一种考虑人岗匹配的强化学习辅助遗传编程算法，用于团队组建问题

Yangyang Guo, Hao Wang*，Lei He，Witold Pedrycz，*IEEE*终身会士，P. N. Suganthan，*IEEE*会士，Yanjie Song*

arXiv:2304.04022v1 [cs.NE] 8 Apr 2023

摘要—**高效的团队对于公司成功完成新项目至关重要。为解决考虑人岗匹配的团队组建问题（TFP-PJM），构建了一个0-1整数规划模型，该模型同时考虑人岗匹配以及团队成员的沟通意愿对团队效率的影响，其中人岗匹配分数采用直觉模糊数计算。随后，提出一种强化学习辅助的遗传编程算法（RL-GP），以提升解的质量。RL-GP采用集成种群策略。在每一代进行种群进化之前，智能体根据所获得的信息从四种种群搜索模式中选择一种，从而实现探索与利用之间的良好平衡。此外，算法中使用代理模型对个体生成的组建方案进行评估，从而加快算法学习过程。随后，开展了一系列对比实验，以验证RL-GP的整体性能以及算法内部改进策略的有效性。通过高效学习获得的超启发式规则可在组建项目团队时用作决策辅助工具。本研究揭示了强化学习方法、集成策略以及应用于遗传编程框架的代理模型的优势。搜索模式的多样性与智能选择，以及快速适应性评估，是使RL-GP能够部署于现实世界企业环境中的显著特征。**

索引词—**团队组建、强化学习、遗传编程、直觉模糊数、集成种群策略、代理模型**

一、引言

IN近年来，众多现代企业的发展呈现出技术密集型的特点，人才在推动企业成长与成功方面发挥着至关重要的作用。然而，许多企业在制定发展战略时未能优化人才资源，往往忽视了招聘和培训等人力资源机制的落实。这导致企业无法及时响应市场需求，进而阻碍了自身的发展。因此，企业必须认识到人力资源对组织发展的重要性，因为人力资源是核心资源，对竞争力有着重大影响 [1]。重视人才管理有助于企业应对复杂的社会环境和激烈的市场竞争，最终保障企业的经济效益。

在团队中为候选人选择并分配不同工作岗位，是人力资源管理的重要环节。这个问题可称为考虑人岗匹配的团队组建问题（TFP-PJM）。在职位匹配问题中，决策者需要根据现有信息，科学地将候选人分配到合适的岗位，充分发挥候选人的价值，使人与岗位的匹配度最大化。将候选人与合适的工作岗位有效匹配，可以提高工作满意度和生产力，进而提升组织绩效。相反，错误地分配候选人可能导致人才流失，并给企业造成重大损失 [2]。此外，许多公司需要组建团队来完成特定项目，而团队中的候选人必须彼此合作。不同候选人的沟通意愿可能存在差异，这会影响团队效率。因此，为确保团队的协同效率，决策者需要考虑如何从候选人库中选出一定数量的候选人，使其既满足组建团队所需的技能要求，又能最大限度地提高候选人之间的协作程度。

团队组建是现代组织不可或缺的一部分，许多学者从不同方面对其进行了研究。文献 [3] 对团队组建问题的相关研究进行了全面综述，并将其分为两类：基于任务分配的团队组建和基于社区的团队组建。在这些研究中，考虑了最大化板球队得分表现 [4],最大化团队收益[5], 最小化沟通成本[6], 等目标函数。许多研究采用了不同的方法来量化技能评分，包括精确数评分 [7], 直觉模糊数 [8],概率变量 [9],等。然而，很少有研究关注人岗匹配和人员关系对团队效率的共同影响。

团队组建问题是NP难问题 [3], ，研究人员已采用多种启发式方法或元启发式方法来求解该问题

* 通讯 p作者 g： Hao Wan g, Yan jie Son g Yangyang Guo 任职于中国国防科技大学系统工程学院和中国大连海事大学交通工程学院。Hao Wang、Lei He 和 Yanjie Song 任职于中国国防科技大学系统工程学院。Witold Pedrycz 任职于加拿大阿尔伯塔大学电气与计算机工程系。P. N. Suganthan 任职于卡塔尔卡塔尔大学工程学院 KINDI计算研究中心，以及新加坡南洋理工大学电气与电子工程学院。[1] L ei He and Y angyang G uo contr ibuted equally to t his art icle.

，例如遗传算法 [10],、粒子群算法 [5], [11], 和局部搜索算法 [12]。此外，还有一些混合算法，它们结合多种搜索策略或将多种算法结合起来，着重提升全局与局部搜索性能，从而不断优化团队组建方案。然而，这些算法仍存在一些问题，例如搜索效率低、对算法参数依赖性强。此外，由于TFP-PJM问题涉及候选人和岗位的多种信息，很难用简单规则表示该问题的所有属性。人工设定的启发式规则可能不易调整，也可能忽视影响算法性能和搜索解质量的重要属性。

相比之下，遗传编程（GP）作为一种超启发式方法，能够充分利用 TFP-PJM 问题中候选人和岗位的特征，并将其组合到特定表达式中。在种群进化过程中，每个个体会结合不同的底层规则，形成超启发式规则，以获得候选人的完整排序并生成团队组建方案。此外，许多研究还探索了将遗传编程与多种策略相结合，以解决各种组合优化问题，例如作业车间调度问题[13], 、不确定容量弧路径问题 [14],和在线资源分配问题 [15]。这些成功经验可为未来的算法设计提供指导。然而，传统遗传编程容易陷入搜索模式单一的缺陷，可能过分强调全局搜索，无法有效挖掘局部解空间。此外，使用 GP 求解 TFP-PJM 等组合优化问题时，需要评估每个个体生成的团队组建方案并检查各种约束，这会导致适应度评估的计算成本较高。因此，为解决上述问题，我们改进了传统 GP，并提出一种强化学习辅助遗传编程算法（RL-GP）来求解 TFP-PJM问题。首先，在算法设计中引入多种群集成策略，并通过强化学习选择不同的搜索模式来辅助种群搜索，从而打破 GP 搜索模式单一的局限。智能体动作选择采用一种 ε-贪心策略，根据随机数与阈值之间的关系，选择 Q 值最大的动作，或随机选择一个动作。此外，我们采用智能体建模，利用机器学习模型快速计算适应度函数值，从而降低计算成本并增强算法的学习能力。为提高代理模型的准确性，在经过一定数量的代之后，使用真实模型评估个体的适应度函数值，并在线更新代理模型。本研究的主要贡献如下：

1. 针对考虑人岗匹配的团队组建问题，构建了一个0-1整数规划模型。由于一个人是否与某个工作岗位匹配是一个模糊概念，因此适合使用面向模糊集的数学工具来处理此类问题。在在数学模型中，使用直觉模糊数评估人员，以获得人岗匹配度，并使用沟通矩阵计算所选人员之间的合作度。在此基础上，提出了一个旨在最大化团队效率的优化函数。此外，该数学模型还考虑了职位限制和人员能力约束等约束条件。

2. 本文提出了一种强化学习辅助遗传编程算法（RL-GP）。该算法采用集成种群策略，并通过强化学习从集成策略中选择搜索模式进行种群搜索。此外，RL-GP算法引入代理模型，以加快适应度计算并提高算法学习效率。

3. 大量实验表明，RL-GP算法中的改进策略能够有效地进行学习，并获得优于构造启发式算法的超启发式规则。此外，通过比较其他搜索算法生成的解的质量，证明RL-GP算法在求解TFP-PJM问题方面具有优势。实验还表明，基于强化学习的搜索模式选择策略有助于充分利用种群搜索。

本文结构安排如下：第二节介绍团队组建问题以及遗传编程在求解组合优化问题方面的相关工作。第三节描述TFP-PJM问题，并概述模型构建过程。第四节详细介绍用于解决TFP-PJM问题的强化学习辅助遗传编程算法。第五节通过实验验证所提算法的有效性。最后一节总结主要工作，并提出未来研究方向。

II. 相关工作

本节介绍与团队组建问题以及遗传编程解决组合优化问题相关的研究。

A. 团队组建问题

目前，团队组建问题已引起广泛关注，许多学者从不同角度研究了相关问题，提出了构建模型的各种方法，并使用不同算法解决该问题。Ju´arez, J 等人[3] 提出，模型构建可分为两类，即基于任务分配的团队组建和基于社区的团队组建。Bart H. Boon 等人提出的模型[16] 可用于足球队选拔候选人以组建最优团队，并计算新加入球员对团队的价值。Fitzpatrick, E. L 等人[17]在确保团队成员技能满足要求的基础上，将个体本能倾向建模为团队选拔的绩效指标。Xu 等人[18] 基于三个重要特征组建团队，即候选人在相关领域的专业知识水平、技能多样性以及候选人的意愿

在候选人之间进行合作。Baykasoglu, A 等人[19] 采用模糊评分方法计算候选人的技能适应度，并提出了一种模糊多目标优化模型，考虑项目的时间和预算约束以及候选人之间的人际关系。Liemhetcharat, S 等人[9] 将团队组建转化为不同技能的智能体相互协作以完成任务的问题，并提出了一种学习算法，能够在没有先验知识的情况下学习良好的协同图以组建团队。

许多学者研究了人际关系对团队效率的影响，并提出了不同的建模方法。Wang 等人[20] 通过社交网络图计算候选人之间的沟通成本，并提供了一个公共平台，用于实现多种团队组建算法。Guti´errez, J. H 等人[12] 使用沟通矩阵设定了候选人之间合作意愿的三种可能情景，并研究了多项目团队的效率公式，这有助于解决许多现实案例。

许多算法已被用于解决团队组建问题。Zhang 等人 [11] 使用 MBTI 测量工具评估人员之间的合作情况，并采用多目标粒子群优化算法，获得团队组建备选方案的帕累托解。Strnad、D 等人[21] 开发了一种模糊遗传分析模型，将多个标准结合起来，定义单一的综合目标函数，并提出了一种采用混合交叉的岛屿遗传算法，以优化团队组建方案。Bhowmik、A 等人[22] 将团队组建问题建模为无约束的子模函数最大化问题，并提出了一种基于模拟退火的随机逼近方案。

B. 用于求解组合优化问题的遗传编程

本文使用遗传编程生成团队组建方案。遗传编程具有超启发式算法的优势，能够适应与不同组合优化问题相关的多种状态特征。Xu 等人[23] 利用遗传编程算法进行路径和排序决策，并结合三种延迟路由策略来解决动态柔性作业车间调度问题。Wang 等人[24] 采用遗传编程算法解决不确定容量弧路径问题，并引入生态位技术以简化路径策略，从而提高了算法效率和性能。

此外，遗传编程具备整合多种策略以提升算法效率和质量的能力，并已应用于多个领域。Zhang 等人[25] 将遗传编程与代理模型相结合，以解决动态柔性作业车间调度问题；通过充分利用从不同调度任务中学习到的知识，提升了调度启发式算法的质量。Ardeh 等人[14] 研究了不确定容量弧路径问题，并通过遗传编程引入知识迁移，从而提升了.....

通过将探索限制在未探索的领域来提高算法搜索效率。

总而言之，在研究团队组建问题时，很少有研究同时量化人岗匹配的两项指标以及团队成员之间的合作水平。本文旨在通过同时考虑这两个维度对团队效率的影响来弥补这一差距，从而提高模型和求解方案的准确性。许多用于求解团队组建问题的算法都需要人工设计启发式规则，这存在一定的局限性。遗传编程的优势在于能够自动选择底层规则进行组合，从而减少对专业领域知识的需求。此外，目前尚未有研究将遗传编程与强化学习相结合来解决该问题。

III. 模型

本节详细描述TFP-PJM问题，并介绍模型中涉及的变量与符号、目标函数以及约束的设置。

A. 问题描述

对于任何公司而言，在着手开展一个新项目时，如何组建团队都是一项重要决策。作出这一决策时，应考虑各种因素。首先，候选人的技能匹配度是一个重要因素，因为如果员工缺乏必要的技能，项目可能会延误，甚至无法完成。因此，需要从候选人池中选出技能符合要求且能够胜任工作岗位的人选。其次，有效的沟通与协作是团队项目取得成功不可或缺的要素。团队成员之间的沟通效率会直接影响项目的项目进度和质量。因此，应慎重考虑选择沟通技能较强的候选人，以确保与其他团队成员之间建立顺畅的合作关系 [26]。总的来说，只有综合考虑诸多因素，公司才能组建一支高效且具有协作能力的团队，为项目的成功提供强有力的支持 [27]。

当公司需要承担一个新项目时，TFP-PJM问题强调，应选择拥有不同技能的候选人，并将他们安排到符合相应岗位技能要求的合适岗位上。组建的团队必须在和谐的氛围中协同工作，以确保实现最佳生产力。技能评估的准确性在TFP-PJM问题中也很重要。有些技能可能被过分强调，而另一些则可能被忽略。因此，公司需要根据项目需求和工作内容，明确不同岗位所需的技能权重，然后据此评估候选人的技能。此外，如图1所示，某些任务可能需要多名拥有相似技能的成员参与，不同候选人拥有的技能也可能重叠，这正是为了解决这个问题。然而，由于

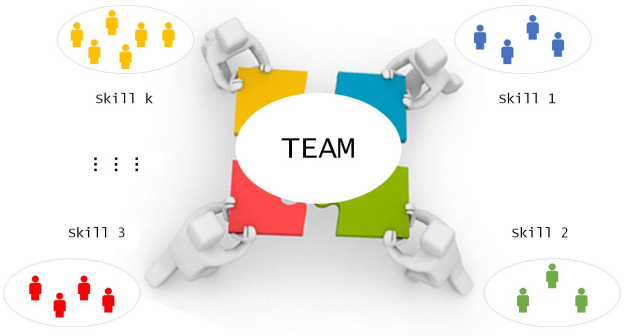


图1：由具备不同技能的成员组成的团队

由于人格和文化差异等各种因素，候选人之间的沟通意愿各不相同。总体而言，有些候选人具有合作心态，有些表现出中立态度，还有些不太愿意合作 [12]。当团队成员具有强烈的沟通意愿时，团队的总体效率自然会提高。

模型中需要决定的是如何从可用候选人池中选出候选人，并将其分配到各个团队位置。这一决策不仅能够满足不同位置的技能要求，还能够满足新项目设定的技能要求。同时，成员可以高效合作，使团队效率达到最大值。

B. 变量与符号

C: 候选人集合，$C = \{c_1, c_2, ..., c_{|C|}\}$，$c_i$ 代表候选人 i;

S：所有候选人的技能集合，$S = \{s_1, s_2, ..., s_{|S|}\}$,$s_k$ 表示技能 k;

T：执行任务所需的技能集合，$T = \{s_m, ..., s_n\} \subseteq S$;

$S(c_i)$:候选人 i 所拥有的技能集合, $S(c_i) \subset S$;

$TS(c_i)$：任务 T 所需的技能集合，由候选人 i 拥有，$TS(c_i) \subset S$;

$C(s_k)$：具备技能 k 的候选人集合，$C(s_k) \subset V$;

$n(s_k)$：完成任务 T 所需的具备技能 k 的成员数量;

POS：团队中的职位集合，$POS = \{pos_1, pos_2, ..., pos_{|POS|}\}$，$pos_j$ 表示岗位 j;

$RS(pos_j)$：岗位 j所需的技能集合，$RS(pos_j) = \{s_y, ..., s_z\}$, $RS(pos_j) \subseteq S$;

W：岗位评估能力权重矩阵，w_{jl}表示岗位 j 在第 l个能力维度上的评估权重，$\sum_{l=1}^{|L|} w_{jl} = 1$，$\forall j = 1, 2, ... |POS|$;

A: 直觉模糊数的集合，$A = \{A_1, A_2\}$, $A_1 \cup A_2 = A$ $A_1 \cap A_2 = \emptyset$ $A = \{ \mu_{ijl}\ v_{ijl} \mid i = , , (,) ,,,;$ 1 2 ... $|C|$ j = 1 2 ... $|POS|$ l = 1 2 ... $|L|\}$ A_1 = ,,,; ,,,$\{\mu_{ijl} | i = 1, 2, ..., |C|;\ j = 1, 2, ..., |POS|;\ l = 1, 2, ..., |L|\}$, $A_2 = \{v_{ijl} | i = 1, 2, ..., |C|;\ j = 1, 2, ..., |POS|;\ l$ =1, 2, ..., $|L|\}$, $\alpha_{ijl} =< \mu_{ijl}, v_{ijl} >$, α_{ijl}表示候选人 i 在岗位 j的第 l个能力维度上的直觉模糊数评分，其中 μ_{ijl}, v_{ijl} 分别表示隶属度和非隶属度;

ez_{ij}：候选人 i 与岗位 j的匹配分数;

R：人员关系矩阵，矩阵的大小为 $|C| \times |C|$，$r_{ii'}$表示候选人 i和 i'进行沟通的意愿。$r_{ii'}$的取值如下所示:

$$r_{ii}^{'} = \begin{cases} -1 & \text{unwilling to cooperate} \\ 0 & \text{neutral} \\ 1 & \text{willing to cooperate} \end{cases} \quad (1)$$

γ：团队的沟通效率。

决策变量：

x_{ij}：候选人 i 是否被分配到岗位 j。x_{ij} 的取值如下:

$$x_{ij} = \begin{cases} 1 & \text{if the candidate is chosen} \\ 0 & \text{else} \end{cases} \quad (2)$$

C. 模型

为更好地聚焦于团队组建问题本身的特点，我们作出以下假设:

假设：

1. 每位候选人至少具备团队所需的一项技能，且所有候选人都有可能被选中。
2. 候选人人数多于所需的团队成员人数。
3. 团队组建完成后，在项目期间成员不会离开或被替换。
4. 团队和每个岗位所需的技能不会发生变化。
5. 团队岗位的数量在选择候选人之前确定，之后不会增加或减少岗位。
6. 每位候选人只能运用自己的特定技能来完成任务。
7. 候选人彼此沟通的意愿是预先确定的，并且在整个项目期间保持不变。
8. 团队所需的某些技能需要多名具备这些技能的人员协同工作，而每个岗位所需的所有技能只需一人即可完成 [12]。

传统的人岗匹配评估方法通常使用精确数值来表示影响匹配结果的指标，但由于实际匹配问题的复杂性和不确定性，很难准确量化指标值。为了将选定的候选人安排到最合适的岗位，充分发挥各自的能力，即实现最大的人岗匹配，我们将直觉模糊集引入模型构建中。首先，我们介绍IFS的概念和定义:

假设 X 是一个非空经典集合，定义在 X 上的 IFS A 可表示为:

$$A = \{< x, \mu_A(x), v_A(x), h_A(x) > | x \in X\} \quad (3)$$

其中， $\mu_A(x)$、 $v_A(x)$和 $h_A(x)$分别表示隶属度、非隶属度和犹豫函数 [28]。直觉模糊数是直觉模糊集理论的基本元素，可更简洁地表示为 $\alpha = \langle \mu,\ v \rangle$，以及 $\mu,\ v,\ \mu + v \in [0,1]$。

该模型将直觉模糊数引入候选人评估过程。在每个岗位的评价维度下，决策者进行0-1决策，以判断某位候选人是否能够满足特定岗位的要求，或者可以选择弃权。评估后，将得到多个决策者的直觉模糊矩阵。由于隶属度在决策-制定过程中可能为0，参见 [29], 模型采用直觉模糊幂交互加权平均（IFPIWA）算子，通过为每个岗位的评价维度设置不同权重来整合模糊信息，从而获得聚合直觉模糊数和人岗匹配分数。具体计算公式如下:

$$d(\alpha_{ijl}, \alpha_{ijl'}) = \frac{1}{2}(|\mu_{ijl} - \mu_{ijl'}| + |\mathrm{v}_{ijl} - \mathrm{v}_{ijl'}|) \tag{4}$$

$$\sup(\alpha_{ijl}, \alpha_{ijl'}) = 1 - d(\alpha_{ijl}, \alpha_{ijl'}) \tag{5}$$

$$T(\alpha_{ijl}) = \sum_{l'=1, l' \neq l}^{|L|} w_{jl'} \sup\left(\alpha_{ijl'}, \alpha_{ijl}\right), l = 1, 2, ..., |L| \tag{6}$$

$$\rho_{ijl} = \frac{w_{jl}(1 + T(\alpha_{ijl}))}{\sum_{l=1}^{|L|} w_{jl}(|1 + T(\alpha_{ijl}))} \tag{7}$$

这里给出一个示例，以便于理解该定义。图2展示了岗位 pos_1 的技能要求以及候选人 c_1所具备的技能属性。假设为该岗位设置了三个维度，即教育水平、编程能力和领导能力，其权重分别为0.4、0.35和0.25，对应于 w_{11}、 w_{12}、 w_{13}的值。经专家评分后，将得到候选人在该岗位各维度上的直觉模糊数，分别为 $\alpha_{111} =< 0.4, 0.6 >$ $\alpha_{112} =< 0.6, 0.4 >, \alpha_{113} =< 0.8, 0.2 >$。通过这些信息，可以计算人岗匹配分数。首先，可以计算三个直觉模糊数的直觉模糊汉明距离。

$$d(\alpha_{111}, \alpha_{112}) = \frac{1}{2}(|0.4 - 0.6| + |0.6 - 0.4|) = 0.2 \tag{9}$$

$$d(\alpha_{111}, \alpha_{113}) = \frac{1}{2}(|0.4 - 0.8| + |0.6 - 0.2|) = 0.4 \tag{10}$$

$$d(\alpha_{112}, \alpha_{113}) = \frac{1}{2}(|0.6 - 0.8| + |0.4 - 0.2|) = 0.2 \tag{11}$$

然后，计算它们之间的支持度。

$$\sup(\alpha_{111}, \alpha_{112}) = 1 - 0.2 = 0.8 \tag{12}$$

$$\sup(\alpha_{111}, \alpha_{113}) = 0.6, \sup(\alpha_{112}, \alpha_{113}) = 0.8 \tag{13}$$

$$T(\alpha_{111}) = 0.43, T(\alpha_{112}) = T(\alpha_{113}) = 0.52 \tag{14}$$

$$\rho_{111} = \frac{w_{11}(1 + T(\alpha_{111}))}{\sum_{l=1}^{3} w_{1l}(1 + T(\alpha_{11l}))} = 0.39 \tag{15}$$

$$\rho_{112} = \frac{w_{12}(1 + T(\alpha_{112}))}{\sum_{l=1}^{3} w_{1l}(1 + T(\alpha_{11l}))} = 0.36 \tag{16}$$

$$\rho_{113} = \frac{w_{13}(1 + T(\alpha_{113}))}{\sum_{l=1}^{3} w_{1l}(1 + T(\alpha_{11l}))} = 0.26 \tag{17}$$

$$\text{IFIWA}(\alpha_{111}, \alpha_{112}, \alpha_{113}) =< 0.61, 0.39 > \tag{18}$$

最后，候选人 c_1 与岗位 pos_1 之间的匹配得分计算为 $ez_{11} = 0.22$。

此外，团队内部的沟通效能会受到成员之间沟通意愿的影响。参见 [12], , 沟通矩阵用于表示成员之间的合作程度。该矩阵包含 1、0 或 -1，分别表示两人愿意合作、持中立态度或不愿合作。基于沟通矩阵设计了团队整体沟通效率的计算公式，具体如下:

$$\gamma = a\left(b + \frac{\sum_{i,i' \in C} r_{ii'} \cdot x_{ij} \cdot x_{i'j'}}{[\sum_{s_k \in T} n(s_k)]^2}\right) \tag{19}$$

团队的沟通效率通过汇总所有团队成员的沟通意愿获得。如果候选人 i 和 i' 愿意合作，且二者均被选入团队并被分配到不同的位置，则公式中的分子将增加 1，从而提升团队的沟通效率。在该问题中，我们设置$a = \frac{1}{2}$, $b = 3$，将γ 的区间从 $[-1, 1]$ 转换为 $[1, 2]$, , 这便于后续设置目标函数 [12]。

模型的目标是通过仔细选择合适的候选人并将他们安排到正确的位置来最大化团队生产力。这需要同时考虑人岗匹配分数和沟通效率分数，目标函数如下所示:

目标函数:

$$max \sum_{i \in C} \sum_{j \in POS} ez_{ij} \cdot x_{ij} \cdot \gamma \tag{20}$$

其中， ez_{ij} 表示所选候选人与其各自岗位的匹配分数， γ 表示团队中人员的沟通效率，这两个因素是影响整个团队效率的重要因素。 γ 的区间值使得将团队沟通效率与人岗匹配分数相乘成为可能，从而表示团队生产力。具体而言，计算得到的沟通效率值越大，将其与人岗匹配分数相乘后得到的团队生产力值就越大。

该模型主要考虑人员的工作岗位约束和技能约束，具体表述如下:

约束:

1. 每位候选人最多只能被分配到一个岗位。

$$\sum_{j=1}^{|POS|} x_{ij} \leq 1, \forall i = 1, 2, ..., |C| \tag{21}$$

$$\text{IFPIWA}(\alpha_{ij1},\alpha_{ij2},\ldots,\alpha_{ij|L|}) = \left(1-\prod_{l=1}^{|L|}(1-\mu_{ijl})^{\rho_{ijl}}, \prod_{l=1}^{|L|}(1-\mu_{ijl})^{\rho_{ijl}} - \prod_{l=1}^{|L|}(1-(\mu_{ijl}+v_{ijl}))^{\rho_{iji}}\right) \tag{8}$$

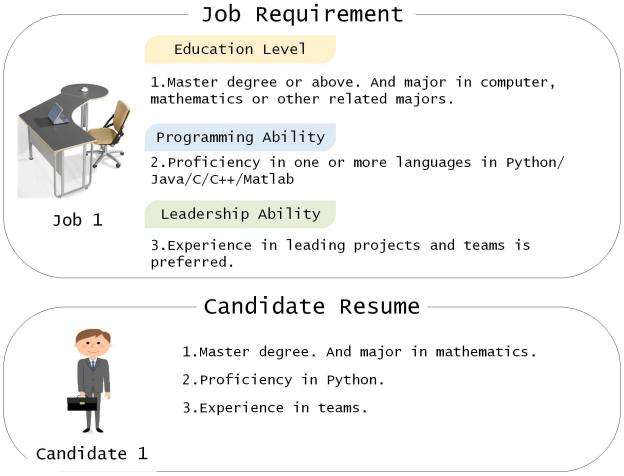


图 2：人岗匹配示例

2. 每个职位恰好分配一个个体。

$$\sum_{i=1}^{|C|} x_{ij} = 1, \forall j = 1,2,...,|POS| \tag{22}$$

3. 团队中每项必需技能都有足够数量的成员。

$$\sum_{c_i \in C(s_k)} x_{ij} \geq |n(s_k)|, \forall s_k \in T \tag{23}$$

4. 所有入选候选人具备的技能集合不少于团队所需的技能。

$$\sum_{c_i \in C} \sum_{m \in TS(c_i)} S_m \cdot x_{ij} \geq |T| \tag{24}$$

5. 入选候选人满足相应岗位的技能要求。

$$(1-x_{ij}) \cdot RS\left(pos_j\right) \leq S(c_i), \forall i \in C, j \in POS \tag{25}$$

6. 决策变量的取值范围。

$$x_{ij} \in \{0,1\}, \forall i = 1,2,...,|C|, j = 1,2,...,|POS| \tag{26}$$

IV. 所提方法

本节介绍 RL-GP 的总体框架及相关改进策略。

A. 框架

TFP-PJM问题包含大量候选人的特征和职位特征，难以使用简单规则同时对其进行描述。相比之下，GP通过将种群中的个体采用的不同规则相结合并进行迭代演化，从而找到最优解 [30]。此外，为了突破GP单一搜索模式的限制，采用了集成种群策略，并利用强化学习从集成种群策略中选择搜索模式，以支持种群搜索。此外，在个体适应度评估阶段使用代理模型，以提高RL-GP的学习速度。

所提出算法的流程图如图 3 所示。我们分析 TFP-PJM 问题的特征，并选择一些低层启发式规则作为问题的终端集，同时选择一些符合要求的基本操作构成函数集。GP 中个体的表示方式将终端集和函数集组合成不同的优先级函数，用于对候选人进行排序，从而得到相应的团队组建方案。算法初始化后，遗传编程框架用于持续迭代，以搜索理想的团队组建解。在 RL-GP 种群进化过程中，算法采用集成种群策略和强化学习方法。具体而言，算法设置了四种种群搜索模式，分别为大小为 N的种群（记为 P_1）、大小为N 的单个精英个体种群（记为 P_2）、大小为 2 的双精英个体种群（记为 P_3），以及由 P_1 和 P_3 组成、大小为 $N+2$的种群（记为 P_4）。上述四种搜索模式作为动作，由智能体在每代种群迭代搜索之前进行选择。强化学习方法使用 Q表记录智能体的状态，并采用 ε-贪婪策略进行动作选择 [31]。 ε-贪婪策略通过比较随机数和参数，选择 Q值最大的动作，或在特定时刻随机选择一种动作。该策略充分利用搜索信息，同时探索新的动作，以提升算法的探索性能。

确定搜索模式后，RL-GP种群通过选择、交叉和变异操作进行进化，以生成子代种群。新生成的种群使用代理模型评估适应度函数值，随后根据其优化性能更新Q表。为便于快速评估个体的适应度值，采用K近邻（KNN）模型作为代理模型 [32]。经过一定数量的代搜索后，算法将选择表现最佳的个体，并将其生成的团队组建方案确定为最终方案。

介绍 RL-GP 的算法框架后，还需要进一步说明一些方法和策略。本节第一部分介绍 RL-GP 中的个体表示及其相应的编码和解码过程。第二部分将介绍强化学习方法。随后，将介绍集成种群策略。

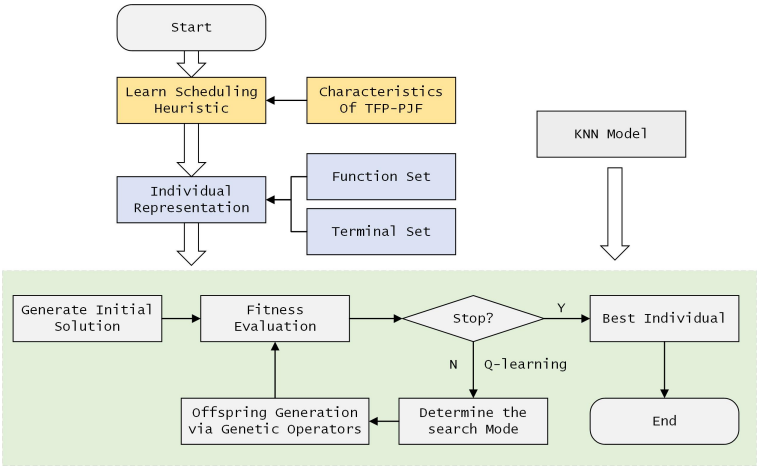


图 3：RL-GP 的流程图

第三部分对其进行了详细描述。第四部分阐述了代理模型的构建，而最后一部分介绍了RL-GP 的学习过程。

B. 个体表示

与其他进化算法的编码方式不同，遗传编程使用终端集和函数集对个体的染色体进行编码，二者分别由值和运算符组成 [33]。终端集由TFP-PJM问题的相关状态特征组成，如表I所示。我们的RL-GP采用经典的树结构。函数集中的运算符充当树的非终端节点。个体表示过程是按照一定顺序组合TFP-PJM问题的若干状态特征，以获得优先级函数。优先级函数允许对所有候选人进行排序，并依次进行选择，以组建项目团队。为了为候选人选择合适的岗位，启发式方法将在可用岗位中选择匹配度最高的岗位。

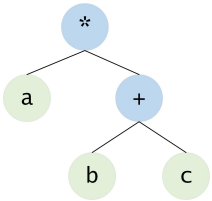


图4：RL-GP中的个体结构示例

图4展示了RL-GP中的一个个体结构示例。它是一个由三个终端（例如，a、b和c）以及两个函数（例如，*和 +）组成的优先级函数$a * (b + c)$。将所有候选人的相应特征属性代入优先级函数后，即可获得各自的优先级值。通过排序，优先级值最高的候选人将被选中加入团队。

在解码过程中，将每位候选人的特征属性代入个体组合规则，以获得不同候选人的排名，并选出优先级最高的候选人。之后，使用直觉模糊数计算该候选人与所有岗位的匹配分数。根据计算结果，将该候选人安排到匹配分数最高的岗位。安排好第一个成员后，修改个体中的低层启发式规则，从而得到新的候选人优先级排序，并根据更新后的排序从剩余候选人中选出下一名成员。随后，将新选出的候选人安排到可用岗位中匹配分数最高的岗位。重复优先级排序和工作岗位选择过程，直至满足团队组建的所有要求。在个体适应度评估阶段，RL-GP 将使用代理模型估算个体的适应度值，并根据奖励和搜索性能更新智能体的状态值。以下部分将介绍强化学习中不同种群搜索策略的集成，以及使用强化学习指导遗传编程种群进化的具体过程。

C. 强化学习方法

在所提出的算法中，我们采用强化学习方法，通过从集成种群策略中选择合适的搜索模式，提高算法种群搜索的效率。强化学习涉及一个学习过程，在此过程中，智能体与其环境进行交互，获取反馈信号和奖励，以优化其决策过程并最大化累积奖励 [34]。所提出的算法在强化学习中采用了一种简单的结构化 Q-learning 方法，其中智能体的状态记录在 Q表中。种群搜索根据所选搜索模式执行，智能体的奖励则根据搜索发现的团队组建新规则进行计算，并记录在 Q表中，以指导后续决策。在选择搜索模式时，引入了 ε-贪婪策略，通过利用现有信息并持续探索尚未尝试的动作，增强算法的探索能力。

图5展示了 RL-GP 种群与 Q学习协作进化的原理，包括两个关键

表 I：终端集合表

符号	描述
NPT	团队设置的岗位数量
RNP	剩余岗位数量
SC	候选人拥有的技能数量
SCN	候选人所具备且团队需要的技能数量
ANS	候选人的平均技能数量
NSR	团队所需的技能数量
NCP	具备团队所需某项技能的候选人人数
NCR	团队所需某项技能的候选人人数
SCW	现有团队成员沟通意愿分数之和
SMP	所安排的人岗匹配的分数总和
WEC	岗位对胜任力指标评估的权重
RAND	一个介于 0 和 1 之间的随机数

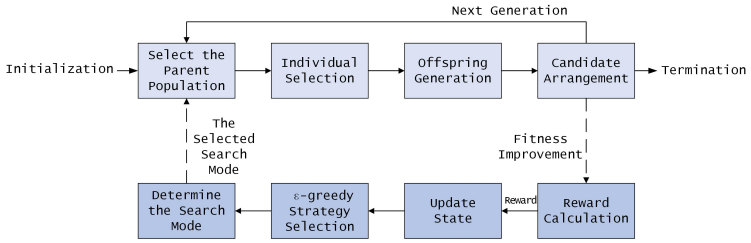


图 5：Q学习与遗传编程之间的数据流

过程。强化学习辅助 RL-GP 种群确定搜索模式，使种群能够更专注于潜在的最优解，从而提高搜索效率和质量。同时，GP 种群的进化结果会反馈给智能体，以更新 Q表，进一步指导后续搜索决策，从而获得更优的搜索结果 [35]。这两个过程持续循环交互，使算法能够找到最优规则并获得高质量的团队组建规则。在此过程中，父代种群按照特定搜索模式进行进化，并通过交叉和变异等进化算子生成子代种群。子代种群中的最佳个体输出一个新解，其对应的适应度值是当前代的最佳值。智能体比较相邻两代中最佳个体的适应度值，计算在该状态下选择某一搜索模式所获得的奖励，并更新 Q表。随着不断尝试和更新，智能体通过 Q表结合 ε-贪心策略做出最优决策，从而更高效地引导种群进化。每一代的进化结果都会经过评估，并进一步优化搜索策略，直至找到近似最优解或达到预定的迭代次数。该交互式算法旨在利用种群搜索所获得的信息来寻找最优策略。

RL-GP种群搜索仅与当前状态相关，并满足构建马尔可夫决策过程（MDP）的条件 [36]。MDP由四个组成部分构成：状态、动作、奖励和价值函数。RL-GP算法将智能体的状态与每一代种群中最优个体适应度值的提升情况相关联。这种关联可分为两类：有提升和无提升。RL-GP中的动作设置包含四种种群搜索模式，这些模式被整合到种群策略中。每一代种群进化后，都会选出表现最佳的个体，并根据不同时间点的个体适应度值之差计算奖励。 R_t 的计算遵循以下关系：

$$R_t = f_t^*(S_t, A_t) - f_{t-1}^*(S_{t-1}, A_{t-1}) \tag{27}$$

其中， $f_t^*(S_t,\ A_t)$表示智能体在时间点 t选择搜索模式 A 后，处于状态S 的RL-GP种群通过种群进化获得的最优个体的适应度值。它与时间点 $t-1$ 的最优个体适应度值之差，就是时间点 t的即时奖励 R_t 。

在种群搜索过程中，智能体根据当前状态选择不同的搜索模式，并根据搜索性能计算奖励。智能体将计算结果存储在 Q表中，并根据新状态选择下一步动作。Q值通过以下公式更新：

$$\begin{aligned} Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha[R_t + \\ \gamma \max_{a \in A} Q(S_{t+1}, a) - Q(S_t, A_t)] \end{aligned} \tag{28}$$

该函数有两个可调参数， α 表示学习率， γ 表示折扣因子。更新过程是一种基于贝尔曼方程的动态规划方法，称为贝尔曼方程 [37]。该方程根据当前状态给出最优决策值，该值等于后续状态下最优决策值的期望值与

当前状态下获得的即时奖励之和。如图 6 所示，当前状态下的 RL-GP 种群选择特定搜索模式后，便会确定下一个状态。公式中的最大值 $\max_{a \in A} Q(S_{t+1}, a)$指的是下一个状态下所有动作可达到的最大 Q 值。

在 RL-GP 算法中，智能体采用 ε-贪心策略选择动作，并选择合适的搜索模式。该策略通过设置一个取值范围为 0 到 1 的常数阈值 ε，来控制随机程度。当 ε 等于 0 时，采用 ε-贪心策略，这意味着动作完全依据当前已知的最优策略进行选择，这种策略称为 ε-贪心策略。相反，当 ε 为 1 时，表示完全随机地选择动作，称为完全随机策略。对于 ε的其他取值，智能体会在每次决策前生成一个 0 到 1 之间的随机数。当随机数小于 ε 时，智能体会从所有种群搜索策略中随机选择一种；反之，则选择 Q 值最高的种群策略进行搜索。 ε-贪心策略的伪代码如算法 1 所示。

算法 1： ε-贪心策略

输入： 状态 S_t，动作集 A，Q表 Q，参数 ε。
输出： 动作 a

```
rand ←生成一个 0 到 1 之间的随机数;
如果 ε ≤ rand，则
    a ←从动作集中随机选择一个动作 A;
否则
    a ←从 Q表中 Q 根据状态 S_t选择 Q 值最大的动作;
```

D. 集成种群策略

种群搜索是一种应用广泛的全局优化方法，涵盖多种元启发式和超启发式搜索，包括遗传编程。与传统的个体搜索相比，通过初始化一个由潜在问题解构成的种群，并对其进行持续迭代和协同进化，该方法能够降低算法收敛到局部最优的概率，并提高算法的搜索效率。

然而，对于遗传编程而言，在求解复杂组合优化问题时，仅使用单一种群搜索模型 [38]更难找到高质量解。为提高算法的搜索性能，RL-GP [39]采用了集成种群策略。具体而言，在每一代种群搜索开始之前，算法会使用强化学习方法为当前搜索选择一种种群搜索模式。RL-GP算法中的四种种群策略的集成如图 7 所示。

- $\mathbf{P_1}$: (普通种群的种群规模为N。) 这是最常见的种群进化形式遗传编程算法中用于初始化的方法。该方法通过选择、交叉和变异等进化操作生成一个子代种群。

- $\mathbf{P_2}$：（单个精英个体种群的种群规模为 N。） 为加快 RL-GP的收敛速度，我们使用全局最优个体随机替换 P_1 中的一个个体，以形成 P_2。在每一代种群进化之后，我们将当前最优个体的适应度与全局精英个体进行比较。如果最优个体优于精英个体，我们使用新的精英个体替换原始 P_2 种群中的精英个体。

- $\mathbf{P_3}$：（双精英个体种群的种群规模为 2。） P_3 种群由全局最优个体和上一代的最佳个体组成。这种种群搜索模式类似于邻域搜索，能够在局部解空间中挖掘高质量解。

- $\mathbf{P_4}$：（普通种群的种群规模为 $+$，双精英个体种群的种群规模为 $N + 2$。） P_4 种群由 P_1 和 P_3 共同组成。这样的种群在全局与局部搜索能力之间取得了平衡。因此，该种群的算法搜索需要更多的计算资源。

四种种群搜索策略被整合到遗传编程框架中，并通过强化学习方法进行选择，使RL-GP算法具备很强的搜索自适应能力。智能体可以根据当前搜索性能动态调整搜索策略，以提高搜索效果，从而增加RL-GP算法找到最优解的机会。

E. 代理模型

由于TFP-PJM问题具有NP难度，生成团队组建方案需要进行复杂的约束检查。约束评估与验证过程的计算量很大，会显著影响算法 [25]的学习效率。随着问题规模增大，算法的学习时间也会相应增加。鉴于适应度函数评估的计算开销很大，因此有必要寻找替代方法。代理建模为快速评估个体适应度函数提供了一种解决思路。参见 [32], ，我们使用K近邻（KNN）模型来近似适应度函数值。该简单的机器学习模型通过计算个体之间的距离，求出与新个体最近的 k 个个体的适应度值平均值，从而估算新个体的预测值。这种适应度评估方法的时间复杂度较低，能够加快迭代种群搜索过程。

由于 RL-GP算法采用的个体表示不同于其他基于种群的搜索算法，树结构不允许将基因型直接输入代理模型。因此，参考 [32], ，我们生成

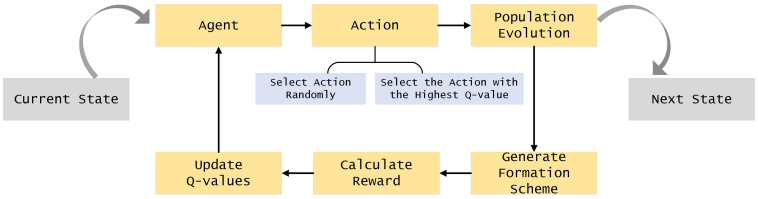


图 6：Q学习 方法的决策过程

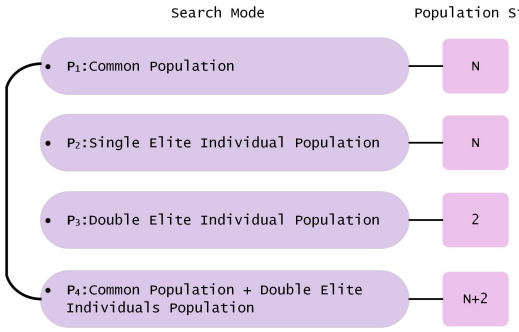


图 7：四种搜索模式

一个决策向量，方法是使用参考规则和其他规则。其中，参考规则表示员工对工作岗位的适合度，并按降序排列；其他规则则源自 RL-GP学习。为构建每个工作维度的决策向量值，使用参考规则和 RL-GP学习得到的规则获取候选人的排名结果。之后，选取学习得到的规则中排名第一的候选人，并将其在参考规则下的排名结果作为该维度的决策向量值。使用通过 RL-GP学习获得的规则 A 和 B 生成决策向量$\mathbf{d^A}$ 和$\mathbf{d^B}$后，计算这两个向量之间的距离（向量包含i个位置），如下所示：

$$D(\mathbf{d^A}, \mathbf{d^B}) = \sqrt{\sum_{i=1}^{k} \left(d_i^A - d_i^B\right)^2} \tag{29}$$

通过 KNN 模型计算预测个体与训练集中其他个体之间的距离。找出与该个体距离最近的 k 个个体，并计算其适应度值的平均值，作为该个体的预测值。对于 k 的选择，需要考虑训练集中实际个体的数量，并可在模型运行过程中灵活调整[40]。为确保 KNN 模型的预测效果和计算效率，采用阈值 μ 来决定是否更新代理模型。具体而言，每隔 μ 代，RL-GP 都会通过真实评估模型评估搜索过程中此前每一代找到的最优个体的表现。随后，将这些个体及其对应的真实适应度函数值添加到训练集中，并更新 KNN 模型。通过在线学习，可以持续提高模型预测的准确性，从而确保算法最终生成的最优个体的质量

算法最终生成的最优个体的

F. RL-GP 的学习过程

RL-GP 的框架和一些策略已在前文中介绍。本部分通过算法 2 给出了 RL-GP 的学习过程。

算法 2： RL-GP 的学习过程

输入： 种群规模 N_p，学习率 α，折扣因素 γ，动作集 A、ε、k，阈值 μ

输出： $gobal\ best\ indi$。

1 $P \leftarrow$初始化种群;
2 初始化代理模型;
3 初始化 Q表并设置 $t \leftarrow 0$;
4 设置 $iter \leftarrow 0$;
5 **while** 未满足终止条件**do**
6 $a \leftarrow$ 使用 IV-D 中的搜索模式进行选择算法 1 中的 ε-贪心策略;
7 $P \leftarrow$ 根据情况调整种群组成到动作 a;
8 从 P 中选择个体;
9 $P' \leftarrow$ 通过使用交叉和变异算子进行种群进化（P）;
10 使用 IV-E 中的代理模型评估适应度;
11 $\Omega \leftarrow$记录子代中的最优个体代;
12 $R_t \leftarrow$使用公式（27）计算奖励;
13 $Q_{t+1} \leftarrow$计算 Q值并更新 Q表，使用公式 (28);
14 **如果** $local_{best} \geq gobal_{best}$ **则**
15 $gobal_{best} \leftarrow local_{best}$;
16 $gobal_best_indi \leftarrow local_best_indi$;
17 **如果** $mod(iter,\ \mu)$ == 0 **则**
18 $f' \leftarrow$ 使用 III (Ω) 中的模型评估适应度;
19 更新代理模型（f'）;
20 $\Omega \leftarrow [\ ]$;
21 $iter \leftarrow 0$;
22 $t \leftarrow t + 1$;
23 $iter \leftarrow iter + 1$

如算法 2 所示，RL-GP 使用代理模型完成个体适应度评估后，记录种群中表现最佳的个体（第 9 行）。随后利用这些记录的个体计算真实适应度函数（第 16 行），以更新代理模型（第 17 行）。

V. 实验

在本节中，将介绍实验设置、实验结果和讨论。

A. 实验设置

实验环境：本节通过一系列实验验证 RL-GP 算法的有效性。所有算法均在配备 Core(TM) i5-8265U CPU 1.60GHz 的 Windows 10 操作系统计算机上运行，代码环境为 Python 3.11。

实验场景：目前没有针对 TFP-PJM 问题的公开基准测试，因此我们采用随机生成的方法，构造不同规模的场景。为便于表示不同实例，我们采用“P-ID”的形式表示实例，其中“P”表示工作岗位数量，“ID”表示同一工作岗位规模下的内部编号。

对比算法：为验证 RL-GP 算法的解性能，我们在实验中采用了多种遗传编程算法、构造启发式和搜索算法。我们选择传统遗传编程（记为 BGP）、使用外部存储库的遗传编程（记为 GP-E）、两种构造启发式算法（分别记为 CH1 和 CH2）、改进遗传算法（IGA） [41],和变邻域搜索算法（VNS） [42] 作为对比算法。构造启发式算法 CH1 和 CH2 根据不同标准生成候选人的选择偏好。CH1 根据候选人在团队中满足的技能数量，按从高到低的顺序生成候选人选择偏好。CH2 则根据人岗匹配分数生成候选人选择偏好，同样按降序排列。

算法参数设置：经过对参数进行交叉验证实验后，我们作出如下设置。在 RL-GP 中，种群规模设为 100，最大迭代次数设为 100，初始化方法采用 ramped-half-and-half，交叉采用两点交叉，变异采用单点变异，交叉概率设为 0.9，变异概率设为 0.1，折扣因子设为 0.9，学习率设为 0.01，k 设为 5。函数集包含 $\{+,-,*,/$,正弦、余弦、最大值、最小值$\}$。值得注意的是，当分母为 0 时，除法运算通过保护策略返回结果 1。其他遗传编程算法的参数设置与 RL-GP 相同，搜索算法的参数设置与相关文献保持一致。

评估指标：对于学习过程，不同实例规模下算法的学习时间长度可以反映算法的学习速度。目标函数值用于评估遗传编程算法在测试过程中的表现。对于其他搜索算法，由于搜索过程存在不确定性，取10次运行的最大值（记为最大值）和平均值（记为平均值）。此外，还使用所提算法与对比算法之间的差值百分比（记为差距）。用于评估该算法生成超启发式规则的性能。

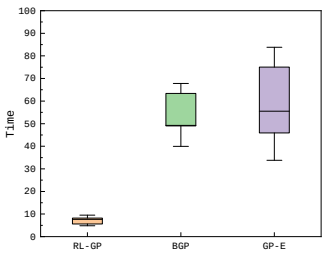
B. 实验结果

实验结果分别比较了 RL-GP 与其他遗传规划算法、构造启发式算法以及其他搜索算法。随后，还分析了所学习的编程启发式方法、岗位与技能之比对算法性能的影响。

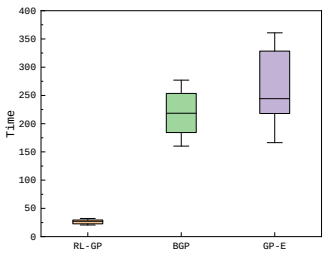
1）与其他遗传规划算法的比较：作为遗传编程的改进形式，需要通过实验验证所采用的改进策略能否在时间和解质量两方面对 RL-GP 的性能发挥积极作用。首先，比较不同遗传编程算法所需的学习时间。如图8所示，RL-GP 消耗的学习时间显著短于 GP 和 GP-E 算法，这反映出该算法使用的代理模型能够显著降低评估个体适应度值的计算成本。随着问题规模增大，这种适应度评估方法的优势也更加明显。例如，100个工作实例中的候选人数是50个工作实例中的两倍。选择其中一部分候选人组成团队，约束判断次数的增加会使每次解生成耗时更长。然而，随着问题规模增大，使用代理模型计算适应度函数值并不会导致计算量显著增加。值得一提的是，代理模型计算并不能完全取代实际解评估方法。因此，经过一定数量的代之后，RL-GP 会计算真实适应度函数值，并更新最优解。

随后，我们将 RL-GP 与另外两种用于求解问题的同类 GP 算法进行了性能比较。如表II所示，RL-GP 能够获得优于 BGP 和 GP-E 的匹配解，其中 TE 表示总效率。在 RL-GP 中，利用强化学习方法选择种群搜索模式，可以帮助算法根据搜索过程中获得的信息调整搜索策略。如果问题的解空间极其复杂，组合搜索方法能够使算法更容易平衡探索与利用。在种群中使用精英个体，更有可能推动搜索朝着良好的方向发展。此外，在大多数情况下，GP-E 的表现优于 BGP。

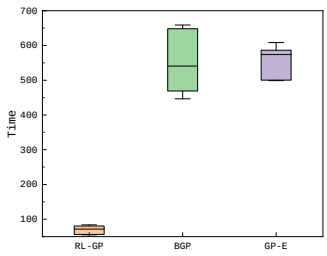
2）与构造启发式算法的比较：为了评估通过RL-GP算法训练得出的启发式规则在求解TFP-PJM问题时的性能，本文采用两种构造启发式方法作为基准进行比较。构造启发式算法和RL-GP算法都能快速获得团队组建解。本节重点根据解的总体效率评估算法的解性能。如图9所示，RL-GP算法能够获得优于CH1和CH2算法的团队解。算法之间明显的差距体现了种群迭代搜索在寻找合适策略方面的有效性



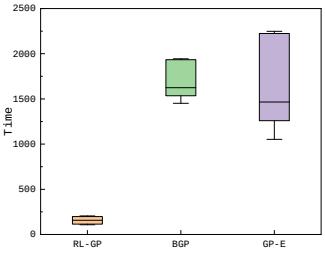
(a) 25 个岗位实例学习时间的箱线图



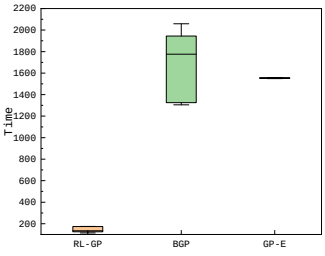
(b) 50 个岗位实例学习时间的箱线图



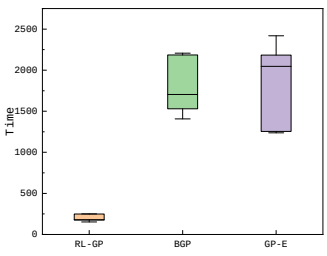
(c) 具有75个位置的实例的学习时间箱线图



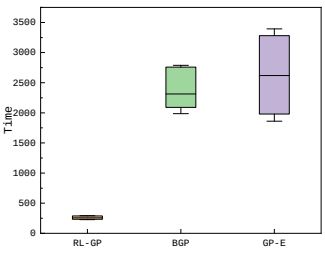
(d) 实例学习时间的箱线图
包含100个位置



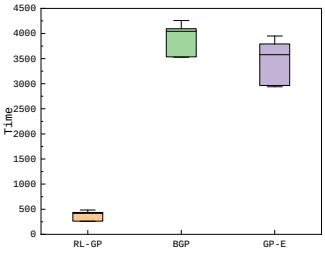
(e) 实例学习时间的箱线图
包含125个位置



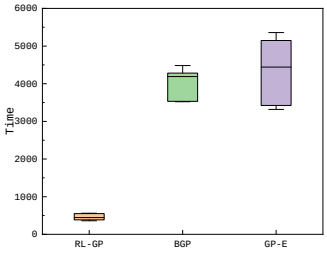
(f) 150 个岗位实例学习时间的箱线图



(g) 实例学习时间的箱线图
有 175 个位置



（h）各实例学习时间的箱线图
包含200个位置



（i）实例学习时间的箱线图
具有225个位置

图8：不同规模岗位实例学习时间的箱线图

。使用代理模型评估RL-GP的适应度函数值，必然会导致遗漏一些可能获得高匹配度解的个体。然而，对于TFP-PJM这类复杂的组合优化问题，必须在学习时间和解的准确性之间取得平衡。因此，求解此类问题时，无法保证得到最优解。相反，即使解不是最佳值，只要是高质量解，也同样可以接受。

3）与其他搜索算法的比较：为进一步增强算法解的性能的说服力，我们还将其与两种搜索算法进行了比较。如表 III 所示，IGA 在部分实例中经过长时间运行后能够找到高效团队。与此同时，RL-GP 在某些实例中优于这两种搜索算法，所得团队组建方案同样出色或更优。此外，VNS 的表现不如 IGA。通过学习，RL-GP 获得了启发式方法，不仅具备泛化能力，还能够组建高效团队。强化学习方法提高了学习过程的效率，其中，每种搜索模式的状态决定了后续学习过程中要采取的动作。算法在学习过程中获得的信息能够引导种群找到更有可能取得高效率值的组建策略。

4）学习得到的调度启发式方法：本节展示了针对规模为100和200的实例学习得到的一些启发式规则。图10和图11分别展示了用于组建包含100个和200个岗位的团队的高性能启发式方法。可以看出，在两种实例规模下，能够生成高效团队解的启发式方法具有部分相似的结构。这表明，智能体在搜索过程中倾向于根据状态信息采取相似的动作，而以这种方式进行的个体进化也会因所选动作而朝着彼此接近的方向发展

表II：RL-GP 与 BGP 和 GP-E 的比较结果

实例	RL-GP	BGP		GP-E	
		TE	差距（%）	TE	差距（%）
25-1	**14.50**	**14.50**	0.00	**14.50**	0.00
25-2	**14.75**	**14.75**	0.00	**14.75**	0.00
25-3	**14.50**	13.86	-4.43	13.86	-4.43
50-1	**25.02**	24.88	-0.56	**25.02**	0.00
50-2	**24.65**	24.64	-0.05	24.52	-0.56
50-3	**25.02**	25.04	0.11	24.93	-0.36
75-1	**37.66**	36.64	-2.72	36.73	-2.49
75-2	**37.66**	36.32	-3.56	36.53	-3.02
75-3	**36.86**	35.88	-2.68	35.88	-2.68
100-1	**47.55**	**47.55**	0.00	**47.55**	0.00
100-2	47.48	46.89	-1.23	**47.79**	0.66
100-3	**47.51**	47.36	-0.31	46.61	-1.89
125-1	**59.16**	57.68	-2.50	59.08	-0.15
125-2	**58.68**	57.41	-2.17	58.64	-0.07
125-3	**59.16**	58.97	-0.33	58.57	-1.00
150-1	**69.69**	67.14	-3.66	69.66	-0.04
150-2	**69.38**	67.41	-2.83	68.43	-1.36
150-3	**68.68**	68.57	-0.17	67.95	-1.07
175-1	**80.38**	79.98	-0.49	79.99	-0.48
175-2	**78.88**	78.57	-0.39	78.57	-0.39
175-3	**79.86**	78.56	-1.63	79.44	-0.53
200-1	**91.18**	**91.18**	0.00	**91.18**	0.00
200-2	**90.52**	89.78	-0.81	88.72	-1.98
200-3	**90.42**	88.76	-1.84	89.96	-0.51
225-1	**102.42**	98.62	-3.72	98.44	-3.89
225-2	**100.66**	99.39	-1.26	99.97	-0.69
225-3	**100.94**	100.55	-0.39	100.93	-0.02

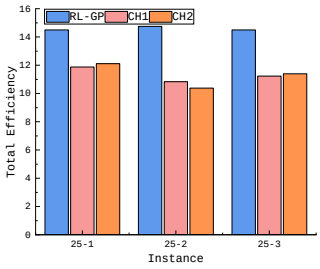
。对于较大问题规模，最优启发式方法更为复杂，因为包含更多终端节点和函数的个体结构能够更轻松地表示有关候选人的有用信息。由此生成的优先级排序使系统能够从大量候选人中选出合适人选，从而组建高效团队。

5）技能与职位之比对算法性能的影响：我们还分析了不同技能要求的数量与位置数量之间的比例关系对算法生成的解的效率值的影响。图12展示了多个实例在不同技能与位置数量比（记作 S/P）下的团队效率结果对比。技能与位置之比在一定程度上影响问题的求解难度。技能与位置之比较高，意味着组建团队时可供选择的候选人更加多样，从而扩大了问题的解空间。选择满足要求的候选人的难度也随之增加。

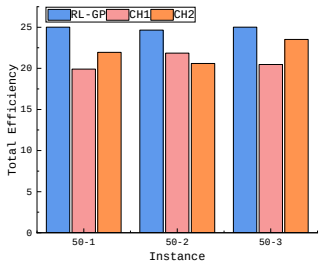
*6)*讨论：上述实验综合评估了所提出算法在解决TFP-PJM问题方面的性能，并使用各种遗传编程算法、构造启发式算法和搜索算法进行比较。例如，RL-GP算法的学习时间不到其他遗传编程算法的一半，并且与两种构造启发式方法相比，RL-GP算法获得的解的效率也有所提高。总之，该启发式方法可用于辅助决策-在实际问题场景中生成团队组建解时发挥作用。然而，搜索的随机性会对生成的高质量个体的深度产生很大影响。一般而言，深度较小的启发式方法具有较强的泛化能力，但不一定能产生最高效的团队。相反，深度较大的启发式方法往往能在个别实例中取得良好的规划性能，但未必能有效应对所有情况。因此，后续研究应进一步分析 GP 生成的个体结构中相同终端节点或函数的深度和出现频率对结果的影响。此外，个体的一些有效信息可以通过迁移直接应用于其他个体的学习策略，这也可能是未来研究的一个有前景的方向。

VI. 结论

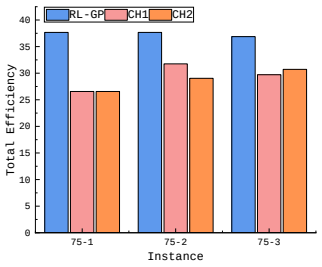
在使用强化学习辅助遗传编程解决团队组建问题时，我们提出了一种包含四种搜索模式的种群集成策略，并采用强化学习方法进行搜索模式选择，以平衡种群搜索中的探索与利用。此外，我们使用K近邻代理模型，快速评估新生成种群的迭代搜索性能。在规则学习过程中，代理模型根据获取的信息进行在线更新，以提高模型预测的准确性。大量实验表明，RL-GP所需时间显著少于 $1/5$ 至 $1/10$ 的



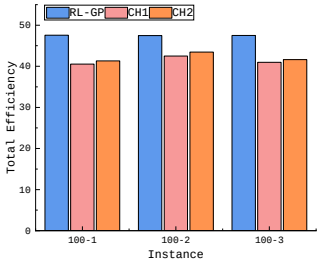
(a) 具有25个位置的实例的结果



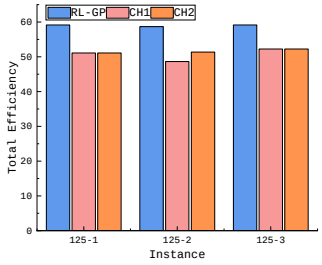
(b) 50 个岗位实例的结果



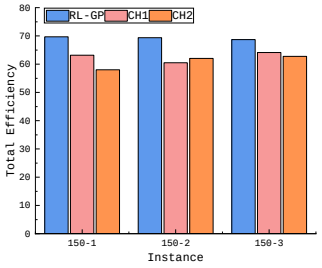
(c) 具有75个位置的实例的结果



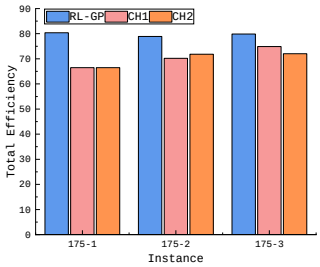
(d) 包含 100 个位置的实例结果



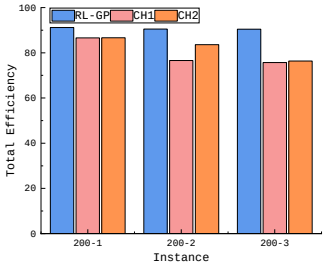
(e) 125 个岗位实例的结果



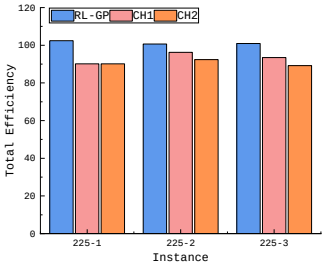
(f) 针对包含 150 个位置的实例的结果



(g) 175个位置的实例结果



(h) 200个位置的实例结果



(i) 225个位置的实例结果

图9：RL-GP与构造启发式算法的比较结果

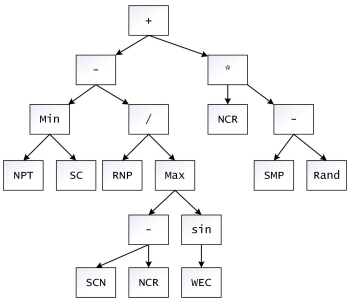


图10：创建包含100个位置的团队的良好启发式示例

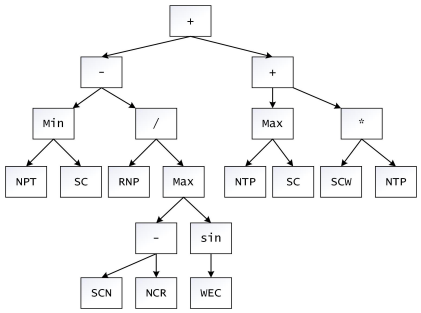


图11：创建包含200个位置的团队的良好启发式示例

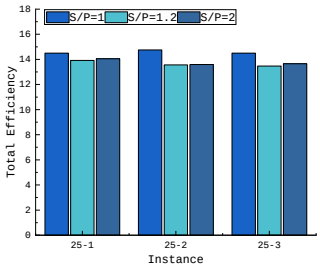
其他遗传编程方法所需的时间，便完成了学习过程。此外，RL-GP生成的团队组建规则优于实验中用于比较的已构建启发式算法。另外，一些实验表明，RL-GP能够生成团队组建

其生成的解与遗传算法和变邻域搜索算法生成的解相当或更优。

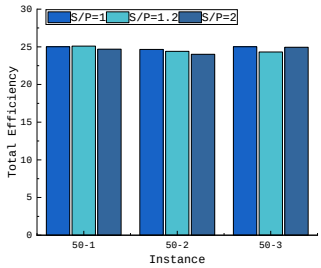
未来，我们将探索更复杂的团队组建问题，例如多团队组建和在线调整

表 III：RL-GP 与 IGA 和 VNS 的比较结果

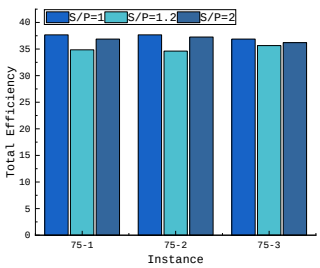
实例	RL-GP	IGA			VNS		
		Best	Ave	差距（%）	Best	Ave	差距（%）
25-1	**14.50**	**14.50**	14.39	0.00	13.01	10.75	-10.30
25-2	**14.75**	**14.75**	14.44	0.00	11.06	10.72	-25.01
25-3	**14.50**	13.86	13.70	-4.43	12.30	11.56	-15.14
50-1	**25.02**	24.83	24.09	-0.75	24.08	21.73	-3.73
50-2	24.65	**25.16**	24.21	2.04	23.48	22.61	-4.78
50-3	**25.02**	24.94	24.05	-0.33	22.57	22.23	-9.77
75-1	37.66	**37.72**	36.54	0.15	32.82	30.87	-12.86
75-2	**37.66**	36.74	36.22	-2.44	33.41	33.25	-11.29
75-3	**36.86**	36.74	35.70	-0.35	33.32	32.59	-9.61
100-1	**47.55**	46.93	45.28	-1.30	42.99	41.91	-9.59
100-2	**47.48**	46.89	45.21	-1.24	44.99	41.71	-5.25
100-3	**47.51**	47.48	45.35	-0.07	43.58	41.78	-8.27
125-1	**59.16**	59.08	57.67	-0.15	54.52	53.94	-7.84
125-2	**58.68**	**59.21**	57.17	0.90	51.80	50.07	-11.74
125-3	**59.16**	58.20	56.83	-1.63	53.23	50.21	-10.02
150-1	**69.69**	68.29	67.43	-2.02	63.89	61.70	-8.33
150-2	**69.38**	69.03	67.76	-0.49	64.08	62.66	-7.64
150-3	**68.68**	**69.73**	67.70	1.53	66.75	62.75	-2.82
175-1	**80.38**	**80.38**	79.09	0.00	76.70	75.33	-4.57
175-2	**78.88**	**78.94**	78.33	0.07	74.80	72.71	-5.17
175-3	**79.86**	79.72	78.53	-0.18	74.58	72.02	-6.61
200-1	**91.18**	90.42	88.62	-0.83	82.81	80.00	-9.17
200-2	**90.52**	89.78	88.89	-0.81	84.08	83.50	-7.11
200-3	**90.42**	**90.66**	88.95	0.26	84.56	82.16	-6.48
225-1	**102.42**	**102.42**	99.33	0.00	90.58	89.31	-11.56
225-2	**100.66**	**100.99**	99.74	0.33	95.70	93.10	-4.93
225-3	**100.94**	100.93	99.92	-0.02	94.24	92.55	-6.64



(a) 25 个岗位实例的结果



(b) 具有 50 个位置的实例的结果



(c) 具有 75 个位置的实例的结果

图12：不同 S/P 比率影响的结果

团队组建方案，并且模型中的约束将考虑更多因素。此外，进一步深入探索遗传编程算法与数据挖掘、集成学习和迁移学习等其他方法相结合的方式也很有价值。这将进一步提升遗传编程算法的性能和适用性，并有效处理更加复杂的组合- 优化问题。

参考文献

[1] J. Liu、J. Huang、T. Wang、L. Xing 和 R. He，《基于工作专长的员工发展数据驱动分析》，*IEEE Transactions on Computational Social Systems*，第 8 卷，第 2 期，第 410–422 页，2021年。[2] M. He、D. Shen、T. Wang、H. Zhao、Z. Zhang 和 R. He，《自注意力多字段特征表示与交互学习人岗匹配，”*IEEETransactions onComputational Social Systems*，2021年。[3] J. Ju´arez、C. Santos 和 C. A. Brizuela，“团队组建问题的全面综述及分类方案”，*ACM Computing Surveys (CSUR)*，第 54 卷，第 7 期，第 1–33 页，2021年。[4] F. Ahmed、K. Deb 和 A. Jindal，“板球队选拔中的多目标优化与决策方法”，《应用软计算》，第 13 卷，第 1 期，第 402–414 页，2013年。[5] H. M. Abdelsalam，“集成产品开发项目的多目标团队组建优化”，*Foundations of Computational Intelligence Volume 3:* 全局优化，第 461–478 页，2009年。[6] M. Niveditha、G. Swetha、U. Poornima 和 R. Senthilkumar，“团队组建中三目标优化的遗传方法”，载于*2016*年第八届先进计算国际会议*(ICoAC)*。IEEE，2017年，第 123–130 页。[7] M. ´A. P´erez-Toledano、F. J. Rodriguez、J. Garc´ıa-Rubio 和 S. J. Iba˜nez，“使用多目标进化算法，基于绩效指数评分选拔篮球队球员”，*PLOS ONE*，第 14 卷，第 9 期，第 e0221258 页，2019年。

[8] T. Okimoto、T. Ribeiro、D. Bouchabou和K. Inoue，“面向任务的稳健多团队组建及其在机器人救援仿真中的应用”，载于第二十五届国际人工智能联合会议（IJCAI-16），2016年。[9] S.Liemhetcharat 和 M.Veloso，“ 异构智能体团队组建中的协同效应建模与学习 ”，载于第十一届自主智能体与多智能体系统国际会议论文集—第1卷，2012年，第365–374页。[10] H.Wi、S.Oh、J.Mun 和 M. Jung，“基于知识与协作的团队组建模型”，《专家系统与应用》，第36卷，第5期，第9121–9134页，2009年。[11] L.Zhang和 X. Zhang，“面向新产品开发的多目标团队组建优化”，《计算机与工业工程》，第64卷，第3期，第804–811页，2013年。[12] J. H.Guti´errez、C.A. Astudillo、P. Ballesteros-P´erez、D.Mora-Meli`a和 A. Candia-V´ejar，“使用社会测量学的多团队组建问题”，《计算机与运筹学》，第75卷，162，2016年150页。–.[13] F. Zhang、Y. Mei、S.Nguyen、K.C.Tan 和M. Zhang，“基于多任务遗传编程的生成式超启发式：动态调度案例研究”，《IEEE控制论汇刊》，第52卷，第10期，第10 515–10 528页，2021年。[14] M. A.Ardeh、Y.Mei和 M. Zhang，“结合知识迁移与引导式搜索的遗传编程，用于不确定容量弧路径问题”，IEEE进化计算汇刊，第26卷，第4期，第765–779页，2021年。[15] B.Tan、H. Ma、Y.Mei和M.Zhang，“用于基于容器的云中在线资源分配的协同协进化遗传编程超启发式方法”，《IEEE云计算汇刊》，第10卷，第3期，第1500–1514页，2020年。[16] B.H.Boon 和G.Sierksma，“团队组建：匹配优质供给与优质需求”，《欧洲运筹学杂志》，第148卷，第2期，第277–292页，2003年。[17] E. L. Fitzpatrick 和 R. G. Askin，“结合多功能技能要求组建有效员工团队”，《计算机与工业工程》，第48卷，第3期，第593–608页，2005年。[18] Y. Xu、J. Ma和 X. Guo，“为研究团队组建建模研究人员特征”，2012年。[19] A. Baykasoglu 、 T. Dereli 和 S. Das，“ 使用模糊优化方法进行项目团队选择”，《控制论与系统：国际期刊》，第38卷，第2期，第155–185页，2007年。[20] X.Wang、Z.Zhao 和 W. Ng，“社交网络中团队组建的比较研究”，载于《高级应用数据库系统：第二十届国际会议，DASFAA2015，越南河内，2015年4月20日至23日，论文集》，第I部分20。Springer，2015年，第389–404页。[21] D.Strnad 和 N. Guid，“用于项目团队组建的模糊遗传决策支持系统”，《应用软计算》，第10卷，第4期，第1178–1187页，2010年。[22] A. Bhowmik、V.Borkar、D.Garg 和M. Pallan，“团队组建问题中的次模性”，载于《2014年SIAM国际数据挖掘会议论文集》。SIAM，2014年，第893–901页。[23] B. Xu、Y. Mei、Y.Wang、Z. Ji和M.Zhang，“结合延迟路由的遗传编程，用于多目标动态柔性作业车间调度”，《进化计算》，第29卷，105，2021年1，第75页。– .[24] S. Wang、Y.Mei、M.Zhang 和 X. Yao，“结合生态位技术的遗传编程，用于不确定容量弧路径问题”，IEEE进化计算汇刊，第26卷，87，2021年1，第73页。–.[25] F. Zhang、Y. Mei、S. Nguyen、M.Zhang和 K.C. Tan，“用于动态柔性作业车间调度的代理辅助进化多任务遗传编程”，IEEE进化计算汇刊，第25卷，第4期，第651–665页，2021年。[26] A. Farasat 和 A.G.Nikolaev，“团队组建中的社会结构优化”，《计算机与运筹学》，第74卷，142，2016年127页。– .[27] M. Fathian、M.Saei-Shahi 和 A.Makui，“考虑专家协作网络的可靠团队组建问题新优化模型”，《IEEE工程管理汇刊》，第64卷，第4期，第586–593页，2017年。[28] Z. Xu，“直觉模糊聚合算子”，《IEEE模糊系统汇刊》，第15卷，第6期，第1179–1187页，2007年。[29] X.He、Y. Wu和 D. Yu，“直觉模糊多准则决策及其在求职中的应用：比较视角”，《智能与模糊系统杂志》，第30卷，1946，2016年4，第1935页。– 。[30] Y. Mei、Q. Chen、A.Lensen、B.Xue和 M. Zhang，“基于遗传编程的可解释人工智能：综述”，*IEEE*进化计算汇刊，2022年。[31] J. Kallestad、R. Hasibi、A. Hemmati和K. S¨orensen，“用于求解组合优化问题的通用深度强化学习超启发式框架”，《欧洲运筹学杂志》，2023年。[32] T.Hildebrandt 和 J. Branke，“关于在遗传编程中使用代理模型”，《进化计算》，第23 卷，367，2015年3，343 第 – 页。[33] P. G. Espejo、S. Ventura 和 F. Herrera，“遗传编程在分类中的应用综述”，*IEEE*系统、人类与控制论汇刊*C*辑（应用与评论），第40卷，第 2 期，第 121–144 页，2009年。[34] M. Dabbaghjamanesh、A. Moeini 和A. Kavousi-Fard，“使用Q学习技术基于强化学习的电动汽车充电站负荷预测”，*IEEE*工业信息学汇刊，第 17 卷，第6 期，第 4229–4237 页，2020年。[35] L. M. Da Silva、M.F. Torquato 和 M. A.Fernandes，“面向现场可编程门阵列的强化学习Q学习技术并行实现”，*IEEE Access*，第7 卷，第 2782–2798 页，2018年。[36] S. Doltsinis、P.Ferreira 和 N.Lohse，“用于生产站爬坡优化的基于MDP模型的强化学习方法：Q学习分析”，*IEEE*系统、人类与控制论汇刊：系统，第 44 卷，第9 期，第 1125–1138 页，2014年。[37] J. Moon，“广义风险敏感最优控制与哈密顿–雅可比–贝尔曼方程”，*IEEE*自动控制汇刊，第 66 卷，第 5 期，第2319–2325页，2020年。[38] W.Ye、W. Feng 和 S. Fan，“具有动态学习策略的新型多群粒子群优化”，《应用软计算》，第 61卷，第832–843 页，2017 年。[39] G. Wu 、 R.Mallipeddi 和 P.N.Suganthan，“基于种群的优化算法的集成策略——综述”，《群智能与进化计算》，第 44 卷，第695–711 页，2019年。[40] S. Zhang、X. Li、M.Zong、X. Zhu 和 R. Wang，“具有不同近邻数量的高效KNN分类”，*IEEE*神经网络与学习系统汇刊，第 29 卷，第5 期，第1774–1785 页，2017年。[41] H. Wang、J. Li、Y. Song、J.Huang、J. Li 和 Y. Chen，“团队组建问题的改进遗传算法”，载于*2022*年 *IEEE*计算智能研讨会系列（*SSCI*）。IEEE，2022年，第 774–781 页。[42] P. Hansen、N.Mladenovi´c、J.Brimberg 和 J. A.M. P´erez，变邻域搜索。Springer，2019年。