Deep Reinforcement Learning for Multiobjective Optimization

Chen Huaneng

2025 年 10 月 15 日

1 Deep Reinforcement Learning Based Multiobjective Optimization Algorithm (DRL-MOA)

这个研究基于两个关键背景：

- 多目标优化问题（Multiobjective Optimization Problem, MOP）的基本困境：传统方法（如 NSGA-II, MOEA/D）通过迭代更新种群寻找 Pareto 最优解，但面对大规模问题时（如 200 城市的多目标旅行商问题 MOTSP）时，迭代次数多、计算效率低，且问题稍有变化（如城市位置微调）就需要重新计算；
- 深度强化学习（Deep Reinforcement Learning, DRL）的优势：DRL 能通过试错学习训练一个"黑箱模型"，训练后只需要一次前向传播就能输出解，无需迭代，且泛化能力强（能够处理未见过的问题实例）。

文章[1] 提出的 DRL-MOA 框架，本质是用"分解思想"（来自 MOEA/D[2]）拆解多目标优化问题，用"DRL + 神经网络"（来自 Pointer Network[3]、Actor-Critic[4-5]）求解每个子问题，再用"参数迁移"加速训练。

2 General Framework

通用框架是 DRL-MOA 的"骨架"，解决了"如何将多目标问题转化为 DRL 可处理的单目标问题"和"如何高效训练多个子问题的模型"这两个核心问题。分为分解策略和领域参数迁移策略两部分。

2.1 Decomposition Strategy

Decomposition Strategy: 文章中采用 weighted sum approach[6] 进行多目标优化问题的分解，也可以采用其他 scalarizing methods，比如 Chebyshev 和 the penalty-based boundary intersection (PBI) method[7-8]。首先，生成一组均匀分布的权重向量（uniformly spread weight vector）$\lambda^1, \lambda^2, \ldots, \lambda^N$，其中 N 为子问题的数量，比如对于双目标问题（$M = 2$），可以取权重向量为 $(1, 0), (0.9, 0.1), \ldots, (0, 1)$，每个向量表示对不同目标的"重视程度"。对第 j 个权重向量 $\lambda^j = (\lambda^j_1, \lambda^j_2, \ldots, \lambda^j_M)^{\mathrm{T}}$，$M$ 表示目标

函数的个数，通过 weighted sum approach，可以将 MOTSP 分解为 N 个单目标优化子问题（scalar optimization subproblems）。第 j 个子问题的目标函数为：

$$\min g^{ws}(x \mid \lambda_i^j) = \sum_{i=1}^{M} \lambda_i^j f_i(x) \tag{1}$$

其中 $f_i(x)$ 是原 MOP 的第 i 个目标函数，$g^{ws}(x \mid \lambda_i^j)$ 是第 j 个子问题的“加权和成本”（单目标）。

分解后每个子问题的解都是原 MOP 的 Pareto 最优解，这是因为权重向量的不同权衡，使得每个子问题的最优解对应 PF（Pareto Front）上的一个“权衡点”。通过将 MOP 分解成子问题，可以将每个子问题的“加权和成本”作为 DRL 的“奖励信号（比如奖励 = -加权和成本，因为 DRL 通常最大化奖励，而 MOP 需要最小化成本）。这样就通过将 MOP 拆解为多个标量子问题，每个子问题对应一个“权重向量”，求解所有子问题的解就可以组成 PF。

2.2 Neighborhood-Based Parameter-Transfer Strategy

Neighborhood-Based Parameter-Transfer Strategy: 采用领域参数迁移的策略的核心在于，如果每个子问题都“从头训练”一个神经网络，计算量会非常大（N 个子问题需要 N 次独立训练）。但文章根据公式 (1) 和 Zhang 的研究[2] 发现，相邻权重向量对应的子问题，其最优解和最优模型参数非常相似，比如在双目标问题中，权重向量为 $(0.8, 0.2)$ 和 $(0.7, 0.3)$ 的子问题对于目标的权衡接近，最优路径和模型参数也接近。因此，借鉴 MOEA/D 的“领域更新”思想[2]，提出了领域参数迁移策略，即用前一个子问题的最优模型参数，作为当前子问题的初始参数，避免从头训练，减少计算成本。

其具体过程为：假设已经训练好第 $i-1$ 个子问题的最优模型参数 $[w^*_{\lambda^{i-1}}, b^*_{\lambda^{i-1}}]$（$w$ 为权重，b 为偏置），在训练第 i 个子问题时，使用 $[w^*_{\lambda^{i-1}}, b^*_{\lambda^{i-1}}]$ 作为初始参数（$[w_{\lambda^i}, b_{\lambda^i}] = [w^*_{\lambda^{i-1}}, b^*_{\lambda^{i-1}}]$）进行训练。然后在此基础上用 Actor-Critic 进行微调，快速收敛到第 i 个子问题的最优参数 $[w^*_{\lambda^i}, b^*_{\lambda^i}]$。重复该过程，直到所有 N 个子问题都训练完毕。

领域参数迁移策略的优势在于无需为每个子问题初始化随机参数，从而减少了收敛时间，降低了训练复杂度；同时，由于相邻子问题的模型参数平滑过渡，避免 PF 上出现“跳跃”的解，保证了解的一致性和多样性。

2.3 Pseudo Code of General Framework of DRL-MOA

DRL-MOA 的通用框架伪代码如 algorithm 1所示。每个子问题的训练核心是 Actor-Critic 算法，负责将子问题的“加权和成本”转化为模型的优化信号。训练完成之后，对于新的 MOP 实例，只需要一次前向传播（forward propagation）就能得到对应的 Pareto 最优解，无需重新训练。

Algorithm 1: General Framework of DRL-MOA

Input: The model of the subproblem $\mathcal{M} = [\mathbf{w}, \mathbf{b}]$, weight vectors $\lambda^1, \ldots, \lambda^N$
Output: The optimal model $\mathcal{M}^* = [\mathbf{w}^*, \mathbf{b}^*]$

$[\omega_{\lambda^1}, \mathbf{b}_{\lambda^1}] \leftarrow$ Random_Initialize
for $i \leftarrow 1$ **to** N **do**
  if $i == 1$ **then**
    $[\omega^*_{\lambda^1}, \mathbf{b}^*_{\lambda^1}] \leftarrow$ Actor_Critic$([\omega_{\lambda^1}, \mathbf{b}_{\lambda^1}], g^{\text{ws}}(\lambda^1))$
  else
    $[\omega_{\lambda^i}, \mathbf{b}_{\lambda^i}] \leftarrow [\omega^*_{\lambda^{i-1}}, \mathbf{b}^*_{\lambda^{i-1}}]$
    $[\omega^*_{\lambda^i}, \mathbf{b}^*_{\lambda^i}] \leftarrow$ Actor_Critic$([\omega_{\lambda^i}, \mathbf{b}_{\lambda^i}], g^{\text{ws}}(\lambda^i))$
  end if
end for
return $[\mathbf{w}^*, \mathbf{b}^*]$
/* *Given inputs of the MOP, the PF can be directly calculated by* $[\mathbf{w}^*, \mathbf{b}^*]$. */

3 Modeling the Subproblem of MOTSP

文章的实验实例是多目标旅行商问题 MOTSP : The multiobjective traveling salesman problem (MOTSP), where given n cities and M cost functions to travel from city i to j, one needs to find a cyclic tour of the n cities, minimizing the M cost functions.

3.1 Formulation of MOTSP

One needs to find a tour of n cities, that is, a cyclic permutation ρ, to minimize M different cost functions simultaneously.

$$\min z_k(\rho) = \sum_{i=1}^{n-1} c^k_{\rho(i),\rho(i+1)} + c^k_{\rho(n),\rho(1)}, \quad k = 1, 2, \ldots, M \tag{2}$$

where $c^k_{\rho(i),\rho(i+1)}$ is the k-th cost of traveling from city $\rho(i)$ to $\rho(i+1)$. The cost functions may, for example, correspond to tour length, safety index, or tourist attractiveness in practical applications.

参考文献

[1] LI K, ZHANG T, WANG R. Deep Reinforcement Learning for Multiobjective Optimization[J/OL]. IEEE Transactions on Cybernetics, 2021, 51(6): 3103-3114. DOI: 10.1109/TCYB.2020.2977661.

[2] ZHANG Q, LI H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition[J/OL]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731. DOI: 10.1109/TEVC.2007.892759.

[3] VINYALS O, FORTUNATO M, JAITLY N. Pointer Networks[C]//Advances in Neural Information Processing Systems: Vol. 28. Curran Associates, Inc., 2015.

[4] NAZARI M, OROOJLOOY A, SNYDER L, et al. Reinforcement Learning for Solving the Vehicle Routing Problem[C]//Advances in Neural Information Processing Systems: Vol. 31. Curran Associates, Inc., 2018.

[5] BELLO I, PHAM H, LE Q V, et al. Neural Combinatorial Optimization with Reinforcement Learning: arXiv:1611.09940[A/OL]. 2017. arXiv: 1611.09940.

[6] MIETTINEN K. Nonlinear multiobjective optimization: Vol. 12[M]. Springer Science & Business Media, 1999.

[7] WANG R, ZHOU Z, ISHIBUCHI H, et al. Localized weighted sum method for many-objective optimization[J/OL]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 3-18. DOI: 10.1109/TEVC.2016.2611642.

[8] WANG R, ZHANG Q, ZHANG T. Decomposition-based algorithms using pareto adaptive scalarizing methods[J/OL]. IEEE Transactions on Evolutionary Computation, 2016, 20(6): 821-837. DOI: 10.1109/TEVC.2016.2521175.