

基于神经网络的降维 用于具有数百万个变量的大规模二元优化

摘要

二元优化在背包问题、最大割问题和关键节点检测问题等实际应用中具有广泛的意义。数学规划、启发式、进化计算和神经网络等现有技术已被用于解决二元优化问题 (BOP)，然而，它们面临着优化大量二元变量的挑战。本文提出了一种基于神经网络的降维方法来辅助进化算法求解大规模 BOPs 问题。该方法将大量二元变量的优化问题转化为少量网络权值的优化问题，大大降低了搜索空间的维度。最重要的是，该方法避免了训练过程，从而消除了对先验知识的要求，提高了搜索效率。

关键词：二元优化: 大规模优化: 降维: 进化算法: 神经网络

1 引言

二元优化问题 (BOP) 代表了一类值得注意的组合优化问题，在软件工程、推荐系统、物流、生物信息学和电气系统等领域具有广泛的应用。有趣的是，组合优化中的 21 个经典 NP 完全问题可以重构为 BOP。二元优化的实践和理论重要性引起了人们的极大兴趣，激发了人们提出各种解决 BOP 的优化方法。广义上，BOP 旨在从候选项中识别最佳子集以优化特定目标，其中解决方案 $\mathbf{x} = (x_1, \dots, x_i, \dots, x_n)$ 中的每个决策变量 $x_i \in \{0, 1\}$ 表示是否选择第 i 项。因此，二元优化也称为零一编程和子集选择。离散搜索空间使得 BOP 不可微，并且通常会产生大量局部最优值，从而使得有效识别全局最优值成为一项具有挑战性的任务。虽然 BOP 通常是 NP 难的，但数学规划方法、启发式、进化算法和神经网络已被建议用于在多项式时间内逼近全局最优值。

2 相关工作

2.1 二元优化问题

BOP 的一个定义特征是二元决策向量，其中每个决策变量只能为零或一，指示相应的项目是否被选择。一般来说，无约束 BOP 可以在数学上写为

$$\begin{aligned} &\text{Minimize} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ &\text{Subject to} \quad \mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n \end{aligned} \quad (1)$$

其中 x 表示对应于 n 个项目的 n 个二元变量，例如背包问题中的项目、最大割问题中的节点、实例选择问题中的实例以及关键节点检测问题中的节点。求解 BOP 的目标是最小化 m 个目标 $f(x)$ ，例如背包问题中的总利润的负值，最大割问题中的互连边数的负值，实例选择问题中的训练损失，以及关键节点检测问题中的成对连通性。不同的 m 和 n 取值可能对应不同类型的 BOP。具体而言， $m = 1$ 表示单目标 BOP， $m \geq 2$ 表示多目标 BOP， $n < 100$ 表示小规模 BOP， $n \geq 100$ 表示大规模 BOP。在这项工作中，我们考虑具有大量变量的单目标和多目标 BOP。此外，我们不对目标的形式做出任何假设，其中的函数可以是非凸的、不可微的，甚至没有明确的公式。BOP 通常由数据驱动，其中实例的功能是根据特定的数据集计算的。相应地，进化算法擅长处理黑盒函数，而神经网络擅长从数据集中挖掘有用信息，因此成为二值优化的主流。

2.2 二元优化的进化算法

受生物进化机制的启发，进化算法是处理各个领域复杂问题的主流元启发法。特别是，遗传算法中提出了许多二元算子，并在求解 BOP 方面显示出有效性。例如，单点交叉和位翻转变异是最经典的二元算子 [11]，已被许多单目标和多目标进化算法所采用。虽然进化算法的随机搜索在处理大量决策变量时收敛缓慢，但一些多目标进化算法已经被提出用于大规模连续优化和二元优化，称为稀疏进化算法 [13]。通过考虑最优解的稀疏性（即大多数决策变量为零），这些算法在高维离散搜索空间中开发出具有快速收敛速度的新搜索策略。例如，[20] 中提出了稀疏交叉算子和稀疏变异算子，它们以相同的概率翻转零变量和非零变量，从而更好地生成具有更多零变量的解。[17] 中提出了一种降维方法，该方法使用受限玻尔兹曼机学习二元变量的稀疏分布。文献 [16] 提出了一种进化模式挖掘方法，根据当前种群挖掘零变量的最大和最小候选集。[5] 中设计了三种动态知识，用于指导二元交叉和变异算子中的后代生成。[14] 中提出了一种快速聚类方法，它以线性时间复杂度减少了数百万个二元变量。

2.3 用于二元优化的神经网络

与以黑盒方式搜索的进化算法相比，神经网络由于使用梯度方法而收敛得更快。由于 BOP 的函数是不可微的，因此使用梯度方法来优化网络权重以学习有用的映射，这主要包含三种不同的想法。给定特定类型的 BOP，第一个想法是直接学习数据集与其最优解之间的映射。一些数据集的最优解是通过精确的方法训练神经网络得到的，从而神经网络可以预测未知数据集的最优解。例如，为了解决背包问题，使用带有注意力的序列到序列模型来预测问题的最优解 [10]。为了解决双层背包问题，一种称为主邻域聚合的图神经网络用于预测上层问题的最优解 [8]。虽然很难提前获得足够的训练样本，但第二种想法是通过强化学习生成训练样本，它定义了马尔可夫决策过程（即增量添加或删除项目）并学习状态和动作之间的映射。状态通常表示选定或剩余项目的特征，动作表示要选择或删除的项目。例如，指针网络用于解决背包问题 [2]，S2V-DQN 用于解决最大割问题 [7]，GIN 用于解决最大独立集问题 [1]。在多目标优化方面，强化学习方法将目标分解框架与神经网络相结合，例如指针网络 [9]。另一方面，第三个想法无需训练样本即可学习数据集与其最优解之间的映射。这些方法通过优化网络权重来搜索最优解，从而将离散搜索空间转换为连续搜索空间。例如，Hopfield 神经网络用于解决最大割问题 [3]，尖峰神经网络用于解决二次二元优化问题，图卷积网络用于解决最大割问题和最大独立集问题。

除了直接输出最优解之外，神经网络还被用来辅助其他优化方法尤其是进化算法的使用。这些包括种群初始化、交配选择、后代生成、进化方向学习 [6]、环境选择 [12]、局部搜索 [22]、代理模型 [19]、降维 [17]、搜索空间转换 [4]、算子选择 [15]、算法推荐 [18] 和算法迁移 [21]。尽管这些想法中的许多仅在文献中应用于连续优化，但由于进化算法的高度灵活性，它们可以很容易地扩展到二元优化。

3 本文方法

3.1 本文方法概述

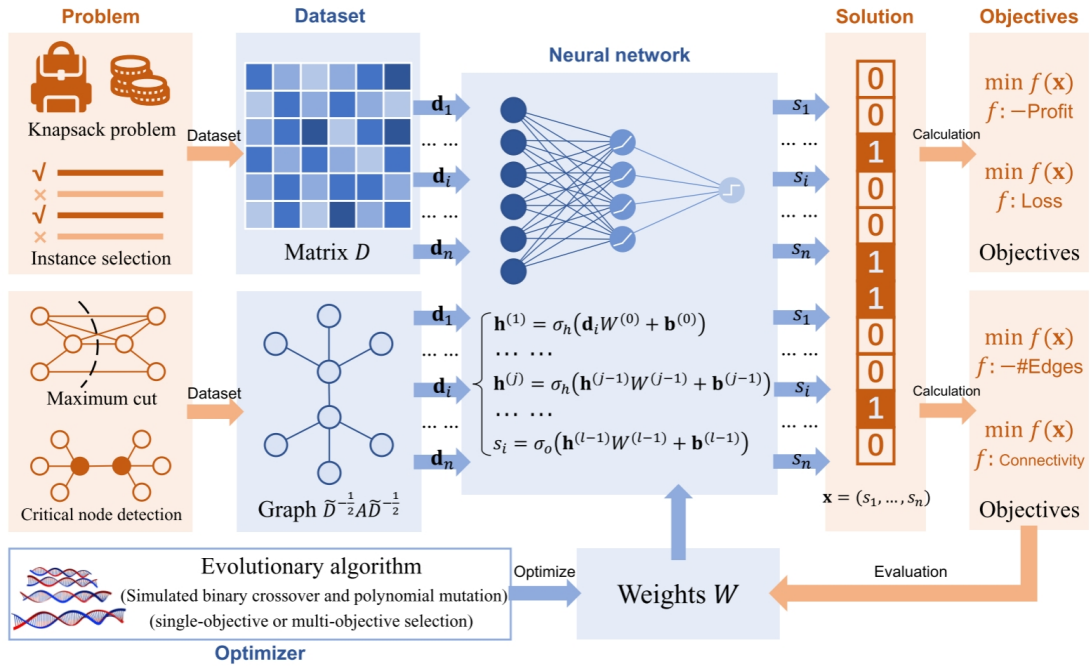


图 1. 所提出的神经网络辅助进化算法 NNDREA 的机制。从特定问题的数据集中提取项目的特征并将其输入神经网络，神经网络输出要由目标评估的解决方案。进化算法优化网络权重，从而可以改变输出的解决方案以最小化目标。

基于神经网络的降维方法，提出了一种进化算法，简称 NNDREA。NNDREA 的机制如图 1 所示，详细过程如算法 1 所示。该算法首先从特定问题的数据集中提取特征，然后随机初始化 N 个权重集，其中每个权重集对应一个独特的神经网络。通过计算每个神经网络的输出，可以获得一组解决方案。在每一代，首先使用单目标或多目标交配选择策略选择 N 个亲本权重集，并使用连续遗传算子生成 N 个后代权重集。然后，获得后代解的群体。之后，从当前和后代权重集的组合中保留 N 个权重集，并从当前和后代解决方案的组合中保留 N 个解决方案，这些解决方案是使用单目标或多目标环境进行的选择策略。

尽管所提出的 NNDREA 在优化过程中演化了两个种群，但这两个种群具有相同的目标值，并且每一代仅消耗 N 个函数评估。因此，相同的策略用于权重集和解决方案的选择。为简单起见，采用二元锦标赛选择作为交配选择策略，其中采用目标值作为单目标 BOP 的标准，采用非支配前沿数和拥挤距离值作为单目标 BOP 的标准。多目标 BOP。此外，这些标

Algorithm 1 Procedure of NNDREA

Input: N (population size), D (dataset), Δ (proportion of generations using dimensionality reduction)

Output: P (final population)

- 1: $I \leftarrow$ Extract the features of all items from D ;
 - 2: $P_W \leftarrow$ Randomly initialize N weight sets;
 - 3: $P_X \leftarrow$ Obtain solutions based on P_W by Eq. (8);
 - 4: **while** the termination criterion is not fulfilled **do**
 - 5: **if** the proportion of consumed function evaluations is smaller than Δ **then**
 - 6: $P'_W \leftarrow$ Select N parent weight sets from P_W via single- or multi-objective mating selection strategy;
 - 7: $Q_W \leftarrow$ Generate N offspring weight sets based on P'_W via continuous genetic operators;
 - 8: $Q_X \leftarrow$ Obtain solutions based on Q_W by Eq. (8);
 - 9: $P_W \leftarrow$ Select N weight sets from $P_W \cup Q_W$ via single- or multi-objective environmental selection strategy;
 - 10: $P_X \leftarrow$ Select N solutions from $P_X \cup Q_X$ via single- or multi-objective environmental selection strategy;
 - 11: **else**
 - 12: $P'_X \leftarrow$ Select N parent solutions from P_X via single- or multi-objective mating selection strategy;
 - 13: $Q_X \leftarrow$ Generate N offspring solutions based on P'_X via binary genetic operators;
 - 14: $P_X \leftarrow$ Select N solutions from $P_X \cup Q_X$ via single- or multi-objective environmental selection strategy;
 - 15: **end if**
 - 16: **end while**
 - 17: **return** P_X ;
-

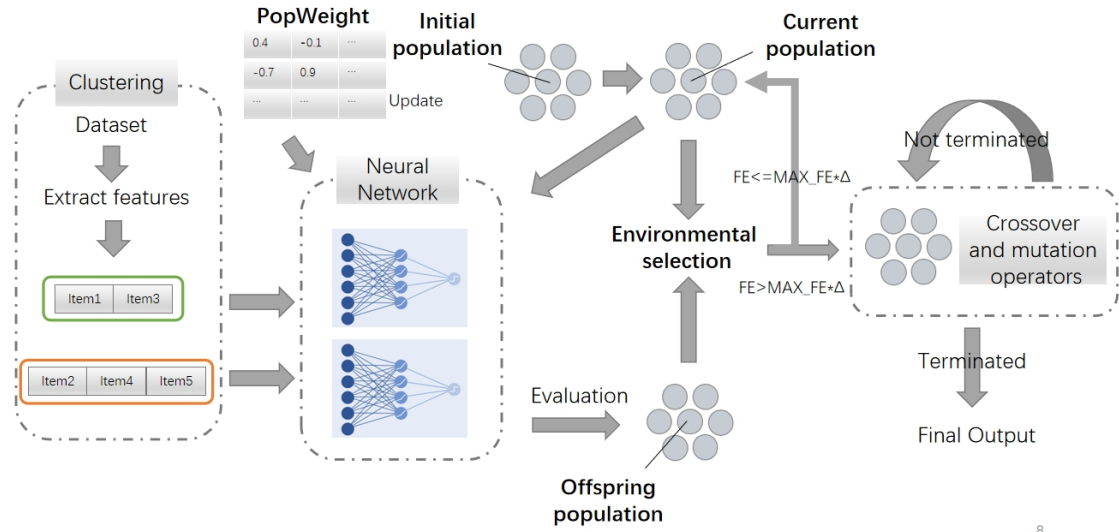


图 2. 聚类算法流程

准还被用于单目标和多目标 BOP 的环境选择。

值得注意的是，参数 Δ 用于控制使用降维的代的比例。经过 Δ 代百分比后，NNDREA 在原始高维搜索空间中演化出解决方案，无需神经网络的帮助。这可以减轻陷入低维搜索空间中的局部最优的风险，从而改进解决方案以更好地逼近全局最优。为了优化低维搜索空间中的连续权重，采用模拟二元交叉和多项式变异作为遗传算子。此外，为了优化高维搜索空间中的二元变量，采用均匀交叉和位翻转变异作为遗传算子。

4 复现细节

4.1 与已有开源代码对比

在 PLATEMO 平台上参考了 NNDREAMO 算法代码(<https://github.com/BIMK/PlatEMO>)。

4.2 实验环境搭建

在 PLATEMOv4.0 上运行。

参数设置与原始论文中概述的相同，函数评估总数固定为 $10n$ 。群体大小设置为 50，交叉概率设置为 1，变异概率设置为 $1/n$ ，其中 n 表示决策变量的数量。NNDREA 参数 Δ 设置为 0.5，交叉和变异的分布指数设置为 20。

5 实验结果分析

5.1 创新点

论文中仅使用一个神经网络去判断是 0 还是 1。创新点：增加聚类方法，每个类使用一个神经网络去判断。设计了 5 中算法与论文作比较。

M1: k-means 对多目标背包问题的物品做 k-means 聚类， $k=2$ 。

M2: 首尾聚类 (112233332211)

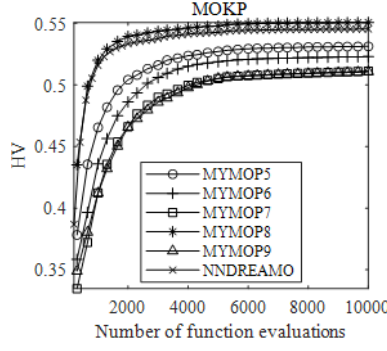


图 3. 各个算法的 HV 曲线 ($k=2, D=1000$)

基于性价比，对所有物品进行排序，排序以后，对所有物品基于性价比分成 $2N$ 份。假设为 $(c_1, c_2, \dots, c_{2n})$ 。然后将 c_1 和 c_{2N} 中的物品聚为一类， c_2 和 c_{2N-1} 一类， c_3 和 c_{2N-2} 一类，以此类推。（这样每个类中都有性价比高和性价比低的，两端的性价比区分度会很大，中间的区分度会更小）。

M3: 间隔 $D/2$ 聚类 (112233112233)

基于性价比，对所有物品进行排序，排序以后，对所有物品基于性价比分成 $2N$ 份。假设为 $(c_1, c_2, \dots, c_{2n})$ 。然后将 c_1 和 c_{N+1} 中的物品聚为一类， c_2 和 c_{N+2} 一类， c_3 和 c_{N+3} 一类，以此类推。

M4: 从当前种群中提取稀疏知识，添加种群非零率特征。

从当前种群中获取有用信息，如非零率特征，即非零值所占种群大小的比例。作为特征，输入到神经网络中去。

M5: M3+M4

间隔 $D/2$ 聚类方法与添加种群非零率特征方法结合。

表 1. 与 NNDREA 算法对比 ($k=2$)

Problem	Dec	M1	M2	M3	M4	M5	NNDREA
MKP1	1000	5.3127e-1 (1.70e-2)-	5.2292e-1 (1.81e-2)-	5.1124e-1 (2.23e-2)-	5.5068e-1 (1.05e-2)=	5.1068e-1 (3.12e-2)-	5.4539e-1 (1.73e-2)
MKP2	5000	5.3674e-1 (2.71e-2)-	5.2814e-1 (3.63e-2)-	5.3925e-1 (1.22e-2)-	5.5890e-1 (1.13e-2)=	5.3164e-1 (1.51e-2)-	5.6141e-1 (2.71e-2)

表 2. 与 NNDREA 算法对比 ($k=5$)

Problem	Dec	M1	M2	M3	M4	M5	NNDREA
MKP1	1000	5.0800e-1 (2.09e-2)-	4.6567e-1 (2.26e-2)-	4.6482e-1 (2.44e-2)-	5.5087e-1 (1.11e-2)=	4.5738e-1 (2.39e-2)-	5.4539e-1 (1.73e-2)
MKP2	5000	5.2981e-1 (1.60e-2)-	4.8670e-1 (2.58e-2)-	4.8850e-1 (3.10e-2)-	5.5986e-1 (9.92e-2)=	4.8439e-1 (2.80e-2)-	5.6141e-1 (1.00e-2)

从表 1 和表 2 可以看出，所用的聚类方法对算法性能没有提升，分类越多性能会恶化。其中 M4 的实验结果与原论文中的结果没有显著性差异。可以考虑进一步探索更多特征，如物品的相关性、历史选择数据等，并将其整合到神经网络的输入中，以提升模型的决策能力。

6 总结与展望

论文提出的基于神经网络的降维方法为解决大规模二进制优化问题提供了一种新的思路。通过结合深度学习和降维技术，能够有效地减少计算复杂度，并在不牺牲精度的情况下

加速优化过程。实验结果验证了该方法的高效性和适用性，尤其在处理具有数百万变量的问题时，相比传统方法具有明显优势。

未来的改进方法可以考虑以下：

1. 聚类算法的多样化：除了当前使用的几种聚类方法外，还可以引入更多先进的聚类算法，如密度聚类或基于图的聚类方法，以进一步提高分类的准确性。
2. 动态调整聚类参数：当前的聚类方法参数（如 K 值或 N 值）是固定的，但在不同的实验条件下，适应性地调整这些参数可能会带来更好的效果。
3. 深度学习方法的结合：结合深度学习中的自适应机制（如自注意力机制、强化学习等），可以进一步提升模型对复杂物品关系的建模能力。
4. 高维特征的引入：进一步探索更多特征，如物品的相关性、历史选择数据等，并将其整合到神经网络的输入中，以提升模型的决策能力。

参考文献

- [1] Kenshin Abe, Zijian Xu, Issei Sato, and Masashi Sugiyama. Solving np-hard problems on graphs with extended alphago zero. arxiv 2019. *arXiv preprint arXiv:1905.11623*.
- [2] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.
- [3] Fuxi Cai, Suhas Kumar, Thomas Van Vaerenbergh, Rui Liu, Can Li, Shimeng Yu, Qiangfei Xia, J Joshua Yang, Raymond Beausoleil, Wei Lu, et al. Harnessing intrinsic noise in memristor hopfield neural networks for combinatorial optimization. *arXiv preprint arXiv:1903.11194*, 2019.
- [4] Roman Denysiuk, António Gaspar-Cunha, and Alexandre CB Delbem. Neuroevolution for solving multiobjective knapsack problems. *Expert Systems with Applications*, 116:65–77, 2019.
- [5] Zhuanlian Ding, Lei Chen, Dengdi Sun, and Xingyi Zhang. A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Swarm and Evolutionary Computation*, 73:101119, 2022.
- [6] Yi Jiang, Zhi-Hui Zhan, Kay Chen Tan, and Jun Zhang. Knowledge learning for evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 2023.
- [7] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.
- [8] Sunhyeon Kwon, Hwayong Choi, and Sungsoo Park. Solving bilevel knapsack problem using graph neural networks. *arXiv preprint arXiv:2211.13436*, 2022.

- [9] Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.
- [10] Hazem AA Nomer, Khalid Abdulaziz Alnowibet, Ashraf Elsayed, and Ali Wagdy Mohamed. Neural knapsack: a neural network based solver for the knapsack problem. *IEEE access*, 8:224200–224210, 2020.
- [11] Jeffrey R Sampson. Adaptation in natural and artificial systems (john h. holland), 1976.
- [12] Ke Shang, Weiduo Liao, and Hisao Ishibuchi. Hvc-net: Deep learning based hypervolume contribution approximation. In *International Conference on Parallel Problem Solving from Nature*, pages 414–426. Springer, 2022.
- [13] Yansen Su, Zhongxiang Jin, Ye Tian, Xingyi Zhang, and Kay Chen Tan. Comparing the performance of evolutionary algorithms for sparse multi-objective optimization via a comprehensive indicator [research frontier]. *IEEE Computational Intelligence Magazine*, 17(3):34–53, 2022.
- [14] Ye Tian, Yuandong Feng, Xingyi Zhang, and Changyin Sun. A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization. *IEEE/CAA Journal of Automatica Sinica*, 10(4):1048–1063, 2022.
- [15] Ye Tian, Xiaopeng Li, Haiping Ma, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1051–1064, 2022.
- [16] Ye Tian, Chang Lu, Xingyi Zhang, Fan Cheng, and Yaochu Jin. A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE transactions on cybernetics*, 52(7):6784–6797, 2020.
- [17] Ye Tian, Chang Lu, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE transactions on cybernetics*, 51(6):3115–3128, 2020.
- [18] Ye Tian, Shichen Peng, Xingyi Zhang, Tobias Rodemann, Kay Chen Tan, and Yaochu Jin. A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. *IEEE transactions on artificial intelligence*, 1(1):5–18, 2020.
- [19] Ye Tian, Shangshang Yang, Lei Zhang, Fuchen Duan, and Xingyi Zhang. A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(2):106–116, 2018.

- [20] Ye Tian, Xingyi Zhang, Chao Wang, and Yaochu Jin. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(2):380–393, 2019.
- [21] Xiaoming Xue, Cuie Yang, Yao Hu, Kai Zhang, Yiu-Ming Cheung, Linqi Song, and Kay Chen Tan. Evolutionary sequential transfer optimization for objective-heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 26(6):1424–1438, 2021.
- [22] Haoran Ye, Jiarui Wang, Zhiguang Cao, Helan Liang, and Yong Li. Deepaco: neural-enhanced ant systems for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36, 2024.