

深度强化学习引导的多种群协同进化 超多目标优化算法

许莹¹⁾ 刘佳¹⁾ 陈斌辉²⁾ 刘益萍¹⁾ 刘志中¹⁾

¹⁾(湖南大学信息科学与工程学院 长沙 410082)

²⁾(顺丰科技有限公司 广东 深圳 518052)

摘 要 超多目标优化问题因高维决策空间与复杂计算成本等特点而极具挑战。作为求解方法之一,多种群协同进化算法通过协同机制在求解此类问题时有很好的效果,但仍存在计算成本高、搜索效率低等局限性。近年来,强化学习因其卓越的决策能力被引入进化算法框架,成为提升算法性能的关键技术。因此,本文提出了一种深度强化学习引导的多种群协同进化超多目标优化算法 DQNMaOEA,用于求解复杂的超多目标优化问题。为了有效引导大规模决策空间的搜索,提高算法在高维目标空间的搜索能力,本文提出了一种基于深度强化学习模型的自适应子种群选择方法,通过强化学习与环境进行交互选择具有更高潜力的子种群,然后与基于效用值选择的子种群进行协同进化,产生具有更优多样性与收敛性的子代解。此外,为了降低计算成本,提高算法的搜索效率,本文进一步提出了一种自适应子种群计算资源分配策略,根据当前子种群对整个种群优化过程的效用值改进贡献,动态分配子种群的适应值评估次数。为了验证算法及相关策略的性能,本文在大量基准测试集问题及实际物流大规模超多目标车辆路径问题实例上,与现有的不同类型前沿算法进行了大量对比实验。实验分析表明,本文提出的算法在求解性能与解质量上显著优于大部分对比算法。具体表现为:在评估解收敛性与多样性的综合指标上,DQNMaOEA在80%以上的基准测试实例中取得最优结果,较现有最佳算法的平均性能指标提升达1.2~2.0倍。而在计算效率方面,算法的平均运行时间较对比算法降低约25%。特别地,在7个实际物流问题实例中,算法在解的性能指标上获得6项最优结果,且求解效率显著优于对比算法。这些结果充分验证了该算法在解质量、计算效率和实际应用潜力上的综合优势。

关键词 超多目标优化;超多目标进化算法;自适应种群选择;自适应计算资源分配;强化学习

中图分类号 TP311 **DOI号** 10.11897/SP.J.1016.2025.02371

A Deep Reinforcement Learning Guided Multi-Population Cooperative Co-Evolutionary Many-Objective Optimization Algorithm

XU Ying¹⁾ LIU Jia¹⁾ CHEN Bin-Hui²⁾ LIU Yi-Ping¹⁾ LIU Zhi-Zhong¹⁾

¹⁾(College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082)

²⁾(SF Technology Ltd., Shenzhen, Guangdong 518052)

Abstract When solving real-world problems, it usually needs to optimize multiple conflicting objectives at the same time. Such problems are referred to multi-objective optimization problems (MOPs), if the number of objectives is more than three, the MOPs become the challenging Many-objective optimization problems (MaOPs). Many-objective optimization problems are extremely challenging due to the characteristics of high-dimensional decision space and complex

收稿日期:2025-03-06;在线发布日期:2025-07-08。本课题得到湖南省自然科学基金面上项目(2024JJ5095)资助。许莹,博士,副教授,中国计算机学会(CCF)杰出会员,主要研究领域为智能计算、多目标优化、智能优化与调度算法。E-mail: hnxy@hnu.edu.cn。刘佳(通信作者),硕士研究生,中国计算机学会(CCF)会员,主要研究领域为多目标优化、车辆路径规划问题、人工智能。E-mail: liujia0406@hnu.edu.cn。陈斌辉,博士,高级算法工程师,主要研究领域为运筹算法、目标优化。刘益萍,博士,副教授,主要研究领域为计算智能及其应用、多目标优化、进化计算与药物发现。刘志中,博士,副教授,主要研究领域为计算智能、机器人和具身智能。

computational cost. As one of the solution methods, the Multi-population Cooperative Co-evolutionary Algorithms (MPCCAs) select sub-populations through relevant strategies and guides the co-evolution of sub-populations, which performs well in solving various multi-objective optimization problems. However, the existing MPCCAs still suffer from the limitations of high computational cost and low search efficiency, etc. In recent years, reinforcement learning (RL) has been introduced into the framework of evolutionary algorithms due to its superior decision-making ability, which has become a key technology to improve the performance of algorithms. Therefore, this paper proposes a Deep Reinforcement Learning guided Multi-population Cooperative Co-evolutionary Many-objective Optimization Algorithm based on Deep Q-Network, namely DQNMaOEA, for solving complex multi/many-objective optimization problems. In order to effectively guide the search in a large-scale decision space and improve the search capability of the algorithm in high-dimensional objective space, this paper proposes an adaptive sub-population selection method based on the Deep Q-Network (DQN) model. Through the interaction between reinforcement learning and the environment, this method considers the characteristics of problem instances and the current states of sub-populations to select sub-populations with better performance. To further enhance the diversity of solutions, another sub-population is selected based on utility values to form the mating pool together, which can generate offspring solutions with both better diversity and convergence, and improve the adaptability of algorithm to solve different problem instances. Additionally, in order to reduce computational costs and improve the search efficiency of the algorithm, we propose an adaptive sub-population computational resource allocation strategy which dynamically allocates the fitness evaluation times of sub-population based on their utility value contributing to the entire population optimization process. Experimental results on a set of benchmark problems and a set of real-world many-objective vehicle routing problem of logistics from SF Technology Ltd. show that DQNMaOEA achieves better performance than other state-of-the-art algorithms and has high capability to solve real-world problems. This paper is organized as follows. Firstly, we systematically review the relevant research background and related work, and concisely summarizes the primary contributions of the proposed algorithm. In the second section, we comprehensively outlines and elaborates on the fundamental theoretical knowledge and key techniques involved in this study. Then, next section details the collaborative sub-population selection approach and introduces the Deep Reinforcement Learning guided Multi-population Cooperative Co-evolutionary Many-objective Optimization Algorithm based on Deep Q-Network. This section provides an in-depth analysis of the design principles of the algorithm's core components and conducts a rigorous theoretical and computational performance analysis of the proposed algorithm. In the fourth section of this article, we present comprehensive comparative experiments across five distinct aspects to validate the effectiveness of the DQNMaOEA algorithm and its key strategies. The experimental results are presented and analyzed in depth. Finally, we conclude the research findings and outlines potential directions for future work.

Keywords many-objective optimization; many-objective evolutionary algorithm; adaptive sub-population selection; adaptive computing resource allocation; reinforcement learning

1 引 言

在现实世界中,大部分优化问题都需要同时优

化多个目标函数,这些目标之间相互制约,并且无法得到唯一最优解,其结果通常是一组综合考虑多个目标的折中解,即非支配解^[1]。在学术界,这类问题通常被称为多目标优化问题(Multi-objective

optimization problems, MOPs)^[2]。对于目标数大于3的MOP,称之为超多目标优化问题(Many-objective optimization problems, MaOPs)^[3]。

由于多目标进化算法(Multi-objective evolutionary algorithms, MOEAs)具有全面且灵活的求解能力,能够帮助解决现实世界中的复杂问题,并提供多个平衡优化目标之间的最佳选择,因此经常被用于解决MOPs。然而,随着目标数量的增加,逐渐超过3个时,MOEAs对目标向量的处理速度会急剧下降,并且无法很好地平衡种群的收敛性和多样性^[4],超多目标进化算法(Many-objective evolutionary algorithms, MaOEAs)的提出解决了这个问题。

传统的进化算法(Evolutionary algorithm, EA)通常依赖于单一群体来探索解决方案的可能性空间。这种单一的群体结构可能导致随着搜索过程的深入,群体中的个体逐渐变得相似,算法陷入局部最优的可能性增加,这使得搜索真正的帕累托前沿(Pareto front, PF)变得困难^[5]。多种群协同进化超多目标优化算法作为一类新的MaOEAs,因其更注重种群与环境、种群与种群之间的协调,并同时考虑竞争机制和协作机制以实现共同进化的性质,进而克服这一问题并增加群体中个体的多样性^[6]。这些子群体之间可以通过竞争、合作,或者两者的结合方式相互作用。因此,选择合适的子群体以及其内部个体的合理配置对于多群体算法在解决优化问题时能否取得理想结果具有决定性影响。

研究者们已经成功地将一些经典策略应用于协同进化算法中的子种群选择。这些包括单一最佳合作者选择^[7-9]、最佳+随机合作者选择^[10-11]、轮盘赌/锦标赛合作者选择^[12-13]、精英合作者选择^[14]以及链接合作者选择^[15]。除了这些基本策略,还有三种专门针对MOPs的策略。第一种策略是随机方法,即随机选择多个非支配解作为子种群的代表性合作者^[16-18]。如果产生多个完全解,则使用非支配解来估计子种群个体的适应度^[19]。第二种方法是基于偏好的方法,即从每个子种群中选择非支配个体作为代表,可以是膝点^[20]、最优超体积值的解^[21]、较优拥挤距离的解^[22-23],或者是纳什均衡个体^[24],形成完整的解。选取的非支配个体即子种群的代表成员通常具有最好的多样性和收敛性。第三种方法是一种基于竞争的方法,该方法在文献^[25]和文献^[26]中提出,用于选择具有代表性的子种群个体进行合作。每个子种群与其他子种群竞争以表示共享变量,较好的子种群被赋予较高的概率。

然而在所有优化问题上,并不存在一个优于其他任何子种群选择策略的策略,在求解特定超多目标优化问题、设计对应的进化算法时应谨慎选择策略^[27]。这也意味着,当求解不同问题实例,甚至是问题实例发生细微变化时,设计的算法策略不能很好地解决新的实例,该算法便可能不再适用,这在一定程度上降低了算法的问题求解能力。由于强化学习(Reinforcement learning, RL)在各种复杂和随机环境下的决策问题中表现出卓越的性能,尤其是与深度神经网络相结合时^[28]。这种整合大大增强了该方法在处理不确定的连续决策问题时的推理和学习能力。因此,许多研究已将RL集成到EA框架中,建立了强化学习辅助进化算法(RL-EA),这些算法已被证明能成功解决各种复杂的优化问题^[29]。

基于此,为了有效引导大规模决策空间的搜索,合理分配计算资源,提高其搜索效率与求解能力,以更好地解决各类不同的问题实例,本文提出一种深度强化学习引导的多种群协同进化超多目标进化算法(Deep Q-network guided multi-population cooperative co-evolutionary many-objective evolutionary algorithm, DQNMaoEA),算法框架如图1所示。

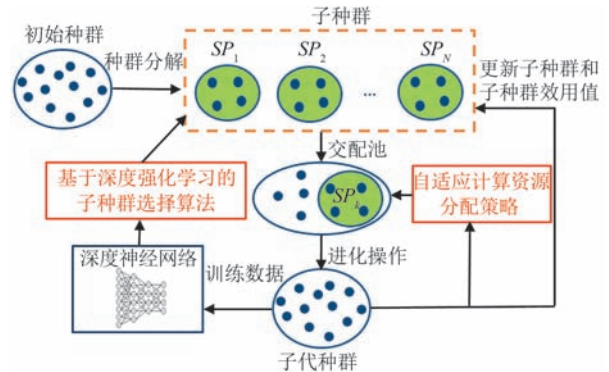


图1 DQNMaoEA算法框架图

本文主要贡献包括:

(1)提出了一种新的自适应子种群选择方法,该方法基于深度强化学习DQN模型。方法通过强化学习与环境的交互,考虑问题实例本身的特点与各子种群当前的状态,选择具有更高性能的子种群。为进一步考虑解的多样性,与基于效用值选择的子种群共同组成交配池,进行协同进化,产生具有更优多样性与收敛性的子代解,也提高了算法的不同问题实例求解能力;

(2)为了有效调度计算资源,算法定义子种群适应度相对改进率来评估子种群对当前种群的贡

献。自适应子种群计算资源分配策略则根据每个子种群的适应度改进率,动态分配子种群适应值评估次数,合理分配了算法的计算资源,在一定程度上提高算法的搜索效率,降低了算法的时间成本;

(3)基于以上策略,提出基于深度强化学习引导的多种群协同进化超多目标优化算法 DQNMaoEA,用于求解复杂的超多目标优化问题。在提出的 MaoEA 中,智能体通过迭代更新深度神经网络来指导子种群的选择。所提出的 MaoEA 在具有大规模决策空间和高维目标空间的复杂 MaOPs 上表现出高通用性,并在实验中获得了比最先进的 MaoEA 更好的性能。同时,算法也基于顺丰科技的支线物流运输问题的一组实际案例上进行了实验,证明了所提出算法求解实际问题的能力。

本文的其余部分结构如下。第二节对相关工作与知识点进行了全面的综述。第三节详细介绍了所提出的深度强化学习引导的多种群协同进化超多目标进化算法 DQNMaoEA 及相关策略。实验结果在第四节中进行了报告和分析。最后,第五部分对整个研究进行了总结。

2 相关工作

2.1 超多目标优化问题

一般来说,以一个同时最小化多个目标函数的 MOP 为例,其数学模型可以表示为:

$$\begin{aligned} &\text{minimize } F(x) = [f_1(x), f_2(x), \dots, f_M(x)] \\ &\text{subject to } x \in \Omega \end{aligned} \quad (1)$$

其中, M 表示目标数量, $x = (x_1, x_2, \dots, x_D) \in \Omega$ 是一个 D 维的决策向量, $x_k (k \in \{1, 2, \dots, D\})$ 表示决策变量。 $\Omega \subseteq \mathbb{R}^D$ 表示决策变量的搜索空间, $F(x)$ 是由 M 个子目标函数 $f_i(x), i \in \{1, \dots, M\}$ 决定的目标向量,且各目标函数之间相互冲突。在这些情况下,不存在一个单一的解能够同时最小化所有目标函数。因此,解决方案往往是一个帕累托最优解集,称为帕累托集(Pareto Set, PS)。其中每个解都是不可支配的,即不存在另一个解能够在所有目标上都优于它,称为非支配解或帕累托最优解。PS 在目标空间中的映射称为帕累托前沿 PF。因此,多目标进化算法求解多目标优化问题的目标是找到能够均匀且紧密地逼近 PS/PF 的代表性解集^[30]。

当一个 MOP 所要优化的目标函数大于 3 时,这个优化问题被称为超多目标优化问题 MaOP,并具

有以下特点:

(1)解空间的维度显著增加:随着目标数目的增多,解空间变得极为庞大,寻找帕累托最优前沿变得更加困难。

(2)决策的偏好表达更加复杂:很难为如此多的目标函数找到一个满意的折中点。

(3)计算资源需求增加:解决 MaOPs 需要更多的计算资源和时间。

正是由于大规模决策空间和高维目标空间带来的维度灾难^[31],随着目标数量的增加,问题变得更加复杂,导致传统 MOEAs 在求解 MaOPs 时性能往往急剧恶化。因此,研究如何高效且准确地求解 MaOPs 是十分必要的。为了实现这一目标,涌现了大量的相关策略与算法,并将其称为超多目标优化算法。

2.2 超多目标优化算法

在超高维目标空间中,传统基于排序的多目标优化算法的失效本质源于以下核心问题^[32]:

(1)支配关系失效:随着目标空间维度的不断增大,Pareto 支配机制在高维空间中呈现显著局限性。其目标维度每增加一维,种群中非支配解占比呈现指数级跃升,直至严格支配关系的概率趋近于零。此时,由于选择压力不足,算法无法有效逼近 Pareto 前沿,收敛速度大幅下降。

(2)多样性维护瓶颈:传统基于距离的多样性维护策略遭遇维度灾难。高维空间中距离计算呈现“趋同性”特征,且传统拥挤度机制等无法平衡收敛性与多样性,导致解的分布特性难以通过常规几何度量准确刻画。出现种群聚集或分布不均现象,无法有效维持多样性,使得高维下多样性维护成为瓶颈。

(3)计算复杂度骤增:传统基于排序的多目标优化算法在处理超高维目标空间时,其计算复杂度呈现双重维度敏感性,具体表现为非支配排序的阶乘级复杂度膨胀与多样性维护机制的隐性成本两方面。随着目标数的增加,传统算法计算资源消耗剧增,处理高维数据时效率低下,甚至无法完成排序。

针对传统基于排序的多目标优化算法在超高维目标空间中的核心失效问题,可通过设计支配关系重构、分解协同策略、指标驱动优化、机器学习增强等改进策略实现性能突破,形成适用于求解超高维目标空间问题的超多目标优化算法^[33]。如采用 ϵ -支配、模糊支配等建立动态阈值比较机制,或引入角度支配构建目标空间超锥体选择模型以重构支配关

系;结合基于分解多目标优化算法框架,将高维问题分解为带权重子问题簇,并设计相应参考向量自适应生成算法或开发混合分解策略,以实现分解协同策略;构建基于高维空间目标指标R2、IGD+等的搜索导向机制或采用超体积子梯度等驱动的进化算子等相关策略以实现指标驱动超多目标优化;构建深度强化学习驱动的动态选择压力调节器或开发基于Transformer的解集分布预测网络等利用基于学习的方法替代超多目标优化框架中的组件,以实现机器学习增强优化算法。

基于上述内容与改进方法,为了解决超多目标优化问题,研究人员提出了多种超多目标优化算法,大致可分为基于支配关系的MaOEAs,基于分解的MaOEAs以及基于指标的MaOEAs:

(1)基于支配关系的MaOEAs通过改进的非支配排序机制评估个体之间的相对优劣,并据此选择下一代种群。Deb等人为了求解NSGA-II^[34]在求解MaOPs时,拥挤距离导致解在非优势层分布不均匀,陷入局部最优的问题,设计了专为求解MaOPs的NSGA-III^[35],通过定义一组参考点来指导搜索过程,并保持种群的多样性。

(2)基于指标的MaOEAs是一类通过定义和使用特定的性能指标来引导优化过程的算法。这类算法通常通过最大化或最小化某个性能指标来指导搜索过程,从而使算法能够有效地找到高质量的帕累托最优解集。Gu等人^[36]提出了一种新的超多目标进化算法PSEA,用于求解具有不同PF形状的MaOPs。该算法采用基于单位向量指标的种群预选策略,在目标空间中划分出一个有前景的区域,选择收敛性好且不损失多样性的解进入下一轮迭代。

(3)基于分解的超多目标优化算法则是通过将多目标优化问题分解为多个单目标子问题来解决的方法,并允许算法同时优化不同的子目标,适合于处理具有超多目标优化问题。MOEA/D^[37]通过使用一组参考向量来指定PF的多个搜索方向,算法得到的最终解可以很好地覆盖PF。在MOEA/D的框架下,许多算法被开发出来。RVEA^[38]通过预定义的参考点和角度惩罚距离(Angle-penalized distance, APD)度量,在收敛性和多样性之间取得了很好的平衡。MOEA/D-DD^[39]则是结合了解析和支配策略来平衡算法的收敛性和多样性,取得了很好的效果。

此外,大多数MaOEAs通常依靠单一种群来搜索解空间,这种单一的种群结构可能会导致随着搜

索的深入,个体之间逐渐变得相似,从而增加算法陷入局部最优的风险。同时,复杂且冗余的搜索也会降低算法的搜索精度与效率。Zhan等人^[40]提出了一种新的协同进化技术,称为多种群多目标技术(Multi-population multi-objective, MPMO),并基于MPMO技术设计了CMPISO算法,对每个子种群采用PSO粒子群优化,并通过存档机制共享信息协同优化。MPMO是一种通用技术,其基本思想为将整体种群进行划分,让每个子种群仅对应一个目标,并通过相应的目标对每个子种群的个体适应度进行分配,其中每个子种群的优化都可以使用现有的优化算法。最近的一些研究表明,MPMO框架可以在一定程度上提高MaOEA算法的收敛性与多样性。Tian等人^[41]提出了MPMMEA算法,利用多个子群体探索决策空间的不同区域,并通过自适应更新的引导向量引导子群体的搜索行为,每个子种群的引导向量可以在巨大的搜索空间中提供有效的收敛,有效求解大多数超多目标优化问题。

然而,现有的超多目标优化算法尽管在性能上取得了很大的进步,仍面临收敛速度慢以及动态问题适应能力差等巨大挑战。

2.3 强化学习

强化学习方法是机器学习(Machine learning, ML)方法的一个重要分支,用于在多样化和不确定环境中熟练的策略决策,其目标是学习最优的动作策略,以最大化期望的累积奖励^[42]。RL可分为两类,即基于策略的方法和基于价值的方法。基于策略的方法通过直接学习和优化随机策略的方法来处理具有连续动作空间的问题^[43],而基于价值的方法则是通过学习状态或状态-动作对的价值函数,来指导智能体的动作决策,进而达到优化长期回报的目的,适用于处理离散动作空间的问题^[44]。

RL的显著特点是从与环境的交互中学习,将其与有监督和无监督学习方法区分开来。在RL中,智能体(Agent)通过主动与环境交互以做出最优决策来最大化累积回报。除了智能体Agent,RL系统的完整框架还包括状态空间 S 、动作空间 A 、状态转移函数 P 、奖励函数 R 和折扣因子 γ 。因此,一个五元组 $\langle S, A, P, R, \gamma \rangle$ 作为Agent决策的基础,将强化学习中的Agent决策过程建模为一个马尔科夫决策过程(Markov Decision Process, MDP)^[45]。如图2所示,强化学习通过学习一个策略 π ,使得Agent根据当前时刻 t 的状态 s_t 选择动作,随后,通过与环境的交互来评估动作的性能进行动作决策,从而获得

奖励 r_t 。然后, Agent 综合考虑状态变化和策略, 决定下一步的动作 a_t 进行决策, 并转移到下一个状态 s_{t+1} 。这样, 策略 π 就通过状态和动作的结合来构建。此外, 记录元组 $\langle s_t, a_t, r_t, s_{t+1} \rangle$ 作为样本来训练网络, 使得 Agent 不断更新策略 π , 以最大限度地获得期望的奖励。

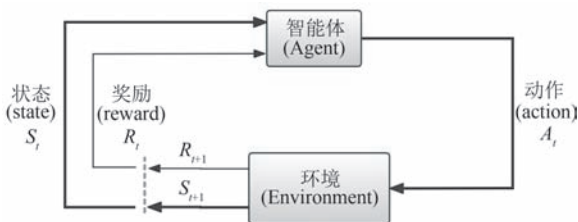


图2 强化学习示意图

为了解决传统进化算法存在的局限性, 如收敛速度慢和跨场景适应能力差等, 学者们将强化学习作为 EA 框架中的一个组件, 对算法结构、算子、搜索模式等方面进行改进, 以提高算法的搜索效果, 近年来表现出了优越的性能^[43]。已有的相关工作将 RL 技术应用于多目标进化算法的非支配解生成^[46]、目标函数学习^[47]、自适应算法/种群/算子选择^[48-50]、参数调整^[51]等方面, 但尚未应用于多种群协同多目标优化算法的协作子种群选择。在文献^[50]中, Jia 等人提出了一种基于多种群的算法 MPMA-QL, 该算法包含 7 个采用不同的交叉策略的子种群, 并使用专门针对问题特征的 Q 学习方法, 为每个子种群选择其个体组成。在搜索中使用的子种群可以通过 RL 由基于竞争的多种群 EA 确定。Guo 等人^[52]将 RL 和集成种群策略嵌入到遗传编程框架中, 其基本思想为在每轮迭代搜索中, 利用 RL 选择一个子种群。这些方法均与本文使用的深度强化学习方法完全不同, 且并不是为了选择求解 MaOPs 的协同进化子种群而量身定做的。

Deep Q-Network (DQN) 作为一种基于深度学习和强化学习的算法, 已被成功用于处理离散动作空间中的强化学习任务^[53]。由于参考向量引导的子种群选择也可以看作是一个离散的动作空间, 因此, 在我们的工作中, 将 DQN 用于子种群的选择, 提出了一种融合深度强化学习 DQN 模型的自适应子种群选择方法的多种群协同超多目标优化进化算法。

2.4 车辆路径规划问题求解

车辆路径规划问题 (Vehicle routing problem, VRP) 最初由 Dantzig 等人^[54]于 1959 年以卡车调度问题的形式进行了阐释。经典的 VRP 可以描述为: 给

定一个仓库和多个已知需求的客户节点, 在满足特定的约束条件下, 以最小化总运输成本或距离为目标, 为一组车辆分配一条从仓库出发, 经过一系列客户节点并最终返回仓库的最佳配送路线, 使得所有客户节点都被访问且仅被访问一次^[55]。随着路径规划问题在现实场景中的广泛应用, 根据现实需要, VRP 需要纳入更多的约束条件和目标函数考量, 由此衍生出了众多 VRP 变体。根据优化目标的不同, 车辆路径规划问题可以分为单目标车辆路径规划问题和多目标车辆路径规划问题。其中, 单目标车辆路径规划问题是指仅考虑一个优化目标, 求解相对简单。而多目标车辆路径规划问题则需要同时考虑多个优化目标。因其涉及多个优化目标, 且这些目标之间存在相互冲突或相互制约的关系, 求解过程相对复杂。在实际应用场景中, 许多车辆路径问题均为多目标情况下的决策优化问题。此外, 当优化目标数大于 3 时, 问题变为超多目标车辆路径规划问题。

由于 VRP 的 NP-难特性^[56], 求解 VRP 一直被视为一项极具挑战性的优化任务。为此, 研究人员正不断努力, 致力于开发各种高效的方法来解决这一难题。目前, 求解 VRP 的主要方法大致可分为经典方法、基于学习的端到端方法以及混合方法三类。

其中, 经典方法可进一步细分为精确方法、启发式算法以及元启发式算法。精确方法依赖于严密的数学逻辑与计算流程, 旨在找到车辆路径规划问题的最优解。如通过将 VRP 建模为双指标车流量公式, 并不断添加有效的不等式约束以加强问题的建模, Naddef 等人^[57]设计了相应求解的分支切割算法。启发式方法则是建立在经验与直观理解之上的一类方法, 它能够在合理的时间框架内找到车辆路径规划问题的近似最优解。根据解的生成机制不同, 可将启发式算法进一步分为构造启发式与改进启发式。Mole 等人^[58]提出了一种基于插入法的节约算法, 在合并两条路径的同时, 允许将可行节点插入路径中。而元启发式方法则是一类基于自然过程或生物行为的优化算法, 通过迭代搜索, 得到问题的高质量近似最优解。以是否基于种群为划分标准, 可以将元启发式算法分类为基于单一解的元启发式算法与基于种群的元启发式算法。Vincent 等人^[59]提出了一种新的模拟退火算法变体, 该变体引入了自适应邻域移动选择策略, 以增强算法的性能。Sabar 等人^[60]则开发了一种基于种群的迭代局部搜索方法, 该算法在运行过程中维持一个包含潜在优秀解的种群, 自适应地执行包含各种进化算子的局

部搜索算法,以求解动态VRP。

随着人工智能和深度学习技术的发展,研究人员在利用基于深度学习的模型端到端求解VRP方面获得了新的启发,其中训练好的网络模型代表了输入和最终解之间的近似隐式映射函数,求解时用于从输入实例直接输出解,该方法在车辆路径规划领域得到广泛应用并取得了突破性的进展,进一步提高了求解效率与通用性。在基于学习的端到端方法中,可根据学习模式的不同进一步分为监督学习(Supervised learning, SL)、无监督学习(Unsupervised learning, UL)以及强化学习(Reinforcement learning, RL),三者均是仅利用学习的方法直接生成从起点到终点的最优路径规划结果。Kong等人^[61]提出了一种基于注意力机制的指针网络(Attention-based pointer network, A-Ptr-Net),旨在解决无人机物流配送中的轨迹优化问题,兼顾路径质量与实时性。Joshi等人^[62]则提出了一个基于图增强和动态路由机制的深度学习框架,用于学习TSP的最优解。此外,Jiao等人^[63]提出了一种基于角度密度的K-means聚类遗传算法K-means-AD,以尽可能均衡每辆车的行驶距离。为了求解取送货车车辆路径规划问题,Li等人^[64]在编码层引入异构注意力机制,优先处理高价值节点,并使用强化学习Actor-Critic算法^[65]对模型进行训练。

而后,研究人员进一步将传统算法框架与机器学习方法相融合,提出混合求解方法,与端到端方法不同的是,训练好的模型充当改进算子的辅助工具,在迭代过程中对解进行优化改进,或利用传统方法对网络模型进行加速,更好地应用于处理复杂优化问题和具有多约束条件的场景。根据混合方法解的结构和求解流程不同,可以将其分为两大类:学习辅助传统方法以及传统方法加速学习的优化算法。Lima等人^[66]和Alipour等人^[67]分别使用Q-learning与多智能体强化学习生成初始解,然后由遗传算法对生成的初始解进行迭代优化。Sun等人^[68]则通过支持向量机量化每条边属于最优路径的可能性,有效缩小了启发式算法的搜索空间以简化问题复杂度,实现学习辅助传统算法。Joe等人^[69]提出了一种基于深度强化学习的解决方法DRLSA,该方法主要包括训练和运行两个阶段,通过将基于时序差分学习与经验回放的深度强化学习算法和基于模拟退火算法的路由元启发式相结合,实现了传统方法加速基于学习的优化算法。

车辆路径规划问题作为物流与运筹学领域的核

心问题,经过数十年的发展,已形成了丰富的理论与方法体系。然而,随着实际应用场景的复杂化与多样化,现有方法在求解效率、动态适应性与实时在线求解等方面仍存在诸多挑战。

3 DQNMaeOA 算法

在本节中,我们首先介绍本文提出的深度学习引导的多种群协同进化超多目标优化算法DQNMaeOA的框架,然后分别介绍DQNMaeOA算法中自适应子种群选择方法和自适应子种群计算资源分配策略,并对算法的相关性能与科学性进行了详细分析。

3.1 算法框架

图3给出了DQNMaeOA算法的流程图。首先,采用基于分解的思想,算法随机初始化 V 个参考向量、 N 个候选解,并将候选解与对应的参考向量 λ' 相关联,每个子种群 SP_i 与一个参考向量 λ' 关联,实现多种群协同优化。当算法未满足终止条件时进入循环步骤,而若满足终止条件,则算法结束运行,并输出最终种群。

本文提出的算法DQNMaeOA的伪代码如算

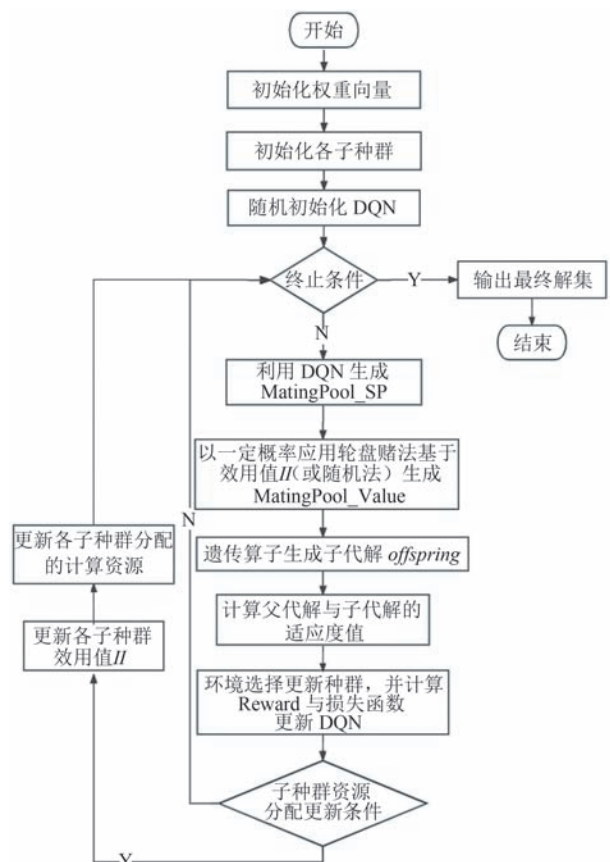


图3 算法整体流程图

法1所示,首先对参数进行初始化:(1)随机生成有 N 个个体的种群 P ;(2)生成 V 个 M 维在目标空间均匀分布的参考向量集 λ ;(3)关联种群 P 与参考向量 λ ,初始化各子种群 SP_i ,每个子种群 SP_i 与一个参考向量 λ^i 关联,实现多种群协同优化;(4)随机参数化一个DQN;(5)设置各子群效用值 Π_k 为1;(6)初始化各参数值:目标最小值 Z 、经验回放池 T 、奖励队列 Rq 。

当算法未满足终止条件时进入循环步骤,主要过程如下:(1)根据当前种群与参考向量的状态,与DQN进行交互,完成交配池 $MatingPool_SP$ 的选择,自适应选择父代子种群 P_SP ;(2)利用结合轮盘赌法的基于效用值 Π 子种群选择方法,产生交配池 $MatingPool_Value$ 。同时,为避免陷入局部最优,算法以一定概率抽取随机个体,共同生成协同父代子种群 P_Value ;(3)由交配池中的父代解生成子代解;(4)环境选择策略中,根据Tchebycheff切比雪夫聚合函数的适应度值大小更新种群 P ,同时,在该过程中计算并返回DQN该阶段的奖励 $reward$;(5)利用更新后的种群和参考向量的状态以及奖励 $reward$ 更新DQN,同时更新目标最小值 Z 、经验回放池 T 以及奖励队列 Rq ;(6)判断并更新子种群的资源分配。若满足终止条件,则算法结束运行,并输出最终种群 P 。

算法1. 1DQNMaOEA算法框架

输入: N (种群数量), M (目标数量)

输出: P (最终种群)

1. $P = Initialization(N)$; //随机初始化种群 P
2. $\lambda = UniformPoint(V, M)$; //生成 V 个均匀分布的参考向量 λ
3. $SP = UpdateAssociate(\lambda, V)$; //关联种群 P 与参考向量 λ ,生成各子种群
4. $rl = RecRL_SP(\lambda, N, length(SP))$; //初始化DQN
5. $\Pi = ones(length(SP), 1)$; //初始化各子种群的效用值 Π
6. $Z \leftarrow MiniminumObj(P)$; //计算种群目标的最小值
7. $T \leftarrow \emptyset$; //初始化经验回放队列
8. $Rq \leftarrow \emptyset$; //初始化奖励队列
9. $Obj \leftarrow P.object$; //记录当前种群的目标值
10. $sub_FE = ones(length(SP), 1)$; //记录每一轮迭代中各子种群的最大评估次数
11. While 未满足结束条件 do
12. $P_SP = MatingPool_SP(P, \lambda, SP, M)$; //利用基于DQN的子种群选择策略,生成交配

池 $MatingPool_SP$,详见算法4

13. $P_Value = MatingPool_Value(P, \Pi, SP)$; //利用基于效用值的子种群选择策略,生成交配池 $MatingPool_value$,详见算法5
14. While operation no. $\leq sub_FE(no, P_SP)$ do
15. $p_sp \leftarrow P_SP(operation\ no. ++)$; //从 P_SP 中选取一个解作为父代解
16. $p_value \leftarrow ProbabilitySelect(P_Value,)$; //从 P_Value 中随机选取一个父代解
17. $Offspring = GenerateOffsprings(p_sp, p_value)$; //遗传算子生成子代解
18. $Z = Minimize(Z, Offspring.obj)$; //更新目标最小值
19. $[P, Rq, reward] = Selection(Offspring UP(P_value) UP p_sp)$; //环境选择策略更新种群,见算法2
20. $rl \leftarrow UpdateDQN(P, \lambda, reward)$; //更新DQN,见算法3
21. End While
22. If 满足计算资源更新条件 then
23. $\Pi \leftarrow UpdateUtilities(P, \lambda, SP, Z)$; //利用公式(8)(9)(10)更新各子群效用值
24. $sub_FE \leftarrow UpdateSubFE(\Pi)$; //利用公式(11)更新各子群计算资源的分配
25. $Obj \leftarrow P.object$; //更新 Obj 目标值
26. End If
27. End While
28. Return P

3.2 自适应协作子种群选择策略

在算法的优化过程中,每次迭代生成交配池时父代解的选择是超多目标优化算法产生较优解的一个关键问题。为了有效引导大规模决策空间的搜索,提高算法的搜索效率和跨场景适应能力,本文采用基于DQN的自适应子种群选择策略,在每次迭代中选择子种群,并与依据子种群的效用值选择的解共同形成交配池,实现多种群协同进化,以更好地生成更好的子代解。

3.2.1 基于DQN的自适应子种群选择策略

确定给定任务的状态、动作和奖励是使用强化学习的关键问题,以便让Agent更明确地与环境进行交互,做出更优的决策。为了引导种群在每次迭代过程中产生兼顾多样性和收敛性的更好子代,该方法采用基于分解的MOEAs框架,从而计算了聚合函数值的改进。这里,解的适应度值计算采用Tchebycheff聚合函数:

$$\text{Minimize } g^{rch}(x, \lambda^x, Z^*) = \max_{1 \leq i \leq M} \lambda_i^x (f_i(x) - z_i^*) \quad (2)$$

其中 $\lambda^x = (\lambda_1^x, \dots, \lambda_M^x)$ 是解 x 对应的参考向量, $Z^* = (z_1^*, \dots, z_M^*)$ 为由总体中的最小目标值组成的理想点。

基于该适应度值计算函数, 后代解 x 在生成后与父代解 y 进行比较, 解 x 对解 y 的适应值改进率为:

$$FI(x, y) = \max \left\{ 1 - \frac{g^{rch}(x, \lambda^y, z^*)}{g^{rch}(y, \lambda^y, z^*)}, 0 \right\} \quad (3)$$

其中, λ^y 表示 y 的权重向量。

然而, 后代解可以替换父代解集中不止一个的解, 因此其总适应度改善 SFI_x 进一步计算为被后代解替换的父代解集 Y 的适应度改善之和:

$$SFI_x = \sum_{y \in Y} FI(x, y) \quad (4)$$

其中, Y 为被 x 替换的解的集合。

此外, 为了进一步提高算法的搜索效率, 算法定义一个包含记录 (sp, SFI_x) 的先进先出队列 Rq , 保存给定时间窗内子种群更新解的 SFI , 其中 SFI_x 是由子种群 sp 生成的后代解 x 的适应度改善。因此, 在进化操作生成子代解集后, 可采用如下方式计算子种群 sp 的奖励值 $reward$:

$$reward_{sp} = \max_{(sp, SFI_x) \in R} SFI_x \quad (5)$$

即子种群 sp 的 $reward$ 可定义为队列 R 中保存 SFI 的最大值。由于普遍认为罕见但重大的改进比频繁且小的改进更重要。因此, 将最大值而不是平均值视为奖励, 也使得每个子种群的奖励可以在一段时间内保持稳定。所提出的信用分配策略如算法 2 中所示。

一旦产生子代解, 计算奖励后, 将元组 $(\{p, \lambda^p\}, sp_k, reward_{sp_k}, \{x, \lambda^x\})$ 保存在另一个经验回放队列 T , 其中 $\{p, \lambda^p\}$ 为当前解 p 及其参考向量 λ^p , sp_k 为所选择的子种群, $\{x, \lambda^x\}$ 为生成的子代解 x 及其参考向量 λ^x 。即深度强化学习 DQN 的状态由解的决策变量及其参考向量组成 (即 $s = (x_1, \dots, x_D, \lambda_1, \dots, \lambda_M)$), 动作作为所选择的子种群 (即 $a = sp_k$)。

算法 2. 信用分配策略

输入: P (当前种群), x (子代解), sp (选择的子种群), λ (关联的参考向量), Y (被 x 替换的解集), Rq (奖励队列)

输出: P (更新后的种群), Rq (更新后的奖励队列), $reward_{sp}$ (选择子种群的奖励)

1. $SFI_x = 0$; // 初始化 SFI_x

2. For each $y \in Y$ do
3. $FI(x, y) \leftarrow \text{UpdateFI}(x, y, \lambda)$; // 利用公式(2)(3)计算适应度提升值
4. $SFI_x = SFI_x + FI(x, y)$
5. $\text{Update}(P, x, y)$; // 更新种群
6. End For
7. If $\text{length}(Rq) > \text{最大阈值}$ then
8. $\text{DeQueue}(Rq, \text{the oldest})$; // 删除 Rq 中队列首元素
9. End If
10. $\text{EnQueue}(Rq, (sp, SFI_x))$; // 将 (sp, SFI_x) 加入 Rq 队列中
11. $reward_{sp} \leftarrow \text{UpdateReward}(SFI_x)$; // 公式(5)计算奖励值
12. return $P, Rq, reward_{sp}$;

算法的原理如图 4 所示, 深度强化学习 DQN 将 $(\{p, \lambda^p\}, sp_k, reward_{sp_k}, \{x, \lambda^x\})$ 作为输入数据, 将动作 a 设置为所选择的子种群 sp_k 。算法进化过程中, 通过训练深度强化学习模型 DQN 来自适应选择子种群, 提高算法搜索的效率。

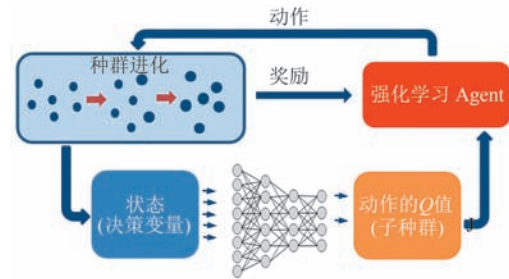


图4 DQNMaOEA算法的深度强化学习模型DQN

同时, 采用损失的梯度下降方法训练 DQN:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} (Q(s_i, a_i) - q_i)^2 \quad (6)$$

其中, \mathcal{T} 为训练集, $Q(s_i, a_i)$ 是神经网络在输入 s_i 时第 a_i 个输出的神经元。 q_i 是在状态 s_i 时动作 a_i 的 Q 值, 计算如下:

$$q_i = r_i + \gamma \max_{a' \in \mathcal{A}} Q(s_{i+1}, a') \quad (7)$$

其中, $\gamma \in [0, 1]$ 是折扣因子, $\max_{a' \in \mathcal{A}} Q(s_{i+1}, a')$ 为下一个动作集 \mathcal{A} 中最大的 Q 值。

基于以上陈述, 我们采用了一个简单的反向传播深度神经网络作为 DQN, 该 DQN 架构的详细参数设置如表 1 所示。

当 DQN 被训练或更新时, 将执行算法 3 的过程。基于经验回放池 T , 随机采样小批量经验形成训练样本 \mathcal{T} 。其中, 如表一中批量大小计算公式所

表1 DQN架构采用的详细参数设置

参数	值
DQN隐藏层数: <i>Layer</i>	<i>Layer</i> = 5
DQN输入层节点数: <i>inDim</i>	<i>inDim</i> = $\text{length}(\{p, \lambda^p\})$
DQN输出层节点数: <i>outDim</i>	<i>outDim</i> = $\text{length}(SP)$
经验回放池大小: <i>Capacity</i>	<i>Capacity</i> = 512
训练数据批量大小: <i>batch_size</i>	<i>batch_size</i> = $8 * \text{num}(\text{Update_Parent})$
学习率: <i>LR</i>	<i>LR</i> = 0.1
网络更新总频次: <i>Update_iter</i>	<i>Update_iter</i> = $0.8 * \text{MaxEpoch}$
激活函数	ReLU

示, *batch_size* 根据本次迭代中生成的子代解替换父代解的数目进行调整计算而得。随后, 通过(6)-(7)计算损失函数并利用反向传播来训练一个深度神经网络, 该神经网络可用于在每次迭代中选择交配池的子种群。

算法3. DQN训练方法

输入: *T*(经验回放队列), *Q*(训练的网络)

输出: *Q'*: (更新后的网络)

1. 从经验回放池 *T* 中随机采样 *batch_size* 批量经验 *T*;
2. If 满足更新参数条件 then
3. $q_t = r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a')$; // 计算目标 *Q* 值
4. $q_{eval} = Q(s_t, a_t; Q)$; // 计算预测 *Q* 值
5. $L = \frac{1}{|T|} \sum_{t \in T} (Q(s_t, a_t) - q_t)^2$; // 计算损失函数
6. $Q' = \text{DLUpdate}(Q, L, \text{grad})$; // 采用损失的梯度下降方法反向传播更新原网络 *Q* 中的参数
7. End If
8. Return *Q'*

在生成交配池之前, 根据输入的状态 $\{p, \lambda^p\}$ 计算每个子种群的 *Q* 值, 然后通过轮盘赌法, 根据它们的 *Q* 值选择适合的子种群。需要注意的是, 如果一个子种群在经验回放池 *T* 中不存在, 它将不能再被选择; 在这种情况下, 子种群将被强制选择, 而不是通过轮盘赌选择。本文提出的基于深度强化学习的自适应子种群策略的伪代码如算法4所示。

算法4. 基于深度强化学习的自适应子种群策略

输入: *P*(当前种群), λ (关联的参考向量), *SP*(候选子种群集合), *T*(经验回放队列), *Q*(训练的网络)

输出: *MatingPool_SP*(选择的交配池)

1. $state = \{p, \lambda^p\}$;
2. $Qvalue \leftarrow \text{Zeros}(1, |SP|)$; // 初始化 *Q* 值
3. For each $sp_i \in SP$ do
4. If $(\bullet, sp_i, \bullet) \notin T$ then
5. return sp_i ;

6. Else
7. $Qvalue_{sp_i} \leftarrow Q(state, sp_i)$;
8. End For
9. $sp_k \leftarrow \text{RouletteWheelSelect}(Qvalue, SP)$; // 轮盘赌选择
10. $MatingPool_SP \leftarrow sp_k$;
11. return *MatingPool_SP*;

3.2.2 基于效用值的子种群选择策略

为了衡量子种群优化性能, 算法定义子种群效用值向量 $\pi = [\pi_1, \pi_2, \dots, \pi_N]$, 其中 $\pi_i \in [0, 1]$ ($i = 1, 2, \dots, N$) 表示子种群 sp_i 的优化性能, 计算过程如下:

首先, 计算各个子种群 sp_k 的适应度改进率 FIR_{SP_k} :

$$FIR_{sp_k} = \frac{g^{ich}(best_{sp_k}, \lambda_{sp_k}, z^*) - g^{ich}(best'_{sp_k}, \lambda_{sp_k}, z^*)}{g^{ich}(best_{sp_k}, \lambda_{sp_k}, z^*)} \quad (8)$$

其中, $best_{sp_k}$ 是子种群父代的最优解, $best'_{sp_k}$ 是子种群子代的最优解。

进而得到 *N* 个子种群的适应度改进率最大值:

$$MAXFIR = \text{Max}(FIR_{SP_i} | i \in \{1, \dots, N\}) \quad (9)$$

最终计算得到每个子种群的效用值, 以衡量子种群对当前整个种群的贡献:

$$\pi_k = \frac{0.9FIR_{SP_k} + 0.1}{MAXFIR} \quad (10)$$

基于公式(8)、(9)、(10), 计算得到每个子种群的效用值, 然后以一定概率通过轮盘赌法, 根据各子种群的效用值选择或从整体种群中随机选择的子种群。同时, 为了避免算法陷入局部最优, 以一定概率抽取随机个体, 共同生成协同父代子种群 *P_Value*, 具体策略过程如算法5所示:

算法5. 基于效用值的子种群解策略

输入: *P*(当前种群), Π (各子种群效用值), *SP*(子种群)

输出: *MatingPool_value*(生成的交配池)

1. $P_value = \text{RouletteWheelSelect}(SP, \Pi)$; // 根据子种群效用值 Π , 利用轮盘赌法生成 *P_value*
2. While operation no. $\leq P_value.length$ do
3. If $\text{rand} < \zeta$ then
4. $p_value \leftarrow P_value(\text{operation no.}++)$;
// 从 *P_value* 子种群中选取一个父代解
5. Else
6. $p_value \leftarrow \text{Random}(P)$; // 从整体种群中随机选取一个父代解
7. End If
8. $MatingPool_value \cup p_value$

9. End While

10. return *MatingPool_value*

各子种群的效用值可以衡量子种群优化性能,量化子种群对当前整个种群的贡献,因此,根据子种群的效用值,利用轮盘赌法,更好地选择能产生更优子代解的子种群进入交配池。值得注意的是,为避免优化过程陷入局部最优,算法以一定概率从当前整体种群选取一个随机解加入父代解集 *MatingPool_value* 中,在保证收敛性的同时,扩大种群的多样性。

3.2.3 多种群协同进化策略

如图5所示,基于分解的思想,利用参考向量形成

并引导多个子种群。在算法进化过程中,根据效用值子种群选择策略产生的解集与DQN选择的子种群生成交配池,协同进化生成兼顾收敛性与多样性的子代解集,以更新当前解集。协同进化过程如图6所示。

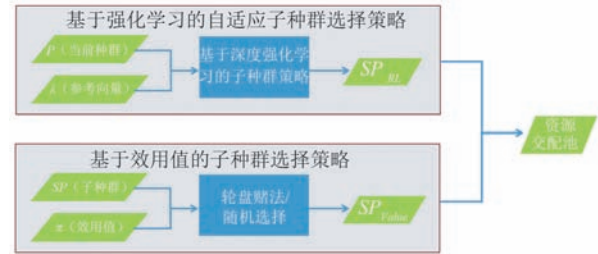


图5 多种群协同优化示意图

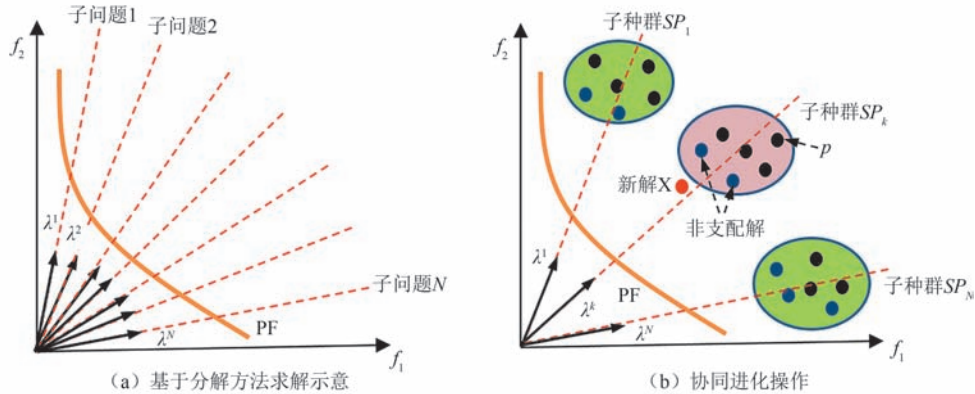


图6 DQNMaoEA算法的多种群协同优化示意图

根据所描述的方法,其信用分配策略和两种子种群选择策略相较于现有方法具有以下优势:

一般而言,现有信用分配策略将奖励设置 q_t 为对历史奖励的考虑,即 $r_t + \gamma q_{t-1}$,而新提出的策略将奖励设置为对未来奖励的考虑,即 $r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a')$ 。由于选择的子种群是在生成子代解时起作用,而不是对过去进化过程,因此新策略更有前景。当然,在优化过程中,未来回报是未知的,因此需要通过训练一个深度神经网络来推断。

现有的子种群选择策略均是根据具体问题设计并基于经验选择相关策略,所提出的基于深度强化学习的选择策略,利用深度强化学习与环境进行交互,通过向网络输入一个解及其权重向量来计算各子种群的 Q 值,基于 Q 值自适应选择子种群。也就是说,所提出的策略在进化过程中考虑了待解决问题的自身特点以及当前解的状态,可以为不同的实例与当前解集,在协同进化过程中建议不同的子种群选择,为子种群选择提供了更为精细和变通的策略。

所定义的效用值计算方法,可以有效地衡量各子种群的优化性能。因此,采用基于效用值的协作子种群策略,可以对当前整个种群的贡献更高的子种群,以产生更好的子代解。同时,为了避免种群进化陷入局部最优,以一定概率在其他空间进行搜索,同时兼顾种群进化的收敛性和多样性。将基于深度强化学习与基于效用值选择策略结合,形成最终的交配池,利用协同进化方法,让算法在泛化性能、可扩展性、多样性、收敛性以及搜索效率等方面都得到了很大的提升。

3.3 自适应子种群计算资源分配策略

由于各子种群对整体种群进化过程中所提供的贡献度不同,如果对贡献度低的子种群进行相同的计算资源分配,在每次迭代中进行相同次数的交叉变异、适应度评估以及环境选择等相关进化操作,会大大降低算法的整体搜索效率。因此,根据各子种群的搜索进展以及对整体种群进化的贡献度,对其资源分配进行调整是有很必要性的。自适应子种群计算资源分配可以使算法具有更好的灵活性和适应性,从而更好地满足问题的需求。同时,通过合理分

配计算资源,可以使得每个子种群都能够充分利用资源进行搜索和优化,从而加速整个算法的收敛速度和搜索效率。此外,自适应资源分配也可以提高算法的鲁棒性和稳定性。在面对不同的问题实例和不同的目标函数时,自适应分配可以使得算法更加鲁棒和稳定,从而更好地适用于多样化的问题领域和目标函数。

综上所述,自适应分配各子种群的计算资源可以帮助多种群协同进化多目标算法更好地适应多样化的问题和目标函数,提高算法的搜索效率、收敛速度和稳定性。

本文提出一种自适应子种群计算资源分配策略,基本思想是根据每个子种群对当前整个种群优化的贡献,动态分配子种群适应值评估次数,其具体分配过程如下:

(1)各子种群的效用值 II 可以有效地衡量各子种群的优化性能,因此利用公式(8)、(9)、(10),计算得到每个子种群的效用值 II 。

(2)根据每个子种群对当前整个种群优化的贡献,计算得到每个子种群的计算资源分配。设 $MaxFE$ 为 N 个子种群每轮迭代允许的最大适应值评估次数,子种群适应值评估次数为:

$$FE_{SP_i} = \frac{0.9FIR_{SP_i} + 0.1}{MAXFIR} \times MaxFE \quad (11)$$

如果 $MAXFIR=0$,设置每个子种群的适应值评估次数为 $FE_{SP_i}=MaxFE$ 。

基于上述过程,根据子种群不同的特点、状态和贡献度,最终得到各子种群的资源分配情况。

3.4 算法性能分析

3.4.1 时间复杂度分析

依据算法1所示,DQNMaeOEA的时间复杂度主要由生成每个子代解的5个步骤决定,即子种群选择、子代生成、种群更新、信用分配和神经网络训练。对于神经网络的一个推理过程,基于深度强化学习的子种群选择的时间复杂度为 $O(D^2)$,其中 D 为表示网络层规模的决策变量维度。在每一次进行协同进化操作中,对于利用轮盘赌法的一个选择过程,基于效用值的子种群选择的时间复杂度为 $O(|SP_k|)$ 。对于所有的子种群选择,子代生成的时间复杂度均为 $O(D)$ 。计算聚合函数时,种群更新的时间复杂度为 $O(M)$ 。计算适应度改进的信用赋值时间复杂度为 $O(N)$ 。对于一个反向传播过程,模型训练的时间复杂度为 $O(D^2)$ 。因此,DQNMaeOEA一轮迭代的时间复杂度为 $O(ND^2)$ 。

3.4.2 理论与性能分析

(1)理论基础

本文提出的多种群协同进化策略通过深度强化学习与基于效用值的子种群选择策略的有机结合,形成了“动态适应-历史评估-资源优化”的协同进化机制。其中,DQNMaeOEA的核心思想源自基于分解的多种群协同优化理论以及强化学习的马尔可夫决策过程MDP建模的严谨性。

协同进化策略基于分解式超多目标进化算法框架,通过参考向量将超多目标优化问题分解为多个子问题,每个子种群对应一个参考向量。各子种群聚焦于优化一个由Tchebycheff聚合函数定义的单目标子问题,将高维帕累托前沿的逼近转化为多个子问题的协同优化。这种分解方法在协同进化理论上被证明能有效处理超多目标场景,通过子种群间的信息交互,平衡解的收敛性与多样性。其中,子种群选择作为算法的核心环节,本质上构成子问题优先级的动态调度机制,其策略设计直接影响算法的全局搜索效率与解的不同问题实例求解能力。而强化学习通过环境交互与奖励驱动机制,在复杂决策、动态优化及超多目标协同等场景中展现出显著性能优势。基于环境反馈的深度强化学习驱动自适应策略,根据问题特征自适应调整,优先挖掘高潜力子问题区域的同时,保留低贡献子问题的探索可能性,进一步提高了算法的求解与跨场景适应能力。

作为强化学习的基础,马尔可夫决策过程是一种用于描述在不确定环境中智能体进行序列决策的数学框架,其核心假设是当前状态包含决定未来的全部关键信息(马尔可夫性),即下一状态和奖励仅依赖当前状态与动作,与历史无关。因此,强化学习任务通常可以规范化为一个马尔可夫决策过程。一个问题若要定义为马尔可夫决策过程,需满足以下核心条件:

①状态的马尔可夫性:当前状态包含了决策所需的全部历史信息,未来状态仅依赖当前状态和所选动作,与更早的历史状态无关。

②序列决策与动态演化:问题需通过多步决策完成,其决策具有时间依赖性,目标是优化长期累计奖励,而非单步最优。环境状态随时间变化,每一步选择动作后,状态按转移概率演化,并获得即时奖励。

③明确的状态与动作空间:状态与动作的可表征性,即状态和动作空间需有限或可处理。其中状态空间需能完整描述环境当前信息的表示,动作空

间则需包含了智能体在每个状态下可执行的动作集合。

④目标导向性:存在可量化的奖励信号,并能够评价动作的长期影响。智能体需学习策略,使得从任意初始状态出发的期望累积折扣奖励目标最大化。

如表2所示,DQN的MDP框架与子种群选择在状态设计、动作空间、奖励函数及探索机制上高度适配。基于DQN的自适应子种群选择策略满足

DQN求解特点:状态信息完备且满足马尔可夫性、离散动作空间与问题特性契合、奖励与目标函数有效引导策略优化,以及种群进化过程符合环境动态演化特点,DQN通过与种群进化环境的交互学习,渐近优化子种群选择策略。根据强化学习收敛定理,当奖励有界且状态转移可估计时,通过渐近一致探索和合理学习率,DQN的Q值函数逼近能够实现策略的渐近优化,为算法的科学性提供了理论保证。

表2 DQN架构采用的详细参数设置

MDP要素	具体实现	合理性分析
状态	$s=\{p, \lambda^p\}$	状态信息全面刻画了当前各子种群状态以及优化进程的多样性与收敛性。 根据当前种群生成新种群的进化过程仅依赖当前种群分布,满足马尔可夫性
动作	$a=sp_k$	子种群选择本质是离散决策,动作空间离散且规模可控,适合DQN处理离散动作的特性
奖励	$reward_{sp}=\max_{(sp, SFIx) \in R} SFI_x$	奖励直接量化子种群对种群优化的贡献,最大化累计奖励目标有效引导DQN优先选择高效子种群
状态转移	$\{p, \lambda^p\} \rightarrow \{p', \lambda^{p'}\}$	环境动态变化。子种群选择后,通过进化算子生成后代解,更新种群并重新关联参考向量,形成新的状态

(2)性能分析

DQNMaOEA通过深度强化学习与多种群协同进化的深度融合,在收敛性、多样性、效率、鲁棒性等方面展现其性能优势。此外,模块化设计和自适应机制使其具备强大的跨场景适应能力与可扩展性,适用于复杂、动态、高维的超多目标优化问题。

①收敛性分析

自适应子种群选择策略加速收敛:由于DQN能够学习到解的状态与子种群选择之间的复杂映射关系,引导算法朝着更优的方向搜索。因此,基于DQN的子种群选择策略能够根据当前解的状态信息自适应地选择具有潜力的子种群进行进化。在迭代过程中,DQN可以有效识别那些能使解快速趋近帕累托前沿的子种群,在更少的迭代次数内让解更靠近真实的帕累托前沿。

奖励机制聚焦有效搜索方向:算法采用以子代解对父代解的适应度改进为核心的奖励机制,通过计算总适应度改进来衡量子代解的优劣,并将其作为DQN学习的奖励信号,使得DQN能够动态选择对收敛贡献最大的子种群。这种机制使得算法能够集中探索那些能够带来实际优化效果的区域,避免在无效的搜索空间中浪费计算资源,从而显著加快了收敛进程。在复杂的超多目标优化中,该奖励机制能帮助算法有效跳出局部最优,朝着全局最优解前进。

②多样性分析

双策略协同保障多样性:DQNMaOEA结合了

基于DQN的子种群自适应选择策略和基于效用值的子种群选择策略。DQN策略侧重于利用学习经验并利用基于Q值的轮盘赌法选择最优子种群进行开发,以兼顾解的收敛性与多样性;而基于效用值的策略则以一定的概率随机选择子种群或利用基于效用值的轮盘赌法进行探索,有效平衡探索与开发,维持多样性。这种双策略协同的方式在维持算法收敛性的同时,有效增强了解集的多样性。

参考向量分解实现解空间覆盖:算法运用均匀分布的参考向量将超多目标优化问题分解为多个单目标子问题,并关联对应的子种群,确保种群在目标空间中的均匀分布。其中,每个子问题对应一个子种群。同时,每个子种群关联对应参考向量,各子种群通过优化其对应的子问题,推动整体种群逼近Pareto前沿。这种分解方式确保了算法能够在目标空间的各个方向上进行搜索,使得解集在目标空间中实现均匀覆盖,进一步提升了解集的多样性。对于具有复杂目标空间结构的问题,参考向量分解能够引导算法探索到更多不同的有效解。

③求解效率分析

自适应资源分配降低计算成本:算法根据子种群的效用值动态分配计算资源,即对那些在进化过程中表现出更高适应度改进能力的子种群分配更多的计算资源,而对表现较差的子种群减少资源分配。这种自适应资源分配策略有效避免了对所有子种群进行资源平均分配的盲目性,大大减少了不必

要的计算,从而降低了整体的计算成本。

经验回放与批量训练提升学习效率:算法采用经验回放和批量训练的方法来训练 DQN。经验回放机制将智能体的经验存储在经验回放池中,并在训练时对其进行随机采样学习,这样可以打破数据之间的相关性,提高学习的稳定性和效率。而批量训练则可以充分利用计算资源,并行处理多个样本,加速了 DQN 的训练过程。通过这些技术,DQN 能够更快地学习到最优的子种群选择策略,从而提高整个算法的求解效率。

④鲁棒性分析

边界条件的鲁棒处理:资源分配失效处理,即当所有子种群的适应度改进率为零时,算法自动切换为均匀资源分配,避免因贡献度计算失效导致的搜索停滞。未探索子种群的强制选择,即若子种群未在经验回放池中出现,直接强制选择该子种群进行探索,确保所有子种群至少被评估一次,有效避免了因初始探索不足导致的策略偏差问题。

避免局部最优陷阱:由于算法结合了探索和开发策略,并且在目标空间的各个方向上进行搜索,因此能够有效地避免陷入局部最优陷阱。在处理复杂的超多目标优化问题时,DQNMaOEA 能够找到多个不同的帕累托最优解,为决策者提供更多的选择。

⑤通用问题求解能力分析

深度强化学习的策略学习能力:算法通过深度强化学习的方式学习子种群选择策略,能够在环境变化时迅速做出调整,保证系统的高效运行,使得其在不同的问题实例上具有较好的求解能力。DQN 在训练过程中学习到的是解的状态与子种群选择之间的通用模式,而不是针对特定问题的局部最优策略。因此,当面对新的超多目标优化问题时,算法能够快速调整并找到有效的解决方案。

通用问题抽象与状态编码:算法基于参考向量将超多目标问题分解为多个子问题,该框架不依赖具体目标函数形式,适用于不同类型的超多目标优化问题。此外,DQN 的输入状态为解的决策变量与参考向量,通过结构化编码,有效捕捉问题的本质特征,而非依赖特定问题的局部细节。同时,基于子代解对父代解的相对改进奖励函数设计,通过非绝对性能指标的相对度量避免了对特定问题尺度的依赖。

⑥可扩展性分析

可演进的技术架构:按照功能模块对算法进行组件划分,可将算法分为多个独立的组件,分析每个

组件的功能、可扩展性以及它们之间的交互方式,从而构建出可灵活扩展的超多目标优化框架,其核心优势在于各核心组件的独立可替换性与协同机制的稳定性。根据其功能的不同,大致可划分为问题定义、种群初始化、选择策略、进化操作、环境选择以及资源分配组件。这种基于功能解耦、组件独立、接口标准化的设计不仅为现有超多目标优化提供了高效解决方案,更构建了可演进的技术架构,使算法能够随着强化学习、进化计算的前沿进展持续升级,保持长期的技术竞争力。

可扩展性:在问题规模扩展方面,算法能够轻松扩展到处理更多目标和更高决策变量维度的超多目标优化问题。只需相应调整参考向量的数量和分布,以及 DQN 网络的输入与输出维度,就能继续有效地进行优化。在应用领域扩展方面,适用于多个不同的行业领域,如工程设计、能源管理、智能交通等。在不同领域中,可根据具体问题特点选择合适的组件组合和参数设置。还能与其他技术融合,如机器学习、深度学习中的其他先进网络模型与技术以及前沿优化算法等。在算法结构扩展方面,可以扩展为多智能体协同的形式,多个智能体分别负责不同的子种群或优化任务,通过协同合作提高整体优化性能。也可以与其他优化策略相结合,形成混合优化算法,在算法不同阶段采用不同策略或在局部区域采用传统优化算法进行精细搜索等。

4 性能评估

在本节中,首先介绍实验中的基本参数设置。同时,为了验证本文提出的算法 DQNMaOEA 以及相关策略的有效性,将从以下几个方面进行对比实验验证:

(1)基于深度强化学习的自适应子种群选择策略的验证;

(2)自适应子种群计算资源分配策略的验证;

(3)多种群协同优化方法的验证;

(4)算法 DQNMaOEA 总体性能比较:a)对比算法在 IGD 指标上的表现;b)对比算法在 HV 指标上的表现;c)对比算法在测试实例上的最终种群分布效果图;

(5)算法 DQNMaOEA 在实际问题实例上的验证。

4.1 实验设计

本文主要选用 DTLZ、WFG 以及 MaF 系列测

试问题,并在5、10、12目标数量条件下进行对比实验。在目标数量为 M 、决策变量维度为 D 的DTLZ系列测试问题中,对于DTLZ1,设置 $D=M+4$;对于DTLZ2-DTLZ6,设置 $D=M+9$;对于DTLZ7,设置 $D=M+19$ 。对于目标数为 M 的WFG系列问题,设置WFG1-WFG7的决策变量维度 $D=M+9$ 。在MaF系列测试问题中,当目标数量为 M 、决策变量维度为 D 时,对于MaF1-MaF6,设置 $D=M+9$;对于MaF7,设置 $D=M+19$ 。同时,在各目标数量下设置不同的种群数量,其中5目标的种群数量为100,10、12目标的种群数量为275。在本文中,使用种群最大评估次数 $MaxFE$ 作为算法的结束条件,所有算法在不同测试问题以及目标数量的情况下,均将种群最大评估次数设置为100 000,且每个测试实例运行30次。最后,我们使用两个广泛使用的综合评价指标来评估算法性能,即超体积指标(Hypervolume Metric, HV)和反转世代距离指标(Inverted Generational Distance, IGD)。

如表3所示,本文选取了目前最新且具有代表性的超多目标优化算法、多种群协同进化算法以及使用强化学习的超多目标优化算法等作为实验的对比算法,分别包括:MOEA/D-DQN^[70]、CMOCSO^[71]、DKCA^[72]、MOEA-NZD^[73]、NRV-MOEA^[74]、TriP^[75]、MaOEA/SRV^[76]、DRLOS-EMCMO^[77]、APSEA^[78]和MaOEA-PDE^[79],以验证算法DQNMaOEA的有效性。实验所涉及的算法均使用MATLAB实验,并嵌入到进化多目标优化平台PlatEMO^[80]中。

表3 对比算法总览		
类型	年份	算法
非学习型超多目标优化算法	2022	CMOCSO ^[71]
	2022	TriP ^[75]
	2024	DKCA ^[72]
	2024	MaOEA-PDE ^[79]
	2025	APSEA ^[78]
基于学习的超多目标优化算法	2022	MaOEA/SRV ^[76]
	2023	MOEA/D-DQN ^[70]
	2024	MOEA-NZD ^[73]
	2024	NRV-MOEA ^[74]
	2024	DRLOS-EMCMO ^[77]

此外,除了对比较算法的统一设置外,DKCA、TriP、MaOEA/SRV以及APSEA需要自定义的参数如表4所示。其中,在本文3.3.2小节提到,在基

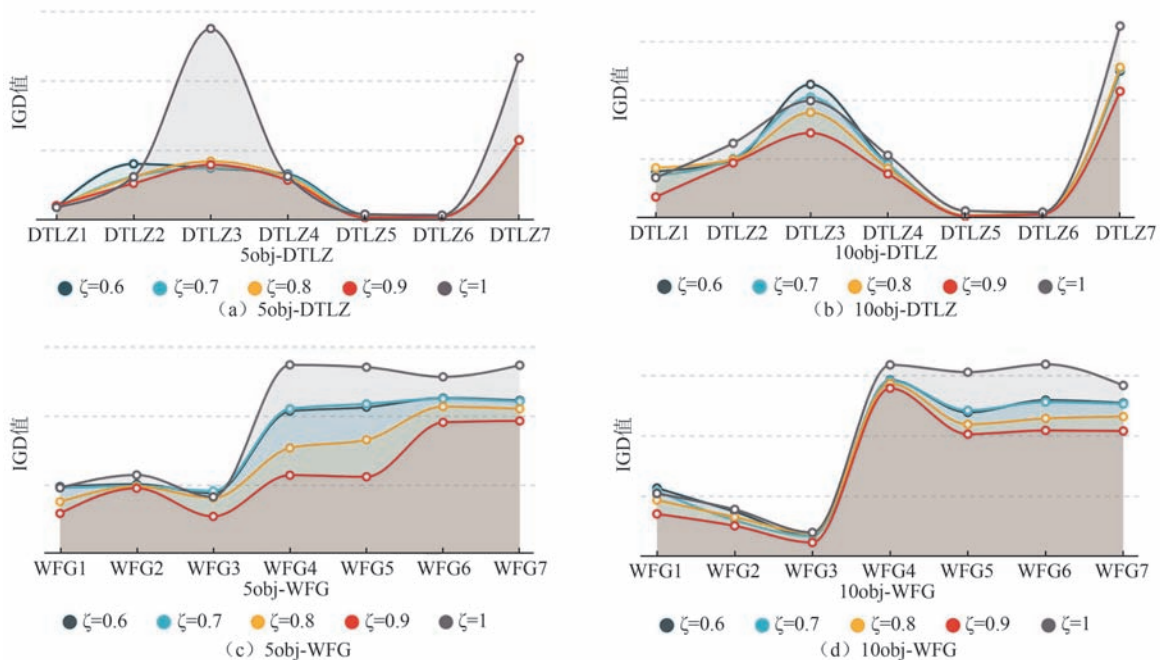
于效用值的子种群选取策略中,为避免陷入局部最优,算法以一定概率从当前整体种群随机选取一个解加入父代子种群中。为进一步确定该概率超参数 ζ 的最优设置,在保证算法整体框架与关键策略不变的情况下,进行实验对比了算法在 $\zeta=0.6、0.7、0.8、0.9、1$ 五种不同超参数设置下的性能。图7展示了各超参数值算法在求解5、10目标DTLZ与WFG系列问题上的IGD值对比情况。结果表明,随着 ζ 逐渐增大,其IGD值大致呈现先降后升的趋势。同时,在涵盖4组基准问题实例中, $\zeta=0.9$ 在绝大多数场景下均展现出显著优势。基于上述对比分析,最终确定 $\zeta=0.9$ 为该策略的最优参数配置,其平衡探索与开发的能力可有效适配多样化问题特性。

表4 实验参数设置	
参数	值
DKCA进行降维后所选决策变量比例: t	$t=0.3$
DKCA进行种群稀疏度评价的代数: k	$k=4$
在TriP中局部选择父母的概率: δ	$\delta=0.9$
TriP中父代解被各子代解替换的最大数目: nr	$nr=2$
在MaOEA/SRV中触发SRV的迭代轮数间隔: τ	$\tau=1$
APSEA种群1中可行解的最小占比: α	$\alpha=0.05$
APSEA种群1中目标的最小变化率: β	$\beta=0.05$
APSEA的动态边界下降趋势: cp	$cp=5$
DQNMaOEA基于效用值选择策略中概率超参数: ζ	$\zeta=0.9$

4.2 基于深度强化学习的自适应子种群选择策略的有效性验证

为验证所提出的基于深度强化学习的自适应子种群选择策略的有效性,在不修改DQNMaOEA环境选择、效用值更新、计算资源分配等关键方法的情况下,设计实验比较了算法采用不同的子种群选择策略的性能,包括DQNMaOEA_R使用随机选择子种群策略、DQNMaOEA_V使用基于效用值的子种群选择策略、DQNMaOEA_B使用最优目标值的子种群选择策略以及DQNMaOEA使用本文提出的子种群选择策略。

算法分别在5、10目标DTLZ1-7上分别运行,并对IGD结果进行比较,如表5所示。表中用平均值和标准差来衡量算法性能和稳定性,并在三线表中记录算法实验结果,采用“平均值(标准差)+/-/ \approx ”的形式表示。加粗突出每个测试问题中最好的结果,并统计整个测试问题集的总体性能差异并采用Wilcoxon秩和检验方法来估计算法的差异。实验结果统计表中的符号“-”表示本文给出

图7 概率超参数 ζ 的最优设置实验对比图

的算法明显强于其他算法,而符号“+”表示所比较的算法强于本文算法,而符号“ \approx ”表示两种算法之间没有显著差异。

通过表5的结果可知,在仅修改了子种群选择策略的情况下,DQNMaOEA算法比其余三种固定策略算法总体上表现更佳。在12个测试实例中的

IGD值取得了10个最优结果,特别是在PFs为均匀分布的情况下表现优秀,仅在不规则Pareto前沿问题的5目标DTLZ7、10目标DTLZ1两个测试实例中结果次优。综上所述,结合实验结果比较,DQNMAOEA总体表现更优,因此验证了本文提出的自适应子种群选择策略的有效性。

表5 自适应子种群选择策略有效性实验结果

问题	M	D	DQNMaOEA_B	DQNMaOEA_R	DQNMaOEA_V	DQNMaOEA
DTLZ1	5	9	2.0537e-1 (2.47e-1) -	1.2247e-1 (4.91e-3) -	1.4685e-1 (1.64e-1) -	9.9656e-2 (1.69e-3)
	10	14	2.2677e-1 (2.87e-2) -	1.2747e-1 (5.26e-3) +	2.1464e-1 (2.39e-1) -	1.7429e-1 (1.40e-1)
DTLZ2	5	14	3.4296e-1 (2.80e-2) -	2.9698e-1 (5.26e-3) -	3.8947e-1 (1.69e-2) -	2.5972e-1 (5.24e-3)
	10	19	7.1659e-1 (6.14e-2) -	4.9088e-1 (1.93e-2) -	6.1395e-1 (2.68e-2) -	4.8791e-1 (2.66e-2)
DTLZ3	5	14	6.6053e-1 (1.56e-1) -	3.9596e-1 (1.93e-2) -	4.0236e-1 (1.56e-2) -	3.0352e-1 (2.66e-2)
	10	19	1.0039e+0 (8.82e-2) -	1.0466e+0 (8.14e-1) -	1.3655e+0 (7.92e-1) -	5.1865e-1 (5.50e-2)
DTLZ4	5	14	4.1555e-1 (7.38e-2) -	3.0836e-1 (1.51e-3) -	3.7901e-1 (9.77e-2) -	2.8279e-1 (5.89e-3)
	10	19	5.2260e-1 (1.05e-1) -	4.4184e-1 (1.14e-1) -	4.1966e-1 (3.03e-2) -	3.7017e-1 (4.04e-3)
DTLZ5	5	14	2.4123e-2 (3.91e-3) -	3.7544e-2 (2.20e-3) -	3.0682e-2 (7.83e-4) -	1.3673e-2 (2.15e-3)
	10	19	5.1863e-2 (2.97e-3) -	3.7573e-2 (5.36e-3) -	2.6275e-2 (4.79e-3) -	1.0994e-2 (1.54e-3)
DTLZ6	5	14	1.2844e-1 (2.04e-1) -	1.1781e-1 (2.18e-2) -	9.1385e-2 (2.75e-2) -	1.9705e-2 (5.60e-3)
	10	19	3.5357e-2 (1.49e-2) -	1.9083e-1 (5.97e-2) -	3.9709e-2 (1.85e-2) -	2.7284e-2 (7.77e-3)
DTLZ7	5	24	6.1991e-1 (4.12e-2) -	5.7077e-1 (7.46e-3) =	5.4930e-1 (4.54e-2) +	5.7048e-1 (6.53e-4)
	10	29	1.3213e+0 (4.42e-2) -	1.2685e+0 (2.33e-2) -	1.1519e+0 (3.78e-2) -	1.0727e+0 (2.00e-2)
+/-/ \approx			0/14/0	1/12/1	1/13/0	

4.3 自适应子种群计算资源分配策略的有效性验证

本文提出的自适应子种群计算资源分配策略,

通过计算子种群的效用值,衡量各子种群对当前整个种群优化的贡献,进而基于此合理分配子种群的计算资源,以提高算法的稳定性、鲁棒性和搜索效

率。为了验证自适应资源分配操作的性能,本文将比较DQNMaOEA在采用本文提出的资源分配策略和不添加资源分配策略条件下的结果,并将其记为DQNMaOEA_non。实验分别对比并分析了在5和10个目标DTLZ问题上的IGD指标以及运行时间。

表6比较了添加资源分配操作的DQNMaOEA和不添加资源分配操作的DQNMaOEA_non在DTLZ测试问题上的IGD结果。实验结果显示,DQNMaOEA在十二个测试实例中的IGD值上取得了9个最优结果。在其余测试实例中,虽然没有在IGD值上取得最佳性能表现,但是最终种群的分

布接近真实Pareto前沿。此外,图8可视化了两个算法在目标数为5、10的DTLZ测试问题下的运行时间对比。结果表明本文提出的算法DQNMaOEA在维持较优的搜索效果的情况下,通过自适应资源分配策略降低算法的运行时间,进一步优化了算法的搜索效率。同时,实验对比了DQNMaOEA与其他算法在求解10目标下DTLZ系列问题的平均运行时间,并将对比结果在图9中进行可视化展示,结果表明,本文提出的算法在求解不同问题时,求解速率较优,算法效率与性能相对稳定,进一步验证了自适应子种群计算资源分配策略在提高整体算法求解效率方面的有效性。

表 6 自适应子种群计算资源分配策略有效性实验结果

问题	M	D	DQNMaOEA_non	DQNMaOEA
DTLZ1	5	9	1. 2139e-1 (4. 30e-3) -	9. 9656e-2 (1. 69e-3)
	10	14	1. 2235e-1 (6. 33e-3) +	1. 7429e-1 (1. 40e-1)
DTLZ2	5	14	2. 9042e-1 (1. 73e-2) -	2. 5972e-1 (5. 24e-3)
	10	19	4. 6372e-1 (2. 31e-2) +	4. 8791e-1 (2. 66e-2)
DTLZ3	5	14	3. 9966e-1 (2. 00e-2) -	3. 0352e-1 (2. 66e-2)
	10	19	1. 0664e+0 (6. 33e-1) -	5. 1865e-1 (5. 50e-2)
DTLZ4	5	14	2. 9322e-1 (1. 40e-2) -	2. 8279e-1 (5. 89e-3)
	10	19	3. 7017e-1 (4. 04e-3) +	4. 1028e-1 (7. 54e-3)
DTLZ5	5	14	3. 2856e-2 (6. 99e-3) -	1. 3673e-2 (2. 15e-3)
	10	19	6. 9413e-3 (5. 74e-4) +	1. 0994e-2 (1. 54e-3)
DTLZ6	5	14	9. 8155e-2 (3. 68e-2) -	1. 9705e-2 (5. 60e-3)
	10	19	1. 4305e-2 (6. 74e-3) +	2. 7284e-2 (7. 77e-3)
DTLZ7	5	24	5. 9839e-1 (2. 78e-2) -	5. 7048e-1 (6. 53e-4)
	10	29	1. 2393e+0 (2. 55e-2) -	1. 0727e+0 (2. 00e-2)
+/-/≈			5/9/0	

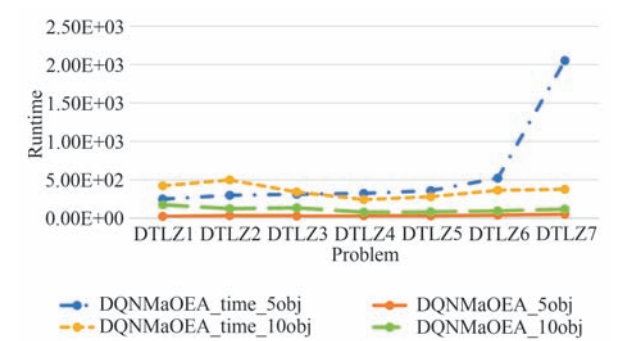


图8 DTLZ下M为5、10的消融实验运行时间对比

4.4 多种群协同优化方法的有效性验证

为验证多种群协同进化方法的有效性,在不修改所提出算法框架中其余重要环节与策略的情况下,设计协同策略消融分析对比实验,对比单一子种群选择策略以及独立种群进化与本文多种群协同优

化策略算法DQNMaOEA在5目标与10目标的DTLZ1-7测试问题上的性能差异。其中,对比算法包括仅通过DQN网络选择协作子种群的DQNMaOEA_I、仅通过基于效用值的子种群选择策略的DQNMaOEA_II,以及双策略子种群独立进化方法的DQNMaOEA_III。

从表7的HV实验数据可见,在14个测试问题中,本文提出的算法DQNMaOEA对比其余算法共获得了13个HV最优。表8则记录了实验结果的IGD值,在14个测试问题中,DQNMaOEA同样取得了13个最优IGD值,仅在10目标的DTLZ1问题上性能略逊。上述结果表明多种群协同优化方法在绝大多数问题上具有统计学意义的整体性能优势。

其中,独立进化方法DQNMaOEA_III因缺乏子种群间的信息交互和资源共享,导致消融后性能大

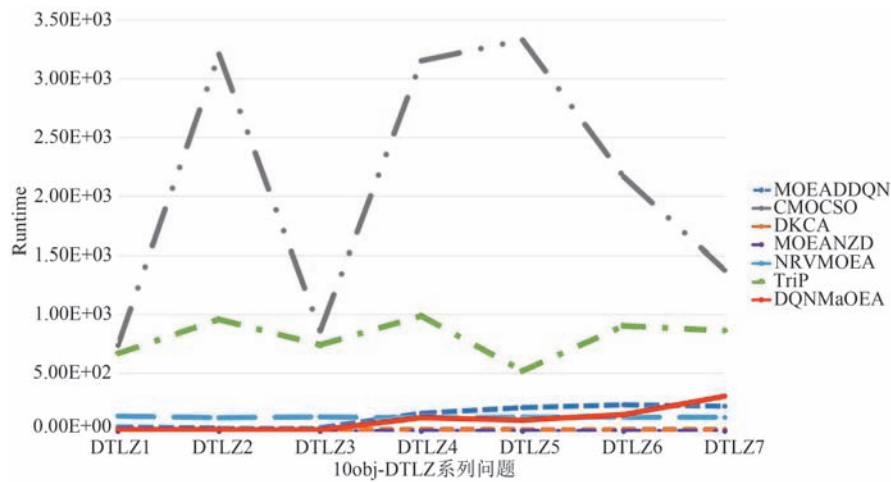


图9 DTLZ下M为10的各算法运行时间对比

表 7 多种群协同进化方法有效性 HV 实验结果

问题	M	D	DQNMaoEA_ I	DQNMaoEA_ II	DQNMaoEA_ III	DQNMaoEA
DTLZ1	5	9	6. 9306e-1 (1. 63e-1) -	7. 1788e-1 (1. 74e-2) -	6. 2960e-1 (2. 31e-2) -	7. 2333e-1 (1. 68e-2)
	10	14	9. 0240e-1 (1. 73e-2) -	8. 3190e-1 (1. 96e-1) -	8. 3117e-1 (1. 33e-2) -	9. 5489e-1 (1. 07e-2)
DTLZ2	5	14	6. 6108e-1 (1. 82e-2) -	6. 5504e-1 (1. 97e-2) -	2. 7588e-1 (2. 32e-2) -	6. 9287e-1 (2. 27e-2)
	10	19	5. 9678e-1 (2. 35e-2) -	5. 8876e-1 (3. 35e-2) -	4. 7381e-1 (2. 52e-2) -	8. 6551e-1 (2. 79e-2)
DTLZ3	5	14	3. 8272e-1 (2. 75e-2) -	3. 8679e-1 (2. 63e-2) -	2. 7507e-1 (1. 68e-2) -	4. 1839e-1 (3. 15e-2)
	10	19	3. 4149e-3 (1. 21e-2) -	2. 0068e-1 (1. 86e-1) -	2. 3680e-1 (1. 49e-1) -	7. 7513e-1 (3. 99e-2)
DTLZ4	5	14	7. 2370e-1 (1. 39e-2) -	7. 1453e-1 (2. 65e-2) -	2. 9420e-1 (1. 87e-2) -	7. 2917e-1 (1. 40e-2)
	10	19	7. 8335e-1 (1. 65e-2) -	7. 6596e-1 (1. 78e-2) -	6. 4603e-1 (1. 81e-2) -	9. 7385e-1 (1. 17e-3)
DTLZ5	5	14	1. 2042e-1 (9. 35e-4) -	1. 2070e-1 (9. 86e-4) -	5. 1112e-2 (1. 58e-2) -	1. 2187e-1 (9. 84e-4)
	10	19	7. 2485e-2 (1. 80e-2) -	6. 0181e-2 (1. 24e-2) -	4. 3374e-2 (2. 28e-2) -	1. 0036e-1 (2. 52e-4)
DTLZ6	5	14	1. 0719e-1 (7. 97e-2) +	1. 0890e-1 (6. 39e-2) +	5. 6494e-2 (1. 77e-2) -	1. 0397e-1 (2. 77e-2)
	10	19	7. 0915e-2 (1. 59e-2) -	7. 6839e-2 (1. 26e-2) -	5. 5847e-2 (3. 09e-2) -	1. 0049e-1 (3. 40e-4)
DTLZ7	5	24	4. 8110e-2 (1. 42e-2) -	4. 5813e-2 (1. 17e-2) -	2. 3945e-2 (3. 06e-2) -	9. 8425e-2 (5. 17e-2)
	10	29	8. 6709e-4 (3. 00e-3) -	2. 3632e-3 (8. 99e-3) -	3. 0552e-3 (4. 22e-3) -	8. 8848e-3 (2. 74e-3)
+/-/≈			1/13/0	1/13/0	0/14/0	

表 8 多种群协同进化方法有效性 IGD 实验结果

问题	M	D	DQNMaoEA_ I	DQNMaoEA_ II	DQNMaoEA_ III	DQNMaoEA
DTLZ1	5	9	1. 2175e-1 (6. 36e-3) -	5. 0436e-1 (1. 95e-1) -	1. 4584e-1 (9. 39e-3) -	9. 9656e-2 (1. 69e-3)
	10	14	1. 5644e-1 (9. 16e-3) +	2. 1464e-1 (2. 39e-1)-	1. 6601e-1 (8. 51e-3) +	1. 7429e-1 (1. 40e-1)
DTLZ2	5	14	3. 1568e-1 (1. 56e-2) -	3. 8947e-1 (1. 69e-2) -	4. 8088e-1 (2. 19e-2) -	2. 5972e-1 (5. 24e-3)
	10	19	5. 9563e-1 (2. 05e-2) -	6. 1395e-1 (2. 68e-2)-	6. 6428e-1 (2. 70e-2) -	4. 8791e-1 (2. 66e-2)
DTLZ3	5	14	3. 9073e-1 (2. 44e-2) -	1. 7117e+1 (4. 63e+0) -	4. 8330e-1 (2. 83e-2) -	3. 0352e-1 (2. 66e-2)
	10	19	2. 7812e+0 (1. 09e+0) -	1. 3655e+0 (7. 92e-1)-	8. 4096e-1 (1. 26e-1) -	5. 1865e-1 (5. 50e-2)
DTLZ4	5	14	3. 1731e-1 (1. 76e-2) -	4. 5334e-1 (6. 10e-2) -	5. 8811e-1 (1. 14e-1) -	2. 8279e-1 (5. 89e-3)
	10	19	5. 9593e-1 (9. 78e-3) -	6. 0911e-1 (1. 91e-2)-	6. 6735e-1 (2. 35e-2) -	3. 7017e-1 (4. 04e-3)
DTLZ5	5	14	3. 9204e-2 (3. 76e-3) -	3. 6708e-2 (3. 85e-3) -	1. 5873e-1 (2. 67e-2) -	1. 3673e-2 (2. 15e-3)
	10	19	5. 7317e-2 (1. 47e-3) -	6. 5463e-2 (1. 33e-3) -	9. 4480e-2 (2. 63e-3) -	1. 0994e-2 (1. 54e-3)
DTLZ6	5	14	1. 0694e-1 (5. 79e-2) -	8. 5444e-2 (3. 66e-2) -	1. 6585e-1 (4. 62e-2) -	1. 9705e-2 (5. 60e-3)
	10	19	5. 3212e-2 (1. 47e-2) -	6. 1960e-2 (1. 82e-2)-	1. 0059e-1 (2. 37e-2) -	2. 7284e-2 (7. 77e-3)
DTLZ7	5	24	6. 5592e-1 (4. 80e-2) -	1. 1793e+0 (2. 23e-1) -	1. 3628e+0 (2. 37e-1) -	5. 7048e-1 (6. 53e-4)
	10	29	2. 3658e+0 (8. 11e-1) -	2. 0860e+0 (5. 40e-1) -	3. 1463e+0 (3. 77e-1) -	1. 0727e+0 (2. 00e-2)
+/-/≈			1/13/0	0/14/0	1/13/0	

幅下降,在几乎所有问题上 HV 与 IGD 值均显著劣于其他算法,且差距随问题复杂度增加而扩大,无法通过其他技术弥补。通过实验验证与理论分析,因其不可替代性,协同进化框架对资源分配和种群进化至关重要,是 DQNMaoEA 的技术核心与性能基石。

对比基于单一子种群选择策略的算法,结果表明单一策略的使用可能导致算法过早收敛,易于陷入局部最优,从而降低算法的收敛性,又因探索不足导致多样性下降。实验有效证明了双策略多种群协同的互补效应平衡了求解的多样性与收敛性,并显著提高了算法的稳定性与跨场景适应能力。此外,将两种不同单一策略的结果进行比较,与 DQN 选择相比,效用值策略的收敛性波动较大,在不同类型的问题上表现差异较大,问题适应性与鲁棒性较差,而基于 DQN 的自适应子种群选择策略在不同问题求解的性能方面显优。进一步说明深度强化学习引导的选择策略通过环境交互选择高潜力子种群,是自适应子种群选择策略的关键使能技术。而效用值评估与随机探索策略作为多种群协同策略的组成部分,则通过与 DQN 策略结合实现其收敛性维护与多样性保障价值。基于深度强化学习环境交互选择与效用值评估,共同形成双策略协同机制,是应对复杂高维目标空间的关键创新。

综上所述,多种群协同进化策略是本文的核心技术,通过信息交互与双策略多种群协同,显著提升了算法的稳定性、探索与开发能力以及复杂场景的适应性,突破了传统超多目标优化算法的性能瓶颈。

4.5 算法总体性能比较

为了验证本文提出的 DQNMaoEA 算法的性能,将 DQNMaoEA 与 MOEA/D-DQN、CMOC SO、

DKCA、MOEA-NZD、NRV-MOEA、TriP、MaOEA/SRV、DRLOS-EMCMO 和 APSEA 在 5 目标、10 目标和 12 目标的 MaF、DTLZ 以及 WFG 系列测试问题上进行对比实验。所有算法在测试问题上分别独立运行 30 次,最后取 HV 和 IGD 指标的平均值做 Wilcoxon 秩和检验,在 0.05 的显著性水平上比较所有算法性能。

(1)DQNMaoEA 在 DTLZ 基准测试问题上的性能验证。

表 9 记录了算法的 HV 值结果、表 10 记录了算法的 IGD 值。表 9 中,在 0.05 的显著性水平上,21 个测试实例中,DQNMaoEA 的 HV 值分别比 MOEA/D-DQN、CMOC SO、DKCA、MOEA-NZD、NRV-MOEA、TriP、MaOEA/SRV、DRLOS-EMCMO 和 APSEA 优 18、20、19、16、16、20、21、19、19 个。总体来说 DQNMaoEA 共获得了 14 个最优结果。在表 10 中,DQNMaoEA 的 IGD 值比 MOEA/D-DQN、CMOC SO、DKCA、MOEA-NZD、NRV-MOEA、TriP、MaOEA/SRV、DRLOS-EMCMO 和 APSEA 分别有 18、20、18、17、19、19、20、20、19 个较优结果,更是在总体上获得了 17 个最优结果。在表现不佳的 DTLZ7 问题上,由于 DTLZ7 的 Pareto 前沿是直线和超平面的组合,导致算法搜索到了一定冗余的非支配解,而这些非支配解并不存在于 Pareto 前沿上。结果显示,DQNMaoEA 在 DTLZ1-6 上大部分测试实例上均为最优,特别是在 DTLZ5 与 DTLZ6 上 DQNMaoEA 的 HV 和 IGD 值全部都是最优,兼顾了解的多样性和收敛性。因此,本文提出的算法总体上在 DTLZ1-7 上较其他算法更优。此外,图 10 给出本文提出的 DQNMaoEA 与其余对比算法在 DTLZ4 上的 5、10、12 目标测试实例的最终输出种群,以可视化比较算法的性能。

表 9 在 DTLZ 系列测试问题上的 HV 值统计结果

问题	M	D	MOEA-D/ DQN	CMOC SO	DKCA	MOEA- NZD	NRV- MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaoEA
DTLZ1	5	9	6.5159e-1 (4.24e-1)-	4.6429e-1 (4.76e-1)-	6.2591e-5 (3.43e-4)-	7.1402e-1 (1.62e-1)-	7.4250e-1 (1.06e-2)-	6.0562e-1 (3.79e-1)-	0.0000e+0 (0.00e+0)-	9.1508e-1 (9.26e-3)+	9.6217e-1 (3.08e-3)+	7.2333e-1 (1.68e-2)
		14	9.4904e-1 (2.23e-1)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	4.5504e-1 (4.11e-1)-	0.0000e+0 (0.00e+0)-	5.6815e-1 (4.54e-1)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	9.5489e-1 (1.07e-2)
	10	14	9.6879e-1 (1.57e-2)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	6.2559e-1 (3.48e-1)-	0.0000e+0 (0.00e+0)-	4.0376e-1 (4.43e-1)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	9.8497e-1 (6.32e-3)
		16	6.5501e-1 (1.91e-2)-	4.3285e-1 (7.48e-2)-	6.3447e-1 (1.81e-2)-	5.7464e-1 (3.52e-4)-	6.8581e-1 (1.67e-3)-	5.8208e-1 (8.58e-2)-	1.8763e-1 (2.87e-2)-	3.7516e-1 (7.31e-2)-	6.8078e-1 (2.52e-2)-	6.9287e-1 (2.27e-2)
	12	16	6.5501e-1 (1.91e-2)-	4.3285e-1 (7.48e-2)-	6.3447e-1 (1.81e-2)-	5.7464e-1 (3.52e-4)-	6.8581e-1 (1.67e-3)-	5.8208e-1 (8.58e-2)-	1.8763e-1 (2.87e-2)-	3.7516e-1 (7.31e-2)-	6.8078e-1 (2.52e-2)-	6.9287e-1 (2.27e-2)
		14	6.5501e-1 (1.91e-2)-	4.3285e-1 (7.48e-2)-	6.3447e-1 (1.81e-2)-	5.7464e-1 (3.52e-4)-	6.8581e-1 (1.67e-3)-	5.8208e-1 (8.58e-2)-	1.8763e-1 (2.87e-2)-	3.7516e-1 (7.31e-2)-	6.8078e-1 (2.52e-2)-	6.9287e-1 (2.27e-2)

表 10 在 DTLZ 系列测试问题上的 IGD 值统计结果 (续表)

问题	M	D	MOEA/D- DQN	CMOCSO	DKCA	MOEA- NZD	NRV- MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaOEA
DTL3	10	19	4.9915e-1	1.5720e+0	1.8466e+0	5.4272e-1	4.8958e-1	7.5575e-1	7.8945e-1	1.4921e+0	2.3271e+0	4.8791e-1
			(3.67e-2)-	(1.11e-1)-	(1.55e-1)-	(3.43e-3)-	(4.27e-3)-	(4.47e-2)-	(3.25e-2)-	(1.16e-1)-	(2.39e-1)-	(2.66e-2)
			5.6019e-1	2.0168e+0	1.9792e+0	4.9617e-1	5.1070e-1	1.0867e+0	4.6387e-1	1.7653e+0	2.4421e+0	6.4894e-1
	12	21	(3.55e-2)+	(2.15e-1)-	(1.31e-1)-	(2.30e-3)+	(6.05e-3)+	(1.23e-1)-	(2.26e-2)+	(1.70e-1)-	(2.06e-1)-	(4.78e-2)
			4.7749e+0	7.9785e+1	6.0337e+1	1.3562e+0	8.0965e-1	1.8209e+1	3.8295e+2	4.9896e+1	1.6661e+0	3.0352e-1
			(9.68e+0)-	(6.19e+1)-	(3.28e+1)-	(9.53e-1)-	(7.27e-1)-	(2.90e+1)-	(6.13e+1)-	(2.51e+1)-	(1.50e+0)-	(2.66e-2)
	10	19	5.6046e-1	3.8799e+2	2.2299e+2	7.4165e+1	9.4316e+1	3.2274e+1	3.6179e+2	7.1655e+2	1.2858e+3	5.1865e-1
			(1.20e-1)-	(1.26e+2)-	(8.56e+0)-	(1.56e+1)-	(2.10e+1)-	(3.32e+1)-	(6.34e+1)-	(1.51e+2)-	(1.71e+2)-	(5.50e-2)
			1.2341e+0	6.7897e+2	2.2567e+2	9.2716e+0	1.1224e+2	6.0121e+1	3.4808e+2	9.1022e+2	1.5618e+3	7.5211e-1
	12	21	(1.19e+0)-	(1.34e+2)-	(6.55e+0)-	(4.59e+0)-	(2.39e+1)-	(4.19e+1)-	(5.54e+1)-	(1.42e+2)-	(7.14e+1)-	(1.97e-1)
			3.0225e-1	4.0305e-1	2.4178e-1	2.2716e-1	2.1666e-1	3.3476e-1	1.0959e+0	4.3438e-1	2.6815e-1	2.8279e-1
			(6.56e-2)-	(4.91e-2)-	(5.32e-3)+	(5.55e-2)+	(1.37e-3)+	(6.00e-2)-	(6.55e-2)-	(2.80e-2)-	(1.59e-2)+	(5.89e-3)
DTL4	10	19	4.6165e-1	1.5552e+0	1.2157e+0	4.8073e-1	4.2433e-1	5.5548e-1	9.7752e-1	1.2356e+0	2.0911e+0	3.7017e-1
			(1.12e-2)-	(1.15e-1)-	(6.53e-2)-	(5.02e-2)-	(5.23e-3)-	(1.72e-2)-	(5.33e-2)-	(9.96e-2)-	(3.77e-1)-	(4.04e-3)
			6.0805e-1	2.1400e+0	1.3220e+0	5.6337e-1	4.9911e-1	8.3475e-1	7.7299e-1	1.4163e+0	2.4061e+0	4.6540e-1
	12	21	(2.00e-2)-	(2.06e-1)-	(6.83e-2)-	(7.51e-2)-	(1.26e-2)-	(8.33e-2)-	(6.29e-2)-	(1.17e-1)-	(2.30e-1)-	(6.71e-3)
			9.3280e-2	4.3553e-1	1.1997e-1	1.7813e-1	8.3636e-2	1.5200e-1	3.3921e-1	3.3708e-1	2.4999e-1	1.3673e-2
			(1.39e-2)-	(1.73e-1)-	(2.36e-2)-	(5.27e-2)-	(1.61e-2)-	(2.61e-2)-	(4.29e-2)-	(8.40e-2)-	(5.56e-2)-	(2.15e-3)
DTL5	10	19	1.9922e-1	1.2406e+0	3.3264e-1	4.5979e-1	1.7852e-1	7.5607e-2	3.3174e-1	8.8609e-1	1.3476e+0	1.0994e-2
			(2.62e-4)-	(3.98e-1)-	(1.12e-1)-	(8.98e-2)-	(2.83e-2)-	(1.82e-2)-	(4.49e-2)-	(2.77e-1)-	(4.00e-1)-	(1.54e-3)
			2.0593e-1	1.7028e+0	3.7489e-1	4.2287e-1	2.0894e-1	1.1257e-1	2.5632e-1	1.0385e+0	1.3599e+0	1.2265e-2
	12	21	(8.83e-5)-	(3.32e-1)-	(1.95e-1)-	(1.32e-1)-	(2.94e-2)-	(3.30e-2)-	(2.75e-2)-	(3.04e-1)-	(4.35e-1)-	(1.46e-3)
			9.1510e-2	4.9840e-1	1.9167e-1	2.4377e-1	1.5801e-1	2.5502e-1	8.4576e+0	4.3278e-1	2.6111e+0	1.9705e-2
			(8.30e-3)-	(2.08e-1)-	(4.15e-2)-	(9.52e-2)-	(4.88e-2)-	(2.92e-1)-	(9.11e-2)-	(1.40e-1)-	(1.31e+0)-	(5.60e-3)
DTL6	10	19	1.9944e-1	4.3517e+0	1.9881e-1	2.7014e-1	3.8780e-1	6.9111e-2	8.3670e+0	5.2796e+0	9.8934e+0	2.7284e-2
			(2.72e-4)-	(5.85e-1)-	(7.19e-2)-	(7.49e-2)-	(1.16e-1)-	(2.29e-2)-	(1.64e-1)-	(1.02e+0)-	(4.08e-2)-	(7.77e-3)
			2.0590e-1	6.1173e+0	2.2637e-1	3.0782e-1	6.2279e-1	1.1735e-1	8.1926e+0	7.3263e+0	9.9358e+0	2.0692e-2
	12	21	(2.11e-5)-	(2.55e+0)-	(5.22e-2)-	(8.72e-2)-	(2.03e-1)-	(4.21e-2)-	(1.35e-1)-	(2.22e+0)-	(3.21e-2)-	(9.30e-3)
			5.9405e-1	3.3062e-1	3.7650e-1	3.7948e-1	2.9721e-1	5.2045e-1	3.4980e+1	3.3827e-1	3.0314e-1	5.7048e-1
			(4.87e-2)=	(1.34e-2)+	(1.21e-2)+	(1.56e-2)+	(2.33e-2)+	(2.01e-1)+	(3.38e+0)-	(1.18e-2)+	(7.49e-3)+	(6.53e-4)
DTL7	10	29	1.5742e+0	1.4223e+0	1.0477e+0	9.0473e-1	7.9540e-1	1.3456e+0	2.2656e+1	1.5923e+0	1.8699e+0	1.0727e+0
			(6.69e-2)-	(4.96e-2)-	(1.53e-2)+	(4.51e-2)+	(1.32e-2)+	(2.25e-2)-	(1.63e+0)-	(5.23e-2)-	(1.43e-1)-	(2.00e-2)
			2.1660e+0	1.9254e+0	1.2478e+0	1.5428e+0	1.5085e+0	1.7222e+0	1.3189e+1	1.8499e+0	2.2746e+0	1.3554e+0
	12	31	(3.52e-1)-	(6.67e-2)-	(1.57e-2)-	(1.19e-1)-	(1.87e-2)-	(3.49e-2)-	(1.21e+0)-	(4.00e-2)-	(2.39e-1)-	(1.62e-2)
			+/-/≈	1/18/2	1/20/0	3/18/0	4/17/0	1/19/1	1/19/1	1/20/0	1/20/0	2/19/0

(2) DQNMaOEA 在 WFG 基准测试问题上的性能验证。

实验结果的 HV 值保存在表 11 中, IGD 值保存在表 12 中。总体来看, DQNMaOEA 的 HV 值在 21 个测试实例中优于其他比较算法的个数分别为 15、19、20、19、15、20、19 和 16。IGD 值分别优于 MOEA/D-DQN、CMOCSO、DKCA、MOEA-NZD、NRV-MOEA、TriP、MaOEA/SRV、DRLOS-EMCMO 和 APSEA 的测试实例为 18、13、18、13、14、13、18、13 和 13 个。得益于多种群协同优化策略, DQNMaOEA 在 HV 值上表现优秀, 使得种群具

有良好的多样性。在求解特征为多模态的 WFG4、不可分解的 WFG6 以及参数独立的 WFG7 时, DQNMaOEA 虽然没有在 IGD 值上取得最佳性能表现, 但是最终种群的分布接近 Pareto 真实前沿。而在 WFG1-3、WFG5 上, DQNMaOEA 有出色的表现, 不仅能寻找到 PFs, 而且在解空间中分布较为均匀。综上所述, 在求解 WFG 系列问题上的较优效果证明了本文提出算法 DQNMaOEA 的有效性。并以 WFG2 测试问题为例, 图 11 对 5、10 以及 12 目标种群的最终搜索结果进行了可视化, 更加鲜明地对各算法的性能进行比较。

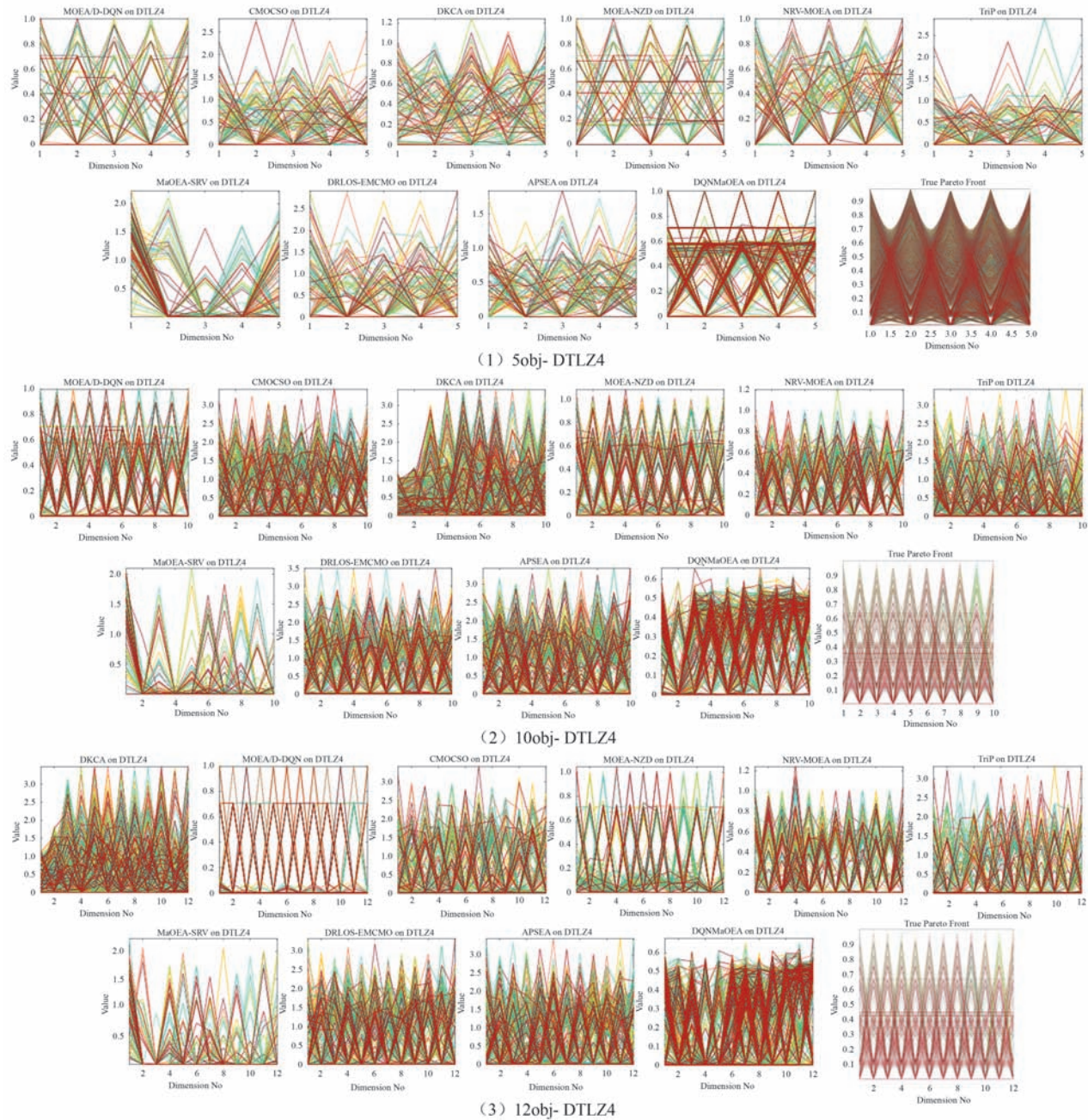


图10 5、10和12目标DTLZ4的最终输出种群示意

表 11 在WFG系列测试问题上的HV值统计结果

问题	$M D$	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA- NZD	NRV- MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaoEA
WFG1	5 14	9.9210e-1 (2.70e-3) +	7.9909e-1 (2.98e-1) +	9.5214e-1 (2.21e-2) +	9.7470e-1 (2.74e-2) +	9.8667e-1 (3.77e-3) +	6.9178e-1 (3.05e-1) =	0.0000e+0 (0.00e+0) -	7.7885e-1 (6.94e-2) +	9.9550e-1 (7.52e-4) +	5.8459e-1 (5.51e-2)
	10 19	9.9839e-1 (1.79e-3) =	3.3609e-1 (2.34e-2) -	7.4716e-1 (5.90e-2) -	5.6431e-1 (4.12e-2) -	4.9846e-1 (3.66e-2) -	4.5214e-1 (3.72e-2) -	4.9421e-2 (2.14e-2) -	2.7474e-1 (1.63e-2) -	4.8970e-1 (4.24e-2) -	9.9994e-1 (5.47e-5)
	12 21	8.0208e-1 (1.92e-1) -	2.5402e-1 (1.20e-2) -	6.8531e-1 (6.86e-2) -	8.4597e-1 (8.82e-2) -	4.3142e-1 (4.23e-2) -	3.6773e-1 (4.38e-2) -	4.7055e-2 (1.19e-2) -	2.4942e-1 (1.62e-2) -	4.6897e-1 (3.82e-2) -	9.9983e-1 (8.00e-5)
WFG2	5 14	9.0022e-1 (2.89e-2) -	9.4279e-1 (8.64e-3) -	9.4707e-1 (4.47e-3) -	9.4224e-1 (6.19e-3) -	9.4582e-1 (2.04e-3) -	9.1232e-1 (1.85e-2) -	6.3402e-1 (1.59e-2) -	9.1081e-1 (9.35e-3) -	9.4821e-1 (1.55e-3) =	9.4872e-1 (9.68e-3)

表 11 在 WFG 系列测试问题上的 HV 值统计结果（续表）											
问题	$M D$	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA- NZD	NRV- MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaoEA
WFG3	10 19	9.8733e-1 (7.67e-3)	9.7325e-1 (3.98e-3)-	9.7423e-1 (8.95e-3)-	9.6810e-1 (1.30e-2)-	9.7687e-1 (4.48e-3)-	9.8704e-1 (4.59e-3)-	5.9052e-1 (3.37e-2)-	9.0157e-1 (1.78e-2)-	9.7915e-1 (4.54e-3)-	9.9873e-1 (5.07e-4)
	12 21	9.1714e-1 (4.25e-2)-	9.2375e-1 (1.50e-2)-	9.6987e-1 (1.02e-2)-	9.7472e-1 (1.09e-2)-	9.7492e-1 (6.27e-3)-	9.5598e-1 (1.11e-2)-	5.9639e-1 (3.11e-2)-	8.6107e-1 (1.91e-2)-	9.7387e-1 (6.31e-3)-	9.9463e-1 (1.76e-3)
	5 14	6.2495e-2 (3.34e-2)-	1.1901e-2 (1.23e-2)-	1.0030e-1 (1.54e-2)-	1.1296e-1 (2.44e-2)-	1.3099e-1 (2.24e-2)-	4.0540e-2 (2.19e-2)-	1.4632e-3 (4.63e-3)-	1.0187e-1 (3.02e-2)-	1.1104e-1 (2.92e-2)-	1.3708e-1 (2.29e-2)
	10 19	6.0570e-4 (2.47e-3)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	8.0115e-2 (4.92e-3)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-	9.8085e-2 (1.00e-2)
	12 21	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)
	WFG4	5 14	5.3556e-1 (4.05e-2)-	5.9823e-1 (1.76e-2)-	6.5199e-1 (1.44e-2)+	7.6129e-1 (2.28e-3)+	7.6966e-1 (2.49e-3)+	6.1102e-1 (1.06e-2)-	3.9878e-1 (1.19e-2)-	5.9316e-1 (2.96e-2)-	6.7838e-1 (1.08e-2)+
10 19		7.2132e-1 (3.24e-2)-	5.2280e-1 (2.04e-2)-	6.3150e-1 (2.37e-2)-	8.9446e-1 (5.97e-3)+	8.5951e-1 (9.88e-3)+	8.1515e-1 (6.66e-3)+	3.8598e-1 (2.24e-2)-	5.9927e-1 (2.06e-2)-	7.3229e-1 (1.42e-2)-	7.9937e-1 (2.53e-2)
12 21		7.9639e-1 (2.25e-2)-	4.4762e-1 (1.67e-2)-	6.3682e-1 (2.19e-2)-	8.1514e-1 (5.60e-3)-	8.3203e-1 (1.25e-2)-	7.4207e-1 (1.91e-2)-	3.9290e-1 (1.26e-2)-	5.7613e-1 (1.81e-2)-	7.0306e-1 (2.28e-2)-	8.3257e-1 (1.67e-2)
5 14		4.9016e-1 (2.21e-2)-	6.0222e-1 (1.88e-2)-	6.0212e-1 (1.38e-2)-	7.2307e-1 (1.36e-3)-	7.3310e-1 (2.10e-3)-	5.8567e-1 (1.34e-2)-	3.6029e-1 (5.00e-3)-	6.4171e-1 (1.75e-2)-	6.7126e-1 (6.83e-3)-	7.3476e-1 (1.46e-2)
10 19		7.2594e-1 (2.06e-2)-	7.0428e-1 (1.49e-2)-	6.2128e-1 (1.34e-2)-	7.8863e-1 (2.32e-3)=	7.5976e-1 (5.23e-3)-	7.6628e-1 (9.26e-3)-	3.6622e-1 (1.68e-2)-	6.7008e-1 (1.80e-2)-	6.6593e-1 (1.70e-2)-	7.8879e-1 (6.44e-3)
12 21		6.8071e-1 (4.83e-2)-	6.6950e-1 (2.03e-2)-	6.1691e-1 (1.92e-2)-	6.7515e-1 (4.63e-3)-	7.2206e-1 (1.48e-2)-	7.2368e-1 (1.51e-2)-	3.6803e-1 (1.95e-2)-	6.4099e-1 (1.89e-2)-	6.3336e-1 (2.71e-2)-	7.5283e-1 (9.91e-3)
WFG6	5 14	6.1360e-1 (7.95e-2)+	5.3786e-1 (1.02e-2)-	4.8816e-1 (9.90e-3)-	5.9855e-1 (3.52e-4)+	7.1332e-1 (1.62e-2)+	6.5149e-1 (2.66e-2)+	3.2002e-1 (7.31e-3)-	5.2351e-1 (2.39e-2)-	6.6411e-1 (1.83e-2)+	5.6661e-1 (1.21e-2)
	10 19	7.1416e-1 (9.53e-2)-	6.4104e-1 (1.52e-2)-	5.7793e-1 (2.04e-2)-	7.4103e-1 (2.43e-4)-	7.2890e-1 (2.03e-2)-	7.4730e-1 (3.99e-2)=	3.2135e-1 (1.28e-2)-	4.9272e-1 (4.02e-2)-	6.5366e-1 (3.57e-2)-	7.4752e-1 (1.53e-2)
	12 21	6.5784e-1 (3.64e-2)-	6.4578e-1 (1.48e-2)-	5.7392e-1 (1.72e-2)-	6.4262e-1 (3.97e-4)-	6.9500e-1 (2.38e-2)-	6.5716e-1 (2.05e-2)-	3.2373e-1 (1.62e-2)-	4.5250e-1 (3.03e-2)-	6.0548e-1 (3.46e-2)-	7.1331e-1 (1.39e-2)
	5 14	7.0660e-1 (3.02e-2)+	5.9335e-1 (3.77e-2)-	6.3972e-1 (1.74e-2)-	7.6525e-1 (1.62e-3)+	7.8419e-1 (1.88e-3)+	6.8810e-1 (1.63e-2)+	3.9359e-1 (6.28e-3)-	5.7550e-1 (3.09e-2)-	6.6505e-1 (3.18e-2)-	6.4747e-1 (4.93e-3)
	10 19	8.4249e-1 (2.37e-2)-	4.6879e-1 (2.45e-2)-	6.4399e-1 (2.21e-2)-	7.1060e-1 (1.49e-2)-	8.0096e-1 (1.28e-2)-	7.8509e-1 (1.58e-2)-	3.8759e-1 (1.66e-2)-	5.1168e-1 (1.80e-2)-	6.3520e-1 (3.06e-2)-	8.9885e-1 (1.55e-2)
	12 21	8.0491e-1 (3.66e-2)+	4.0798e-1 (2.06e-2)-	6.3833e-1 (2.72e-2)-	9.0934e-1 (1.51e-2)+	8.3562e-1 (1.86e-2)+	7.1982e-1 (2.29e-2)-	3.8620e-1 (1.75e-2)-	4.9629e-1 (1.50e-2)-	6.0480e-1 (2.84e-2)-	7.7112e-1 (1.00e-2)
+/-/≈		4/15/2	1/19/1	2/18/1	6/13/2	6/14/1	3/15/3	0/20/1	1/19/1	3/16/2	

表 12 在 WFG 系列测试问题上的 IGD 值统计结果											
问题	MD	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA- NZD	NRV-MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaoEA
WFG1	5 14	7.8963e-1 (1.07e-1)+	1.0552e+0 (6.00e-1)+	9.0691e-1 (9.43e-2)+	5.0102e-1 (1.72e-2)+	4.5510e-1 (9.47e-3)+	1.2293e+0 (5.81e-1)=	2.6151e+0 (4.80e-2)-	9.2160e-1 (1.30e-1)+	6.0470e-1 (1.77e-2)+	1.2200e+0 (1.01e-1)
		1.7843e+0 (1.69e-1)-	2.8575e+0 (1.79e-1)-	1.8542e+0 (7.92e-2)-	1.7968e+0 (1.07e-1)-	1.9315e+0 (1.37e-1)-	2.3896e+0 (1.37e-1)-	3.3855e+0 (5.46e-2)-	3.1005e+0 (1.18e-1)-	2.3681e+0 (2.26e-1)-	1.3925e+0 (5.44e-2)
	10 19	1.9198e+0 (2.69e-1)-	3.5758e+0 (1.79e-1)-	2.0047e+0 (1.12e-1)-	1.5883e+0 (8.54e-2)-	2.3716e+0 (1.99e-1)-	3.0359e+0 (3.55e-1)-	3.6478e+0 (5.04e-2)-	3.5701e+0 (1.98e-1)-	2.5681e+0 (1.91e-1)-	1.5680e+0 (7.62e-2)
		9.6483e-1 (1.06e-1)-	8.1216e-1 (4.66e-2)-	8.6283e-1 (5.23e-2)-	7.0115e-1 (4.42e-3)-	6.8280e-1 (1.62e-2)-	8.0156e-1 (4.25e-2)-	1.4821e+0 (3.15e-1)-	8.3696e-1 (3.28e-2)-	8.1002e-1 (2.67e-2)-	6.6033e-1 (3.07e-2)
	12 21	7.8963e-1 (1.07e-1)+	1.0552e+0 (6.00e-1)+	9.0691e-1 (9.43e-2)+	5.0102e-1 (1.72e-2)+	4.5510e-1 (9.47e-3)+	1.2293e+0 (5.81e-1)=	2.6151e+0 (4.80e-2)-	9.2160e-1 (1.30e-1)+	6.0470e-1 (1.77e-2)+	1.2200e+0 (1.01e-1)
		1.7843e+0 (1.69e-1)-	2.8575e+0 (1.79e-1)-	1.8542e+0 (7.92e-2)-	1.7968e+0 (1.07e-1)-	1.9315e+0 (1.37e-1)-	2.3896e+0 (1.37e-1)-	3.3855e+0 (5.46e-2)-	3.1005e+0 (1.18e-1)-	2.3681e+0 (2.26e-1)-	1.3925e+0 (5.44e-2)

表 12 在 WFG 系列测试问题上的 IGD 值统计结果 (续表)

问题	<i>MD</i>	MOEA-D/ DQN	CMOCSO	DKCA	MOEA- NZD	NRV-MOEA	TriP	MaOEA/ SRV	DRLOS- EMCMO	APSEA	DQNMaoEA
WFG3	1019	1.7417e+0 (9.68e-2) -	1.9782e+0 (1.44e-1) -	1.6112e+0 (6.54e-2) -	1.0902e+0 (1.23e-1) -	1.2302e+0 (7.81e-2) -	1.6666e+0 (1.34e-1) -	3.5548e+0 (7.43e-1) -	2.1343e+0 (1.11e-1) -	2.0348e+0 (1.72e-1) -	9.8866e-1 (2.23e-2)
	1221	1.6928e+0 (1.02e-1) -	2.6334e+0 (3.69e-1) -	1.7219e+0 (8.83e-2) -	1.2640e+0 (8.84e-2) -	1.3185e+0 (5.19e-2) -	2.8263e+0 (5.09e-1) -	3.9946e+0 (8.22e-1) -	2.9993e+0 (4.70e-1) -	2.5877e+0 (4.42e-1) -	1.2623e+0 (4.05e-2)
	514	1.2253e+0 (1.70e-1) -	9.0864e-1 (9.98e-2) -	6.1155e-1 (1.06e-1) -	6.9919e-1 (1.33e-1) -	6.3075e-1 (4.39e-2) -	8.6491e-1 (1.17e-1) -	8.2851e-1 (4.46e-2) -	7.1039e-1 (1.35e-1) -	6.2023e-1 (8.16e-2) -	5.4035e-1 (3.88e-2)
	1019	3.4571e+0 (1.40e-1) -	2.8735e+0 (2.69e-1) -	1.2710e+0 (2.61e-1) -	1.1406e+0 (3.41e-1) -	2.1501e+0 (2.21e-1) -	1.9983e+0 (2.71e-1) -	1.4487e+0 (7.49e-2) -	2.6245e+0 (2.05e-1) -	2.5716e+0 (4.99e-1) -	4.4563e-1 (4.53e-2)
	1221	4.2574e+0 (2.23e-1) -	3.6069e+0 (2.90e-1) -	1.6698e+0 (3.05e-1) -	2.1306e+0 (3.93e-1) -	3.0545e+0 (4.50e-1) -	3.4103e+0 (4.55e-1) -	1.6637e+0 (8.71e-2) -	3.3012e+0 (2.76e-1) -	3.6050e+0 (4.75e-1) -	7.1409e-1 (6.94e-2)
	WFG4	514	2.3016e+0 (1.65e-1) -	1.2331e+0 (1.36e-2) -	1.2894e+0 (2.68e-2) -	1.2256e+0 (1.22e-3) -	1.2491e+0 (1.13e-2) -	1.2528e+0 (1.01e-2) -	2.0383e+0 (1.64e-1) -	1.2271e+0 (2.45e-2) -	1.1487e+0 (7.47e-3) -
1019		5.7785e+0 (2.87e-1) -	4.3939e+0 (3.63e-2) +	4.6172e+0 (6.26e-2) +	4.5824e+0 (2.64e-2) +	4.6163e+0 (8.59e-2) +	4.1337e+0 (2.22e-2) +	8.9771e+0 (8.47e-1) -	4.2782e+0 (2.78e-2) +	4.1720e+0 (4.05e-2) +	5.5646e+0 (1.57e-1)
1221		7.4939e+0 (2.14e-1) -	5.9675e+0 (4.89e-2) +	5.9182e+0 (5.50e-2) +	6.8193e+0 (4.57e-2) -	6.1532e+0 (1.05e-1) +	6.2875e+0 (4.78e-2) +	1.1266e+1 (4.39e-1) -	5.8499e+0 (4.02e-2) +	5.8318e+0 (4.41e-2) +	6.6752e+0 (3.22e-1)
514		2.1420e+0 (1.25e-1) -	1.2315e+0 (1.63e-2) -	1.3145e+0 (2.34e-2) -	1.2135e+0 (1.78e-3) -	1.2485e+0 (1.60e-2) -	1.2565e+0 (1.27e-2) -	1.5729e+0 (6.55e-2) -	1.1597e+0 (1.68e-2) -	1.1333e+0 (6.66e-3) -	1.1143e+0 (2.12e-2)
1019		5.8331e+0 (3.35e-1) -	4.1711e+0 (3.29e-2) -	4.6247e+0 (4.24e-2) -	4.4540e+0 (1.80e-2) -	4.8186e+0 (6.82e-2) -	4.1078e+0 (2.72e-2) -	6.6405e+0 (2.33e-1) -	4.1626e+0 (3.47e-2) -	4.1468e+0 (3.77e-2) -	4.0390e+0 (9.64e-2)
1221		7.2043e+0 (3.07e-1) -	5.7562e+0 (4.82e-2) +	5.8968e+0 (4.77e-2) +	6.6713e+0 (6.29e-2) -	6.4818e+0 (1.41e-1) -	6.1046e+0 (7.40e-2) -	8.8798e+0 (3.87e-1) -	5.6621e+0 (3.71e-2) +	5.6839e+0 (4.23e-2) +	6.0756e+0 (2.12e-1)
WFG6	514	1.5453e+0 (2.42e-1) +	1.2727e+0 (1.81e-2) +	1.4426e+0 (3.78e-2) +	1.2286e+0 (1.82e-4) +	1.2615e+0 (1.35e-2) +	1.2618e+0 (1.54e-2) +	1.6789e+0 (4.19e-2) +	1.2991e+0 (2.92e-2) +	1.1632e+0 (7.25e-3) +	1.9034e+0 (5.77e-2)
	1019	5.1451e+0 (5.95e-1) -	4.2832e+0 (4.38e-2) -	4.6455e+0 (4.51e-2) -	4.5913e+0 (1.85e-2) -	5.0432e+0 (8.36e-2) -	4.1736e+0 (3.91e-2) -	6.9761e+0 (4.12e-1) -	4.4724e+0 (5.50e-2) -	4.1674e+0 (3.38e-2) -	4.1638e+0 (9.56e-2)
	1221	7.2827e+0 (7.97e-1) -	5.7884e+0 (3.84e-2) +	5.9110e+0 (4.39e-2) +	6.7237e+0 (4.28e-2) -	6.7269e+0 (1.49e-1) -	6.2359e+0 (7.40e-2) +	9.3699e+0 (4.38e-1) -	5.9797e+0 (4.73e-2) +	5.7455e+0 (5.75e-2) +	6.5069e+0 (2.14e-1)
	514	1.6666e+0 (1.21e-1) +	1.2545e+0 (2.66e-2) +	1.3584e+0 (4.36e-2) +	1.2277e+0 (2.38e-3) +	1.2738e+0 (1.65e-2) +	1.2499e+0 (1.20e-2) +	1.6123e+0 (5.77e-2) +	1.2727e+0 (2.34e-2) +	1.1753e+0 (1.86e-2) +	1.9248e+0 (4.63e-2)
	1019	5.4595e+0 (2.79e-1) -	4.4249e+0 (3.55e-2) -	4.7346e+0 (6.64e-2) -	4.5030e+0 (1.23e-1) -	4.8152e+0 (1.11e-1) -	4.1389e+0 (2.76e-2) =	7.2877e+0 (3.84e-1) -	4.4060e+0 (3.16e-2) -	4.1948e+0 (4.71e-2) -	4.1344e+0 (1.70e-1)
	1221	7.5425e+0 (3.88e-1) -	5.9527e+0 (6.83e-2) +	5.9888e+0 (6.49e-2) +	6.8314e+0 (2.42e-1) +	6.4706e+0 (1.38e-1) +	6.2346e+0 (6.95e-2) +	9.6727e+0 (5.70e-1) -	5.9430e+0 (4.94e-2) +	5.7800e+0 (4.30e-2) +	6.9673e+0 (2.08e-1)
+/-/≈		3/18/0	8/13/0	8/13/0	5/16/0	6/15/0	6/13/2	3/18/0	8/13/0	8/13/0	

(3)DQNMaoEA 在 MaF 基准测试问题上的性能验证。

为了测试算法 DQNMaoEA 求解更多的超多目标优化问题变体的能力,本组实验在 MaF 基准测试问题上,分别与 MOEA/D-DQN、CMOCSO、DKCA、MOEA-NZD、NRV-MOEA、TriP、DRLOS-EMCMO 和 APSEA 八个对比算法进行了测试,实验结果的 HV、IGD 值分别保存在表 13、表 14 中。其中,每个测试问题均代表着不同

类型变体。实验结果表明,DQNMaoEA 在 MaF1-2 以及 MaF5-6 问题上的 HV 值表现较优,在 MaF1、MaF3-4 以及 MaF7 问题上的 IGD 值总体效果强于其他算法,表明算法能够很好地处理具有倒置 PF、凸 PF、断开连接 PF 以及适应度多模态等特征的超多目标优化问题。而对于求解 PF 具有不良缩放和退化的不规则超多目标优化问题以及提高在不同目标上的并发收敛性等方面,DQNMaoEA 仍需要进行修改,以进一步提高算法的求解性能。

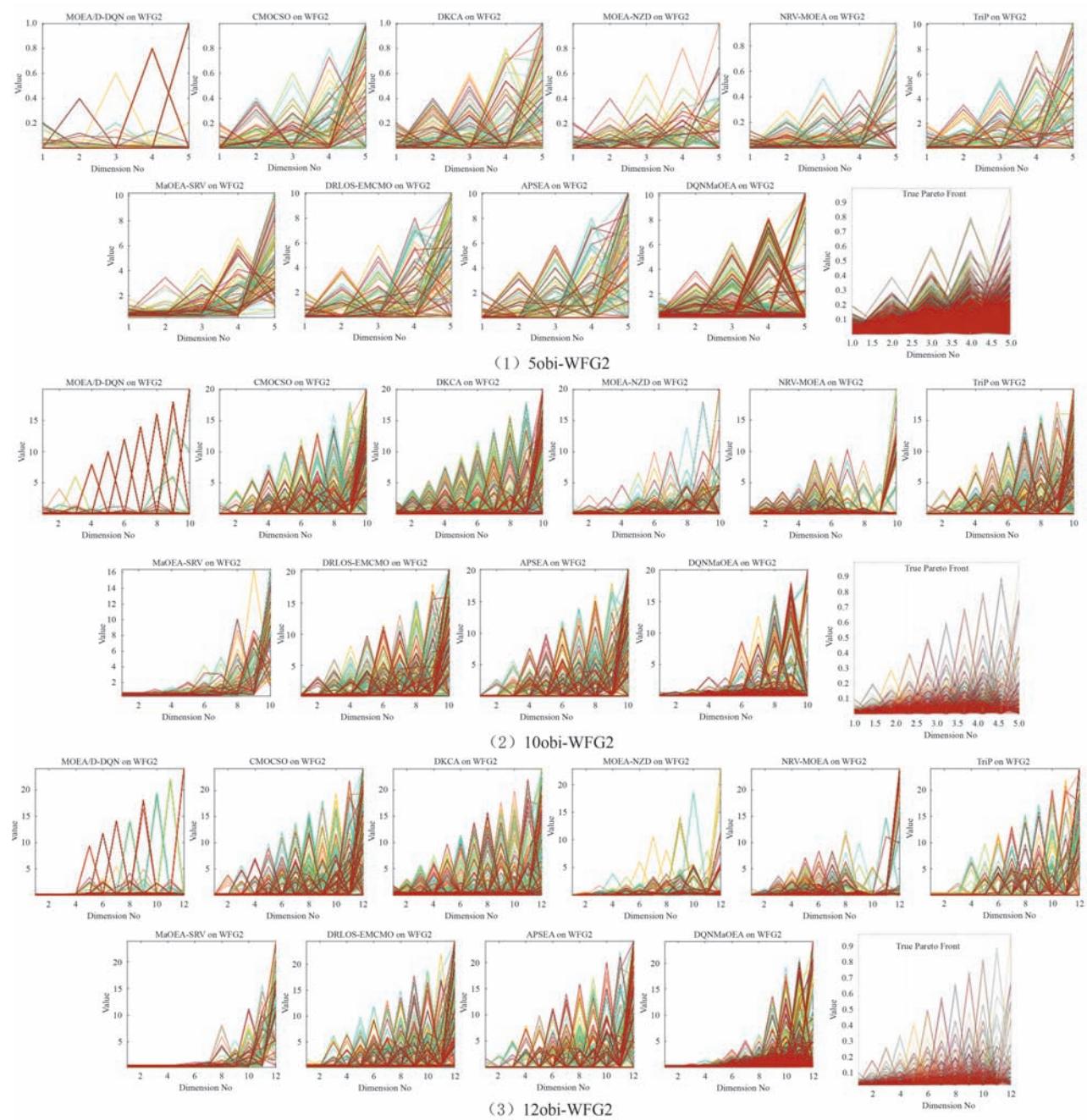


图 11 5、10和12目标WFG2的最终输出种群示意图

表 13 在MaF系列测试问题上的HV值统计结果

问题	M	D	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA- NZD	NRV- MOEA	TriP	DRLOS- EMCMO	APSEA	DQNMaOEA
MaF1	5	14	5.1042e-3 (3.46e-4) -	5.5069e-3 (5.78e-4) -	5.5087e-3 (4.53e-4) -	6.0362e-3 (6.70e-4) -	6.9881e-3 (1.09e-4) -	5.5565e-3 (9.56e-4) -	3.3093e-3 (5.44e-4) -	6.9629e-3 (2.77e-4) -	7.6944e-3 (1.09e-4)
		19	4.7358e-8 (3.57e-8) -	1.0923e-7 (2.28e-7) -	9.6261e-8 (2.97e-7) -	5.2762e-7 (9.07e-8) -	5.8721e-7 (1.17e-7) -	1.1404e-7 (2.81e-7) -	2.4884e-8 (1.11e-7) -	7.7097e-8 (2.44e-7) -	7.2914e-7 (3.18e-7)
	12	21	2.0149e-10 (1.16e-10) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	4.3381e-9 (1.02e-9) -	3.0008e-9 (6.52e-9) -	1.0686e-10 (1.95e-10) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	4.6015e-9 (6.10e-9)
		14	1.1999e-1 (1.20e-2) -	1.2657e-1 (7.46e-3) -	1.4434e-1 (4.22e-3) -	1.5528e-1 (3.51e-3) -	1.8396e-1 (2.20e-3) -	1.4789e-1 (5.61e-3) -	1.3078e-1 (4.49e-3) -	1.4203e-1 (3.85e-3) -	1.8822e-1 (4.04e-3)

表 13 在 MaF 系列测试问题上的 HV 值统计结果（续表）

问题	M	D	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA- NZD	NRV- MOEA	TriP	DRLOS- EMCMO	APSEA	DQNMaOEA
MaF3	10	19	1.3224e-1 (2.84e-3)	1.7457e-1 (3.74e-3) -	1.9703e-1 (3.72e-3) -	2.1014e-1 (5.88e-3) -	1.9209e-1 (3.82e-3) -	1.8995e-1 (4.57e-3) -	1.7460e-1 (4.79e-3) -	1.7640e-1 (4.54e-3) -	2.4666e-1 (1.02e-3)
			9.3207e-2 (1.03e-2) -	1.0049e-1 (5.18e-3) -	1.0758e-1 (6.43e-3) -	1.3456e-1 (1.01e-2) -	1.5648e-1 (6.93e-3) -	1.5256e-1 (5.07e-3) -	1.2547e-1 (4.98e-3) -	1.3236e-1 (5.81e-3) -	2.2033e-1 (2.19e-3)
	12	21	5.9888e-1 (5.02e-1) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	6.4514e-1 (4.57e-1) -	0.0000e+0 (0.00e+0) -	2.9067e-1 (4.24e-1) -	0.0000e+0 (0.00e+0) -	3.8152e-1 (4.56e-1) -	9.2856e-1 (1.03e-2)
			9.9976e-1 (6.06e-4) +	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	5.9011e-2 (1.87e-1) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	9.9480e-1 (1.79e-3)
	12	21	8.7639e-1 (2.93e-1) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	9.9125e-1 (5.48e-3)
			6.3016e-3 (6.80e-3) -	1.8300e-2 (2.34e-2) -	2.4553e-3 (8.92e-3) -	5.7248e-3 (1.26e-2) -	9.1024e-2 (2.46e-2) +	2.2024e-2 (2.21e-2) -	1.5054e-3 (3.29e-3) -	1.1991e-2 (1.30e-2) -	2.5903e-2 (2.98e-2)
MaF4	5	14	9.7637e-8 (1.45e-7) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	7.8984e-7 (2.30e-6) +	1.9068e-6 (2.84e-6) +	0.0000e+0 (0.00e+0) -	5.7749e-9 (2.54e-8) -	2.9221e-7 (3.53e-7)
	10	19	7.8570e-11 (1.13e-10) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	5.3862e-8 (1.94e-7) +	2.0322e-7 (9.09e-7) +	1.8883e-9 (7.58e-9) +	0.0000e+0 (0.00e+0) -	3.4315e-10 (1.44e-9) -	1.0089e-9 (1.06e-9)
	12	21	6.6107e-1 (5.32e-2) +	5.8203e-1 (5.28e-2) +	6.6743e-1 (1.32e-2) +	7.7464e-1 (5.36e-4) +	7.8703e-1 (1.43e-3) +	5.4405e-1 (1.54e-1) -	4.8266e-1 (4.30e-2) -	7.0842e-1 (1.54e-2) +	5.5261e-1 (4.02e-2)
MaF5	5	14	8.8691e-1 (2.76e-3) -	1.9801e-3 (4.27e-3) -	3.5400e-3 (9.55e-3) -	9.5688e-1 (2.59e-2) -	6.0119e-1 (3.12e-2) -	4.5001e-3 (9.59e-3) -	8.4607e-3 (1.41e-2) -	0.0000e+0 (0.00e+0) -	9.7283e-1 (1.02e-3)
	10	19	8.1263e-1 (1.94e-2) -	5.5813e-3 (9.64e-3) -	1.5608e-4 (6.98e-4) -	9.7180e-1 (6.96e-4) -	5.6151e-1 (4.41e-2) -	2.0440e-3 (5.73e-3) -	8.0469e-3 (1.19e-2) -	0.0000e+0 (0.00e+0) -	9.8517e-1 (1.14e-3)
	12	21	1.1415e-1 (5.00e-4) +	7.9333e-2 (6.02e-2) -	1.2926e-1 (3.49e-4) +	1.2294e-1 (2.45e-3) +	1.2881e-1 (6.07e-4) +	1.2170e-1 (8.03e-3) +	1.2886e-1 (4.10e-4) +	1.2900e-1 (2.91e-4) +	8.4107e-2 (1.47e-2)
MaF6	5	14	9.6607e-2 (3.38e-4) -	4.6746e-14 (2.09e-13) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	1.0034e-1 (3.80e-4) =	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	1.0065e-1 (3.01e-4)
	10	19	9.4782e-2 (2.31e-4) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	9.7281e-2 (3.36e-4) =	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	9.7441e-2 (3.71e-4)
	12	21	1.4602e-1 (3.23e-2) -	1.9112e-1 (8.57e-3) +	1.9426e-1 (7.09e-3)	2.3723e-1 (4.10e-3) +	2.5855e-1 (4.61e-3) +	1.2154e-1 (7.67e-2) +	1.9542e-1 (1.05e-2) +	2.1033e-1 (9.91e-3) +	8.3839e-2 (8.59e-2)
MaF7	5	24	1.4645e-1 (6.47e-3) +	9.3096e-4 (2.40e-3) -	9.8768e-2 (2.38e-3) +	1.9446e-1 (3.47e-3) +	1.3358e-1 (8.01e-3) +	3.4242e-3 (1.09e-2) -	7.5653e-7 (2.80e-6) -	4.6075e-5 (1.12e-4) -	9.0136e-3 (2.84e-3)
	10	29	5.3841e-6 (7.87e-6) -	3.8039e-6 (1.69e-5) -	9.2784e-2 (1.03e-3) +	1.6836e-1 (3.58e-3) +	7.9694e-2 (1.11e-2) +	8.0359e-5 (2.60e-4) -	6.2609e-8 (2.80e-7) -	0.0000e+0 (0.00e+0) -	8.2816e-4 (1.70e-3)
	12	31	+/-/≈	4/17/0	2/19/0	4/17/0	6/15/0	8/13/0	4/15/2	2/19/0	3/18/0

表 14 在 MaF 系列测试问题上的 IGD 值统计结果

问题	M	D	MOEA-D/ DQN	CMOCSSO	DKCA	MOEA-NZD	NRV-MOEA	TriP	DRLOS-EM- CMO	APSEA	DQN- MaOEA
MaF1	5	14	2.2898e-1 (1.62e-3) -	1.6457e-1 (5.85e-3) -	1.8158e-1 (6.68e-3) -	2.0003e-1 (9.87e-3) -	1.5446e-1 (1.67e-3) -	1.6885e-1 (8.57e-3) -	1.9944e-1 (7.93e-3) -	1.4324e-1 (1.75e-3) +	1.4883e-1 (7.64e-3)
			4.5956e-1 (3.07e-2) -	2.8503e-1 (6.94e-3) -	2.8647e-1 (6.58e-3) -	2.6383e-1 (9.37e-3) -	2.2692e-1 (9.83e-3) -	2.8452e-1 (3.65e-3) -	3.3137e-1 (7.76e-3) -	2.8087e-1 (5.26e-3) -	1.9167e-1 (4.16e-3)
	12	21	5.1003e-1 (3.56e-2) -	3.6623e-1 (1.58e-2) -	3.1434e-1 (6.92e-3) -	2.8541e-1 (1.30e-2) -	2.6177e-1 (3.70e-2) -	3.6651e-1 (2.01e-2) -	3.9457e-1 (1.50e-2) -	3.2927e-1 (6.47e-3) -	2.4083e-1 (7.13e-3)
			1.8457e-1 (9.86e-3) -	1.3235e-1 (4.17e-3) -	1.4890e-1 (5.34e-3) -	1.4217e-1 (4.47e-3) -	1.2137e-1 (2.31e-3) -	1.2645e-1 (3.63e-3) -	1.3481e-1 (2.99e-3) -	1.2978e-1 (2.50e-3) -	9.4635e-2 (1.85e-2)
	5	14	2.2898e-1 (1.62e-3) -	1.6457e-1 (5.85e-3) -	1.8158e-1 (6.68e-3) -	2.0003e-1 (9.87e-3) -	1.5446e-1 (1.67e-3) -	1.6885e-1 (8.57e-3) -	1.9944e-1 (7.93e-3) -	1.4324e-1 (1.75e-3) +	1.4883e-1 (7.64e-3)
			4.5956e-1 (3.07e-2) -	2.8503e-1 (6.94e-3) -	2.8647e-1 (6.58e-3) -	2.6383e-1 (9.37e-3) -	2.2692e-1 (9.83e-3) -	2.8452e-1 (3.65e-3) -	3.3137e-1 (7.76e-3) -	2.8087e-1 (5.26e-3) -	1.9167e-1 (4.16e-3)
	12	21	5.1003e-1 (3.56e-2) -	3.6623e-1 (1.58e-2) -	3.1434e-1 (6.92e-3) -	2.8541e-1 (1.30e-2) -	2.6177e-1 (3.70e-2) -	3.6651e-1 (2.01e-2) -	3.9457e-1 (1.50e-2) -	3.2927e-1 (6.47e-3) -	2.4083e-1 (7.13e-3)

表14 在 MaF 系列测试问题上的 IGD 值统计结果 (续表)												
问题	M	D	MOEA-D/ DQN	CMOCSO	DKCA	MOEA-NZD	NRV-MOEA	TriP	DRLOS-EM- CMO	APSEA	DQN- MaOEA	
MaF3	10	19	2.9221e-1 (5.31e-3) -	1.6816e-1 (2.27e-3) +	1.6189e-1 (2.15e-3) +	2.1228e-1 (1.91e-2) +	1.5538e-1 (5.62e-3) +	1.6213e-1 (3.41e-3) +	1.6786e-1 (2.24e-3) +	1.6627e-1 (3.35e-3) +	2.7331e-1 (1.66e-2)	
		12	21	3.4669e-1 (2.96e-2) -	2.3133e-1 (8.39e-3) +	2.1000e-1 (1.17e-2) +	2.3238e-1 (2.25e-2) +	1.6175e-1 (4.92e-3) +	1.9442e-1 (5.82e-3) +	2.0124e-1 (4.06e-3) +	1.9396e-1 (7.51e-3) +	3.2445e-1 (2.06e-2)
	5	14	9.3279e+2 (1.85e+3) -	4.1123e+7 (9.53e+7) -	1.0448e+4 (6.25e+3) -	7.9775e-1 (1.48e+0) -	7.4596e+1 (1.11e+2) -	3.7701e+4 (1.15e+5) -	3.1052e+5 (5.31e+5) -	2.9195e+1 (6.90e+1) -	1.8287e-1 (1.61e-2)	
		10	19	1.2638e-1 (1.83e-2) -	1.8205e+11 (7.60e+10) -	8.4069e+4 (7.10e+4) -	1.1684e+6 (6.33e+6) -	2.5835e+10 (1.59e+10) -	4.1818e+6 (1.31e+7) -	5.9162e+11 (1.62e+11) -	1.8883e+12 (3.43e+11) -	1.1243e-1 (7.03e-3)
	12	21	2.9732e-1 (5.22e-1) -	3.9015e+11 (1.49e+11) -	2.5473e+7 (1.38e+8) -	8.0474e+5 (4.37e+6) -	2.3884e+10 (1.46e+10) -	3.2408e+7 (1.31e+8) -	1.0570e+12 (2.92e+11) -	2.3340e+12 (4.59e+11) -	1.2145e-1 (9.86e-3)	
		5	14	3.0385e+2 (4.18e+2) -	1.0683e+3 (1.13e+3) -	5.6098e+1 (3.18e+1) -	2.5494e+1 (2.40e+1) -	2.8355e+1 (9.01e-1) -	9.7648e+1 (1.40e+2) -	3.6497e+1 (4.18e+1) -	2.1989e+1 (3.47e+1) -	9.0721e+0 (1.74e+0)
	10		19	1.7412e+2 (1.20e+1) +	5.4244e+3 (5.42e+3) -	5.2292e+3 (2.67e+3) -	3.3115e+3 (1.77e+3) -	3.6961e+2 (4.03e+2) -	7.5838e+1 (2.49e+1) +	1.9403e+4 (8.30e+3) -	6.6316e+2 (4.64e+2) -	2.7611e+2 (3.47e+1)
		12	21	1.6009e+3 (1.63e+3) -	1.7030e+5 (5.28e+4) -	2.6163e+4 (1.38e+4) -	3.4662e+3 (3.01e+3) -	3.6742e+3 (3.81e+3) -	2.9194e+3 (3.11e+3) -	9.1652e+4 (4.34e+4) -	2.0833e+3 (1.66e+3) -	1.3062e+3 (2.62e+2)
	MaF5		5	14	4.3700e+0 (1.02e+0) +	2.7305e+0 (1.31e-1) +	2.4123e+0 (8.48e-2) +	2.3704e+0 (7.92e-3) +	2.4335e+0 (5.74e-2) +	2.9842e+0 (4.91e-1) +	3.0690e+0 (2.22e-1) +	2.2967e+0 (2.56e-1) +
		10		19	1.2770e+2 (5.43e+0) +	1.6302e+2 (8.84e+0) +	9.8657e+1 (9.29e+0) +	7.8371e+1 (9.34e+0) +	5.8908e+1 (4.09e+0) +	1.6361e+2 (1.58e+1) +	1.5696e+2 (1.10e+1) +	1.7490e+2 (1.27e+1) +
12		21	6.9705e+2 (6.10e+1) +	6.6895e+2 (7.51e+1) +	3.4007e+2 (3.10e+1) +	3.8307e+2 (2.44e+1) +	2.4113e+2 (1.57e+1) +	7.2300e+2 (6.93e+1) +	6.3509e+2 (6.36e+1) +	6.9384e+2 (8.54e+1) +	9.2308e+2 (1.85e+2)	
	MaF6	5	14	6.8429e-2 (3.94e-4) -	7.2968e-1 (1.41e+0) -	5.7885e-3 (3.32e-4) +	6.5711e-2 (2.73e-2) -	4.9354e-3 (2.45e-4) +	1.4158e-2 (1.04e-2) +	4.1206e-3 (3.75e-5) +	4.0877e-3 (3.73e-5) +	3.4419e-2 (1.16e-2)
10			19	9.1849e-2 (3.15e-5) -	2.1454e+1 (3.02e+1) -	3.5884e+0 (1.62e+0) -	3.1694e+0 (1.27e+0) -	4.4127e+0 (1.98e+0) -	2.2027e-3 (4.37e-4) +	1.8792e+1 (1.90e+1) -	2.6198e+1 (3.29e+1) -	8.5503e-3 (3.25e-3)
12		21	1.1857e-1 (4.19e-5) -	3.5998e+1 (3.07e+1) -	4.0483e+0 (1.96e+0) -	2.0197e+0 (1.03e+0) -	6.3465e+0 (2.62e+0) -	2.9514e-3 (2.35e-4) +	3.1637e+1 (3.04e+1) -	7.9628e+1 (4.17e+1) -	1.2053e-2 (1.30e-3)	
	MaF7	5	24	5.9152e-1 (6.65e-2) =	3.3145e-1 (1.79e-2) +	3.7677e-1 (1.06e-2) +	3.7720e-1 (2.49e-2) +	2.9074e-1 (7.68e-3) +	5.0145e-1 (1.68e-1) +	3.3698e-1 (1.05e-2) +	3.0327e-1 (1.10e-2) +	5.9528e-1 (2.57e-2)
10			29	1.5802e+0 (8.77e-2) -	1.4199e+0 (7.12e-2) -	1.2445e+0 (1.14e-2) -	1.1343e+0 (5.12e-2) -	1.9574e+0 (1.02e-2) -	1.3311e+0 (6.03e-2) -	1.5735e+0 (4.01e-2) -	1.8467e+0 (1.41e-1) -	1.0823e+0 (2.40e-2)
12		31	2.1083e+0 (2.92e-1) -	1.8111e+0 (6.22e-2) -	1.3512e+0 (1.64e-2) -	1.4797e+0 (1.00e-1) -	1.4076e+0 (1.53e-2) -	1.7162e+0 (3.73e-2) -	1.8322e+0 (4.19e-2) -	2.2345e+0 (1.72e-1) -	1.2965e+0 (2.27e-2)	
	+/-/≈			4/16/1	6/15/0	7/14/0	6/15/0	7/14/0	9/12/0	7/14/0	8/13/0	

4.6 在实际物流车辆路径问题实例上的验证

为了进一步检验 DQNMaoEA 算法在实际应用中的有效性,本文选择顺丰科技的支线物流真实场景进行应用,研究了所提算法的实际应用能力。

4.6.1 实际问题实例描述

顺丰速运是中国及亚洲最大、全球第四大综合物流服务提供商,其提出了基于“集装”场景的“支线散货”动态调度物流运输问题。该问题是指在集装运输模式下,针对支线运输中散货的物流问题进行动态调度的过程,如图 12 所示。在研究的问题场景中,没有固定规格和包装的货物,通过对货物装笼

(集装)、车辆分配以及运输路线进行灵活动态的安排,最终将中转场的货物通过支线运输到各网点,以达到满足散货时效、减少用车数量、降低路径长度、提高货物装载率以及车辆复用率等目标。集装模式将没有固定规格和包装的货物装至固定且统一规格的容器“笼”中,能够提高货物运输过程的装卸速度,提升车辆载位的管理。在支线运输中,由于货物数量和种类的不确定性,以及运输需求的临时性和多样性,需要进行动态调度来有效地管理和运输这些散货物流。

基于“集装”场景的“支线散货”动态调度物流问

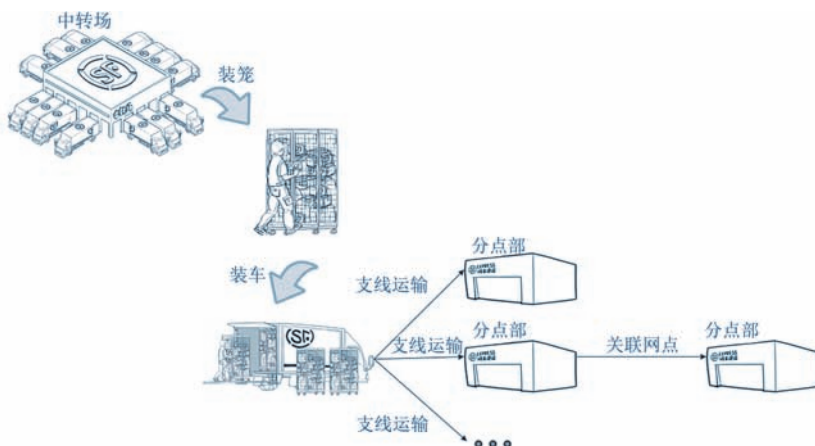


图12 顺丰科技物流车辆路径规划问题业务场景示意图

题作为一个复杂而又实际的物流问题,需要综合考虑货物、运输工具、运输路线等多方面的因素,来实现最优的动态调度和物流管理。因此,可将基于“集装”场景的“支线散货”动态调度物流问题建模为一个带有多个目标的优化问题,具有一定复杂性。通过科学的调度算法和技术手段,求解该超多目标优化问题,可以实现散货的高效运输和物流配送,提高物流运输效率和服务质量。

4.6.2 实例模型假设

(1)模型假设

对待求解实际问题实例模型做出如下假设:
①支线运输车辆从中转场出发,经停各网点后不考虑回场,仅提频车辆涉及返回中转场,重新装载货物进行发运。其中,设定中转场唯一;
②网点的货运需求均小于运输车容量时,一个网点需求被一辆运输车一次停靠满足。若实际网点不满足要求,则进行数据处理工作,将网点需求拆分为多个合规的固定容量需求,并将各拆分需求包装为新的网点;
③各网点的货运需求、路径距离、运输车容量等条件已知;
④装运笼的尺寸及容量已知且唯一;
⑤不考虑各网点之间的货运需求。

(2)数学建模

构建实例模型时,需同时考虑影响模型效率和成本的三个重要因素:货物装载、路径规划、车辆分配,以及各因素之间的可行性和完整性。因此,模型设置了多个相关优化目标以及约束条件,以得到尽可能同时满足不同影响因素需求的最终解集。在货物装载方面,将各网点的货物需求进行装笼,且考虑不同网点的拼笼问题,尽可能减少车辆的未装载量;在路径规划方面,可将问题的定义为弹性时间窗的开放路径优化问题,其中运输货物容量不可超过车

的容量,装笼后的笼数不可超过规定值,且最大限度满足各网点的时间要求,最终实现最小化用车成本、路径成本以及时间惩罚成本;而在车辆分配方面,考虑车辆的完成运输时间以及各网点的时间窗要求之间的关系,可以实现车辆的复用,提高同批次车辆运输的复用率。

①解的表达

如图13所示,共有三条从中转场出发的路径。由于路径 r_1 与 r_3 满足车辆复用条件,两条路径可由同一辆车运输,该车辆完成1、2号网点运输服务后,需返回中转场,进而服务6、7号网点。而路径 r_2 则无需考虑返场问题。

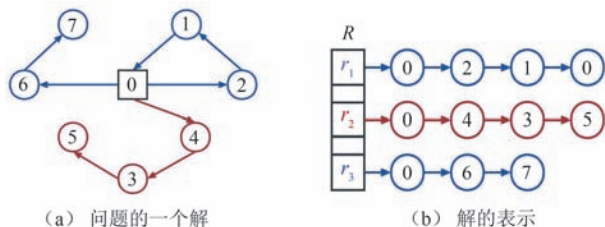


图13 顺丰科技物流车辆路径规划问题求解示意图

②参数描述

在表15中,对问题的参数设置进行了详细阐述:

③目标函数

$$\text{Minimize} \begin{cases} f_1 = K \\ f_2 = \sum_{i=1}^R \sum_{j=0}^{|r_i|-1} d_{j,j+1} \\ f_3 = \alpha \sum_{i=1}^R \sum_{j=0}^{|r_i|} \max(0, b_j - a_j) \\ f_4 = \beta \sum_{i=1}^R \sum_{j=0}^{|r_i|} \max(0, a_j - e_j) \\ f_5 = \sum_{i=1}^R \sum_{j=0}^{|r_i|} C - g_j \\ f_6 = K / |\text{Route}| \end{cases} \quad (12)$$

目标 $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ 中各个目标的定义, f_1 是车辆数量, f_2 是总行驶距离, f_3 是由于提前到达而造成的总等待时间, f_4 是由于迟到而造成的总延迟时间, f_5 是所有路径未装满的容量和, f_6 在数值计算中反映了车辆的重用率。最小化 f_1 旨在降低购买(或租用)和维护车辆的固定成本。车辆路径问题中的可变成本使用一个总旅行距离的函数 f_2 来估计。因此, f_1 和 f_2 可以被视为运输成本。 f_3, f_4 可以看作是网点满意度相关的服务成本。最小化 f_5 可提高每辆车的装载率, 避免浪费笼车分配时的容量资源。 f_6 中 K 表示使用的车辆数, $|R|$ 表示总路径数, 因此, $K/|R|$ 可在数值上反映车辆的复用率, 目标 f_6 值越小, 车辆的复用率越高。

表 15 问题参数设置

参数	描述
$v = \{0, \dots, N\}$	仓库和用户节点编号, 节点 0 是仓库, 其他的是网点
C	车辆容量
K	车辆数
L	最大装车笼数
g_i	网点 i 需求量
s_i	网点 i 的服务时间
l_i	网点 i 的装载笼数
$[b_i, e_i]$	网点服务的时间窗口, 其中, b_i 和 e_i 分别表示网点最早和最晚服务时间
a_i	车辆到达网点 i 的时间
d_{ij}	网点 i, j 之间的运输距离
t_{ij}	网点 i 和 j 之间的运输时间
α	早于 b_i 到达网点 i 并开始服务的惩罚系数
β	早于 e_i 到达网点 i 并开始服务的惩罚系数
$Route = \{r_1, \dots, r_k\}$	包含 n 条车辆路径的总路径, r_1, r_n 是其中的一条路径
$r_n = \langle 0, \dots, v_i, v_j, \dots, [0] \rangle$	车辆路径经过的节点, 0 表示仓库, v_i, v_j 是途经的网点

④约束

约束(a), (b)表示车辆受装载能力限制以及笼数限制, 每辆车的装载量不超过其最大载重量限值, 且装载笼数不能超过最大笼数要求。约束(c), (d)表示所有车辆均由配送中心出发, 每个网点仅由一辆车服务, 且仅提频车辆考虑返场问题, 其余车辆均不考虑。约束(e)表示决策变量的属性。

$$\left\{ \begin{array}{l} \sum_{j=0}^{|r_i|} g_j \leq C \quad (a) \\ \sum_{j=0}^{|r_i|} l_j \leq L \quad (b) \\ v_i! = v_j \quad \forall v_i, v_j \in r_k, r_k \in Route(c) \\ \left(\sum_{i=1}^{|Route|} (|r_i| - 1) \right) - (|Route| - K) \leq N(d) \\ v_i, v_j \in \{1, \dots, N\} (e) \end{array} \right. \quad (13)$$

4.6.3 实验验证

为了验证DQNMaOEA的实际性能, 根据4.5.2小节所述的数学模型, 对支线散货动态调度物流问题进行了建模, 选取了部分真实的物流数据并进行预处理后, 形成了最终的测试问题实例SF1-SF7。需要强调的是, SF1-SF7这7个问题实例均是基于4.5.2小节描述的数学模型设计而来。如表16中的问题规模设置所示, SF1-SF4问题由同一物流片区内, 选取100、150、200、245个不同数量网点的真实数据建模而成; SF5-SF7问题则选取网点需求规模不同的三个片区的真实数据建模而成, 且不同的问题由不同容量的运输车进行服务, 并将网点规模设置为200。在该超多目标优化问题实例上, 对DQNMaOEA与MOEA/D-DQN、CMOCSO、DKCA、MaOEA-PDE、NRV-MOEA、TriP、DRLOS-EMCMO和APSEA进行了比较, 设置实验最大函数评估次数为100 000, 每个算法在测试问题上运行了20次。

表 16 SF1-SF7规模设置

问题	客户节点数量	所属片区
SF1	100	片区1
SF2	150	片区1
SF3	200	片区1
SF4	245	片区1
SF5	200	片区2
SF6	200	片区3
SF7	200	片区4

表17是算法在SF1-SF7实际问题上的IGD结果对比。其中, DQNMaOEA在IGD值上共获得了6个最优结果, 具有良好的收敛性。同时, 求解SF3实际问题的最终结果也近似于最优算法CMOCSO的结果, 搜索到的最终种群的分布接近Pareto真实前沿。总体上, DQNMaOEA取得了6个最优, 较其他算法更优。同时, 在图14中对各算法求解SF系列问题的平均运行时间进行了可

视化对比,结果显示 DQNMaoEA 在求解 SF1-SF7 时,所需求解时间较少,求解速率较优,在一定程度上满足了求解实际问题时的实时性要求。

综上所述,在求解 SF1-SF7 实际问题上的最优效果以及求解速率证明了本文提出算法 DQNMaoEA 求解实际问题的能力。

表 17 在 SF1-SF7 实际测试问题上的 IGD 值统计结果

问题	M	D	MOEA/D-DQN	CMOCSO	DKCA	MaOEA-PDE	NRV-MOEA	TriP	DRLOS-EMCMO	APSEA	DQNMaoEA
SF1	6	100	3.9068e+3 (5.06e+1)-	3.9365e+3 (6.84e+1)-	3.9724e+3 (5.88e+1)-	3.9349e+3 (5.09e+1)-	3.9169e+3 (8.69e+1)-	3.9487e+3 (4.16e+1)-	3.9940e+3 (5.09e+1)-	4.0021e+3 (7.21e+1)-	3.9003e+3 (4.91e+1)
SF2	6	150	6.1829e+3 (1.83e+1)-	6.1551e+3 (4.23e+1)-	6.1479e+3 (6.61e+1)-	6.1219e+3 (7.42e+1)-	6.1320e+3 (9.72e+1)-	6.2205e+3 (8.70e+1)-	6.1999e+3 (8.01e+1)-	6.1839e+3 (1.06e+2)-	6.0950e+3 (6.66e+1)
SF3	6	200	8.2639e+3 (9.43e+1)-	8.1988e+3 (1.22e+2)+	8.2017e+3 (9.24e+1)+	8.2535e+3 (4.53e+1)-	8.2425e+3 (8.41e+1)-	8.2380e+3 (7.46e+1)-	8.3351e+3 (9.23e+1)-	8.3563e+3 (1.03e+2)-	8.2236e+3 (1.19e+2)
SF4	6	250	1.0215e+4 (1.09e+2)-	1.0277e+4 (6.74e+1)-	1.0210e+4 (8.30e+1)-	1.0280e+4 (5.09e+1)-	1.0261e+4 (6.91e+1)-	1.0277e+4 (3.39e+1)-	1.0406e+4 (4.83e+1)-	1.0404e+4 (5.42e+1)-	1.0196e+4 (8.14e+1)
SF5	6	200	7.8741e+3 (7.60e+1)-	7.8470e+3 (6.61e+1)-	7.8363e+3 (5.89e+1)-	7.8580e+3 (7.63e+1)-	7.8250e+3 (5.36e+1)-	7.8104e+3 (8.81e+1)=	7.8730e+3 (6.93e+1)-	7.8949e+3 (9.73e+1)-	7.8102e+3 (7.22e+1)
SF6	6	200	8.4307e+3 (5.95e+1)-	8.4295e+3 (4.74e+1)-	8.4335e+3 (5.50e+1)-	8.4196e+3 (4.35e+1)-	8.3864e+3 (6.56e+1)=	8.4035e+3 (7.68e+1)-	8.4977e+3 (7.01e+1)-	8.5339e+3 (6.08e+1)-	8.3846e+3 (7.99e+1)
SF7	6	200	8.3285e+3 (8.54e+1)-	8.3759e+3 (5.11e+1)-	8.3443e+3 (4.40e+1)-	8.3627e+3 (5.89e+1)-	8.3386e+3 (7.59e+1)-	8.3711e+3 (5.09e+1)-	8.4469e+3 (9.10e+1)-	8.4353e+3 (7.10e+1)-	8.2996e+3 (8.23e+1)
+/-/≈			0/7/0	1/6/0	1/6/0	0/7/0	0/6/1	0/6/1	0/7/0	0/7/0	

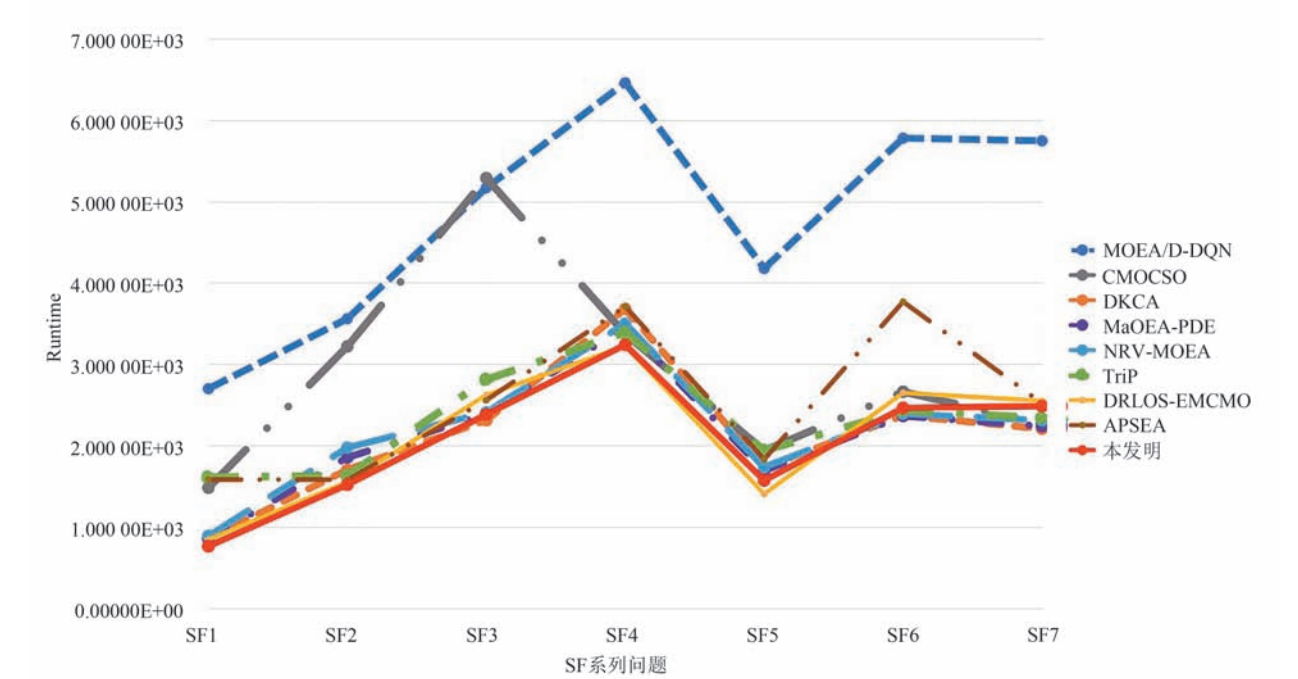


图 14 SF 下各算法运行时间对比

此外,如图 15 所示,为了更清晰地对比算法在各目标函数上的最终结果以及搜索到的最终路径规划,以 SF2 问题为例,可视化了 MOEA/D-DQN、CMOCSO、DKCA、MaOEA-PDE、NRV-

MOEA、TriP、DRLOS-EMCMO、APSEA 以及 DQNMaoEA 九个算法搜索得到的车辆路径规划图,以及对比了各算法求解得到的最终目标值。

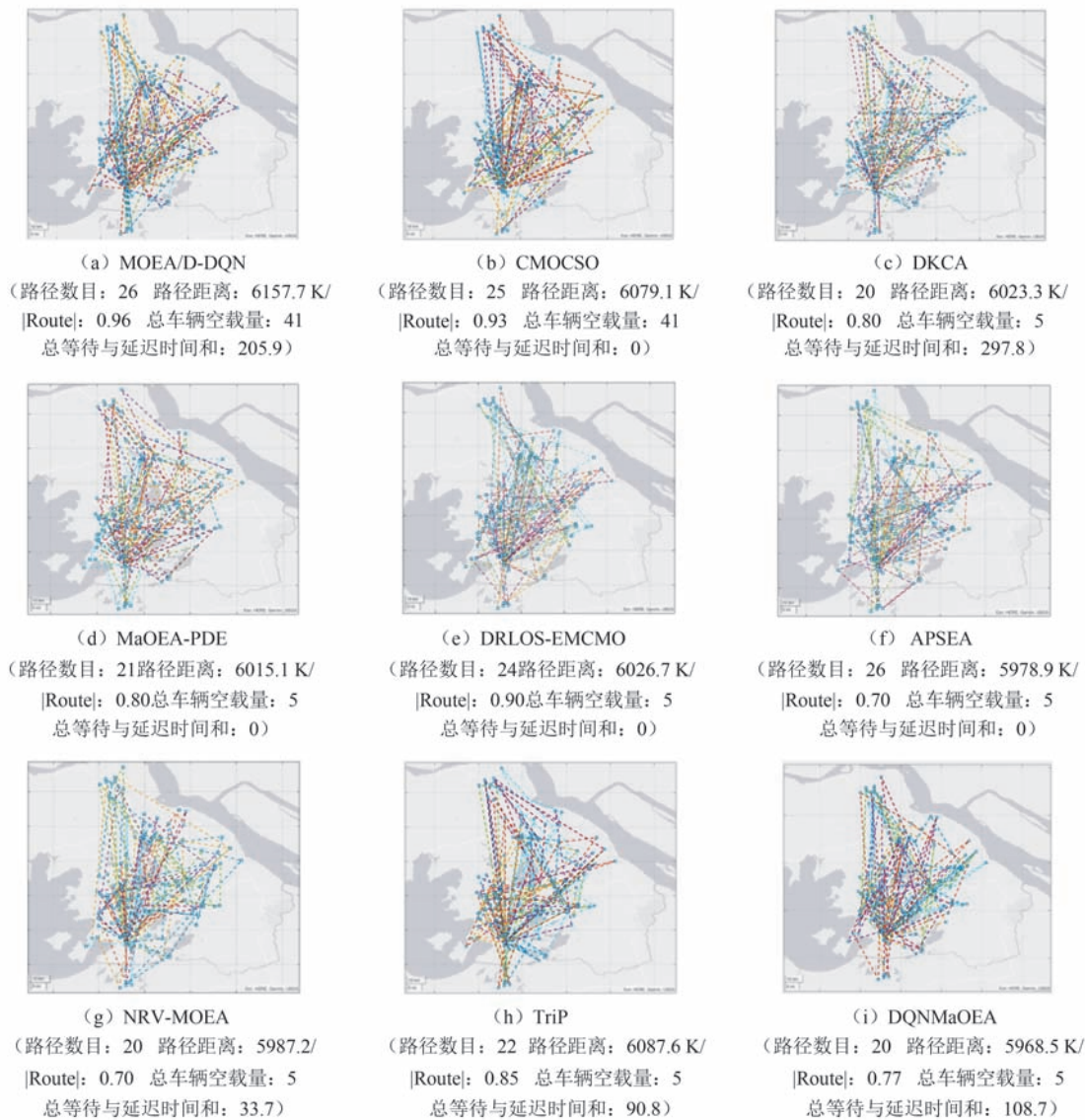


图 15 算法搜索到SF2问题的路径规划图

5 结 论

为高效求解具有挑战性的超多目标优化问题, 本文提出了一种新的深度强化学习引导的多种群协同进化超多目标优化算法 DQNMaoEA。基于深度强化学习的自适应子种群选择策略, 在环境选择时, 选择高质量的父代子种群, 最终产生更具多样性和收敛性的子代解, 并进一步提高了算法在高维目标空间的搜索能力。为了降低计算成本, 提高算法的搜索效率, 提出了一种自适应子种群计算资源分配策略, 根据当前子种群对整个种群优化过程的效用值改进占比, 动态设置子种群的适应值评估次数。

经实验验证, 证明了两种策略的有效性, 提高了算法在求解不同问题实例时的搜索精度以及时间效率。此外, 通过在大量基准测试问题以及实际物流车辆路径问题上的实验结果表明, 与最先进的不同类型算法相比, 本文提出的 DQNMaoEA 取得总体上较优的效果, 为解决超多目标优化问题以及实际问题提供了有竞争力的解决方案。

未来, 我们将进一步研究其他多目标优化算法框架与深度强化学习方法的融合, 并研究算法求解其他具有挑战性的实际问题的性能。此外, 算法在解决不规则 Pareto 前沿问题时效果欠佳, 针对这一问题, 值得进一步深入研究。