arXiv:2601.00770v1 [cs.CE] 2 Jan 2026

用于组合优化有效前沿的大语言模型智能体：投资组合优化

SimonPaquette-Greenbaum **JiangboYu***加
拿大魁北克省蒙特利尔市麦吉尔大学土木工程系

摘要

投资组合优化是各大金融机构都会开展的一项工作。基数约束均值-方差投资组合优化（CCPO）问题表述在投资组合优化中应用广泛。这类投资组合优化问题属于混合整数二次规划（MIQP），其难点在于精确求解器难以求得可行解，因此会采用启发式算法寻找近似的投资组合解。CCPO 涉及许多繁琐复杂的工作流程，也需要投入大量精力开发启发式算法；将多种启发式解汇集起来，则能改善有效前沿。因此，常见做法是开发多种启发式算法。智能体框架在组合优化的诸多问题中展现出广阔前景，因为事实证明，它们在自动化大型工作流程方面同样高效，在算法开发方面也表现出色，有时甚至超越人类水平。本研究为 CCPO 实现了一种新颖的智能体框架，并探索了若干具体架构。在基准问题上，该智能体框架的表现可与最先进算法媲美。此外，该框架减轻了复杂工作流程和算法开发所需的投入；在最差情况下，误差虽较大，但仍处于可接受范围。

1 引言

1.1 背景

智能体大语言模型（LLM）正逐渐成为自动化众多领域大型工作流和决策支持系统的关键要素，例如物流 [Xu et al., 2021], 管理[Chen et al., 2025a], 医疗保健 [Liu et al., 2025], 城市规划 [Yu and Hyland, 2025], 以及交通运输 [Yu, 2025]。因此，大语言模型智能体也被用作相关组合优化问题（如调度）的算法开发工具 [Romera-Paredes et al., 2024]。在用于优化的自然语言处理领域，这类智能体已得到广泛研究[Ramamonjison et al., 2022], , 研究表明，自然语言描述的问题能够有效转化为有价值的数学建模。语言模型智能体同样擅长在给出自然语言描述和数学建模后，为优化问题生成算法解 [Xiao et al., 2024], ；研究还表明，在时间受限的场景中，其表现甚至超过人类专家 [Sun et al., 2025]。

研究人员和业界都对这些框架表现出浓厚兴趣，因为它们为大语言模型的一些重要应用提供了宝贵的基准测试。这些应用展现了大语言模型性能的两个关键方面：自然语言处理（NLP）和编程 [Xi et al.,2025]。然而，语言模型智能体在组合优化问题中的应用

*通讯作者：jiangbo.yu@mcgill.ca

预印本。

一直局限于学术案例，其中的问题构建和描述直接取自或借鉴了人类教育所用的教科书 [Mostajabdaveh et al., 2025]。尽管这些研究为大语言模型智能体提供了结构化基准，但涉及的问题仅有单一目标，且可通过精确求解方法求解。

针对反映现实情况的组合优化，相关研究仍然有限。反映现实情况的问题很少会被写入教科书，也往往无法精确求解，原因可能是计算资源限制，也可能是问题不确定性。此外，现实问题很少不存在权衡 [Deb, 2014], , 而现实中的决策者往往需要了解这些权衡所带来的影响，即多目标优化任务产生的帕累托前沿。已有研究将大语言模型智能体应用于开发启发式算法，以求解精确求解器无法处理的问题 [˙Ibrahim O˘guz Çetinkaya et al., 2026]。然而，针对多目标优化问题的大语言模型智能体解法，相关研究仍然有限。

1.2 理由

本研究旨在开发一种语言模型框架，使其能够生成组合优化问题的算法解，并反映现实最坏情形（与若干基准研究相反，后者将问题设定为最佳情形）。组合优化问题通常因其NP难性以及子集的组合爆炸而难以求解 [Hoffman, 2000],, 因此，探索所有子集以获得精确解也难以实现。此外，现实问题往往涉及多个目标，决策者需要应对相互冲突的业务需求等。

能够开发启发式算法的智能体框架，在此类情形下是一种很有价值的工具。对于精确求解器难以处理的问题，通常需要借助元启发式算法求得近似解 [Peres and Castelli, 2021]。此外，研究表明，多目标优化问题有时能从集成启发式解中获益良多；集成后的启发式解可形成收敛性和覆盖度更高的非支配前沿 [Calderín et al., 2015]。例如，基数约束均值-方差投资组合优化（CCPO） [Chang et al., 2000] 就非常符合这一描述。与标准的马科维茨均值-方差投资组合优化 [Markowitz, 1952], 不同，后者可以通过精确动态规划轻松求解，而CCPO是一个NP难问题 [Moral-Escudero et al., 2006], 其有效前沿非凸且不连续。文献已对CCPO的元启发式解进行了广泛研究 [Kalayci et al., 2019], , 并且研究表明，在这种情况下，集成启发式算法的表现远优于单一启发式算法 [Woodside-Oriakhi et al., 2011]。

1.3 研究贡献

基于上述思路，本研究得出以下几项发现：

- 本研究提出了一种智能体语言模型框架，用于构建求解 NP 难多目标组合优化问题的算法组合。该代理框架不仅大幅减轻了构建算法组合所需的繁重开发工作，还有望发现新型算法。

- 本研究在一系列文献中广泛研究的复杂多目标投资组合优化基准问题上，验证了该智能体框架及其生成的算法组合。通过 OR-Library 中 Beasley 的 CCPO 实例 [1990], 进行评估后发现，该代理框架生成的算法达到当前最佳水平。此外，将其生成的算法组合中的算法汇集起来，能够显著提升解的性能。

2 预备知识

2.1 大语言模型智能体

编程智能体在计算机科学和机器学习领域受到了广泛关注。研究表明，它们在编程能力方面极具潜力

，并能简化复杂的工作流程。由于求解难以处理或 NP-hard问题需要投入大量精力进行算法设计，编程智能体在组合优化领域也受到了广泛关注 [Chen et al., 2025b]。目前已有多个编码代理框架成功应用于组合优化问题的求解，包括 Self-Refine[Madaan et al., 2023], FunSearch [Romera-Paredes et al., 2024], 和 ReEvo [Ye et al., 2024]。

2.1.1 代理框架

本研究采用的编码代理框架是其算法组合开发的核心要素。鉴于迭代代理在算法生成和发现中的成功应用，本研究将以其架构为基础构建框架。研究表明，使用大语言模型进行一次性生成通常会导致算法性能不佳，或出现运行时错误/执行错误 [Madaan 等， 2023]。因此，本研究所采用的此类代理框架依赖于与外部环境进行迭代式细化，这一过程正式称为推理-动作迭代 [Yao 等， 2023]。

本研究将该语言模型（下文称为 $\mathcal{M}$）设定为编码智能体。方程 1 描述了通过 $\mathcal{A}$编码智能体$\mathcal{M}$生成算法的一次生成实例概览，其中$\mathbf{p}$ 是经过设计的提示模板向量，而 a、 f 和 s 分别是前一次迭代中的算法、反馈和分数。

$$\mathcal{M}(\mathbf{p}, a, f, s) \mapsto \mathcal{A} \tag{1}$$

本研究的代理框架基于 [Sun 等， 2025], 所描述的贪心式细化代理框架，而该框架本身又基于 Self-Refine [Madaan 等， 2023], ，通过迭代式地使用贪心改进来生成算法。方程 2 描述了本研究代理框架中的推理-动作迭代。在此，推理需要经过设计的提示模板向量$\mathbf{p} = \{p, p_{\text{PF}}, p_{\text{RA}}, p_{\text{I/O}}\}$，其中包括用于一般迭代指令的 p、用于问题表述的 p_{PF}、用于角色分配的 p_{RA} 以及用于格式化指令的 $p_{\text{I/O}}$。此外， a_t^*、 f_t^* 和 s_t^*分别是在某次迭代 t中，以 ASCII 格式表示的得分最高的前一次迭代的算法、反馈和分数，并在适当位置注入提示模板中。动作则需要问题参数和输入 x （在各次迭代中保持不变），以及生成的算法 $\mathcal{A}$。

$$\mathcal{A}_t \sim \mathcal{M}(\mathbf{p}, a_{t-1}^*, f_{t-1}^*, s_{t-1}^*) \quad \& \quad \{f_t, s_t\} \sim \mathcal{A}_t(x) \tag{2}$$

2.1.2 评分

在本研究中，算法解由语言模型智能体 $\mathcal{M}$在外部进行评分，从而消除自我评估产生幻觉和偏见的可能性。生成的算法 $\mathcal{A}$ 负责为多目标组合优化问题生成元启发式解。因此，推理-动作迭代中的动作部分需要实现超越简单目标适应度得分的指标。在这里，研究多目标优化问题的决策者依赖于有关解的有效前沿的信息，即最接近最优解的非支配解集合。例如，在投资组合优化中，理性的决策者不会选择风险更高、收益却相同的资产组合。反过来，他们也不会选择风险相同、收益却更低的投资组合。本研究将有效前沿，或更正式地说，帕累托前沿，定义为从所有可行解集合 Y中取出的严格非支配解子集 $H \subseteq Y$。方程 3 定义了点 $y' \in Y$在目标维度 n上严格支配另一点 $y \in Y$的方式，方程 4 则定义了严格非支配解子集 H。

$$y' \prec y \iff y'_i < y_i \quad \forall i \in \{1, \ldots, m\} \tag{3}$$

$$H = \{y \in Y \mid \forall y' \in Y,\ y' \not\prec y\} \tag{4}$$

有多种性能指标可用于量化集合 H [Kalayci 等人， 2019]中所包含的生成解的性能。收敛性和覆盖度/多样性是衡量有效前沿性能的主要指标。收敛性表示近似前沿与理论帕累托最优前沿之间的接近程度。覆盖度表示沿有效前沿获得的近似解的分布均匀程度。此外，还可以使用特定领域的指标来衡量帕累托前沿的性能。例如，百分比偏差误差（PE）常用于基准测试 CCPO 的收敛性能

[Chang等人， 2000]。然而，本研究依赖于同时衡量收敛性和覆盖度的混合指标。超体积指标 [Guerreiro等人， 2022]以及反向生成距离（IGD） [Ishibuchi等人， 2015] 通常用于此目的。本研究采用 IGD 作为标准，因为标准的马科维茨投资组合优化会提供无约束有效前沿（UEF），以供评分参考。公式 5 描述了 IGD 指标的计算方法，其中 P（本研究中取 UEF ）表示理论最优帕累托解集合 y^*， H 表示非支配近似帕累托解集合 y，而 $\|y^* - y\|$ 表示近似解 y 与最近的最优解 y^* 之间的欧几里得距离。

$$\text{IGD}(P, H) = \frac{1}{\text{card}(P)} \sum_{y^* \in P} \min_{y \in H} \|y^* - y\| \tag{5}$$

2.2 问题表述

本研究采用 Chang等人 [所述的 CCPO 问题表述 2000] ，因为该问题已在文献中得到广泛研究，可将最先进算法的性能作为参照。CCPO 是标准马科维茨均值-方差投资组合优化的有约束版本。与标准马科维茨方法一样，本研究假设方差足以衡量投资组合相关的风险。

2.2.1 目标

CCPO 可表述为一个多目标优化问题，其目标是在由一组 N 资产构成的潜在投资组合中，最小化投资组合风险并最大化投资组合收益（等价于最小化收益的负值）。与标准马科维茨方法一样，方程 6 将风险定义为投资组合收益的方差，其中 w_i 表示资产 i （$\forall i \in \{1, \ldots, N\}$）的持有比例，$\sigma_{ij}$表示资产 i与 j （$\forall i \in \{1, \ldots, N\}$和 $\forall j \in \{1, \ldots, N\}$）之间的协方差。方程 7 描述的收益，是资产 i （$\forall i \in \{1, \ldots, N\}$）预期收益 μ_i的加权和。

$$\min \quad \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{6}$$

$$\min \quad -\sum_{i=1}^{N} w_i \mu_i \tag{7}$$

鉴于上述 CCPO问题 的计算复杂度，相较于多目标表述，单目标表述更为可取，因为它能够使用单目标元启发式算法；而多目标元启发式算法会给本已颇具挑战的问题带来巨大的计算负担。因此，我们采用相关技术将多目标转换为单一目标。本文出于同样的原因——即上述 CCPO问题 的计算复杂度——选择采用 Bazgan 等人 [提出的加权和（WS）法 2022] 进行目标线性标量化，而非采用 Mavrotas [提出的 ε-约束目标表述 2009],。与ε-约束目标表述不同，采用 WS 目标表述无法覆盖有效前沿的非凸区域；由于其线性特性，它只能找到目标集合凸包上的解点。然而，关注 CCPO 有效前沿的决策者通常并不在意前沿的精确形状，而是满足于其提供的权衡信息。为提高计算效率，本文的目标函数采用 WS 法，不考虑非凸区域。若优化问题要求完整的前沿多样性和覆盖度，则应改用 ε-约束目标表述。

在此实例中，风险与收益目标之间没有已知的先验偏好。这里关注的是投资组合解的有效前沿。要找到有效前沿，必须遍历 WS 方法的线性权重；因为若不引导元启发式算法针对特定目标权衡比率进行搜索，就假设仅凭其探索能力便能覆盖有效前沿是不合理的。对于一般的 n 个目标，可以通过探索由 [Boyd 和 Vandenberghe, 2004] 个权重构成的单位单纯形来找到权重向量 $\boldsymbol{\lambda}$。探索

trad权衡比率只需遍历方程中定义的权重单纯形 Λ^{n-1} 8.

$$\Lambda^{n-1} = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n \;\middle|\; \lambda_i \geq 0,\ \sum_{i=1}^{n} \lambda_i = 1 \right\} \tag{8}$$

使用方程 6 和方程 7 中定义的风险和收益目标（$n = 2$），可以形式化地定义单一目标，其中 $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2\}$。此外，鉴于单位单纯形的线性等式约束，目标可以进一步简化，此时权重向量变为 $\boldsymbol{\lambda} = \{\lambda, 1-\lambda\}$，如方程 9 所述。在风险与收益之间探索权衡比率，只需在 $\lambda \in [0,1]$ 之间进行遍历，其中 $\lambda = 0$ 表示不考虑风险而使收益最大化的情形。相反，$\lambda = 1$ 表示不考虑收益而使风险最小化的情形。

$$\min \quad \lambda \left[\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \right] - (1-\lambda) \left[\sum_{i=1}^{N} w_i \mu_i \right] \tag{9}$$

2.2.2 约束

该形式包含投资组合中所含资产数量的基数约束，其中选定资产的数量固定为 K，如方程 11 所述。因此，为资产引入了零一决策 z 变量，如方程 13 所述。如果持有资产，则对其持有比例的最小值（最低买入门槛）和最大值设置边界约束，其中 ε 和 δ 分别表示最小和最大持有比例，如方程 12 所述。与标准的马科维茨投资组合优化一样，还会按照方程 10 的描述施加预算约束。此外，为了受益于大量依赖 CCPO 基准问题的研究，不考虑交易成本和整手约束。

$$\sum_{i=1}^{N} w_i = 1 \tag{10}$$

$$\sum_{i=1}^{N} z_i = K \tag{11}$$

$$\varepsilon_i z_i \;\leq\; w_i \;\leq\; \delta_i z_i, \quad \forall i \in \{1, \ldots, N\} \tag{12}$$

$$z_i \in \{0,1\}, \quad \forall i \in \{1, \ldots, N\} \tag{13}$$

3 方法

3.1 方法

在本研究中，大语言模型被算法化地构建为编码智能体。本研究设计的多目标组合优化智能体（MOCO–AGENT）算法，在生成 Python 元启发式算法时采用推理-动作迭代 [Yao et al., 2023] 。图 1 展示了所采用的 MOCO–AGENT 算法的一般形式，该形式经过抽象，可适用于任意多目标优化问题。该算法利用外部函数和模板生成提示，作为内部语言模型的输入。本研究定义的 MOCO–AGENT 旨在推理-动作迭代中采用贪心策略：每次迭代（初始化后）都会将生成的 Python 算法中得分最高者的表现（以反馈 f_t^*和分数 s_t^*表示）注入模型提示输入$\mathcal{A}_t^*$。在此例中，选择在推理-动作迭代中采用贪心方法，因为已有研究表明，其简单性和较低的令牌用量，使其在用于组合优化的算法生成任务时，优于更复杂的智能体框架 [Sunetal., 2025] 。

在初始迭代 $t = 0$ 中生成初始算法 $\mathcal{A}_0$，此时模型提示中不包含先前算法 a、反馈 f 和评分 s 信息。不过，针对 CCPO 问题表述 p_{PF}、角色分配 p_{RA} 以及 Python算法格式化指令$p_{\text{I/O}}$ 的精心设计的提示仍会被保留。CCPO 目标和约束以 ASCII 格式写入 p_{PF}。MOCO–AGENT 作为编码智能体的角色定义于 p_{RA}，其中包括关于待开发元启发式算法类型的指令，以及关于算法

混合的指令。格式化指令定义于 $p_{\text{I/O}}$，其中对 Python 代码生成施加了严格约束，确保生成的代码符合外部 MOCO–AGENT算法的功能要求，例如调用用于加载资产全集的库。

算法 1 MOCO–AGENT

需要： 输入 x，模型 $\mathcal{M}$，提示 $\mathbf{p}$，最优前沿 P，迭代次数 T，维度 n，步长 $\Delta\lambda$

$\mathcal{A}_0 = \mathcal{M}(\mathbf{p}, \cdot)$ ▷ 实例化算法（方程 1）
对于所有由 $\Delta\lambda\}$ 离散化的目标向量 $\boldsymbol{\lambda} \in \{\Lambda^{n-1}$ **执行** ▷ 单纯形（方程 8）
  $Y_{\boldsymbol{\lambda}} = \mathcal{A}_0(x, \boldsymbol{\lambda})$
结束 for
$Y_0 =$聚合$(Y_{\boldsymbol{\lambda}\in\Lambda})$
如果可行(Y_0)**那么** ▷ 可行性（例如，方程 10-13）
  $H_0 \subseteq Y_0$ ▷非支配集（公式 3-4）
  $\{f_0, s_0\} =$ IGD(P, H_0) ▷ IGD指标（公式 5）
否则
  $\{f_0, s_0\} = \infty$
结束 if
$\{a_0^*, f_0^*, s_0^*\} =$ 文本化$(\mathcal{A}_0, f_0, s_0)$ ▷初始"最佳"迭代
进行次迭代 $t \in 1, ..., T$ **do**
  $\mathcal{A}_t = \mathcal{M}(\mathbf{p}, a_{t-1}^*, f_{t-1}^*, s_{t-1}^*)$ ▷ 细化算法（方程 2）
  对于所有目标向量 $\boldsymbol{\lambda} \in \{\Lambda^{n-1}$ （由 $\Delta\lambda\}$ 离散化）**执行** ▷ 单纯形（方程 8）
    $Y_{\boldsymbol{\lambda}} = \mathcal{A}_t(x, \boldsymbol{\lambda})$
  结束循环
  $Y_t =$聚合$(Y_{\boldsymbol{\lambda}\in\Lambda})$
  如果 可行(Y_t)**则** ▷ 可行性（例如，方程 10-13）
    $H_t \subseteq Y_t$ ▷非支配集（公式 3-4）
    $\{f_t, s_t\} =$ IGD(P, H_t) ▷ IGD指标（公式 5）
  else
    $\{f_t, s_t\} = \infty$
  结束 if
  如果 $s_t \leq s_{t-1}^*$ **则** ▷ 当前最佳迭代
    $\{a_t^*, f_t^*, s_t^*\} =$ 文本化$(\mathcal{A}_t, f_t, s_t)$
  else
    $\{a_t^*, f_t^*, s_t^*\} =$ 文本化$(a_{t-1}^*, f_{t-1}^*, s_{t-1}^*)$
  结束 if
结束 for
返回 $a^*{}_T$

图 1：代理框架算法。

同样，从第一次常规迭代 $t = 1$到最后一次迭代 $t = T$的常规推理-行动 MOCO–AGENT 迭代采用 p_{PF}、p_{RA} 和 $p_{\text{I/O}}$ 精心设计的提示，但通过提示模板 p 纳入了与最佳评分算法 a_t^*、反馈 f_t^*和评分 s_t^*相关的信息，以提供通用迭代指令。此处，a_t^* 是最佳评分 Python算法的"文本化"版本。此外，其对应的反馈 f_t^*包含可行性和非支配CCPO解的数量，以及任何执行错误。同时也纳入了 IGD分数 s_t^*。生成的精心设计的提示将作为语言模型 $\mathcal{M}$ 的输入，以生成新算法 $\mathcal{A}_t$ （方程 2）。

随后，可以使用 Python元启发式算法$\mathcal{A}_t$，利用特定问题输入 x（例如基数约束参数 K 以及CCPO中的资产全集），在权重单纯形（公式 8）上生成CCPO的近似解。为简单起见，单位权重单纯形

Λ^{n-1} 按照步长 $\Delta\lambda$ 在所有 n目标方向上进行均匀离散化。因此，在 $n = 2$维空间中，所得权重向量 $\boldsymbol{\lambda} = \{\lambda, 1-\lambda\}$ 在 $\lambda \in [0,1]$之间进行遍历。将所有采样权重向量 $\boldsymbol{\lambda}$ 对应的CCPO近似解聚合为一个解集 Y_t。在根据CCPO约束（公式 10-13）检查解的可行性后，可以提取其中的非支配解子集 $H_t \subseteq Y_t$ （公式 3-4）。最后，可使用IGD指标（公式 5），以理论上最优的标准马科维茨投资组合优化UEF作为参考，测量有效前沿的性能分数 s_t。如果发现当前迭代的算法 $\mathcal{A}_t$ 分数 s_t 优于历史最佳值 s_t^*，则更新 a_t^*、 f_t^*和 s_t^*。在参考前沿不易获得的一般情况下，可以使用超体积指标 [（Guerreiro等人， 2022] ）作为合适的替代指标。

3.2 研究设计

本研究将 OpenAI o4-mini 大语言模型设定为编码智能体，并采用 MOCO–AGENT框架。本研究使用 MOCO–AGENT 为 CCPO 生成算法组合。在此过程中，通过修改角色分配提示 p_{RA}，在本研究的多个 MOCO–AGENT算法实例中生成多种算法。然而，鉴于迄今已有的元启发式算法数量不计其数 [Hussain 等人， 2019], 且近期基于隐喻的元启发式算法也受到了一些批评 [Sörensen， 2015], 本研究仅生成少量基于经典早期元启发式方法的算法。

本研究探索生成 10 种算法，每种算法均基于一种独特的元启发式方法。其中包括基于进化的方法，例如遗传算法（GA） [Holland、基于种群的增量学习（PBIL） [Baluja、 1994], 以及差分进化（DE） [Storn 和 Price，1997]；也包括早期的自然启发式方法，例如人工蜂群算法（ABC） [Karaboga，2005] 和萤火虫算法（FA） [Yang， 2009]；还包括基于群体的算法，例如粒子群优化（PSO） [Kennedy 和 Eberhart， 1995] 。最后，还包括基于轨迹的方法，例如爬山算法（HC） [Minsky， 1961],模拟退火（SA） [Kirkpatrick 等人， 1983], 禁忌搜索（TS） [Glover， 1989], 以及贪婪随机自适应搜索过程（GRASP） [Feo 和 Resende， 1995]。此外，除了在推理步骤中指定 MOCO–AGENT 要改编的元启发式方法外，角色分配提示 p_{RA}还明确指出，编码智能体应自行决定如何将生成的算法混合使用，但不得采用精确求解方法。

在 MOCO–AGENT 的动作步骤中，生成的算法在取自 OR-Library [Beasley, 1990] 的资产全集上进行性能测量。本研究采用了 CCPO [Chang等人, 2000] 基础工作中使用的相同恒生指数、德国DAX指数、富时指数、标准普尔和日经指数资产全集。因此，为便于与后续大量关于 CCPO 的文献进行比较，采用了相同的 $K = 10$、 $\varepsilon = 0.01$ 和 $\delta = 1$ 约束参数，以及 $\Delta\lambda = 0.02$ 目标步长。恒生指数资产全集数据由 31 项资产组成，是所采用的 OR-Library 系列中规模最小的资产全集，因此被选为用于算法生成的训练集。每个生成的算法均进行 $T = 32$ 推理-动作迭代；在每次动作迭代中，为该次迭代生成的 Python算法设定 10 分钟的执行时间限制。其余的德国DAX指数、富时指数、标准普尔和日经指数资产全集则保留为测试集，用于评估 a_T^*使用恒生指数训练集生成的评分最高的算法。

4 结果

4.1 生成的算法

表1展示了在恒生指数训练集上评估的 MOCO–AGENT框架生成的10种元启发式算法的性能摘要。生成的算法根据两项性能指标进行评分。第一项指标是平均百分比偏差误差（MPE） [Chang等人， 2000], 它用于衡量相对于 UEF 的有效前沿收敛性，并在解集 $V(\lambda)$ 上计算。该解集表示与每个采样权重向量 $\boldsymbol{\lambda}$ 按$\Delta\lambda$ 步长离散化后对应的最佳目标函数值相关联的解。因此，集合 $V(\lambda)$ 包含51个投资组合解，因为 $\lambda \in [0,1]$ 每隔 $\Delta\lambda = 0.02$ 步进行一次采样。此外，不会剔除被支配解。第二项性能指标是 IGD [Ishibuchi等人， 2015], 它同时用于衡量有效前沿的收敛性和覆盖度。

表1：元启发式算法在恒生指数（$N = 31$）训练集上的表现。

	ABC	DE	FA	GA	GRASP	HC	PSO	PBIL	SA	TS
MPE	1.0983	1.0965	3.4104	1.9444	1.2319	1.0965	1.1165	4.5653	1.4103	20.4934
Rank	3	2	8	7	5	1	4	9	6	10
IGD	0.53E-4	0.69E-4	1.61E-4	1.17E-4	0.88E-4	1.09E-4	0.70E-4	3.31E-4	1.43E-4	18.10E-4
Rank	1	2	8	6	4	5	3	9	7	10
分数	2	2	8	6.5	4.5	3	3.5	9	6.5	10
保留	Yes	Yes	No	No	Yes	Yes	Yes	No	No	No

对于这两项性能指标，基于 FA、PBIL 和 TS 方法生成的算法表现显著逊于其他算法；在恒生指数训练集上的评估中，GA 和 SA 的表现也略逊一筹。因此，保留 ABC、DE、GRASP、HC 和 PSO，用于在测试集上进行评估，其余算法则予以剔除。生成的算法组合由四种现有的元启发式方法类别均等组成：基于进化的、基于自然的、基于群体的和基于轨迹的。此外，表2展示了生成算法的自我评估描述。这些描述是精心设计的提示所产生的副产品；这些提示要求语言模型在每次推理迭代中对生成的模型进行自我评估。

表2：生成算法的自我评估描述。

算法	描述
ABC	“采用偏置初始化和自适应邻域策略的增强型人工蜂群算法，和资产频率引导”。
DE	“结合模因局部搜索的混合自适应差分进化，包-括基于梯度的权重调整和贪心资产交换操作”。
GRASP	“采用簇多样化限制候选列表、自适应大邻域-搜索算子和算子选择学习的增强型 *GRASP*”。
HC	″采用自适应多重交换操作、贪婪交换、交叉多样化和禁忌期限的混合模因爬山法″。
PSO	″采用环形拓扑、莱维飞行变异、增强型差分进化算法和自适应局部交换搜索″

4.2 算法组合性能

在德国DAX指数、富时指数、标准普尔指数和日经指数资产全集上对生成的算法组合进行评估，这些资产全集构成了本研究的测试集。这些取自 OR-Library [Beasley， 1990],的数据集可作为合适的测试集，因为它们无论在资产内容还是全集规模方面，都与恒生指数训练集存在显著差异。例如，恒生指数全集包含 $N = 31$项资产，而日经指数全集包含 $N = 225$ 项资产。将测试集上的性能与当前最佳水平（SOTA）进行同等比较。表3展示了生成的元启发式算法在训练集和测试集上的性能。此外，还将当前最佳水平的性能作为参考。性能采用百分比偏差误差（PE）指标 [Chang等人， 2000] 进行衡量，并以各资产全集对应的效用有效前沿作为理论最优前沿参考；其中报告了所有纳入算法在各集合 V（λ）上的平均值、中位数、最小值和最大值PE。

表现最佳的算法得分在表3中以<u>下划线</u>标出，而通过 MOCO–AGENT框架生成的表现最佳算法得分则以**加粗**显示。对于标准普尔测试集，GRASP的指标留空，因为在所有抽样 $\Delta\lambda$ 步中找到投资组合之前，Python算法代码的执行时间已达到30分钟的限制。表3显示，生成的算法组合在多个性能指标上经常达到最先进水平的表现，偶尔也能超越最先进水平。此处将 Alcazar等人2024]在 [中测得的高种群规模人工蜂群算法（ABC-HP） [Cura 2021] 算法视为最先进水平。然而，生成的算法组合在 PE指标方面大幅优于历史上最先进水平的算法，这些指标是在 V（λ）上测得的，例如 Chang等人 2000], 提出的遗传算法、禁忌搜索和模拟退火算法 [，Deng等人 2012], 提出的粒子群优化算法 [，以及 Lwin和Qu 2013]提出的基于种群的增量学习-差分进化 [。

表3：与 V（λ）的最先进水平比较。

		ABC	DE	GRASP	HC	PSO	最先进水平†
恒生指数（$N = 31$）	Mean	1.0983	**1.0965**	1.2319	**1.0965**	1.1165	1.0873
	中位数	**1.2149**	1.2155	1.2155	1.2154	1.2211	1.2154
	Min	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0224	0.0000
	Max	1.5538	1.5538	8.4637	**1.5537**	1.6241	1.5538
德国DAX指数（$N = 85$）	Mean	2.5920	**2.3232**	2.3778	2.3398	2.5802	2.2898
	中位数	2.6050	**2.5470**	2.5630	2.5630	2.6069	2.5629
	Min	**0.0059**	**0.0059**	**0.0059**	**0.0059**	0.0164	0.0059
	Max	10.3078	5.7669	6.7775	**4.3210**	5.8620	4.0275
富时指数 ($N = 89$)	Mean	1.0042	0.9443	0.9728	**0.8799**	1.1566	0.8406
	中位数	**1.0841**	**1.0841**	**1.0841**	**1.0841**	1.2485	1.0841
	Min	0.0045	**0.0016**	**0.0016**	0.0327	0.0327	0.0016
	Max	4.9484	5.9721	7.5422	**2.6819**	2.8589	2.0670
标准普尔 ($N = 98$)	Mean	1.7416	1.8478	–	**1.4351**	1.6552	1.3464
	中位数	1.1810	1.2144	–	**1.1420**	1.1755	1.1515
	Min	0.0046	0.2655	–	**0.0009**	0.0061	0.0009
	Max	9.8889	8.9263	–	**7.7908**	10.5647	5.4520
日经指数 ($N = 225$)	Mean	0.6787	2.7516	0.7184	**0.5782**	1.2403	0.5665
	中位数	0.6229	1.8011	0.6118	**0.5858**	1.1422	0.5858
	Min	**0.0000**	0.5863	0.0238	0.0003	0.1675	0.0000
	Max	2.7825	7.9270	5.0087	**1.1606**	3.1126	1.1606

† ABC-HP算法 [Cura， 2021] 被测评为最先进水平中表现最佳的方法 [Alcazar等人，2024年].

4.3 集成元启发式算法

除了单个算法的表现外，还可以合并生成的算法组合中的解，以提升有效前沿的表现。图2展示了从 OR-Library [Beasley 1990]获取的恒生指数、德国DAX指数、富时指数、标准普尔和日经指数资产全集中，由生成的元启发式算法得到的投资组合解合并而成的有效前沿 V（λ）。图2中的黑线表示各数据集对应的效用有效前沿。每个抽样 λ 步骤均选取最佳算法组合解。不过，不会剔除不同抽样λ 值对应的被支配解。图2和表4显示，生成的元启发式算法对有效前沿的贡献程度各不相同；不同算法在各资产全集中的贡献存在显著差异。例如，差分进化算法和粒子群优化算法对恒生指数有效前沿贡献很大，而爬山算法没有贡献。相反，爬山算法对日经指数有效前沿贡献很大，而差分进化算法和粒子群优化算法没有贡献。值得注意的是，GRASP算法对每个受评估资产全集的有效前沿都作出了重要贡献。

表4：元启发式算法对集成 $V(\lambda)$ 前沿的贡献。

	ABC	DE	GRASP	HC	PSO
恒生指数（$N = 31$）	–	19/51	8/51	–	24/51
德国DAX指数（$N = 85$）	–	34/51	10/51	1/51	6/51
富时指数（$N = 89$）	3/51	27/51	19/51	1/51	1/51
标准普尔（$N = 98$）	–	3/51	23/51	16/51	9/51
日经指数（$N = 225$）	1/51	–	17/51	33/51	–

还可以通过各算法对改善 IGD 指标的贡献，衡量其在生成的算法组合中的作用。表5显示了纳入特定算法解后，汇总 IGD 的相对改善幅度。例如，在恒生指数资产全集中，将 ABC 解加入由 DE、GRASP、HC 和 PSO 解汇总而成的解集后，汇总 IGD 指标改善了 48.52%。表5还显示了

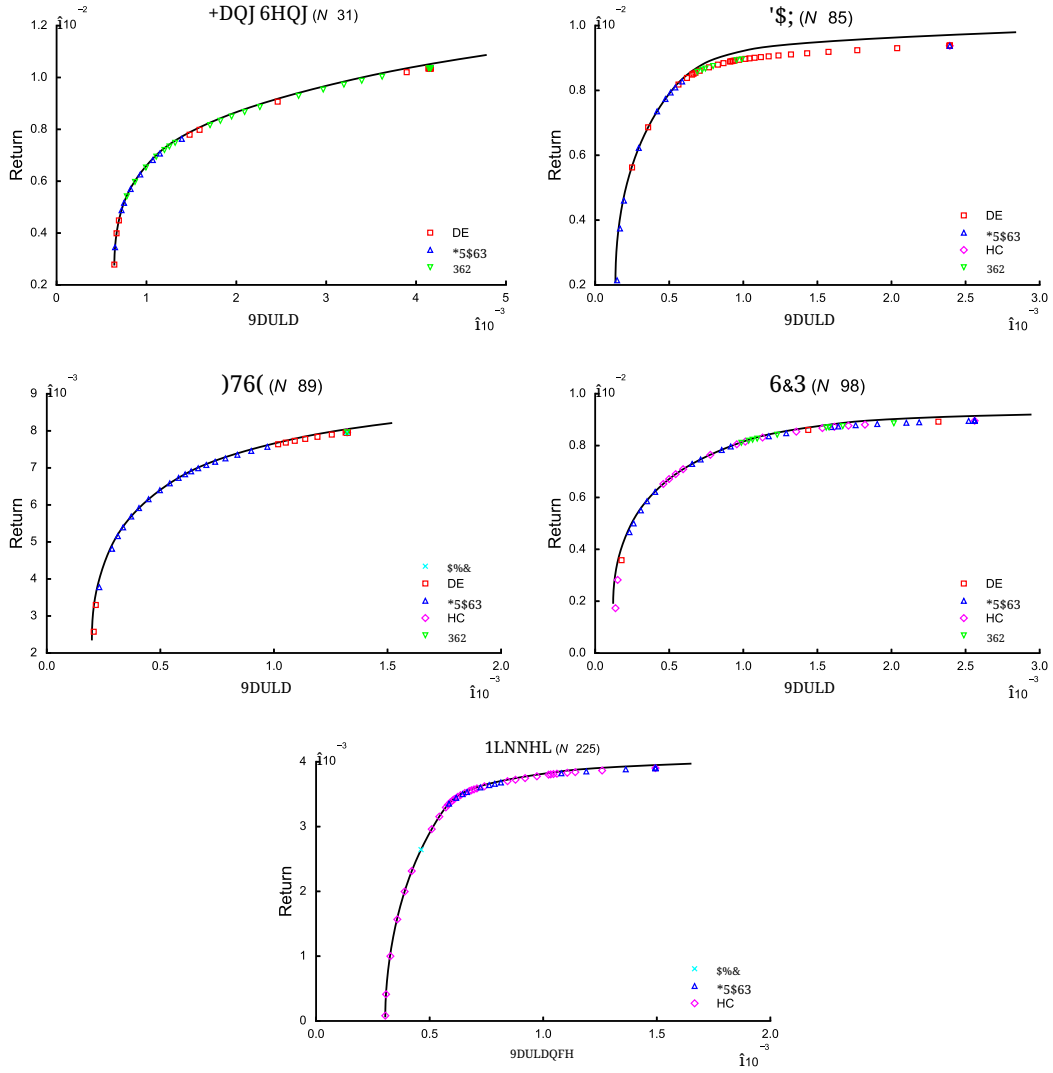


图2：汇总的元启发式算法 $V(\lambda)$ 前沿。

各算法对改善 IGD 指标的贡献程度存在差异。与 PE 指标不同，IGD 指标不仅衡量收敛性，也衡量覆盖度。例如，ABC 算法几乎没有为所评估的 $V(\lambda)$ 前沿作出贡献，却显著改善了所有已评估资产全集的 IGD。这表明，尽管其每个 λ 的收敛性能可能相对较差，但其非支配解集 H 的覆盖度非常出色。相反，DE 或 GRASP 等算法虽然对 $V(\lambda)$ 前沿作出了重要贡献，但对若干已评估资产全集的 IGD 指标影响甚微。这说明，生成的算法组合中包含多样化算法十分重要，因为不同算法对有效前沿收敛性和覆盖度的贡献程度各不相同。

5 结论

总之，本研究成功实现了一种智能体框架，用于生成多目标组合优化的算法组合。该框架应用于 NP-hard CCPO 问题，无需投入大量人工开发工作，便成功生成了一组元启发式算法。随后，我们在基准资产全集上验证了生成的算法组合，并将其与最先进水平进行比较。研究结果表明，智能体框架生成的算法

表5：元启发式算法对提升合并 IGD 的贡献。

	ABC	DE	GRASP	HC	PSO
恒生指数（$N = 31$）	48。52%	0.12%	0.03%	4.42%	0.04%
德国DAX指数 ($N = 85$)	33。21%	10.38%	16.80%	2.09%	35。05%
富时指数 ($N = 89$)	30.32%	11.08%	34。53%	2.16%	3.32%
标准普尔 ($N = 98$)	61。04%	8.89%	0.58%	118.28%	7.62%
日经指数 ($N = 225$)	215.08%	0%	5.63%	32.54%	18。22%

经常能够达到最先进水平的表现，有时甚至更胜一筹。此外，本研究还考察了汇集生成算法所得解的影响，发现每种算法都能为有效前沿收敛和覆盖度的不同区域作出独特贡献。本研究仅针对 CCPO 问题，并采用了一种基于贪心改进的代理框架。未来研究应探索其他智能体框架设计，并将评估范围扩展至其他多目标组合优化问题，以进一步验证其性能。此外，未来研究还应考虑不同生成算法之间的交互，并通过超启发式提高有效前沿生成的效率。尽管如此，研究结果证明了所提出的智能体框架在为 CCPO 生成算法方面的有效性，也展示了所实现的智能体框架整体上的开发效用：汇集不同算法所衍生算法的多样性，能够提升所研究有效前沿的整体表现。

致谢

本研究得到加拿大自然科学与工程研究委员会（NSERC）发现基金以及麦吉尔工程博士奖（MEDA）的支持。计算资源由加拿大数字研究联盟和魁北克计算提供。

参考文献

Liming Xu、Stephen Mak 和 Alexandra Brintrup。机器人会接管供应链吗？重探基于智能体的供应链自动化。《国际生产经济学杂志》，241:108279，2021年。doi: 10.1016/j.ijpe.2021.108279。

Yang Chen、Samuel N. Kirshner、Anton Ovchinnikov、Meena Andiappan 和 Tracy Jenkin。一位管理者和一个人工智能走进酒吧：ChatGPT 会像我们一样做出有偏见的决策吗？《制造与服务运营管理》，27(2):354–368，2025年a。doi: 10.1287/msom.2023.0279。

Fei Liu、Yue Niu、Qihua Zhang、Kai Wang、Zheyi Dong、Io Nam Wong、Linling Cheng、Ting Li、Lian Duan、Kun Li、Gen Li、Tai Wa Hou、Manson Fok、Huiyan Luo、Xiangmei Chen、Kang Zhang 和 Yun Yin。医疗保健领域人工智能代理的基础架构。*CellReports Medicine*，6(10)：102374，2025 年。doi：10.1016/j.xcrm.2025.102374。

Jiangbo Yu 和 Michael F. Hyland。用于人机协同交通规划的城市动态可解释状态空间模型。*TransportationResearch Part B: Methodological*，192：103134，2025年。doi：10.1016/j.trb.2024.103134。

Jiangbo Yu。为人机交通运输系统的智能代理时代做好准备：机遇、挑战与政策建议。*Transport Policy*，171：78–97，2025年。doi：10.1016/j. tranpol.2025.05.030。

Bernardino Romera-Paredes、Mohammadamin Barekatain、Alexander Novikov、Matej Balog、M. Pawan Kumar、Emilien Dupont、Francisco J. R. Ruiz、Jordan S. Ellenberg、Pengming Wang、Omar Fawzi、Pushmeet Kohli 和 Alhussein Fawzi。《利用大型语言模型进行程序搜索的数学发现》。《自然》，625(7995):468–475，2024年1月。doi: 10.1038/s41586-023-06924-6。

Rindranirina Ramamonjison、Timothy Yu、Raymond Li、Haley Li、Giuseppe Carenini、Bissan Ghaddar、Shiqi He、Mahdi Mostajabdaveh、Amin Banitalebi-Dehkordi、Zirui Zhou 和 Yong Zhang。NL4Opt 竞赛：基于自然语言描述构建优化问题。载于*NeurIPS 2022* 竞赛专题论文集，第220卷，《机器学习研究论文集》，第189–203页。PMLR，2022年11月28日—12月9日。

Ziyang Xiao、Dongxiang Zhang、Yangjun Wu、Lilin Xu、Yuan Jessica Wang、Xiongwei Han、Xiaojin Fu、Tao Zhong、Jia Zeng、Mingli Song 和 Gang Chen。《专家链：当大语言模型遇上复杂运筹学问题》。载于第十二届国际学习表征会议，2024年。

Weiwei Sun、Shengyu Feng、Shanda Li 和 Yiming Yang。《CO-Bench：组合优化算法搜索中的语言模型智能体基准测试》。*arXiv*，2025年。doi：10.48550/ ARXIV.2504.04310。

Zhiheng Xi、Wenxiang Chen、Xin Guo、Wei He、Yiwen Ding、Boyang Hong、Ming Zhang、Junzhe Wang、Senjie Jin、Enyu Zhou 等。《基于大语言模型的智能体的兴起与潜力：一项综述》。《中国科学：信息科学》，68(2):121101，2025年。

Mahdi Mostajabdaveh、Timothy Tin Long Yu、Samarendra Chandan Bindu Dash、Rindra Ramamonjison、Jabo Serge Byusa、Giuseppe Carenini、Zirui Zhou 和 Yong Zhang。使用 orqa 评估运筹学领域中的大语言模型推理。*AAAI*人工智能会议论文集，39(23):24902–24910，2025年4月。doi: 10.1609/aaai.v39i23.34673。

Kalyanmoy Deb。多目标优化，第403–449页。Springer US，美国马萨诸塞州波士顿，2014年。doi: 10.1007/978-1-4614-6940-7 15.

Ibrahim Oğuz Çetinkaya、I. Esra Büyüktahtakın、Parshin Shojaee 和 Chandan K. Reddy。利用大语言模型（LLM）发现混合整数规划的启发式方法：单机调度。*Computers&Operations Research*，186:107325，2026年。doi: 10.1016/j.cor.2025. 107325。

Karla L. Hoffman. 组合优化：当前成果与未来发展方向。计算与应用数学杂志，124(1):341–360，2000年。doi: 10.1016/S0377-0427(00)00430-1。《2000数值分析》。第 IV 卷：优化与非线性方程。

Fernando Peres 和 Mauro Castelli。组合优化问题与元启发式算法：综述、挑战、设计与发展。*Applied Sciences*，11(14)，2021年。doi: 10.3390/app11146449。

Jenny FajardoCalderín、AntonioD. Masegosa 和 David A. Pelta。用于动态优化问题的算法组合方案。*International Journal ofComputational Intelligence Systems*，8(4):667，2015年。doi: 10.1080/18756891.2015.1046327。

T.-J. Chang、N. Meade、J.E. Beasley 和 Y.M. Sharaiha。用于基数有约束投资组合优化的启发式方法。*Computers & Operations Research*，27(13):1271–1302，2000年。doi: 10. 1016/S0305-0548(99)00074-X。

Harry Markowitz. 投资组合选择。*The Journal of Finance*，7(1):77–91，1952。

R. Moral-Escudero、R. Ruiz-Torrubiano 和 A. Suarez。具有基数约束的最优投资组合选择。载于*2006IEEE InternationalConference onEvolutionary Computation*，第 2382–2388 页，2006 年。doi: 10.1109/CEC.2006.1688603。

Can B. Kalayci、Okkes Ertenlice 和 Mehmet Anil Akbay。均值-方差投资组合优化确定性模型及其应用的全面综述。*Expert SystemswithApplications*，125:345–368，2019 年。doi: 10.1016/j.eswa.2019.02.011。

M. Woodside-Oriakhi、C. Lucas 和 J.E. Beasley。用于基数约束有效前沿的启发式算法。*European Journalof Operational Research*，213(3):538–550，2011年。doi：10. 1016/j.ejor.2011.03.030。J. E. Beasley。OR-Library：通过电子邮件分发测试问题。*Journal of theOperationalResearch Society*，41(11):1069–1072，1990年。doi：10.1057/jors.1990.166。

Hongzheng Chen、Yingheng Wang、Yaohui Cai、Hins Hu、Jiajie Li、Shirley Huang、Chenhui Deng、Rongjian Liang、Shufeng Kong、Haoxing Ren、Samitha Samaranayake、Carla P. Gomes 和 Zhiru Zhang。HeuriGym：面向组合优化中大语言模型构造的启发式算法的智能体基准。*arXiv*，2025b。doi：10.48550/ARXIV.2506.07972。

Aman Madaan、Niket Tandon、Prakhar Gupta、Skyler Hallinan、Luyu Gao、Sarah Wiegreffe、Uri Alon、Nouha Dziri、Shrimai Prabhumoye、Yiming Yang、Shashank Gupta、Bodhisattwa Prasad Majumder、Katherine Hermann、Sean Welleck、Amir Yazdanbakhsh 和 Peter Clark。《Self-Refine：通过自反馈进行迭代式细化》。载于神经信息处理系统进展，第 36 卷，第 46534–46594 页。Curran Associates, Inc.，2023年。

Haoran Ye、Jiarui Wang、Zhiguang Cao、Federico Berto、Chuanbo Hua、Haeyeon Kim、Jinkyoo Park 和 Guojie Song。《ReEvo：作为超启发式的语言模型与反思式进化》。载于神经信息处理系统进展，第 37 卷，第 43571–43608 页。Curran Associates, Inc.，2024年。doi：10.52202/079017-1381。

Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik R Narasimhan 和 Yuan Cao。ReAct：在语言模型中协同推理与行动。载于第十一届国际学习表征会议，2023年。

Andreia P. Guerreiro、Carlos M. Fonseca 和 Luís Paquete。超体积指标：计算问题与算法。*ACM*计算调查，54(6):1–42，2022年7月。doi: 10.1145/3453474。

Hisao Ishibuchi、Hiroyuki Masuda、Yuki Tanigaki 和 YusukeNojima。世代距离和反世代距离中的修正距离计算。载于进化多准则优化，第110–125页。Springer国际出版公司，2015年。

Cristina Bazgan、Stefan Ruzika、Clemens Thielen 和 Daniel Vanderpooten。加权和标量化在近似多目标优化问题中的威力。计算系统理论，66(1):395–415，2022年2月。doi: 10.1007/s00224-021-10066-5。

George Mavrotas。多目标数学规划问题中 ε-约束方法的有效实现。应用数学与计算，213(2):455–465，2009。doi: 10.1016/j.amc.2009.03.037。

S.P. Boyd 和 L. Vandenberghe。凸优化。剑桥大学出版社，2004年。

Kashif Hussain、Mohd Najib Mohd Salleh、Shi Cheng 和 Yuhui Shi。元启发式研究：全面综述。《人工智能评论》，52(4):2191–2233，2019年12月。doi: 10.1007/s10462-017-9605-z。

Kenneth Sörensen。元启发式——揭示其隐喻。《国际运筹学研究汇刊》，22(1):3–18，2015年。doi: 10.1111/itor.12001。

John H. Holland。《自然与人工系统中的适应》：生物学、控制与人工智能应用导论。麻省理工学院出版社，美国马萨诸塞州剑桥，1992 年。ISBN 0262082136。

Shummet Baluja。基于种群的增量学习：一种整合遗传搜索、基于函数优化和竞争学习的方法。技术报告，美国卡内基梅隆大学，1994 年。

Rainer Storn 和 Kenneth Price。差分进化——一种用于连续空间全局优化的简单高效启发式算法。《全球优化杂志》，11(4)：341–359，1997 年 12 月。doi：10.1023/A:1008202821328。

Dervis Karaboga。基于蜜蜂群的数值优化思想，技术报告——tr06。技术报告，埃尔吉耶斯大学，2005 年 1 月。

Xin-She Yang。用于多模态优化的萤火虫算法。载于《随机算法：基础与应用》，第169–178页，柏林、海德堡，2009年。施普林格·柏林·海德堡。

J. Kennedy 和 R. Eberhart。粒子群优化。载于《*ICNN'95* - 国际神经网络会议论文集》，第4卷，第1942–1948页，第4卷，1995年。doi: 10.1109/ICNN. 1995.488968。Marvin Minsky。迈向人工智能的步骤。*IRE*会议录，49(1):8–30，1961年。doi: 10.1109/JRPROC.1961.287775。S. Kirkpatrick、C. D. Gelatt 和 M. P. Vecchi。通过模拟退火进行优化。《科学》，220 (4598):671–680，1983年。doi: 10.1126/science.220.4598.671。Fred Glover。禁忌搜索——第一部分。*ORSA*计算学杂志，1(3):190–206，1989年。doi: 10.1287/ ijoc.1.3.190。Thomas A. Feo 和 Mauricio G. C. Resende。贪婪随机自适应搜索程序。《全球优化杂志》，6(2):109–133，1995年3月。doi: 10.1007/BF01096763。Tunchan Cura。一种用于投资组合优化的快速收敛人工蜂群算法。《知识型系统》，233:107505，2021年。doi: 10.1016/j.knosys.2021.107505。Javier Alcazar、Mohammad Ghazi Vakili、Can B. Kalayci 和 Alejandro Perdomo-Ortiz。利用经典和量子生成模型增强组合优化。《自然·通讯》，15(1):2761，2024年3月。doi: 10.1038/s41467-024-46959-5。Guang-Feng Deng、Woo-Tsong Lin 和 Chih-Chung Lo。采用改进粒子群优化的基于马科维茨的基数约束投资组合选择。*Expert Systems withApplications*，39(4):4558–4566，2012年。doi: 10.1016/j.eswa.2011.09.129。Khin Lwin 和 Rong Qu。用于约束投资组合选择问题的混合算法。《应用智能》，39(2):251–266，2013年9月。doi: 10.1007/s10489-012-0411-7。