乳胶类文件杂志,卷。 ,不。 ,2022 年 8 月

使用元启发式的自动化设计新将军中的强化学习搜索框架

Wenjie Yi, Rong Qu, Senior Member, IEEE, Licheng Jiao, Fellow, IEEE, Ben Niu, 会员, IEEE

摘要 元启发式算法已被广泛研究以解决高度复杂的组合优化问题。然而,大多数元启发式算法都是由不同专业的研究人员手动设计的,没有一致的框架来支持有效的算法设计。

本文提出了一个通用的搜索框架,以统一的方式制定一系列不同的元启发式算法。该框架定义了通用算法组件,包括选择启发式和进化运算符。统一的通用搜索框架旨在作为自动化算法设计分析算法组件的基础。通过建立新的通用搜索框架,开发了两种基于强化学习的方法,即基于深度 Q 网络和基于近端策略优化的方法,以自动设计一种新的基于通用种群的算法。

所提出的基于强化学习的方法能够在优化过程的不同阶段智能地选择和组合适当的算法组件。所提出的基于强化学习的方法的有效性和泛化性在具有时间窗的有能力车辆路径问题的不同基准实例中得到了全面验证。这项研究有助于通过有效的机器学习支持基础分析的通用框架,朝着自动化算法设计迈出关键一步。

索引词 自动算法设计,通用搜索框架,元启发式,强化学习,车辆路径问题。

一、引言

ADDRESSING 高度复杂的组合优化
具有各种现实世界约束的问题 (COP)
已被证明是当前进化计算中的研究挑战之一。当前最先进的技术包括元启发式算法,这些算法成功地在合理的计算时间内找到了高质量的解决方案。

然而,文献中提出的大多数元启发式算法仅适用于特定的问题实例或解决问题的特定阶段,并且在很大程度上依赖于人类专家的经验。为了解决这个问题,自动算法设计最近引起了研究界的极大关注 [1]、[2]。

W. Yi 和 R. Qu 在英国诺丁汉大学计算机科学学院的计算优化和学习 (COL) 实验室工作(电子邮件:wenjie.yi@nottingham.ac.uk;rong.qu@nottingham .ac.uk)。

L. Jiao 就职于中国西安电子科技大学人工智能学院智能感知与图像理解教育部重点实验室(e-mail:lchjiao@mail.xidian.edu.cn)。

B. Niu 来自中国深圳大学管理学院 (e-mail:drniuben@gmail.com)。

通讯作者:R. Qu, B. Niu。

对于自动化算法设计,设计元启发式问题本身在 [3] 中被定义为组合优化问题,基于不同决策变量的搜索空间,例如算法参数、算法组合或算法组件。因此,根据算法搜索空间中考虑的不同类型的决策变量,该领域的研究可以分为自动算法配置、算法选择和算法组合[3]。第一类旨在自动配置特定类型或算法族的参数。第二类侧重于选择候选算法或结合几个现有算法针对问题/实例特征。与这两个类别相比,通过组合基本算法组件,自动算法组合旨在生成通用算法来解决多个 COP,即生成的算法不属于任何特定的搜索算法,例如遗传算法或粒子群优化, ETC。

算法配置可以确定性能良好的参数设置;然而,它需要关于应该使用哪种特定算法的充分先验知识。算法选择解决了第一类的局限性;然而,它引入了识别问题关键特征的困难和复杂问题。自动化算法组合旨在灵活地组合和生成新的算法;然而,仍然需要一些人类专业知识来预选现有框架中的候选启发式方法。本研究属于第三类,旨在研究在统一框架内自动设计搜索算法的基本组件。

在文献中,强化学习 (RL) [4] 已被用于通过将算法设计问题建模为马尔可夫决策过程 (MDP) 来自动设计算法。 RL 是一种学习技术,代理根据其与环境的交互确定每个状态下的最佳动作。在环境的每个新状态下,代理从一组动作中选择一个动作。基于执行每个选定动作后的奖励或惩罚,它通过反复试验形成状态-动作对,学会智能地选择当前状态下的动作 [5]。

一些研究人员使用最简单的表格 RL 技术,例如 SARSA [4] 和 Q-learning (QL) [6] 进行进化算法设计。应用表格 RL 的一个研究问题与连续状态空间的离散化有关,这会导致不可靠的结果 [7]、[8]。在

本文提出的这项研究,神经网络函数逼近适用于处理连续状态,以解决上述问题。此外,关于 RL 技术的研究较少,以支持进化算法的有效设计,以解决受约束的 COP,例如带时间窗的 Capacitated Vehicle Routing Problem (CVRPTW)。

为了支持自动设计有效的 COP 元启发式算法,首先建立了一个通用搜索框架,在该框架内,学习技术可以应用于算法的设计空间,从而支持自动算法设计。在这个阶段的研究中,我们没有研究所有的算法组件,而是专注于研究对算法性能影响最大的关键演化算子的自动组合这一关键问题。 RL 用于自动算法组合,以根据其性能奖励或惩罚关键进化运算符的组合。研究工作旨在做出以下贡献:

·建立了一个新的通用搜索框架(GSF)来制定不同的基于单一解决方案和基于群体的算法。统一的GSF作为分析算法组件的基础,自动生成有效的CVRPTW搜索算法。 ·自动算法组合过程被制定为MDP。在拟议的 GSF 中研究了两种 RL 方法,即深度 Q 网络 (DQN) [9] 和近端策略优化 (PPO) [10],以解决在过程中自动选择和组合最有效的进化算子的关键问题演化的不同阶段。 CVRPTW 的结果证明了与没有学习的搜索过程相比,经过训练的策略的有效性。 ·通过将训练有素的策略直接应用于新的 CVRPTW 实例,进一步验证了训练策略的泛化。

除了在 DQN 和 PPO 模型中提取和保留的知识外,基于 RL 的技术的训练时间也由从头开发新模型和算法以解决新问题实例所需的时间和专业知识证明是合理的。

本文的其余部分的结构如下。第二部分介绍了现有的自动化算法设计框架和这些框架内的强化学习技术的相关工作。第三部分描述了所提出的通用搜索框架和学习技术。在第四节中,描述了车辆路径问题的优化模型,并在基准数据集上分析了实验结果,而第五节提出了结论并讨论了未来的研究。

二。相关工作

现有文献中的大多数进化算法和元启发式算法都是由具有不同专业知识的研究人员手动设计的,其中许多针对手头的特定问题采用了特别选择的算法。构建通用搜索框架以支持有效算法设计的工作相对较少。

A. 自动化算法设计的现有框架

算法设计问题已正式建模为 COP,即 [3] 中的通用组合优化问题 (GCOP) 模型。基于决策空间的根本区别,自动化算法设计可以分为算法配置、算法选择和算法组合三类。文献中已经开发了许多框架来支持这些不同类别中的自动化算法设计任务。

自动算法配置旨在在给定的一组问题实例中找到目标算法的性能良好的参数设置。为支持此任务而构建的框架包括 ParamILS [11],它利用迭代本地搜索、F-race [12] 和 irace [13],两者都使用赛车机制,以及基于代理的方法,例如 SPOT [14], SMAC [15]、MIP-EGO [16] 和 Hyperopt [17]。

在自动算法选择中,一个特定的算法或一组算法是在逐个实例的基础上自动选择的。开发的框架包括 PAP [18],它集成了不同的进化算法来解决数值优化问题,以及 Hydra [19],具有用于基于投资组合的算法选择的配置技术,以及基于机器学习的算法选择器 [20]。

在自动算法组合中,一组启发式自动组合以生成新算法来解决跨不同问题域的实例。研究最多的技术是超启发式 [21],它广泛涉及在给定情况下智能地选择或生成适当的启发式。开发的框架包括 HyFlex [22]、EvoHyp [23] 和 SHH [24] 等。HyFlex 探索低级启发式或启发式运算符的决策空间(例如,将来自十种知名技术的搜索运算符作为构建块 [25]]) 而 EvoHyp 将进化算法改编为高级策略。 SHH 专门用于自动组合群体智能算法的不同组件 [24]。此外,已经在特定元启发式模板中构建了一些组合框架,例如 CMA-ES [26] 和 PSO-DE [27]。

近来自动化算法组合的快速增长是由于它具有更大的潜力来生成更通用的搜索算法来解决复杂的 COP。它不受现有特定搜索算法模板的限制。本研究侧重于自动算法组合问题,借鉴高级强化学习以进行有效的算法设计。

尽管现有的自动算法组合框架(例如 HyFlex、EvoHyp 和 SHH)已成功用于解决各种 COP,但仍存在一些局限性。

HyFlex 需要一组预定义或特定于问题的启发式方法,而不是基本的算法组件,以针对更广泛的问题生成更通用和更强大的搜索算法。 EvoHyp 预定义了选择算子和进化算子,而 SHH 混合了这两类算子。因此,这些框架建立在算法设计的减少搜索空间之上,然而,导致失去一些基本组件的有利组合,这可能

永远无法获得或探索。

随着算法设计的新标准,即 [3] 中建立的 GCOP,本研究系统地研究了统一 GSF 中的学习技术,以支持自动化算法设计。

B. 自动化算法组合中的强化学习

在最近关于自动算法组合的文献中,一些 RL,如 SARSA [4]、QL [6] 和 DQN [9],已被用于支持智能选择最合适的启发式运算符。他们在搜索过程的不同阶段利用有关操作员绩效的反馈信息。根据动作空间的定义方式,该领域的研究可以分为两类。

自动算法组合中的第一类 RL 技术将特定类型搜索算法中的运算符定义为 RL 代理的可选操作。在文献中,RL 技术主要应用于进化算法,例如遗传算法,以选择有效的变异和交叉算子 [7]。 Travel ling Salesman Problem 和 0-1 Knapsack Problem 的结果证明了这种自动化方法的优越性 [7]、[28]、[29]、[30]。

然而,由于 RL 技术的复杂性,该领域的大多数研究 [7]、[28]、[29] 仅侧重于使用最简单的表格 RL 方法,例如 SARSA 和 QL。很少有研究调查在应用 RL 来选择进化算子时处理连续状态空间的先进技术 [30]。在进化计算中有效和高效的自动化算法设计中,缺乏对高级 RL 的研究。

第二个研究类别将特定问题的启发式方法视为 RL 代理的可选操作。 RL 技术被用作高级策略,在超启发式算法中自动组合不同的低级启发式算法。这些基于 RL 的无人机方法 [31] 和 HyFlex 软件框架 [5] 内的不同 COP 的结果证明了这些方法的有效性。

在这些研究中,几个依赖于搜索的特征,即搜索过程本身的观察,已经被用来表示状态。然而,识别出的特征数量有限且不足以用于学习。此外,使用简单的正/负奖励方案,不能准确反映所选动作的效果。此外,通常不清楚超启发式框架内的 RL 技术是如何设计的,即缺乏对 RL 的三个基本要素(即状态、动作和奖励方案)的明确定义。这一研究领域仍然存在很大的范围和差距,因为重新实施完全相同的方法并随后复制实验通常具有挑战性。

在这项研究中,我们将两种 RL 技术与神经网络函数逼近器应用于自动算法组合的学习中。仔细定义了具有足够特征以进行有效学习的状态空间。动作空间为定义为在通用搜索算法的自动化设计中学习可重用知识的基本算法组件。

此外,还定义了有效的奖励方案以鼓励 RL 系统找到有效的搜索策略。应该注意的是,这项研究采用离线 RL 框架,其中策略是离线训练但以在线方式使用。

这与文献中大多数基于 RL 的自动化算法组合方法不同。

三、在通用搜索框架内学习

A. 通用搜索框架

文献中的进化算法和元启发式算法遵循类似的由选择和繁殖驱动的人工进化的基本哲学。特定元启发式的演化和搜索过程是有区别的,主要取决于选择启发式和演化算子。

在分析元启发式算法基本方案的基础上,开发了通用搜索框架(GSF),如图1所示。该框架由表一所示的五个模块组成,用于更新个人和表三所示的四个档案,用于存储个人。对于这些组件中的每一个,可以选择不同的设置、启发式或参数,如表 V-VII 所示,以在 GSF 中自动组合和设计不同的通用搜索算法。由启发式和运算符的组合表示的算法被设置为输出。

关于初始化,虽然已经开发了一些特定于问题的启发式(hp),但大多数现有研究通常采用“纯随机”(hr)策略。两个最常见的终止标准是计算时间(ht)和种群收敛(hc)。在图 1 中显示的五个模块中,进化选择、进化和替换选择对搜索性能的贡献更大。因此,它们将在下一节中详细讨论。

表一
GSF中的模块

模块	不同的启发式、操作符或参数随机(hr)、问题特
初始化	定(hp)基于概率的操作符(h1、h2、h3)、确
进化选择	定性操作符(h4、h5、h6)变异(Omutation)、交叉(Ocrossover)
进化	
替换逗号选择(h7),加号选择(h8)的选择	
终止计算时间(ht),收敛(hc)	

所提出的 GSF 能够通过为模块和档案设置不同的参数,以统一的方式制定一系列基于单一解决方案的算法和基于人口的算法,如表 II 所示,例如不同的人口规模,四个档案和Selection for Evolution 模块中的启发式集。本文侧重于强化学习对基于人口的搜索算法的自动化设计。

表 III 显示了 GSF 中定义的档案。在进化选择模块中,一些个体在

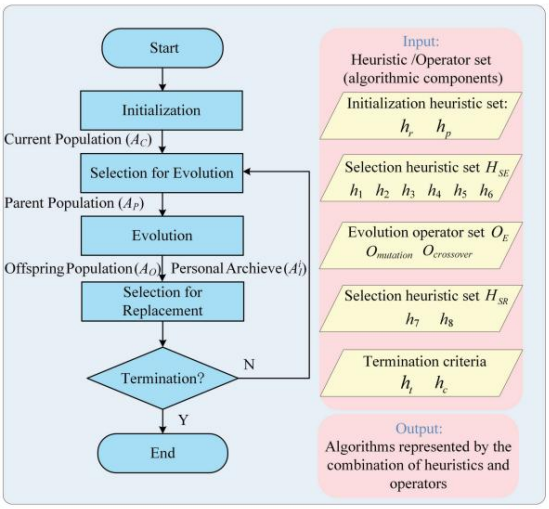

图 1. 通用搜索框架

表二

基于单一解决方案和基于人口的搜索
使用GSF定义的算法

参数/组件	基于单一解决方案的算法	基于人口的算法 > 1
人口规模	1个	
档案	交流= AP = 奥=艾 我	交流= AP = AO = Ai $hr_{我}$
初始化小时,马力		hp h1, h2,
进化h4的选择		h3, h4, h5, h6
进化	突变	Omutation, Ocrossover
替换h7、 h8的选择		h7, h8 ht, hc
终止ht, hc		

表三

GSF中的四个人口档案

存档AC：	描述
当前人口,nP op = |AC |。	进化前在当前种群中选择的个体。
AP :父母群体,μ = |AP |。	在 HSE 中使用启发式选择的个人。
AO:后代种群,λ = |AO|。	进化OE后的后代种群。
艾我: 个人档案,nP op}。我 ∈ {1, 2.· · · ,	个人档案Ai保留个人轨迹。我 对于个人 i th

使用启发式选择法(HSE) 选择当前种群档案 (AC) 并将其存储在父种群档案 (AP) 中。通过使用Evolution 模块中的进化算子(OE)更新或进化种群,并将其存储在后代种群档案(AO) 中。然后通过在替换选择模块中采用选择启发式(HSR)生成当前人口档案。

此外,每个人都有一个个人档案（艾保留个人轨迹。 我） 到

B. 基本 GSF 模块

在 GSF 中,进化选择和替换选择模块根据种群档案中个体的适应性使用各种启发式方法选择个体。

表 IV图 1中GSF内
模块的启发式/运算符集

启发式/运算符集	描述表 V 中定
模块选择的HSE 进化	义的各种启发式算法,用于从当前种群AC中选择父种群AP 。
用于更换模块选择的HSR	表 VI 中定义的各种启发式算法,用于更新基于AP的当前种群AC 。
模块进化的OE	表 VII 中定义的各种算子根据HSE选择的AP生成后代种群AO 。

在不失一般性的情况下,所有选择启发式都被设置为解决优化问题,其目标是最小化目标值。

1)进化选择:进化选择模块中有两种启发式方法。如表 V 所示， h1、 h2和h3是基于概率的,其中个体根据与其适应度相关的概率被选为父母。 h4、 h5和h6以确定性方式而不是概率方式选择一个人作为父母。

表五

HSE :进化模块选择中的启发式

启发式描述tt h1 h1 /h1 b	
	w :锦标赛选择最好/最差的 v ∈ nP op} 个人作为来自AP 的父母候选 {1,· · ·对 ，人。 于每个个体i,被选为亲本候选者的概率p i = 1/nP op。当 v = 1 时:随机选择。 当 v = nP op 时:最佳/最差个体的贪婪选择。
h2	从AP中选择个体 i 作为父母的比例轮盘赌轮盘选择概率与其适应度成正比。
h3	根据与其等级成正比的概率（基于适应度函数的升序)对个体 i 作为父母的排名选择。
h4	选择当前个体本身作为父母。
h5	根据个体的个人档案Ai从AP中选择比当前个体适应度低的所有个体作为父母,将之前最好的位置选择为父母。 我。
h6	

2) Selection for Replacement:进化后,通过使用Selection for Replacement 模块中的选择启发式(h7, h8) 更新种群,如表VI 所示。

表六

HSR:替换模块选择中的启发式方法

启发式描述逗号选择 (nP op, λ)。	
	从后代种群AO 中只选择 nP op 个个体h7 , λ nP op, nP op = |AC | , λ = |AO|。
h8	加选择 (nP op, μ + λ)。从亲本种群AP和后代种群AO 中选择个体， nP op = |AC |, μ = |AP | , λ = |AO|。

3)进化算子:进化模块中的进化算子（OE)包括对一个个体进行操作的Omutation和对多个个体进行操作的Ocrossover 。关于有能力的车辆路径问题

对于时间窗 (CVRPTW),交叉算子很容易出现不可行的解决方案。因此,在本研究中,我们重点研究表 VII 中定义的各种变异算子,以解决 CVRPTW。请注意,这些通用的基本运算符(交换、插入和删除等)可以相应地进行调整,以自动为不同的 COP 设计算法。

表七
OE: CVRPTW的进化算子

操作员	说明 将一个
ochg_in	方案中同一条路径的m、n个节点交换 将一个方案中不同路径的m、n个节点交换 将m个节点插入到一个方案中同一路
ochg_bw	径的其他位置 将m个节点插入到一个方案中不同路径的其他位置 该从解决方案中删除距基本客户预定距离 d 内的
赞成 _	m 个节点。 d的值是根据基节点和离基节点最远的节点之间的距离来设置的。如果存在可容纳被移除节点的可行路
oins_bw	径,则选择等待时间最短的插入位置。否则,将创建一条新路线。
奥林娱乐_	
otwo_opt	在解决方案中交换同一路线中的两个节点 取解决方案
otwo_opt*	中两条路线的末端部分并交换它们以创建两条新路线

C. GSF 中自动算法组合的强化学习

强化学习是一种机器学习技术,其中智能代理根据通过与环境的反复试验交互训练的学习策略采取行动,最大化总奖励。 RL 的环境被认为是一个 MDP,它由一组可能的状态和一组可选择的动作组成。每个状态-动作对都有一个总奖励值(Q 值)。

使用已建立的 GSF,RL 用于自动算法组合,如图 2 所示。这些动作是算法组件(即进化运算符)的可选组合。状态由搜索过程、解决方案和实例的不同特征定义,如表 IX 所示。自动化算法组合过程从观察代理的当前情况(状态)和选择算法组件的组合(动作)开始。所产生的算法组件(选定的动作)的执行通过选定的启发式选择和演化算子导致优化过程(环境)的新状态到当前状态。奖励(或惩罚)被分配给相对于当前选择的动作

状态。

表格 RL 技术,例如 SARSA [4] 和 QL [6],已被用于选择文献中的启发式运算符。
然而,Q 表无法处理连续的状态空间,导致结果不可靠。为了解决这个问题,在本研究中,采用了以神经网络作为值函数逼近器的 RL 技术。

强化学习技术根据其策略大致可分为基于价值的方法和基于策略的方法

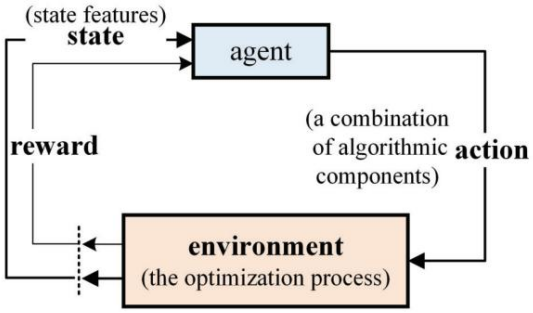

图 2. GSF 中自动算法组合上下文中的强化学习

更新机制[4]。为了全面验证 RL 在自动化算法设计上的有效性,本研究在 GSF 中研究了典型的基于价值的方法和典型的基于策略的方法。

在基于价值的强化学习中,选择了第一种深度强化学习方法 DQN [9]。在 GSF 中自动设计算法的基于 DQN 的方法被命名为 DQN-GSF。在基于策略的 RL 中,选择了优于其他策略梯度方法的 PPO [10],在本研究中命名为 PPO-GSF。

表 VIII 显示了本研究中使用的符号。 DQN-GSF 和 PPO-GSF 的伪代码分别如算法 1 和算法 2 所示。

请注意, h1和h8固定在 Selection for Evolution 和 Replacement 模块中,以解决我们的关键研究问题,即如何使用对进化算法影响最大的进化算子自动设计算法。对于新成立的GSF,现阶段的研究重点是进化的关键模块,而不是同时确定所有模块中的所有组件,以在合理的计算时间内找到最佳结果。通过在修复其他子模块的同时对关键模块进行受控实验,我们可以专注于检查仅由于 Evolution 模块中不同设置的结果。从初步的实验分析可以看出,与进化模块相比,进化选择和替换模块中组件的选择对算法性能的影响更小。因此,选择现有元启发式算法中最常用的组件,即进化选择中的h1和替换中的h8 ,进行重点研究。

表八
DQN-GSF和PPO-GSF中使用的符号

符号说明	
	初始状态s0 st在rt
	时间步 t 的状态
	在时间步长 t 选择的动作
	时间步 t 的奖励值
否	集数
不是	一集中的时间步数

在 DQN-GSF 和 PPO-GSF 中,DQN 和 PPO 这两种 RL 技术首先应用于多个 episode 以训练

DQN-GSF算法 1伪代码

1:初始化内存缓冲区D
2:初始化评价动作价值函数Q网络和
目标动作值函数 Q^ 网络
3:生成初始种群,记录初始状态s0 4:对于episode k = 1到NoE做初始化
状态s0 5:
6:对于时间步长 t = 1 到 NoT,通过计算 7 的值
作,以概率观察当前状态st具有最大Q值的动作,以概率随机选择
使用选择启发式选择父母hi (i = 1, 2, ..., 6) 从HSE （在本研究中固定为
8: h1)通过执行选择的 h1 生成后代种群action at to state st使用来自
HSR的选择启发式hi (i = 7, 8)更新种群（在本研究中固定为h8)观察
基于等式（3)和等式（4)的奖励rt ,以及下一个状态st +1,将经验(st,
9: at, rt, st+1)存储在 D 样本随机小批量经验 [sj , aj , rj , sj+1]中
10:
11:
12:
13:
（J 表示采样小批量的大小)从内存缓冲区 D
中计算
aj) 损失： rj + γmax Q^ (sj+1, aj+1) − Q (sj , ,γ
表示贴现因子。对 Q 网络执行梯
14: 度下降以最小化损失
15: 每 N 个时间步重置 Q^ = Q
16:结束17:结束

算法 2 PPO-GSF 的伪代码

1:初始化内存缓冲区D
2:初始化策略参数θ0,值函数参数
Φ0
3:生成初始种群,记录初始状态s0 4:对于episode k = 1 to NoE do 5:对
于timestep t = 1 to NoT do通过计算值观察当前状态st 6:表IX中的不同
状态特征选择parents使用来自HSE的选择启发式hi (i = 1, 2, ..., 6) （在
代种群。使用来自HSR的选择启发式hi (i = 7, 8)更新种群（在本研究中固定为
h8)根据等式（3)和等式（4)观察奖励rt收集经验（st， at， rt)和
7: 保存在D
8:
9:
10:
11:
12:结束13:通过
最大化 PPO 目标更新策略
θk+1 based on Equation (1)
14: fit value function Φk+1 based on Equation (2)
15: empty memory buffer D 16: end for

GSF 内的政策。之后,训练好的策略用于在线设计搜索算法。训练过程是重点研究的问题,详细描述如下。

如算法 1 所示,DQN-GSF 在每个时间步上进行训练。具体来说,确定性地选择具有最大 Q 值的动作（进化算子)用于开发或随机选择用于探索（第 8 行,算法 1）。设计的具有预定义选择启发式的搜索算法执行一个时间步长（第 9-11 行,算法 1）。确定下一个状态和奖励,并将此经验(st, at, rt, st+1)存储在内存缓冲区中（第 12 行,算法 1）。之后,从内存缓冲区中随机抽取一小批经验来训练评估网络（第 13-14 行,算法 1）。该过程在每个时间步重复,直到剧集结束。在此过程中,目标网络参数与评估网络参数定期同步（第 15 行,算法 1）。

与基于价值的 DQN-GSF 不同,基于策略的 PPO-GSF 在每一集而不是每个时间步上进行训练。如算法 2 所示,首先,根据策略的概率 πθk 、 NoE 和

然后设计的具有预定义选择启发式的搜索算法对应地执行一个情节（第 5-12 行,算法 2）。然后,通过基于等式（1)（第 13 行,算法 2)最大化 PPO 目标来更新策略,并基于等式（2)（第 14 行,算法 2)通过时间微分误差拟合值函数。最后,内存缓冲区设置为空（第 15 行,算法 2）。根据更新后的策略选择一系列动作来执行下一集优化（第 8 行,算法 2）。

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\text{直流}|\text{不是}} \sum_{\tau \in Dc} \sum_{t=0}^{\text{不是}} \text{分钟} \left(r_t(\theta) A^{\pi_{\theta k}}(s_t, a_t), \text{clip}\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right) \right) \quad (1)$$

$$\Phi_{k+1} = \arg\min_{\Phi} \frac{1}{|\text{直流}|\text{不是}} \sum_{\tau \in Dc} \sum_{t=0}^{\text{不是}} \left(V_{\Phi}(s_t) - \hat{R}_t \right)^2 \quad (2)$$

$\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta k}(a_t|s_t)}$ (θ) 表示概率比$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta k}(a_t|s_t)}$。
πθk (st, at)是优势函数的估计量,是一个超参数。 clip $(r_t(\theta), 1-\varepsilon, 1+\varepsilon)$ 时间步根据轨迹 τ 计算：[(st,ta表示通过截断概率比修改后的替代目标。 reward-to-

有关这两个方程的更多详细信息,请参阅 [10]。

1)状态表示：本节首先区分不同的状态特征,包括搜索相关、解决方案相关和实例相关特征。

本文已被 IEEE Transactions on Evolutionary Computation 接受发表。这是作者的版本，尚未完全编辑
内容可能会在最终发布前更改。引用信息:DOI 10.1109/TEVC.2022.3197298

依赖于搜索的特征观察搜索过程，例如对初始解决方案的总体改进。 Solution dependent features与solution encoding scheme相关联，以TSP为例，一个完整的tour的编码可以直接定义为state。 Instance-dependent features 是指实例特定的特征，例如 VRP 的车辆编号或车辆容量。

当依赖于搜索或依赖于实例的特征用于定义状态空间时，学习到的信息可以转移到同一问题的其他实例，甚至转移到其他问题。在许多情况下，依赖于解决方案的特征不能用于开发通用方法，因为它们是特定于问题的。因此，在本研究中，如表 IX 所示，四个搜索相关特征(f1-f4)和四个实例相关特征(f5-f8)用于定义状态空间。

表九
状态空间的定义

特征	描述
	f1相对于初始种群适应度的总体改进程度 种群的多样性，通过种群适应度的标准差来衡量 当前算法阶段，计算为 i/N oT 其中 i 是当前时间步 [illegible] 车辆容量 时间窗口的密度，即时间受限客户的百分比
f3	
f4	
f5	
f6	
f7	
f8	时间窗的紧度，即时间窗的宽度

2) 动作表示：在 DQN-GSF 和 PPO-GSF 中，每个状态下可能的动作集由表 VII 中的演化算子(OE)集定义。一旦选择了一个动作，它就会应用于整个人群。

3)奖励方案：奖励方案鼓励RL系统找到有效的搜索策略，对于RL方法非常重要。在DQN-GSF和PPO-GSF中，奖励是根据当前种群相对于初始种群的适应度的提高来计算的，如等式（3)和等式（4)所示。当种群适应度被优化到一定阈值以上时，同样的适应度提高会得到更大的奖励。

$$f1 = \frac{f_{current}}{初始} \quad (3)$$

$$奖励= \begin{cases} -f1，如果f1 > C \\ -f1 - \log10 (f1)，如果f1 C \end{cases} \quad (4)$$

奖励的设置有两种方法：对f1进行归一化以提高训练效率；在优化过程的后期通过对数函数对相同的适应度改进分配更大的奖励。

文献中的许多简单的正/负奖励方案通过计算成功实现的步数来跟踪适应度的提高。建议奖励 scheme 的设计目的是最大化整体适应度的提高本身，这才是真正需要优化的地方。
与简单的正/负奖励方案相比，所提出的奖励方案不仅反映而且衡量了所选动作的正/负影响。此外，所提出的奖励方案为在优化过程的后期导致适应度改进的动作分配更大的奖励，以解决这种改进在进化的最后阶段通常非常小的问题。

4) Episode Setting：一个episode被定义为整个优化过程。由于在本研究中使用了基于时间的停止标准，因此每个情节的周期等于给定的优化时间tmax。一个 episode 被分成 NoT 个时间步长，所以每个时间步长的周期等于tmax/NoT。

出于训练目的，建议的 DQN-GSF 和 PPO GSF 针对 NoE 剧集执行。出于测试目的，设计的 DQN-GSF 和 PPO-GSF 执行一集。

四、实验与讨论

在本研究中，对新 GSF 中提出的 RL-GSF 方法进行了调查和评估，这是对研究最多的 COP 之一 CVRPTW 进行的。所有实验均使用配备英特尔(R) 至强(R) W-2123 CPU@ 3.60 GHz 处理器和 32.0 GB 内存的计算机进行。

RL-GSF 方法在 Java 环境中实现，开发工具为 IntelliJ IDEA 2020.3.3。

实验研究旨在解决两个研究问题：(1)新的 RL 技术自动生成搜索算法以解决基准 Solomon CVRPTW 数据集的有效性； (2) 将训练有素的策略推广到新问题实例。为了分析 Q 值函数逼近器对学习模型的影响，两种以适应度改进为状态定义的基于值的 RL-GSF 方法，即具有 Q 表的 QL-GSF 和具有神经网络函数的 DQN-GSF 近似器，在第 IV-B1 节中进行了比较。为了分析策略更新机制对学习模型的影响，DQN-GSF 和 PPO-GSF 在第 IV-B2 节中进行了评估。通过在第 IV-C1 节和第 IV-C2 节中将训练过的策略直接应用于新实例，评估训练过的策略在相同类型和不同类型问题实例中的泛化。

A. 问题定义和数据集

车辆路径问题可以说是最重要的运输调度问题之一。在经典模型 CVRPTW 中，车队按最短距离为客户提供服务，满足容量和时间窗口限制。 CVRPTW 已作为评估进化算法和元启发式算法性能的基准问题进行了深入测试 [32]。本文将研究 CVRPTW，以更好地理解所提出的基于强化学习的自动化算法设计方法。