样性良好的不可行解，以弥补其追求可行性所导致的多样性损失。因此，种群能够从多样化的搜索方向向可行区域移动。在半可行阶段，ATM-R 利用多样性或目标函数值良好的不可行解，推动权衡从“可行性与多样性之间的权衡”转变为“收敛性与多样性之间的权衡”。因此，种群能够找到足够多的可行解，并快速逼近约束帕累托前沿。在可行阶段，ATM-R 采用 NSGA-II，以寻找一组收敛性良好、分布均匀且接近约束帕累托前沿的解。此外，本文还提出了一种多阶段交配选择策略，用于自适应地选择合适的交配亲本。在大量约束多目标优化问题上的实验研究表明：

- ATM-R 的性能优于或至少具有与其他代表性约束多目标进化算法高度竞争的表现。
- 对于约束多目标优化，ATM-R 相比 ATM 具有显著优势。
- 不可行阶段和半可行阶段中的更新机制对 ATM-R 的性能都至关重要。
- 多阶段交配选择策略对 ATM-R 的性能至关重要。

未来，我们将扩展 ATM-R，以求解受约束的昂贵多目标优化问题。

参考文献

[1] K. Miettinen，《非线性多目标优化》。Springer Science & Business Media，2012 年，第 12 卷。[2] B.-C. Wang、H.-X. Li、J.-P. Li 和 Y. Wang，“用于约束进化优化的复合差分进化”，《IEEE Transactions on Systems, Man, and Cybernetics: Systems》，第 49 卷，第 7 期，第 1482–1495 页，2018 年。[3] Z. Ma 和 Y. Wang，“用于约束多目标进化优化的基于偏移的罚函数及其应用”，《IEEE〈控制论汇刊〉》，2021 年。[4] Z.-Z. Liu、B.-C. Wang 和 K. Tang，“通过双向协同进化处理约束多目标优化问题”，《IEEE〈控制论汇刊〉》，2021 年。[5] K. Deb、A. Pratap、S. Agarwal 和 T. Meyarivan，“快速精英多目标遗传算法：NSGA-II”，《IEEE Transactions on Evo-lutionary Computation》，第 6 卷，第 2 期，第 182–197 页，2002 年。

[6] Z.-Z. Liu、Y. Wang 和 P.-Q. Huang，“And算法：一种采用基于角度的选择和基于平移的密度估计的多目标进化算法”，Information Sciences，第 509 卷，第 400–419 页，2020 年。[7] H. Jain 和 K. Deb，“一种采用基于参考点的非支配排序方法的进化多目标优化算法，第二部分：约束处理与自适应方法扩展”，IEEE Transactions on Evolutionary Computation，第 18 卷，第 4 期，第 602–622 页，2013 年。[8] W. Ning、B. Guo、Y. Yan、X. Wu、J. Wu 和 D. Zhao，“将约束非支配排序与改进的混合多目标进化算法相结合的约束多目标优化”，Engineering Optimization，第 49 卷，第 10 期，第 1645–1664 页，2017 年。[9] J. G. Hobbie、A. H. Gandomi 和 I. Rahimi，“NSGA-II 中约束处理技术的比较”，Archives of Computational Methods in Engineering，第 1–16 页，2021 年。[10] M. A. Jan 和 R. A. Khanum，“对 MOEA/D 框架中两种无惩罚参数约束处理技术的研究”，Applied Soft Computing，第 13 卷，第 1 期，第 128–148 页，2013 年。[11] T. P. Runarsson 和 X. Yao，“约束进化优化的随机排序”，IEEE Transactions on Evolutionary Computation，第 4 卷，第 3 期，第 284–294 页，2000 年。[12] Z. Fan、W. Li、X. Cai、H. Li、C. Wei、Q. Zhang、K. Deb 和 E. Goodman，“用于求解约束多目标优化问题的推拉搜索”，Swarm and Evolutionary Computation，第 44 卷，第 665–679 页，2019 年。[13] Z. Ma、Y. Wang 和 W. Song，“一种用于约束多目标进化优化、包含两种排序的新适应度函数”，IEEE Transactions on Systems, Man, and Cybernetics: Systems，2019 年。[14] C. Peng、H.-L. Liu 和 F. Gu，“一种采用有向权重的约束多目标优化进化算法”，Applied Soft Computing，第 60 卷，第 613–622 页，2017 年。[15] Z. Liu、Y. Wang 和 B.-C. Wang，“基于指标的约束多目标进化算法”，IEEE Transactions on Systems, Man, and Cybernetics: Systems，第 1–13 页，2019 年。[16] Z. Fan、Y. Fang、W. Li、J. Lu、X. Cai 和 C. Wei，“约束多目标进化算法在约束多目标优化问题上的比较研究”，载于 2017 IEEE Congress on Evolutionary Computation (CEC)。IEEE，2017 年，第 209–216 页。[17] Y. G. Woldesenbet、G. G. Yen 和 B. G. Tessema，“多目标进化优化中的约束处理”，IEEE Transactions on Evolutionary Computation，第 13 卷，第 3 期，第 514–525 页，2009 年。[18] Z.-Z. Liu 和 Y. Wang，“处理决策空间和目标空间均含约束的约束多目标优化问题”，IEEE Transactions on Evolutionary Computation，第 23 卷，第 5 期，第 870–884 页，2019 年。[19] F. Jimenez、A. F. G´omez-Skarmeta、G. S´anchez 和 K. Deb，“一种用于约束多目标优化的进化算法”，载于《2002 Congress on Evolutionary Computation 论文集》。CEC'02（目录号 02TH8600），第 2 卷。IEEE，2002 年，第 1133–1138 页。[20] M. Miyakawa、K. Takadama 和 H. Sato，“用于求解多目标与约束问题的两阶段非支配排序和定向交配”，载于《第 15 届遗传与进化计算年度会议论文集》，2013 年，第 647–654 页。[21] L. Jiao、J. Luo、R. Shang 和 F. Liu，“一种采用可行性引导策略求解约束多目标优化问题的改进目标函数方法”，Applied Soft Computing，第 14 卷，第 363–380 页，2014 年。[22] N. Young，“通过混合排序跨越约束多目标问题中的不可行区域”，载于《计算智能建模、控制与自动化国际会议暨智能代理、网络技术与互联网商务国际会议（CIMCA-IAWTIC'06）》，第 2 卷。IEEE，2005 年，第 191–196 页。[23] Z. Yang、X. Cai 和 Z. Fan，“约束多目标优化问题的 ε 约束法：一些初步结果”，载于《2014 年遗传与进化计算会议伴随出版物论文集》，2014 年，第 1181–1186 页。[24] S. Zapotecas-Mart´ınez 和 A. Ponsich，“通过附加标量化函数在 MOEA/D 中进行约束处理”，载于《2020 年遗传与进化计算会议论文集》，2020 年，第 595–602 页。[25] Z. Fan、H. Li、C. Wei、W. Li、H. Huang、X. Cai 和 Z. Cai，“嵌入 MOEA/D 的改进 ε 约束处理方法，用于约束多目标优化问题”，载于 2016 IEEE Symposium Series on Computational Intelligence (SSCI)。IEEE，2016 年，第 1–8 页。[26] J. Zhou、J. Zou、J. Zheng、S. Yang、D. Gong 和 T. Pei，“一种用于约束多目标进化优化的不可行解多样性维护 ε 约束处理方法”，Soft Computing，第 1–12 页，2021。[27] H. Geng、M. Zhang、L. Huang 和 X. Wang，“约束进化多目标优化中的不可行精英个体与随机排序选择”，载于《亚太模拟进化与学习会议》。施普林格，2006，第 336–344 页。[28] T. Ray、H. K. Singh、A. Isaacs 和 W. Smith，“用于约束优化的不可行性驱动进化算法”，载于《进化优化中的约束处理》。施普林格，2009，第 145–165 页。[29] K. Li、R. Chen、G. Fu 和 X. Yao，“用于约束多目标优化的双档案进化算法”，IEEE Transactions on Evolutionary Computation，第 23 卷，第 2 期，第 303–315 页，2018。[30] Y. Tian、T. Zhang、J. Xiao、X. Zhang 和 Y. Jin，“用于约束多目标优化问题的协同进化框架”，IEEE Transactions on Evolutionary Computation，2020。[31] H. Ishibuchi、T. Fukase、N. Masuyama 和 Y. Nojima，“用于约束多目标进化优化的 MOEA/D 双网格模型”，载于《遗传与进化计算会议论文集》，2018，第 665–672 页。[32] Q. Zhu、Q. Zhang、Q. Lin 和 J. Sun，“采用两种类型权重向量处理约束的 MOEA/D”，载于《2019年IEEE进化计算大会》（CEC）。IEEE，2019，第 1359–1365 页。[33] Y. Yang、J. Liu 和 S. Tan，“一种基于分解和动态约束处理机制的约束多目标进化算法”，Applied Soft Computing，第 89 卷，文章编号 106104，2020。[34] Z. Fan、Z. Wang、W. Li、Y. Yuan、Y. You、Z. Yang、F. Sun 和 J. Ruan，“嵌入 M2M 框架的推拉搜索，用于求解约束多目标优化问题 ”，Swarm and Evolutionary Computation，第 54 卷，文章编号 100651，2020。[35] G. Peng、H. Wu、H. Wu 和 K. Wolter，“协同边缘与云计算中物联网赋能的计算卸载约束多目标优化”，IEEE物联网期刊，2021。[36] K. Yu、J. Liang、B. Qu、Y. Luo 和 C. Yue，“动态选择偏好辅助的约束多目标差分进化”，IEEE Transactions on Systems, Man, and Cybernetics: Systems，2021。[37] Y. Tian、Y. Zhang、Y. Su、X. Zhang、K. C. Tan 和 Y. Jin，“在约束进化多目标优化中平衡目标优化与约束满足”，IEEE《控制论汇刊》，2020。[38] F. Ming、W. Gong、H. Zhen、S. Li、L. Wang 和 Z. Liao，“一种用于约束多目标优化的简单两阶段进化算法”，Knowledge-Based Systems，第 1–14 页，2021。[39] C. Peng、H.-L. Liu 和 E. D. Goodman，“一种基于改进定向权重向量的协同进化框架，用于具有欺骗性约束的约束多目标优化”，IEEE《控制论汇刊》，2020。[40] Y. Wang、Z. Cai、Y. Zhou 和 W. Zeng，“用于约束进化优化的自适应权衡模型”，IEEE Transactions on Evolutionary Computation，第 12 卷，第 1 期，第 80–92 页，2008。[41] I. Das 和 J. E. Dennis，“法向边界交点法：一种在非线性多准则优化问题中生成帕累托曲面的新方法”，SIAM优化期刊，第 8 卷，657，1998 3，631第期，页码 – 。[42] K. Deb、R. B. Agrawal 等，“适用于连续搜索空间的模拟二进制交叉”，Complex Systems，第 9 卷，第 2 期，第 115–148 页，1995。[43] K. Deb、M. Goyal 等，“用于工程设计的组合遗传自适应搜索（GeneAS）”，Computer Science and Informatics，第 26 卷，第 30–45 页，1996。[44] Y. Tian、R. Cheng、X. Zhang 和 Y. Jin，“PlatEMO：用于进化多目标优化的 MATLAB 平台 [教育论坛],”，IEEE计算智能杂志，第 12 卷，第 4 期，第 73–87 页，2017。[45] Z. Ma 和 Y. Wang，“约束多目标进化优化：测试套件构建与性能比较”，IEEE Transactions on Evolutionary Computation，第 23 卷，986，2019 6，第 972期，页码 – 。[46] K. Deb、A. Pratap 和 T. Meyarivan，“用于多目标进化优化的约束测试问题”，载于《进化多准则优化国际会议》。施普林格，2001，第 284–298 页。[47] M. Li 和 X. Yao，“多目标优化中解集质量评价：综述”，ACM计算调查（CSUR），第 52 卷，第 2 期，第 1–38 页，2019。