A Two-stage Genetic Programming Framework for Stochastic Resource Constrained Multi-Project Scheduling Problem under New Project Insertions

HaoJie Chen [1], *Jian Zhang*[1], *Rong Li*[1], *Guofu Ding*[1], *Shengfeng Qin*[2]

HaoJie Chen
e-mail: chenhaojie12138@163.com
Jian Zhang(✉Correspondence author)
e-mail:jerrysmail@263.net
Rong Li(✉Correspondence author)
e-mail: bogiey@home.swjtu.edu.cn
Guofu Ding
e-mail: dingguofu@163.com
Shengfeng Qin
e-mail: sheng-feng.qin@northumbria.ac.uk

1. **School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China**
2. **Department of Design, Northumbria University, Newcastle upon Tyne NE1 8ST, UK**

Abstract: This study proposes a novel hyper-heuristic based two-stage genetic programming framework (HH-TGP) to solve the Stochastic Resource Constrained Multi-Project Scheduling Problem under New Project Insertions (SRCMPSP-NPI). It divides the evolution of genetic programming into generation and selection stages, and then establishes a multi-state combination scheduling mode with multiple priority rules (PRs) for the first time to realize resource constrained project scheduling under both stochastic activity duration and new project insertion. In the generation stage, based on a modified attribute set for multi-project scheduling, NSGA-II is hybridized to evolve a non-dominated PR set for forming a selectable PR set. While in the selection stage, the whole decision-making process is divided into multiple states based on the completion activity duration, and a weighted normalized evolution process with two crossovers, two mutations and four local search operators to match the optimal PR for each state from the PR set. Under the existing benchmark, HH-TGP is compared with the existing methods to verify its effectiveness.

Key words: Multi-state combination scheduling; Genetic programming; Hyper-heuristic; Priority rule; Stochastic resource constrained multi-project scheduling

1 Introduction

The Resource Constrained Project Scheduling Problem (RCPSP) [1] has been widely studied for many years [2] as a NP-hard problem [3], aiming to optimize one or more objective functions such as the minimum makespan with a wide variety of application constraints and extensions [4]. When dealing with a multi-projects application [5,6] in the past decade, the problem becomes the Resource Constrained Multi-Project Scheduling Problem (RCMPSP), thereby referring multiple projects as a *portfolio*. Different from RCPSP, RCMPSP needs to schedule multiple projects, each containing a set of activities with finish-to-start precedence relations, and activities from different projects may require the same resources. Therefore, the concepts of global resources and local resources are defined in RCMPSP [7]. In general, the optimization of RCMPSP is primarily focusing on the minimum makespan and only considering deterministic environment.

However, in the actual environment, due to many unpredictable dynamic events during the execution of a project or portfolio, the actual duration of an project activitymay be longer or shorter than its initial estimate. In order to deal with this uncertainty, the Stochastic Resource-Constrained (Multi-)Project Scheduling Problem (SRC(M)PSP) has been proposed and become a research hotspot in the field of project scheduling in the past decade [8,9]. Generally speaking, the ideas of solving SRC(M)PSP mainly include the following three categories. The first is *pure proactive scheduling*, in which one schedule responding to changes and uncertainties is obtained, that is, by adding some buffers to eliminate the uncertainty [10]. However, due to the actual uncertainty may exceed the extreme of the buffer, its application is limited. The second is *proactive–reactive scheduling* with two execution steps. Before the execution of the project or portfolio, the proactive scheduling creates a baseline schedule in the case of ignoring the dynamic impact, and then the reactive scheduling improves or repairs the schedule to ensure that the whole project or portfolio can be implemented normally with respect to the new baseline [11]. In proactive–reactive scheduling, all information (e.g. the duration of each activity) can be known at

the initial state or each driven time, so it can obtain schedules with better quality. Meanwhile, the reactive scheduling is to repair the initial baseline, so in addition to quality, robustness is also the most common goal of proactive-reactive scheduling [12]. The last one is *stochastic* scheduling, in which randomness is assumed to obey a known distribution, while the execution of the project or portfolio is regarded as a multi-stage decision process, and only part of the information can be obtained in each stage. As a result, this idea does not produce baseline and there is no repair process, and the key of stochastic scheduling is to obtain a so-called scheduling policy [13,14].

Around these three categories, a large number of algorithms have been proposed to solve SRC(M)PSP, mainly PR based heuristics and meta-heuristics (see Section 2.2 for details). However, PRs lack optimization ability and have problem dependence, that is, the optimal PR for different scales and objectives is often inconsistent [15]. Although meta-heuristics can avoid this problem, their responsiveness and robustness are poor due to a large number of iterative calculations and random searches. These PRs limitations and meta-heuristics in stochastic problems lead to the hyper-heuristic proposal, which designs a high-level search mechanism for selecting or generating low-level heuristics to ensure solving problems in a heuristic computational time [16]. Now, hyper-heuristics, especially genetic programming algorithm (GP), have been applied to scheduling fields, such as flow shop scheduling [17], job shop scheduling [18,19] and project scheduling (See Section 2.3 for details).

This paper proposes a novel HH-TGP to solve SRCMPSP-NPI, which extends SRCMPSP to include new project insertions for more practical reasons [20]. By constructing a multi-state PR combination scheduling mode, this method first breaks the fixed use of the same evolved PR for decision-making in existing GP to solve project scheduling. In order to obtain this PR scheduling combination, the whole evolution process is divided into two stages, namely generation stage and selection stage. The function of the generation stage is the same as that of the traditional GP, but HH-TGP is applied to SRCMPSP-NPI, resulting in the need of modifying the attribute set constituting evolved PRs. At the same time, because the final result of the generation stage is a non-dominant PR set rather than a single PR for decision-making, NSGA-II is integrated for PR performance evaluation based on non-dominated relationship. After obtaining the PR set, the selection stage is executed to evolve the PR combination in multiple states, that is, matching the most appropriate PR for each state to improve performance. In order to complete this evolution effectively, the state partition method based on the parameter of completion activity duration, the gene expression of PR combination and the genetic and local search operators are designed. The above key techniques in HH-TGP have been fully verified by SRCMPSP-NPI benchmark under five distributions. Compared with the existing research, the key contributions of this study are:

1. A novel HH-TGP framework is proposed (see Fig.2 for details), which enables GP being applied to stochastic multi-project scheduling. More importantly, it constructs a multi-state PR combined scheduling mode in project scheduling for the first time, which breaks the fixed use of evolved PR at different decision times in existing GP research.

2. In order to obtain the non-dominant PR set, two improvements of modifying the attribute set in the existing GP and combining it with NSGA-II provide a new solution space and an evaluation idea for the subsequent hyper-heuristic research in multi-project or multi-objective scheduling.

3. A new state partition method is designed with two crossovers, two mutations and four local search operators to realize the evolution of selection stage, based on completion activity duration

and the PR expression of each state with integer coding.

The remainder of the paper is organized as follows. Section 2 summarizes the existing related research from the perspective of stochastic project scheduling and hyper-heuristics, and gives the motivation of this paper. Section 3 overviews the mathematical model of SRCMPSP-NPI. The whole framework of HH-TGP and its basic components in the two stages are described in Section 4. Section 5 introduces the details of evolution in the generation and selection stages. Section 6 provides the comparative experimental results and discussion. Section 7 summarizes the findings from this work and proposes some future research directions.

2 Related work

2.1 Existing PRs and meta-heuristics for solving SRC(M)PSP

For solving SRC(M)PSP, the literature on exact algorithm is very scarce. In addition to the branch and bound method for SRCPSP [8], to our best knowledge, only Creemers [21] designed an exact procedure based on Markov chain, and Alipouri et al. [22] proposed a mixed integer linear programming model. The computational time of exact algorithm will increase exponentially. Therefore, researching into PR based heuristics and meta-heuristics attracts a lot of attention. For applying PRs and meta-heuristics to project scheduling, the key two sub problems need to be answered: (1) how to optimize activity sequencing and (2) how to transform activity sequencing into schedule. The solution of the former sub problem is to design a new priority calculation method or search mechanism, while the latter needs to explore new policy classes [23], similar to the schedule generation scheme in RCPSP [24]. In this section, the existing PRs and meta-heuristics applied to SRC(M)PSP are reviewed from these two aspects.

From the perspective of policy class, one of the most used policy classes, is *Resource-based Policy Class* (RB-policy), similar to the parallel SGS in RCPSP [23]. Its role is to make sure that, at any decision-making time, if the resource and precedence constraints are not violated, the decision maker will allow all the unimplemented activities to start. However, RB-policy has the defect of discontinuity and non-monotonicity, which leads to the existence of Graham anomalies [25]. This defect can be eliminated by adding a side start-start constraint to form *Activity-based Policy Class* (AB-policy). At the same time, based on the minimum forbidden set, *Earliest-start Policy Class* (ES-policy) [26] and *Pre-selective Policy Class* (PS-policy) [27] are introduced. The minimum forbidden set means that any two activities in the set do not satisfy the resource constraint, but any subset of the set can break this condition. The difference is that ES-policy adds finish-start constraints, while PS-policy continuously pre-selects from the activity sequence, to break the minimum forbidden set and form a new logical relationship without resource constraints between activities. Furthermore, meta-heuristics are applied on the top of finish-start and start-start constraints, resulting in *Pre-processor Policy Class* (PP-policy) [28] and *Generalized Preprocessor Policy Class* (GP-policy) [29]. In fact, PP-policy only adds finish-start constraints between activities, while GP-policy depends on finish-start and start-start constraints. PP-policy and GP-policy are both showed effective.

Compared with policy class research, many meta-heuristics are proposed to solve SRC(M)PSP. Ginzburg et al. [30] proposed a greedy meta-heuristic composed of three operators

with control, calculation and selection functions for effectively reducing the makespan. With the goal of further improving the performance, Tsai et al. [31] developed a diversified tabu search algorithm combining multiple tabu lists, randomized short-term memory and multiple starting schedules. Similarly, Ballestin et al. [32] described a GRASP-heuristic strategy with improved performance. In addition to greedy and local search, the improved evolutionary algorithm and swarm intelligence are also applied in SRC(M)PSP. Fang et al. [33] improved an estimation of distribution algorithm by combining permutation-based local search, which has a clear dominance under medium or high variance distribution. Ma et al. [34] designed an integrated genetic algorithm for searching the quasi-optimal. Satic et al. [35] analysed and compared the optimization ability and the coping ability facing the increasing uncertainty of five meta-heuristics in SRCMPSP. Sallam et al. [36] designed a control approach based on Q-learning to realize the alternating search of differential evolution and discrete algorithm.

At the same time, PR based heuristics [37-39], due to their simplicity, rapidity, stability, and intuition, have recently attracted more attention in solving RC(M)PSP. Similarly, for SRC(M)PSP, Wang et al [40] established an effective Markov decision process model to solve SRCMPSP by using the existing efficient PRs to reduce the solution space. Chen et al [23] summarized 17 PRs and applied them to schedule different scales of SRCPSP, showing that the optimal PR in stochastic environment is different from that in deterministic environment. Meanwhile, they verified that under medium or high variance distribution, the optimal PR is superior to the five state-of-the-art meta-heuristics for solving SRCPSP. Under the SRCPSP optimization with the goal of net present value, Rezaei et al. [41] also verified the superiority of PRs over existing meta-heuristics. In the same vein, Wang et al [42] analysed the performance of 20 PRs from two aspects of robustness and schedule quality. Chen et al. [20] incorporated new project insertions into SRCPSP-NPI for further exploring the solving ability of the same PRs. Chakrabortty et al [43] proposed a method based on PR to solve the SRCPSP by considering the dynamic resource demand.

2.2 Hyper-heuristics in resource constrained project scheduling

From the research in Section 2.1, when solving SRC(M)PSP, in addition to the large consumption caused by iterations and random searches, the results of the existing meta-heuristic algorithms are worse than the optimal PR under medium and high variance distribution. However, it is noticed that (1) PRs have no optimization ability and (2) there is no single optimal PR that can be applied to all problems when facing different environments (deterministic or stochastic environment), objectives or validation criterion. Meanwhile, due to the need for a lot of experience and data, it is very difficult to manually design or mix PR for improving performance [44]. Therefore, while retaining the advantages of PR decision-making, the hyper-heuristics of selecting or generating PRs with better performance by training show great potential in SRC(M)PSP.

Over the past few years, the hyper-heuristics have limited applications in resource constrained project scheduling. Initially, some local search methods are explored for the high-level search of hyper-heuristics, such as greedy search [45] and threshold accepting [46]. Then, population-based meta-heuristics become popular as high-level search. In addition to a few studies based on swarm intelligence, such as particle swarm optimization [47], GP has almost become the

most mainstream method in hyper-heuristics. For example, Lin et al. [48] adopted GP as the upper-level search and considered 10 neighbourhood structures as the low-level heuristics to optimize the multi-skill RCPSP. Based on the same framework, Zhu et al. [49] further designed a decomposition mechanism to improve population diversity and the search ability of GP. In addition, Chand et al. [50] applied GP to RCPSP, showing that the evolved PRs can be better than the existing state-of-the-art PRs. The similar improved methods are reported to considering the dynamic resource disruptions [51], enhancing the evolution ability with adding Rollout-Justification procedure [52].

Furthermore, when scheduling in stochastic environment, the research on hyper-heuristics is scarcer. Alipouri et al. [53] proposed an adaptive differential evolution hyper-heuristic, which only realized the automatic combination of existing PRs and scheduling policy classes. Kühn et al. [54] used neural network to automatically assign weights to different attributes, to establish hybrid PRs with linear priority combination. Considering the collaborative decision, Chen et al. [55] introduced ensemble learning into the hyper-heuristic framework to schedule SRCPSP by evolving a decision ensemble instead of single PR. This is the first time to realize the application of GP in RCPSP with stochastic duration.

2.3 Motivation

By analysing the above existing literature, the motivation of this paper is as follows:

1. In addition to being fast, intuitive, and stable, PR can produce better results than the existing meta-heuristics for stochastic project scheduling, which gives the great potential to the research of hyper-heuristics. However, to the best of our knowledge, there is scarce literature on GP to generate PRs for SRCMPSP, which is one of the most common cases in practice considering the randomness and the complexity of multiple projects simultaneously. At the same time, the hyper-heuristic studies in Section 2.2 are oriented to single objective or weighted normalized multi-objective optimization. The multi-objective evolution mode based on non-dominated relationship needs to be further explored, although its goal in this paper is to obtain the non-dominant PR set.

2. More importantly, the existing GP with PR generation is applied to obtain an optimal PR through iterative optimization and relies on this PR to calculate the priority during the scheduling process, so that the optimal priority of each decision point is consistent. However, it is well known that the optimal priority will be changed with the execution of a portfolio or project. For example, the resource attributes of the optimal PR should be considered more in the early state of the scheduling process and time level attribute of the optimal PR should be paid more attention in the later stage [20]. With this goal, two challenges need to be addressed in this paper: 1) A multi-state PR combination scheduling mode needs to be constructed for breaking the fixed use of PR in the decision-making process; 2) A novel GP framework needs to be designed for automatically matching the most appropriate PR for each state.

3 The problem description of SRCMPSP-NPI

In actual project management, there is often new project insertion in addition to the stochastic activity duration. For example, in aircraft assembly production, manual assembly is the main

influencing factor of stochastic duration, while emergency orders cause some production tasks (projects) not to exist at the initial scheduling time. Therefore, SRCMPSP-NPI [20] is a new problem formed by adding two dynamic factors, i.e., the stochastic activity duration and the new project insertion, on the basis of RCMPSP, in which the goal is to achieve objective optimization by scheduling a portfolio $\boldsymbol{P}=\{p_1, p_2,\ldots, p_n\}$ composed of n projects. Each project p_i ($i \in \{1, 2, ..., n\}$) can be represented as a directed acyclic graph $\boldsymbol{G_i}(\boldsymbol{V_i}, \boldsymbol{E_i})$, where the nodes $\boldsymbol{V_i} = \{a_{i,0}, a_{i,1}, ..., a_{i,m_i+1}\}$ represents m_i+2 activities, and activity $a_{i,0}$ and activity a_{i,m_i+1} are dummy activities to express the beginning and end of the project. For example, the structure of 14 activities shown in Fig.1. During the execution of $\boldsymbol{P}$, $|\boldsymbol{K}|$ renewable resources are required, each of which has a constant maximum supply R_k. At the same time, the execution of each activity $a_{i,j}$ needs a fixed resource requirement and duration $d_{i,j}$. Suppose $r_{i,j,k}$ is the requirement for resource k, and $st_{i,j}$ refers to the start time of activity $a_{i,j}$. There are two important constraints to be observed in the execution of RCMPSP [20,42].

- *Precedence constraints*: These represent the logical relationships between the activities in each project. In this study, only the finish-start constraint is considered, that is, the activities in the successors $\boldsymbol{S_{i,j}}$ of activity $a_{i,j}$ can only start after $a_{i,j}$ is completed. At the same time, there is no precedence constraint between activities from different projects.
- *Resource constraints*: These limit the number of executable activities, that is, at the same time t, the total activity requirements in the execution activity set $\boldsymbol{A_t}$ for resource k should not exceed R_k. If there is no such a constraint, the makespan lower bound of each project p_i is its critical path CP_i.

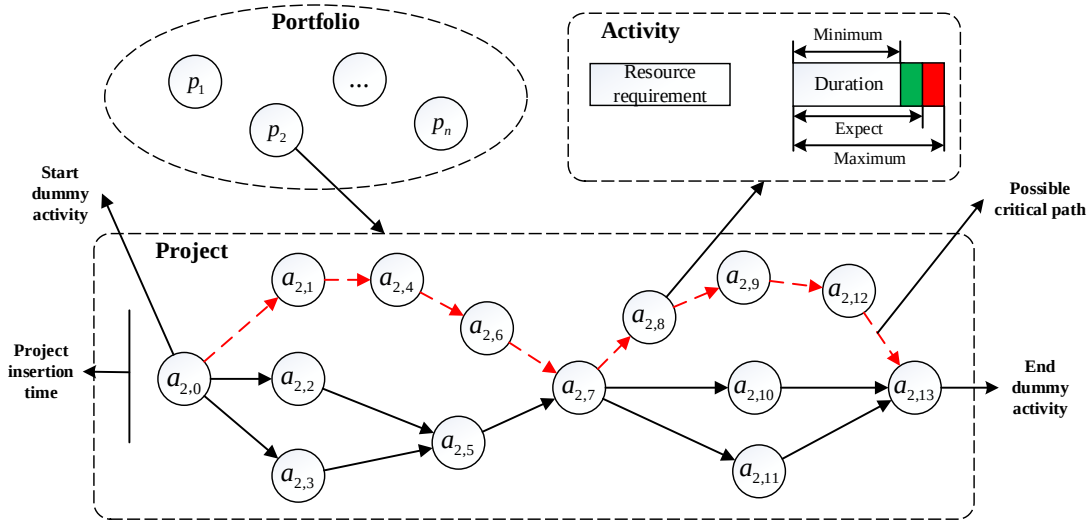


Fig.1 The structure of SRCMPSP-NPI

Unlike RCMPSP, SRCMPSP-NPI is a dynamic problem with time randomness as shown in Fig.1. First, the activity resource requirement and the resource supply are fixed. Secondly, the duration of each activity follows a known distribution over a range, which determines that the project critical path is not fixed. Thirdly, each project has a *start constraint* [20], that is, before the insertion time pst_i of project p_i, the relevant information of the project is unknown, and the start time of all activities in the project cannot be less than pst_i. Considering the evaluation of schedule quality and robustness [20,42], the mathematical model of SRCMPSP-NPI is as follows.

Objective:

$$f_{Q1}=\frac{1}{n}\sum_{p_i\in \boldsymbol{P}}\frac{AD_i-CP_i}{CP_i} \tag{1}$$

$$f_{Q2}=\frac{\max\limits_{p_i\in \boldsymbol{P}} AD_i-\max\limits_{p_i\in \boldsymbol{P}} CP_i}{\max\limits_{p_i\in \boldsymbol{P}} CP_i} \tag{2}$$

$$f_{R1}=\frac{1}{n}\sum_{p_i\in \boldsymbol{P}}\left|\frac{SAD_i-AD_i}{AD_i}\right| \tag{3}$$

$$f_{R2}=\left|\frac{\max\limits_{p_i\in \boldsymbol{P}} SAD_i-\max\limits_{p_i\in \boldsymbol{P}} AD_i}{\max\limits_{p_i\in \boldsymbol{P}} AD_i}\right| \tag{4}$$

s.t:

$$st_{i,j'}-st_{i,j}\geq d_{i,j}\quad \forall a_{i,j'}\in \boldsymbol{S_{i,j}} \tag{5}$$

$$\sum_{p_i\in \boldsymbol{P},a_{i,j}\in \boldsymbol{A_t}} r_{i,j,k}\leq R_k\quad \forall k\in \boldsymbol{K},\forall t\in T \tag{6}$$

$$pst_i\leq st_{i,j}\qquad \forall a_{i,j}\in \boldsymbol{V_i} \tag{7}$$

$$r_{i,0,k}=r_{i,(mi+1),k}=0;\ P(d_{i,0}=0)=P(d_{i,(mi+1)}=0)=1\ \ \forall k\in \boldsymbol{K},\forall p_i\in \boldsymbol{P} \tag{8}$$

$$P(d_{i,j}\leq 0)=0 \tag{9}$$

Eq.(1) to Eq.(4) express the calculation of the four objective functions. f_{Q1} and f_{Q2} respectively express the schedule quality, where AD_i represents the expected makespan of project p_i when the activity duration is equal to its expected value. These two objectives are mainly to test the ability of PR to obtain better makespan under the expected activity duration, in which f_{Q1} is to measure the balance of makespan in each project, while f_{Q2} is to evaluate the overall level (portfolio). f_{R1} and f_{R2} are robustness metrics, where SAD_i represents the average makespan of project p_i derived from Monte Carlo simulations with stochastic activity durations. Similarly, f_{R1} and f_{R2} evaluate the deviation of the actual makespan from the expected value under multiple simulations from the perspective of each project and portfolio, respectively. By referring to [20,42], in the constraints, Eq.(5), Eq.(6) and Eq.(7) are the three key constraints in the problem description respectively, namely precedence constraints, resource constraints and start constraints, where T in Eq.(6) is a maximum positive number. Eq.(8) shows the start and end dummy activity constraints, that is, each dummy activity has no resource requirements and the probability of its duration being 0 is 100%. Eq.(9) indicates the activity duration constraint, where the probability of each activity duration less than 0 is 0%.

4 The framework of HH-TGP

The framework of HH-TGP is designed and shown in Fig.2, in which the key features in the generation stage and the selection stage are highlighted in yellow and green, respectively. With parameters initialization, training set construction and policy class selection, HH-TGP first executes the generation phase to obtain a non-dominant PR set. The improvements in this stage include two points: 1) the calculation of existing attributes is modified and the project level

attribute is added for realizing PR evolution under SRCMPSP-NPI; 2) NSGA-II is combined to evaluate PR based on non-dominant relationships. Upon the completion of the generation stage, the generated non-dominated set is combined with the excellent traditional PRs to form a selectable PR set and enter the selection stage. In the selection stage, the optimal PR scheduling combination is obtained based on the following two improvements: 1) the state partition method based on the completion activity duration and the PR combination gene expression in multiple states are proposed; 2) the crossover, mutation and local search operators are designed to realize genetic evolution under PR combination coding.

Depending on the evolution process in Fig.2, each execution of HH-TGP first trains the optimal PR scheduling combination based on the training set, and then this combination performs scheduling on all test sets for performance evaluation. At the same time, in order to perform the evolution steps of generation stage and selection stage, some basic elements need to be introduced in detail, that is, the encoding and decoding in the two stages.

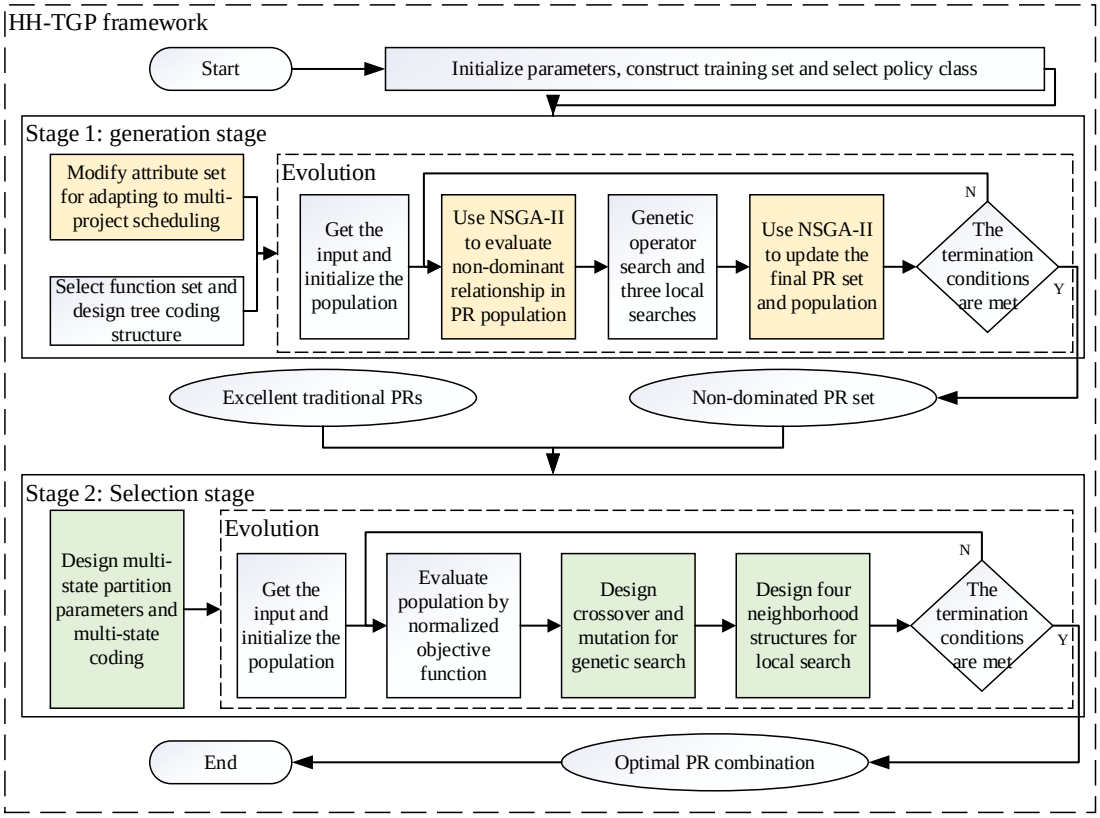


Fig.2 The framework of HH-TGP

4.1 The PR structure in generation stage

As shown in Fig.2, the function of generation stage is to generate a Pareto PR set, so the gene structure of each individual needs to express a PR in this stage. Referring to [50-52], this study selects the tree gene structure to form a PR as shown in Fig.3. The PR structure can be divided into two parts, including the bottom priority expression to construct the basis of priority value calculation and the top discriminant to judge the priority value. In the discriminant *"Jud"*, there are only two elements: *"fall"* and *"rise"*. When the discriminant is "*fall*", it is expressed as minimized, that is, the smaller the value calculated by the priority expression, the higher the

priority, while when it is "*rise*", it is maximized. In the constituent elements of priority expression, there are two symbols: attribute (*"Att"*) and function (*"f"*), that is, the priority expression combines the attributes of different levels (such as activities, resources, etc.) through different functions. Based on existing research [50,55], the function set and attribute set used in this study are shown in Table 1 and Table 2 respectively. The difference is that the project attribute "CP_i" is added in Table 2, and the calculation of all attributes is modified for adapting to multi-project scheduling. At the same time, in the initialization and search process of the PR structure, it cannot be allowed to increase infinitely, so as to prolong the priority calculation time. Therefore, the concept of maximum depth is proposed to control the distance between the topmost node and the lowest node in the priority expression. Fig.3 illustrates the maximum depth of 3.

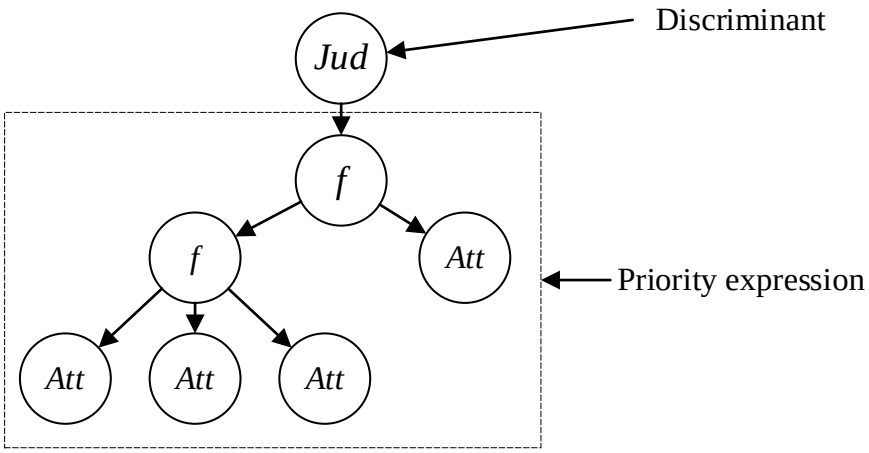


Fig.3 The PR structure

Table 1 The function set

Index	Symbol	Function	Formula	Index	Symbol	Function	Formula
1	+	Add(x,y)	$x+y$	6	−	Sub(x,y)	$x-y$
2	×	Mul(x,y)	$x\times y$	7	Neg	Neg(x)	$-1\times x$
3	Exp	Exp(x)	e^x	8	Abs	Abs(x)	$\begin{cases} x & if\ x\geq 0 \\ -1\times x & otherwise \end{cases}$
4	/	Div(x,y)	$\begin{cases} x/y & if\ y\neq 0 \\ 0 & otherwise \end{cases}$	9	Max	Max(x,y)	$\begin{cases} x & if\ x\geq y \\ y & otherwise \end{cases}$
5	Min	Min(x,y)	$\begin{cases} x & if\ x\leq y \\ y & otherwise \end{cases}$	10	If	If(z,x,y)	$\begin{cases} x & if\ z=0 \\ y & otherwise \end{cases}$

Table 2 The activity attribute set

Attribute	Calculation formula
Early Finish ($EF_{i,j}$)	$\frac{EF_{i,j}}{\max EF_{i',j'}} \quad a_{i,j}, a_{i',j'} \in \boldsymbol{AE}_t$
Late Start ($LS_{i,j}$)	$\frac{LS_{i,j}}{\max LS_{i',j'}} \quad a_{i,j}, a_{i',j'} \in \boldsymbol{AE}_t$
Late Finish ($LF_{i,j}$)	$\frac{LF_{i,j}}{\max LF_{i',j'}} \quad a_{i,j}, a_{i',j'} \in \boldsymbol{AE}_t$

Total Successor ($TS_{i,j}$)	$\frac{\lvert \boldsymbol{TS}_{i,j} \rvert}{\lvert \boldsymbol{V}_i \rvert - 1} \quad a_{i,j} \in \boldsymbol{AE}_t$
Total Successor Duration ($TSD_{i,j}$)	$\frac{1}{\sum_{a_{i,j'} \in \boldsymbol{V}_i} d_{i,j'}} \sum_{a_{i,j'} \in \boldsymbol{TS}_{i,j}} d_{i,j'} \quad a_{i,j} \in \boldsymbol{AE}_t$
Duration ($DT_{i,j}$)	$\frac{d_{i,j}}{\max d_{i',j'}} \quad a_{i,j}, a_{i',j'} \in \boldsymbol{AE}_t$
Resources Required ($RR_{i,j}$)	$\frac{1}{\lvert \boldsymbol{K} \rvert} \sum_{k=1}^{\lvert \boldsymbol{K} \rvert} \begin{cases} 1 & if\ r_{i,j,k} > 0 \\ 0 & otherwise \end{cases} \quad a_{i,j} \in \boldsymbol{AE}_t$
Average Resource Requirement ($AvgRR_{i,j}$)	$\frac{1}{\lvert \boldsymbol{K} \rvert} \sum_{k=1}^{\lvert \boldsymbol{K} \rvert} \frac{r_{i,j,k}}{R_k} \quad a_{i,j} \in \boldsymbol{AE}_t$
Maximum Resource Requirement ($MaxRR_{i,j}$)	$\max \frac{r_{i,j,k}}{R_k} \quad a_{i,j} \in \boldsymbol{AE}_t$
Minimum Resource Requirement ($MinRR_{i,j}$)	$\min \frac{r_{i,j,k}}{R_k} \quad a_{i,j} \in \boldsymbol{AE}_t$
Critical Path Length (CP_i)	$\frac{CP_i - \min CP_{i'}}{\max CP_{i'} - \min CP_{i'}} \quad a_{i,j}, a_{i',j'} \in \boldsymbol{AE}_t$

Because the SRCMPSP-NPI is a dynamic problem, only the activities in the eligible set $\boldsymbol{AE_t}$ at point t can be selected when making decisions (i.e., calculating priority and sorting). The condition for an activity to be eligible is that the project to which it belongs has been started and all its predecessors have been completed. In addition, due to RB-policy integrated with critical path method is selected to transform the schedule (see Section 3.3 for details), the earliest start time of all activities in $\boldsymbol{AE_t}$ is the same at each decision point, so the attribute of earliest start does not exist in Table 2. The normalized attributes in Table 2 have the following meanings:

- $EF_{i,j}$: The earliest finish time of activity $a_{i,j}$, of which calculation ignores resource constraints.
- $LS_{i,j}/LF_{i,j}$: The latest start/finish time of activity $a_{i,j}$, and the resource constraints are ignored when calculating.
- $TS_{i,j}$: The total number of all immediate and non-immediate successors of activity $a_{i,j}$.
- $TSD_{i,j}$: The duration sum of all immediate and non-immediate successors of activity $a_{i,j}$.
- $RR_{i,j}$: The total number of resource types required for activity $a_{i,j}$.
- $AvgRR_{i,j}/MaxRR_{i,j}/MinRR_{i,j}$: The average/maximum/minimum resource requirement of activity $a_{i,j}$ across its duration.
- CP_i: The critical path length of the project to which the activity $a_{i,j}$ belongs.

4.2 The gene expression in selection stage

Based on the selectable PR set formed after the generation stage, the goal of the selection stage is to form a PR combination under multiple states. Therefore, the gene expression at this stage is the matching relationship between different states and the selected PRs. For SRCMPSP-NPI, its basic elements determine that the better way to express its state is the ratio of current resource occupancy to supply and the portfolio current progress. However, the resource occupancy, also affected by RB-policy, can always reach the maximum at each decision time, so

this study uses the ratio of the total completion activity duration and the total activity duration to express the portfolio progress as the state parameter, as shown in Eq. (10).

$$ratio = \left(\sum_{p_i \in \boldsymbol{P}} \sum_{a_{i,j} \in \boldsymbol{AC_i}} d_{i,j}\right) / \left(\sum_{p_{i'} \in \boldsymbol{P}} \sum_{a_{i',j'} \in \mathbf{V_i}} d_{i',j'}\right) \tag{10}$$

Where $\boldsymbol{AC}_i$ represents a set of completed activities in project p_i.

Based on this parameter, the following two steps are needed to form the gene expression in the selection stage. The first is to establish the index of all PRs in the selectable set. To further improve the selection scope, in addition to the PRs evolved in the generation stage, the selectable set also includes some excellent traditional PRs recommended by [20], as shown in Fig.2. The second step is to set the state number to divide the portfolio proportionally. For example, the gene expression under five states is shown in Fig.4 where integer coding is used to correspond to the numbered PR in the selection stage, and the interval length between states is 20% since there are only five stages in total. As shown in Fig.4, when the value of *ratio* is in the range of 0% to 20%, it belongs to state 1, and the PR with index seven is used in this state.

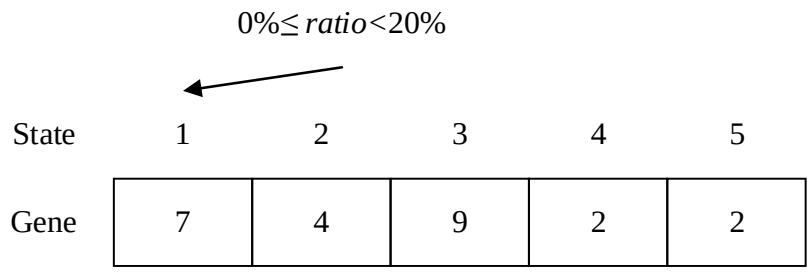


Fig.4 The example of gene expression in selection stage

4.3 The RB-policy for decoding

Because of the function for transforming solution into schedule, the policy plays an important role in HH-TGP with the evaluation in evolution training and decision-making in testing. To realize HH-TGP framework with heuristic computation time, the selection of policy class should also need to avoid iteration or extra computation. Taking the single project instance in PSPLIB as an example [8], the calculation of minimum forbidden set, will increase dramatically from 326 with 30 activities to 243,871 with 120 activities. Therefore, the selection of ES-policy and PS-policy will lead to a lot of extra computation, especially in a larger scale multi-project environment. Similarly, PP-policy and GP-policy generate computation time consumption by iterating. In addition, a large number of experimental results show that good RB-policy is significantly better than good AB-policy in terms of average schedule quality [23], so RB-policy is our final choice.

On this basis, the critical path method is integrated into traditional RB-policy for two reasons. First, the duration of each activity in SRCMPSP-NPI is uncertain, so the priority value calculated initially may change with the progress of decision-making. For example, the critical path in the initial state may not match that of a decision point. At the same time, some projects are inserted during the portfolio execution, so the information about these projects cannot be obtained to calculate the priority before the insertion time. Second, the existing research [38,55] found that dynamic priority calculation will bring better schedule quality. Therefore, the use of the critical

path method is very necessary to combine with its RB-policy for dealing with the decoding in generation and selection states, as shown in Algorithm 1. It is worth mentioning that in the Algorithm 1, each activity $a_{i,j}$ has three judgement variables $Js_{i,j}$, $Jc_{i,j}$ and $Jr_{i,j}$, and their value range is only "true" or "false". When $Js_{i,j}$/$Jc_{i,j}$/$Jr_{i,j}$ is "true", it indicates that activity $a_{i,j}$ has been scheduled/completed/ satisfied the resource constraint.

Algorithm 1 The RB-policy in HH-TGP

Input: the selectable PR set $\boldsymbol{PR_s}$, the selection gene $\boldsymbol{Gene_s}$, the instance $\boldsymbol{P}$, the decision point t;
Output: the scheduled activity $\boldsymbol{AS_t}$;
$\boldsymbol{AC_t}\leftarrow\{\}$, $\boldsymbol{AE_t}\leftarrow\{\}$, $\boldsymbol{AS_t}\leftarrow\{\}$; // $\boldsymbol{AC_t}$/ $\boldsymbol{AE_t}$ means the completed/eligible activity set
// obtain eligible set and completion set
for p_i **in** $\boldsymbol{P}$
  if $pst_i \geqslant t$ **and** $Js_{i,(mi+1)}$==**false**
    for $a_{i,j}$ **in** $\boldsymbol{V_i}$
      if $Jc_{i,j}$==**true**
        $\boldsymbol{AC_t}\leftarrow\boldsymbol{AC_t}\cup\{a_{i,j}\}$;
      else
        Set: $\boldsymbol{PS_{i,j}}\leftarrow$the predecessor set of $a_{i,j}$;
        if (| $\boldsymbol{PS_{i,j}}$ | ==0 **or** all activities in $\boldsymbol{PS_{i,j}}$ are completed) **and** $Js_{i,j}$==**true**
          Update the earliest/latest start/finish time and all attribute values of $a_{i,j}$;
          $\boldsymbol{AE_t}\leftarrow\boldsymbol{AE_t}\cup\{a_{i,j}\}$;
        end if
      end if
    end for
  end if
end for
// Calculate priority and perform scheduling
Normalize the attribute of activities in $\boldsymbol{AE_t}$ based on Table 2;
Set *index*$\leftarrow$0; // the index of selected PR
if |$\boldsymbol{Gene_s}$|>0
  Calculate the *ratio* according to Eq.(10) and obtain the state s;
  index$\leftarrow$The s-th gene in $\boldsymbol{Gene_s}$;
end if
Set *PR*$\leftarrow$The *index*-th PR in $\boldsymbol{PR_g}$
Maximize or minimize sorting activities in $\boldsymbol{AE_t}$ according to *PR* structure;
for $a_{i,j}$ **in** $\boldsymbol{AE_t}$
  Set $Jr_{i,j}\leftarrow$update the resource satisfaction when scheduling $a_{i,j}$;
  if $Jr_{i,j}$==**true**
    Update the resource supply;
    $\boldsymbol{AS_t}\leftarrow\boldsymbol{AS_t}\cup\{a_{i,j}\}$;
  end if
end for
return $\boldsymbol{AS_t}$

5 The evolutions in HH-TGP

Based on the encoding and decoding methods described in Section 4, this section describes the evolution process of the generation and selection stage in detail, including initialization, evaluation, and search operators. Before that, a common evolution characteristic of the two stages needs to be emphasized. Although the objective function of SRCMPSP-NPI includes scheduling quality and robustness, the training of robustness is very difficult due to the following two reasons. Firstly, for a specific distribution, having enough sampling times is necessary, otherwise the sampling results may likely be only applicable to a certain distribution or even smaller range due to randomness. As a result, an individual's evaluation calculation increases dramatically in generation and selection stages, because it needs to schedule all the sampling data. Secondly, the results with good robustness in one distribution are not necessarily suitable for other distributions, which leads to the further increase of sampling. Therefore, in order to reduce the calculation amount and increase the generalization of evolved PR combination, in the training process of HH-TGP, it is assumed that the activity duration is the expected value $d_{i,j}^*$, and the evaluation of PR (combination) only depends on f_{Q1} and f_{Q2}. In this way, each instance in the training set only needs to be scheduled once without violating SRCMPSP-NPI's goal of optimizing the expected makespan. In addition, the PR combination evolved by HH-TGP divides the whole decision-making process into multiple states. Since the stochastic activity duration may only affect some states and the complementary effect of different PRs in different states, the deterioration degree of randomness is reduced, so the robustness is hoped to be compensated by HH-TGP framework in the test decision.

5.1 Non-dominant evolution in generation stage

As shown in Fig.2, in order to obtain a non-dominated PR set, the PR evolution in the generation stage depends on the non-dominated relationship, and the evolution operators of the two parts are described below. The first is population initialization and evaluation. To randomly generate different PR tree structures as shown in Fig.3, initialization includes two steps to obtain discriminant and priority expression respectively. In the discriminant, "*fall*" and "*rise*" are randomly generated with a probability of 0.5, while the priority expression is generated by the classical ramped half-and-half method, in which the tree depth is controlled between 2 and 6 [56]. In the evaluation, since only f_{Q1} and f_{Q2} need to be considered in the evolution of HH-TGP, the evaluation algorithm conducive to more than three objectives is not considered for easy implementation and stable performance. At the same time, NSGA-II can achieve better results in most cases [57], so it can be used as the evaluation of the generation stage in this study.

In addition, the search in generation stage plays a very important role, including selection, crossover mutation in standard GP and additional local search. Based on the analysis in [58], the tournament selection is adopted by HH-TGP to obtain the sub-population performing genetic and local search, that is, continuously executing the comparison cycle between two randomly selected PRs until the PR number of the sub population reaches 90% of the population. In each comparison, the PR meeting one of the following two conditions wins: 1) the PR with smaller number of non-dominated layer; 2) the PR with the same number of non-dominated layer and larger

crowding distance. At the same time, based on existing research [50,55], the search methods in the generation stage are described as follows, and their examples are shown in Fig.5.

- *Crossover*: The crossover is the only search that requires the participation of two parent PRs, in which each parent PR exchanges a part of its own structure tree to form new two offspring PRs, as shown in Fig.5(a).
- *Node replacement local search*: As shown in Fig.5(b), this local search will replace a node of the original PR by a randomly generated node. It should be noted that the replacement node must have the same number of child nodes as the original node. For example, an attribute node can only be replaced by another attribute node. Since only the "if" node in Table 2 has three child nodes, when the selected replacement node is "if", the first judgment child node will change, that is, "0" will become "1".
- *Subtree replacement local search*: Like crossover, this local search also uses a new subtree to replace part of the original PR, but the difference is that the subtree is randomly generated, that is to say, it may produce a new subtree structure that does not exist in the original population.
- *Subtree deletion local search*: Using this local search will delete the subtree of the selected node as the root node to keep another subtree under its parent node. However, if the parent node of the selected node has only one child node, this local search may cause the original PR to leave only one discriminant, so under this condition, an attribute node will be randomly selected to replace the deleted subtree.
- *Discriminant mutation*: In the above three local searches and crossover, almost all search methods are included in the tree structure, but the top-level discriminant has not changed. Therefore, the mutation in HH-TGP will only perform discriminant mutation, that is, if the original discriminant is "fall", it will become "rise".

Based on the above evolution operators, the execution process of the generation stage is shown in the Algorithm 2.

Algorithm 2 The execution process of generation stage

Input: the all portfolios $\boldsymbol{P_{total}}$ in the training set, the mutation rate p_m, the crossover rate p_c, the maximum iteration number Max_{gen}, the population size Pop_{size};
Output: the non-dominated PR set $\boldsymbol{PR_n}$;
1: Randomly initialize Pop_{size} PR to form population $\boldsymbol{PR_t}$;
2: Calculate the objective functions f_{Q1} and f_{Q2} of each PR in $\boldsymbol{PR_t}$ based on $\boldsymbol{P_{total}}$;
3: **for** *gen* **in** {1, 2, …, Max_{gen}}
4: Use NSGA-II for calculating non-dominated stratification and crowding distance;
5: Execute tournament selection operation to form $\boldsymbol{PR_{sub}}$ and randomly shuffle $\boldsymbol{PR_{sub}}$;
6: **for** *i* **in** {1, 2, …, $|\boldsymbol{PR_{sub}}|$-1}
7: **if** $rand<p_c$ // *rand* is a random number from 0 to 1
8: Execute crossover operation;
9: **end if**
10: **end for**
11: **for** *i* **in** {1, 2, …, $|\boldsymbol{PR_{sub}}|$}
12: Generate a random integer *index* from 0 to 2
13: **switch**(*index*)

case 0: subtree replacement local search; **break**;
case 1: node replacement local search; **break**;
case 2: subtree deletion local search; **break**;
end switch
if $rand < p_m$
Execute discriminant mutation;
end if
end for
Update $\boldsymbol{PR_n}$ with non-dominated relationship;
end for
return $\boldsymbol{PR_n}$

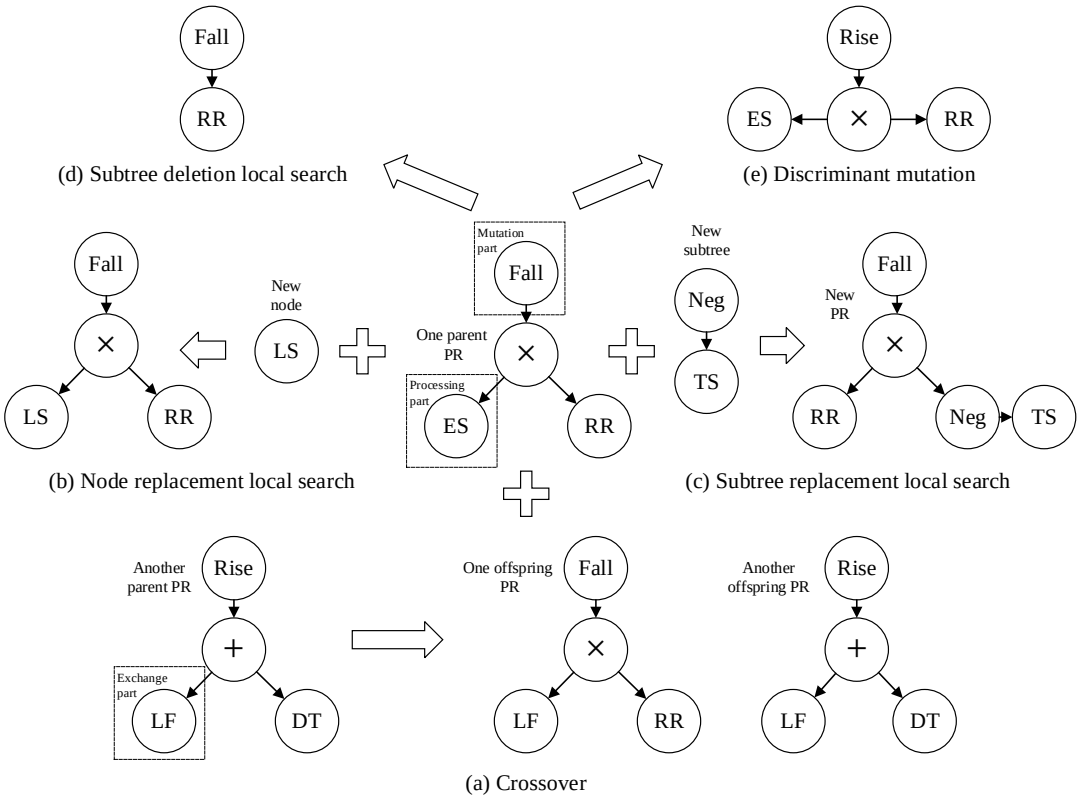


Fig.5 The search operation in generation state

5.2 Normalization evolution in the selection stage

Based on the execution process similar to Algorithm 2, the selection stage aiming at obtaining the optimal PR combination only needs to change the corresponding evolution operator. Firstly, the population initialization in the selection stage is relatively simple, in which only the matching PR needs to be randomly selected for each state to obtain the PR combination shown in Fig.4. Secondly, only the optimal PR combination needs to be obtained in the selection, resulting in the transformation of non-dominated evolution into normalization evolution, that is, the evaluation and fitness function of the PR combination depend on the normalized objective $f_{\bar{O}}$, as shown in Eq.(11). When facing the different demands of decision-maker (focusing on project

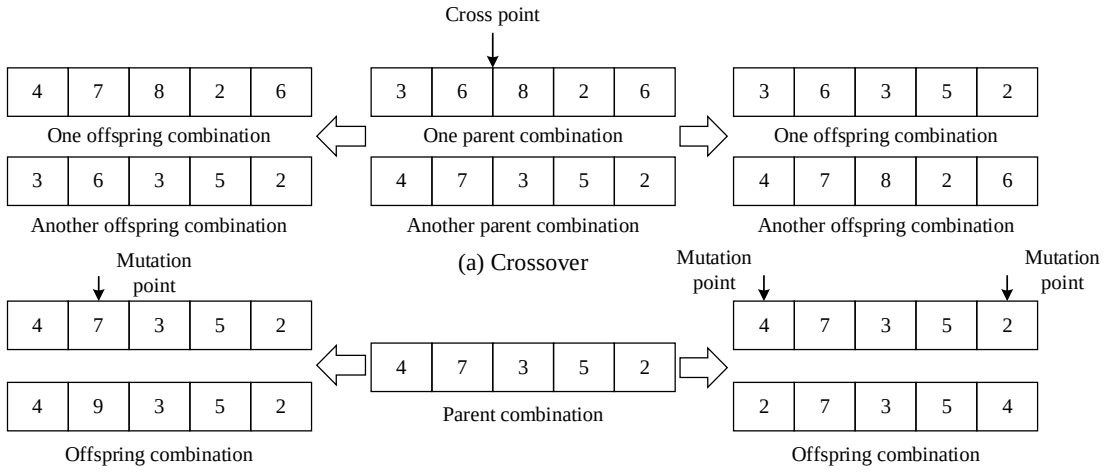
manager or portfolio manager), the selection stage only needs to adjust the corresponding weight value.

$$f_Q = w_1 \times f_{Q1} + w_2 \times f_{Q2} \tag{11}$$

Where w_1 and w_2 represent the weight values of two objectives, respectively, constrained with that their sum is 1 must be satisfied.

In addition, the integer gene structure shown in Fig.4 makes the search operations in Fig.5 inapplicable in selection stage, resulting in crossover, mutation and local search need to be redesigned. In the selection stage, a gene only represents the selected PR index in a certain state, so there is no constrained relationship between genes, resulting in more search methods can be established. From the perspective of genetic search, the crossover and mutation under five states are shown in Fig.6, and the related description are shown as follows:

- *Crossover*: The crossover in selection stage adopts the single-point crossover which is commonly used in integer coding. The difference is that to meet the crossover condition, the generated random number is less than the crossover rate p_c, this study uses a 0.5 probability to determine whether the front or the back part is crossed shown in Fig.6(a), so as to form more different combinations.
- *Mutation*: Similar to crossover, mutation also uses 0.5 probability to control the two mutation modes under the mutation condition shown in Fig.6(b): 1) Randomly select and exchange two points in the original combination gene; 2) Randomly select a point and replace the original PR index with a randomly generated different one.



(b) Mutation

Fig.6 The crossover and mutation in selection stage

From the perspective of local search, there are also four local searches with different structures, whose search methods are different from crossover and mutation to further produce more combinations, as shown in Fig.7.

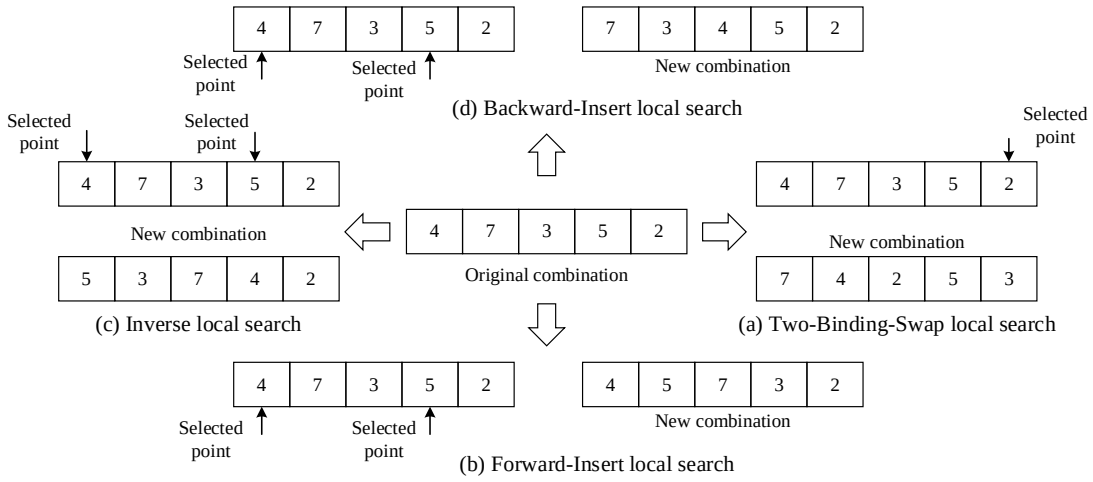


Fig.7 The local searches in selection stage

- *Two-Binding-Swap local search*: This local search will first randomly select a point b, and then swap PR_b with PR_{b+3} and PR_{b+1} with PR_{b+2}. If the selected point causes the index to exceed the gene length, the index will cycle from the beginning. For example, if the selected position is 5 shown in Fig.7(a), it will swap with the third gene through a loop.
- *Forward-Insert/Backward-Insert local search*: In these two local searches, two points b and c ($c>b$) are randomly selected first, and then PR_c is inserted after PR_b in Forward-Insert local search, while in Backward-Insert local search, PR_b is inserted before PR_c.
- *Inverse local search*: Similarly, this local search will first produce two random points, and the difference is that then it will inverse the sub-sequence between the two points, as shown in Fig.7(c).

6 Numerical experiments

In this part, firstly, the experiment preparation is described in Section 6.1, including experimental environment, parameter design, benchmark, traditional PRs participating in comparison and an example of PR combination after HH-TGP evolution. Secondly, HH-TGP is compared with 16 traditional PRs and two state-of-the-art hybrid PRs in Section 6.2 to fully verify its effectiveness. The existing meta-heuristics are not selected and compared due to the following reasons: 1) to our best knowledge, there is no meta-heuristics applied to SRCMPSP-NPI, and the existing meta-heuristics cannot be directly applied to this problem due to the new project insertion constraint with unknown insertion time; 2) the original intention of HH-TGP is to get better results in heuristic computation time, but the use of meta-heuristic scheduling is contrary to it; 3) the existing meta-heuristics have been showed to be inferior to the optimal PR for solving stochastic project scheduling [23,41]. Thirdly, HH-TGP is a hyper-heuristic based two-stage genetic framework. The necessity of the selection stage and generation stage needs to be further verified in Section 6.3, which is also regarded as a comparison between HH-TGP and the state-of-the-art GP for (S)RCPSP. Fourthly, as shown in Eq.(11), the PR combination performance evaluation in the selection stage evolution depends on two weights w_1 and w_2, and the robustness is not considered in the evolution. The influence of w_1 and w_2 on the comprehensive performance of PR combination with robustness considered in the test is further explored in Section 6.4. Finally, in

Section 6.5, taking SPEA2 as an example for comparison [60], the impact of non-dominated evaluation methods in the generation stage on the HH-TGP performance is explored.

6.1 Experimental setup

All experiments are performed on an Intel Core i5-4200 quadcore processor computer with 2.50 GHz clock speed and 8 gigabyte RAM and program code is written in Java on the MyEclipse 2017 compiler. Based on existing research [20,50-52,55], the parameters of HH-TGP are shown in Table 3, of which two aspects need to be emphasized. Firstly, as shown in Fig.5, since the mutation does not change the priority expression of PR and is the only operator to search the discriminant in the generation stage, the p_m of HH-TGP is relatively large based on [55]. Secondly, it can be seen from Eq.(1) and Eq.(2) that f_{Q1} is usually less than f_{Q2} and has a larger fluctuation range. At the same time, due to the limited resources, the two goals are often in conflict, that is, the smaller f_{Q1} usually has the larger f_{Q2}, so a higher weight value is assigned to f_{Q2} for forming more balanced combinations. Therefore, the initial values of w_1 and w_2 are set to 0.3 and 0.7, and the influence of weight distribution on combination performance will be further analysed and discussed in Section 5.4.

Table 3 The parameters of HH-TGP

Variable	Meaning	Value	Variable	Meaning	Value
Pop_{size}	Population size	200	Max_{gen}	Maximum iteration number	25
p_c	Crossover rate	0.9	p_m	Mutation rate	0.2
w_1/ w_2	Weight of f_{Q1}/ f_{Q2}	0.3/0.7	s_{num}	Number of States	5

In addition, the evaluation information in the whole experiment needs to be introduced, that is, the evaluation method and the benchmark. The model of SRCMPSP-NPI comes from [20], so the evaluation criteria and benchmark used for comparison are the same. The benchmark is 1000 instances based on PSPLIB [59] composed of 200 different portfolios with five conditions, in which each condition refers to the insertion of new projects with different structures and different start times. The start time and structure of a new project are random before the decision, so the training set in this study is the 200 basic portfolios, and the test set is the complete 1000 instances. In order to meet the uncertainty duration, five distributions [20,23], including two low variance distributions, two medium variance distributions and one high variance distribution, are adopted to fully verify and compare the performance, as shown in Table 4.

Table 4 Five distributions of activity duration [20,23]

Distribution type	Code	Range	Variance
Uniform distribution	U1	$U(d^*_{i,j}-\sqrt{d^*_{i,j}},\ d^*_{i,j}+\sqrt{d^*_{i,j}})$	$d^*_{i,j}/3$
	U2	$U(0,\ 2d^*_{i,j})$	$(d^*_{i,j})^2/3$
Beta distribution	B1	$B(d^*_{i,j}/2,\ 2d^*_{i,j},\ d^*_{i,j}/2-1/3,\ d^*_{i,j}-2/3)$	$d^*_{i,j}/3$
	B2	$B(d^*_{i,j}/2,\ 2d^*_{i,j},\ 1/6,\ 1/3)$	$(d^*_{i,j})^2/3$
Exponential distribution	E	$E(d^*_{i,j})$	$(d^*_{i,j})^2$

Further, the objective functions in SRCMPSP-NPI involve the comparison of different

perspectives (schedule quality and robustness) and levels (project and portfolio). There is a gap between different objective values. For example, f_{Q1} (f_{Q2}) may reach hundreds or even thousands of times of f_{R1} (f_{R2}). At the same time, from Eq.(1) to Eq.(4), it can be seen that the fluctuation of objectives from the perspective of project and portfolio is inconsistent. If the average values of the four objectives in different instances are directly dependent for evaluating, it is impossible to judge the pros and cons between the strategies with non-dominant relationship, where this probability is high due to the comparison under four objectives. In addition, it is also unreasonable to calculate the mean value between different objectives due to inconsistent fluctuations. For example, an increase of 10 for f_{Q2} may be equal to an increase of 20 under f_{Q1}. Therefore, in order to measure the strategy performance, the superior ranking [20,42] is adopted by performing the following three steps: 1) the $rule_{num}$ strategies participating in the ranking are used to schedule 1000 instances in the test set, and calculate the four objectives under each instance according to Eq.(1) to Eq.(4); 2) In each instance, $rule_{num}$ strategies are sorted and transformed into ranking values from 1 to $rule_{num}$ according to the sequence under each objective; 3) the average ranking values of each strategy from the perspective of scheduling quality, robustness and comprehensive performance are calculated by relying on Eq.(12) to Eq.(14). It is worth mentioning that if the two strategies have the same value under a certain objective, they are given the same ranking. For example, three strategies participate in the ranking. If the f_{Q1} of the first two strategies is the same, their ranking under f_{Q1} is 1, while the third strategy is 3. Meanwhile, in order to calculate the robustness objective, for each instance, SAD_i of Eq.(3) and Eq.(4) is obtained by means of a simulation with 10 replications in the whole experiment.

$$rk_{rule,\,Q} = \Big(\sum_{\boldsymbol{P}\in\boldsymbol{P}_{total}} (rk_{rule,\,Q_1,\,\boldsymbol{P}} + rk_{rule,\,Q_2,\,\boldsymbol{P}})/2\Big)/|\,\boldsymbol{P}_{total}\,| \qquad rule = \{1,\ 2,\ ...,\ rule_{num}\} \tag{12}$$

$$rk_{rule,\,R} = \Big(\sum_{\boldsymbol{P}\in\boldsymbol{P}_{total}} (rk_{rule,\,R_1,\,\boldsymbol{P}} + rk_{rule,\,R_2,\,\boldsymbol{P}})/2\Big)/|\,\boldsymbol{P}_{total}\,| \qquad rule = \{1,\ 2,\ ...,\ rule_{num}\} \tag{13}$$

$$rk_{rule,\,C} = (rk_{rule,\,Q} + rk_{rule,\,R})/2 \qquad rule = \{1,\ 2,\ ...,\ rule_{num}\} \tag{14}$$

Where $rule$ is the $rule$-th strategy, and $rk_{rule,*}$ is the ranking of the $rule$-th strategy under condition *. For example, $rk_{rule,Q1,\boldsymbol{P}}$ represents the f_{Q1} ranking by evaluating portfolio $\boldsymbol{P}$.

Finally, as shown in Fig.2, the selectable PR set in the selection stage includes some excellent traditional PRs in addition to the non-dominated PR set obtained in the generation stage. Based on the traditional PRs as shown in Table 5 and the analysis in [20], seven PRs are selected, including MINSLK, SASP, FCFS, MINLFT, MAXSP, WACRU and MS. They are numbered from 0 to 6 in this order, while PRs in the non-dominated set are numbered backwards from 7. From our experimental results, in most cases, the PRs in the evolved combination all come from the non-dominated PR set obtained in the generation stage, and an example of combination containing traditional PR is shown in the Fig.8, where 5 represents WACRU.

Table 5 The traditional PRs

PR name	Abbreviation	PR name	Abbreviation
Minimum slack	MINSLK	Shortest operation duration first	SOF
Maximum slack	MAXSLK	Maximum operation duration first	MOF
Shortest activity from shortest project	SASP	Minimum late finish time	MINLFT
Longest activity from longest project	LALP	Maximum schedule pressure	MAXSP
Minimum total work content	MINTWK	Minimum worst case slack	MINWCS

Minimum total work content	MAXTWK	Criticality & resource utilization	WACRU
MAXTWK & earliest late start time	TWKLST	Maximum total successors	MS
First come first serve	FCFS	Maximum critical successors	MCS

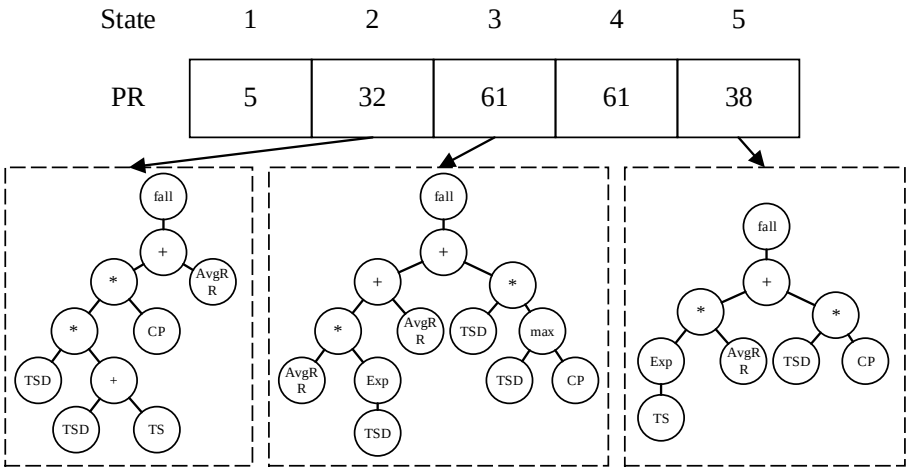


Fig.8 An example of PR combination evolved by HH-TGP

6.2 Comparison with existing PRs

SRCMPSP-NPI with two stochastic factors is a high dynamic problem, which leads to high requirement for response speed, and, to our best knowledge, there are no meta-heuristic methods to solve SRCMPSP-NPI and meet timeliness requirement. Therefore, the existing heuristics (PRs) are selected to compare with HH-TGP, which includes 20 different traditional PRs [20,42] and six hybrid PRs [20]. However, the RB-policy in this paper leads to the fact that the earliest start time of each activity participating in the scheduling is identical, and some PRs in the traditional PRs or hybrid PRs are equivalent, such as EDDF and MINSLK, so only one of them is selected in the comparison. The filtered PRs is shown in Table 5, and the comparative performance ranking of 16 traditional PRs and HH-TGP is shown in Table 6. Because this part data is relatively large, only the average value after 20 experiments is shown in Table 6, and the average ranking comparison between optimal traditional PRs and HH-TGP under each distribution is shown in Fig.9.

Table 6 The average ranking comparison between traditional PRs and HH-TGP

Strategy	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
MINSLK	4.95	7.80	6.37	8.73	6.84	7.93	6.44	9.46	7.21	10.09	7.52
MAXSLK	15.42	9.94	12.68	8.69	12.06	9.87	12.65	8.08	11.75	7.15	11.29
SASP	7.97	8.93	8.45	9.63	8.80	9.10	8.53	10.25	9.11	10.53	9.25
LALP	12.50	7.00	9.75	7.75	10.13	7.09	9.79	8.52	10.51	8.93	10.72
MINTWK	10.58	5.63	8.10	5.30	7.94	5.69	8.14	5.53	8.05	5.13	7.86
MAXTWK	10.72	13.91	12.31	13.54	12.13	13.56	12.14	12.70	11.71	12.51	11.61
TWKLST	10.67	13.96	12.31	13.57	12.12	13.57	12.12	12.67	11.67	12.43	11.55
FCFS	7.30	7.64	7.47	8.52	7.91	8.02	7.66	9.00	8.15	10.05	8.68
SOF	12.03	11.27	11.65	10.28	11.16	11.22	11.63	9.22	10.62	8.28	10.15

MOF	13.78	9.91	11.85	9.18	11.48	9.24	11.51	8.99	11.39	9.08	11.43
MINLFT	6.03	11.35	8.69	10.94	8.49	11.54	8.79	10.21	8.12	10.28	8.15
MAXSP	9.15	5.26	7.20	6.00	7.57	5.47	7.31	6.79	7.97	6.65	7.90
MINWCS	4.94	7.83	6.38	8.80	6.87	8.00	6.47	9.57	7.25	10.13	7.53
WACRU	7.12	8.74	7.93	7.45	7.29	8.58	7.85	6.51	6.81	5.73	6.42
MS	8.72	**4.57**	6.65	**4.98**	6.85	**4.70**	6.71	**5.73**	7.23	**5.42**	7.07
MCS	6.29	7.92	7.10	8.48	7.38	8.02	7.15	9.04	7.66	9.56	7.93
HH-TGP	**2.91**	8.71	**5.81**	9.30	**6.10**	8.80	**5.86**	9.05	**5.98**	9.72	**6.32**

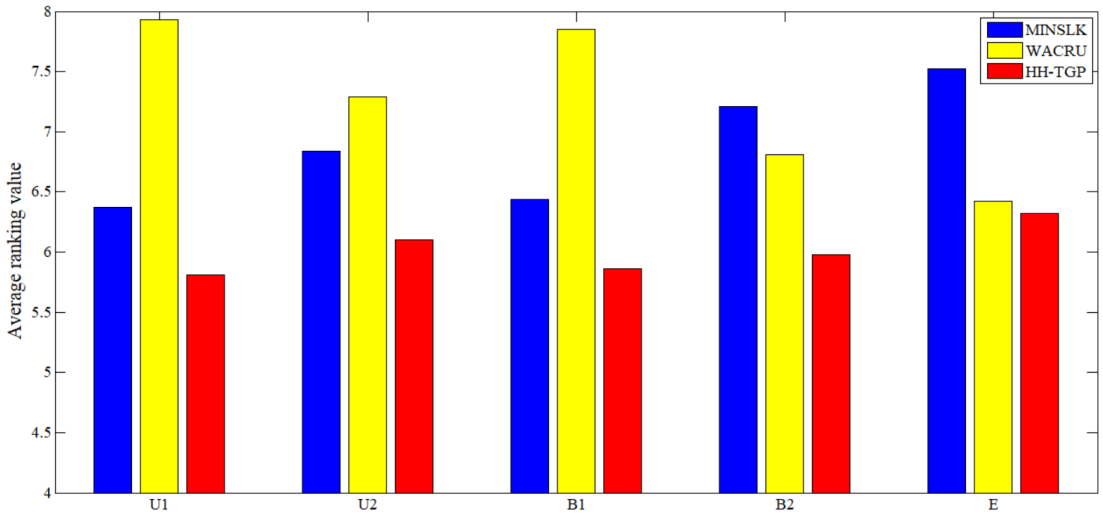


Fig.9 The average $rk_{rule,C}$ comparison between HH-TGP and optimal traditional PRs

The comparison results between traditional PRs shown in Table 6 are similar to [20,42], that is, different traditional PRs tend to effect different objectives, such as MINSLK and MINWCS can get better average $rk_{rule,Q}$, while MS and WACRU perform well in average $rk_{rule,R}$, and the PRs with good robustness often has poor scheduling quality. At the same time, it can be shown from Fig.9 that with the increase of distribution variance, the optimal traditional PR changes from MINSLK to WACRU. This result is also consistent with [20] and shows that PR has problem dependence.

In addition, it can be seen from Table 6 that HH-TGP is significantly better than all traditional PRs on average $rk_{rule,Q}$, which shows that HH-TGP can obtain better expected makespan and mainly leads to that the average $rk_{rule,C}$ of HH-TGP is also better than the optimal traditional PRs under all distributions, as shown in Fig.9. From the perspective of $rk_{rule,R}$, HH-TGP ranks only medium in the comparison of all traditional PRs, because 1) according to Eq.(3) and Eq.(4), the denominator (expected makespan) of HH-TGP is smaller, resulting in greater fluctuation; 2) the complex structure and multiple attributes deteriorate the robustness in scheduling. However, compared with the traditional PRs that also focus on $rk_{rule,Q}$, HH-TGP has better adaptability to randomness changes. For example, the trend curves of the average $rk_{rule,R}$ in MINSLK (best in [20,42]) and HH-TGP under different variance distributions are shown in Fig.10. It can be seen that the average $rk_{rule,R}$ of MINSLK is better than HH-TGP under medium and low variance distributions, and vice versa at high variance. The intersection point between the two curves represents the equal robustness point at the specific variance distribution. In application, when the variance distribution is under this point, MINSLK is more robust scheme while over this

point, HH-TGP is a more robust scheme. Therefore, in general, it can be concluded that HH-TGP *has excellent scheduling quality, effective comprehensive performance and certain randomness adaptability, which shows its effectiveness.*

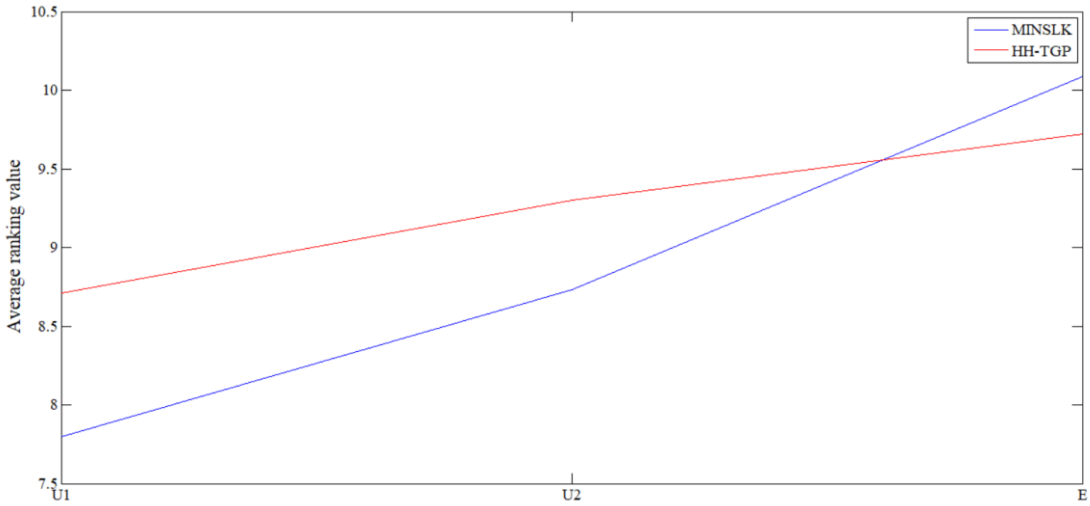


Fig.10 The average $rk_{rule,R}$ trend curves of MINSLK and HH-TGP

Furthermore, based on the performance analysis of traditional PRs, a heuristic combination method is proposed in [20], in which the former stage depends on the earliest start and the latter stage depends on the latest start, so as to produce six hybrid PRs. For the same reason, the two filtered hybrid PRs are compared with HH-TGP, in which FCFS/WCS-SLK represents the mixed result of FCFS/MINWCS and MINSLK. The average ranking comparison are shown in Table 7 after 20 experiments, and the average $rk_{rule,C}$ comparison between hybrid PRs and HH-TGP under each distribution is shown in Fig.11.

Table 7 The average ranking comparison between hybrid PRs and HH-TGP

Strategy	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
FCFS-SLK	2.27	**1.86**	2.07	**1.89**	2.08	**1.88**	2.08	**1.92**	2.10	**1.94**	2.11
WCS-SLK	2.35	1.96	2.15	1.98	2.16	1.95	2.15	2.05	2.20	2.04	2.20
HH-TGP	**1.33**	2.11	**1.72**	2.08	**1.71**	2.10	**1.72**	1.99	**1.66**	1.98	**1.66**

It can be seen from the Table 7 and Fig.11 that the comparison results of HH-TGP and hybrid PRs are similar to those of traditional PRs, that is, it has very excellent scheduling quality and better comprehensive performance, resulting in that the average $rk_{rule,C}$ of HH-TGP is still better than the optimal hybrid PR under all distributions in Fig.11. Further, it can also be seen from Table 7 that HH-TGP has good adaptability, because the gap of average $rk_{rule,R}$ between HH-TGP and hybrid PRs decreases with the increase of distribution variance. These results show that HH-TGP is still effective, and also confirm that this framework is more suitable for solving SRCMPSP-NPI, especially under high variance distribution and comprehensive performance requirements.

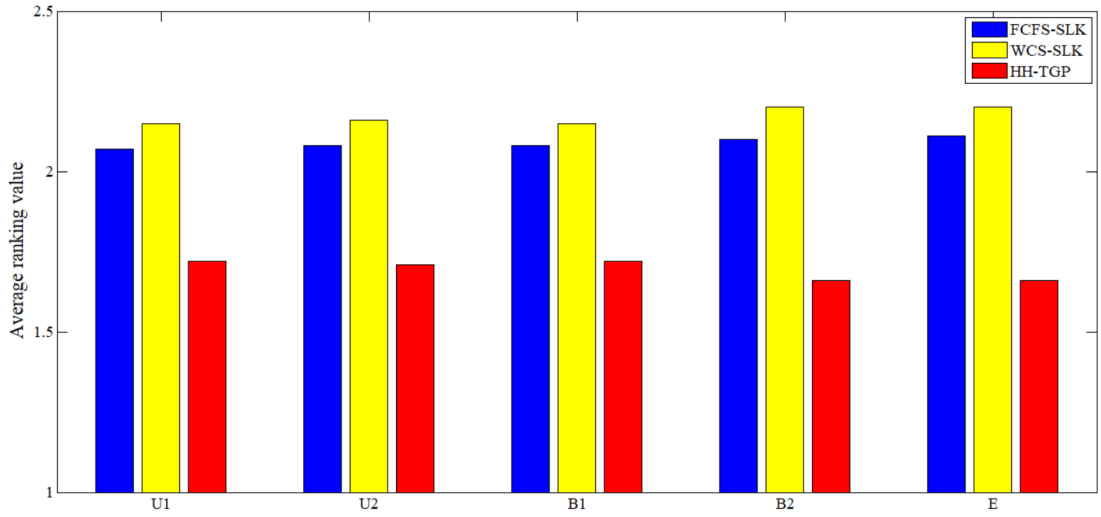


Fig.11 The average $rk_{rule,C}$ comparison between HH-TGP and hybrid PRs

6.3 Validation of the two-stage framework

Compared with the existing PRs, HH-TGP has been shown to be an effective method for solving SRCMPSP-NPI. However, it can be seen from Fig.2 that the training process of HH-TGP has two stages: the first stage generates a group of evolved PRs, and the second stage constructs a combination. Whether the two-stage framework is helpful improving the effectiveness of the HH-TGP needs further verification. Therefore, a two-part experiment, using the generation and the selection stage alone to evolve, is set up as follows.

Firstly, HH-TGP is compared with the GP containing only generation stage (HH-GGP) to verify the effectiveness of the selection stage, that is, the comparison between constructing PR combination and evolving a single optimal PR. HH-GGP is also the evolution mode of existing GP, and in order to ensure the fairness, except that Max_{gen} is expanded to 50 for ensuring the same search (HH-TGP has two stages with Max_{gen} equal to 25), its other parameters and search operators are all consistent with the generation stage in HH-TGP. At the same time, based on the experimental results generated in Section 6.2 and because the ranking calculation depends on the average value of 1000 instances, the fluctuation under each evolution is very small, resulting in only 10 evolutions in the comparison of different GPs in this section. The average ranking comparison of HH-GGP and HH-TGP under 10 evolutions is shown in Table 8 and Fig.12, and their detailed evolution results are shown in the appendix.

Table 8 The average ranking comparison between HH-GGP and HH-TGP

Strategy	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
HH-GGP	1.52	1.55	1.53	1.57	1.54	1.55	1.53	1.54	1.53	1.58	1.55
HH-TGP	**1.47**	**1.43**	**1.45**	**1.42**	**1.45**	**1.43**	**1.45**	**1.45**	**1.46**	**1.41**	**1.44**

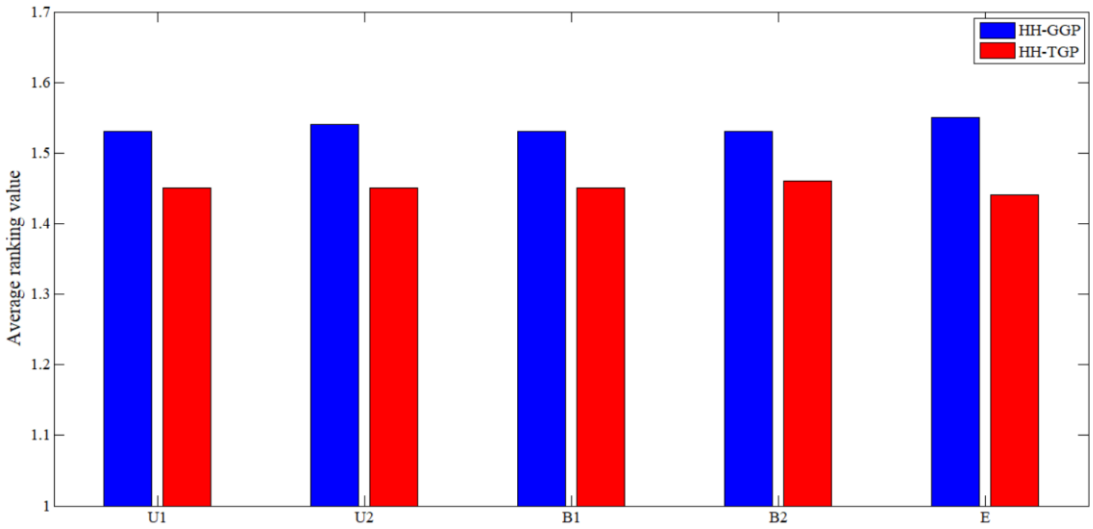


Fig.12 The average $rk_{rule,C}$ comparison between HH-TGP and OPR

As can be seen from Table 8, HH-TGP is better than HH-GGP on average $rk_{rule,Q}$. More importantly, no matter which distribution, the average $rk_{rule,R}$ of HH-TGP is also better than HH-GGP, and the gap increases with distribution variance increasing. Therefore, as shown in Fig.12, the comprehensive performance of HH-TGP is also better than HH-GGP under all distributions, especially under high variance distribution. To sum up, on the basis of further verifying the HH-TGP effectiveness, a conclusion is obtained that *the combined scheduling of multiple PRs can not only further improve the schedule quality compared with a single PR, it can also improve the robustness to deal with stochastic problem.*

Secondly, the GP with only using selection stage (HH-SGP) is compared with HH-TGP to explore the function of generation stage, that is, the comparison between PR combinations including or excluding evolved PRs. Similar to the comparative experiments with HH-GGP, the comparison of the average values between HH-TGP and HH-SGP under 10 evolutions based on the same parameters and search operators is shown in Table 9 and Fig.13, and the detailed data are still shown in the appendix.

Table 9 The average ranking comparison between HH-SGP and HH-TGP

Strategy	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
HH-SGP	1.81	**1.49**	1.65	**1.47**	1.64	**1.48**	1.65	**1.47**	1.65	**1.48**	1.65
HH-TGP	**1.18**	1.50	**1.34**	1.52	**1.36**	1.50	**1.34**	1.51	**1.35**	1.51	**1.35**

It can be clearly seen from Table 9 that the average $rk_{rule,Q}$ of HH-TGP is almost close to 1, which indicates that the schedule quality of HH-TGP is much better than that of the strategy only using the selection stage under the same generation combination mode. At the same time, in Table 9, the average $rk_{rule,R}$ of HH-TGP is inferior to HH-SGP under all distributions, so it can be concluded that the robustness of PR combination is deteriorated when adding PRs with multiple attributes that need to be considered in calculating priority. However, compared with the improvement of scheduling quality, the gap between robustness is very small, resulting in the better comprehensive performance of HH-TGP under all distributions, as shown in Fig.13. In this

part of the experiment, the effectiveness of HH-TGP is verified again, and the main function of generation stage is explained, that is, *for generating more evolved PRs to help the whole framework improve the schedule quality.*

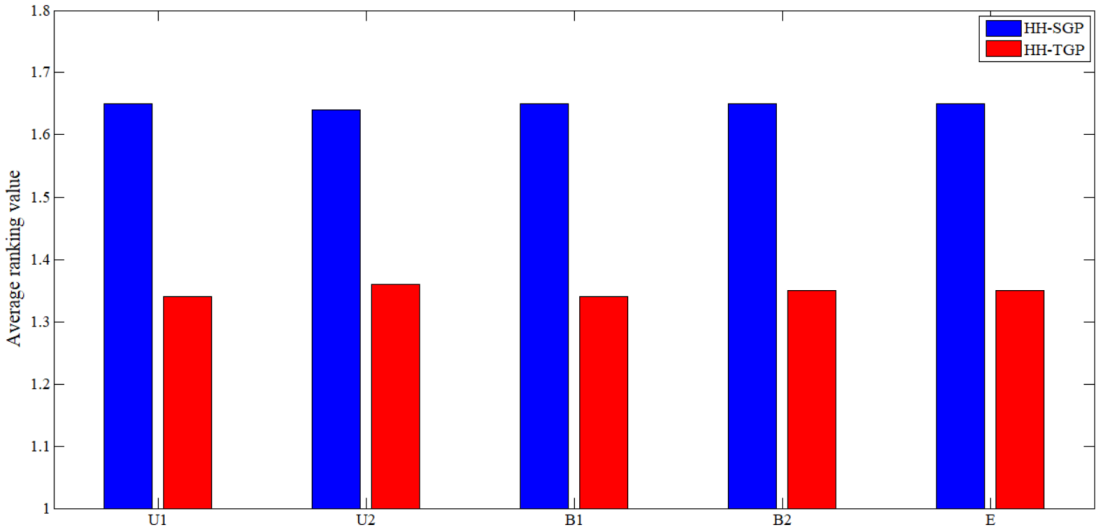


Fig.13 The average $rk_{rule,C}$ comparison between HH-TGP and OC

6.4 The effect of w_1 and w_2 on HH-TGP performance

Eq.(11) shows that in the selection stage, the weights w_1 and w_2 are assigned to the two objectives f_{Q1} and f_{Q2} for achieving the normalized evolution and obtaining the optimal PR combination. Based on the explanation in Section 6.1, in the performance verification in Section 6.2 and Section 6.3, the evolution of HH-GP gives f_{Q1} and f_{Q2} weights of 0.3 and 0.7, respectively. However, how the change of w_1 and w_2 will affect the HH-TGP performance needs to be further explored, so w_1 increases from 0.1 to 0.5 in the range of 0.1 without violating the principle that w_2 is greater than w_1. Therefore, in this part, based on the same non-dominated set evolved by generation stage in each experiment, different weights are allocated to perform selection stage evolution and obtain different optimal PR combinations, and this process is also repeated 10 times. Further, all PR combinations are compared with traditional PRs to calculate the performance ranking, and the average ranking comparison of different weights under 10 evolutions is shown in Table 10 (the detailed results are shown in the appendix). Meanwhile, the average curves of the three rankings with weight changing are shown in Fig.14 to Fig.16, in which robustness and comprehensive ranking have three different variance distributions (U1, U2 and E).

Table 10 The average ranking of HH-TGP under different weights

w_1/w_2	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
0.1/0.9	4.08	**8.33**	6.20	**8.97**	6.52	**8.41**	6.24	9.26	6.67	**9.65**	6.87
0.2/0.8	3.17	8.43	**5.79**	9.13	6.15	8.58	5.88	9.41	6.29	9.85	6.51
0.3/0.7	**2.93**	8.68	5.80	9.26	**6.10**	8.76	**5.85**	**9.01**	**5.97**	9.67	**6.30**
0.4/0.6	3.04	9.07	6.06	9.79	6.41	9.14	6.09	9.34	6.19	10.25	6.64
0.5/0.5	3.26	9.52	6.39	10.35	6.80	9.56	6.41	9.71	6.48	10.79	7.03

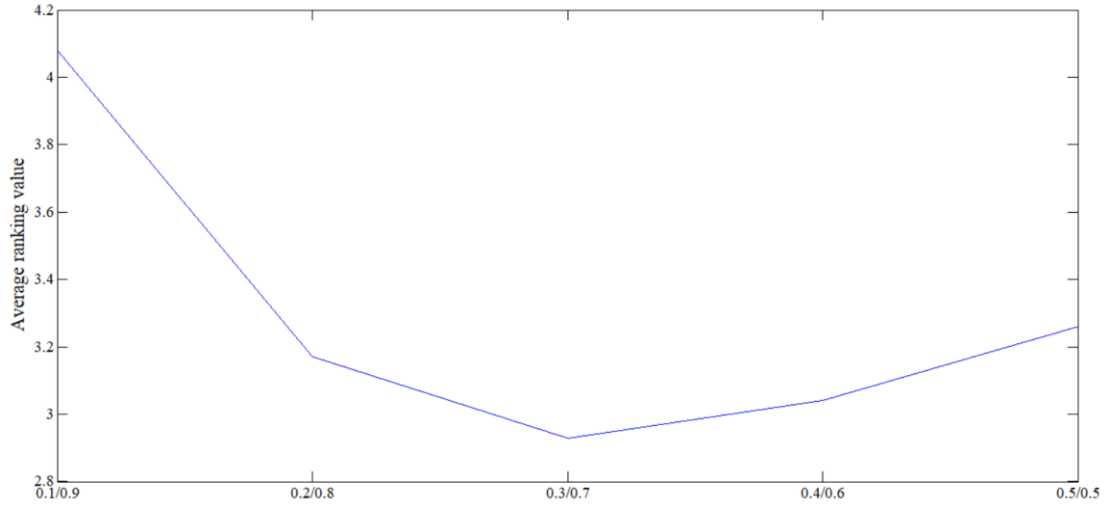


Fig.14 The average $rk_{rule,Q}$ curve with different weights

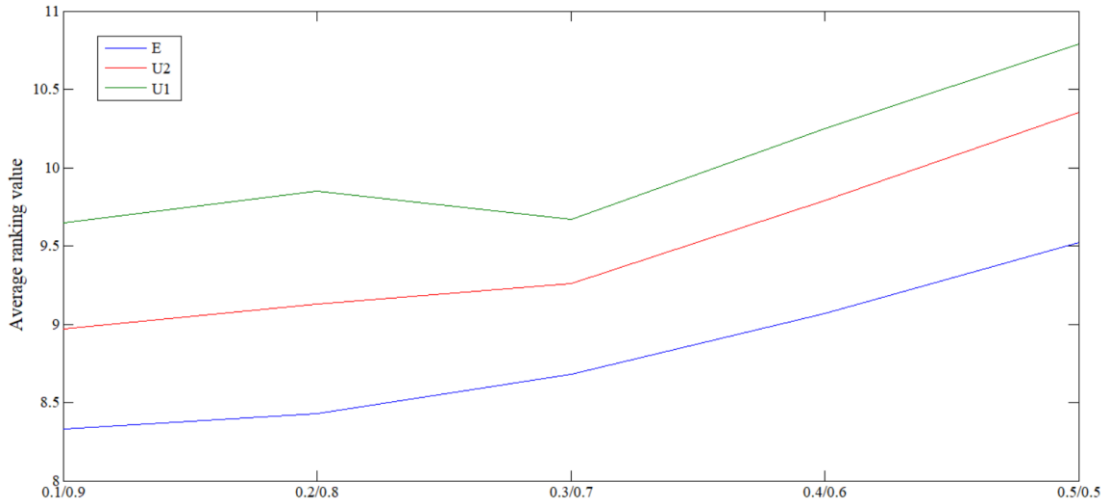


Fig.15 The average $rk_{rule,R}$ curve with different weights under three distributions

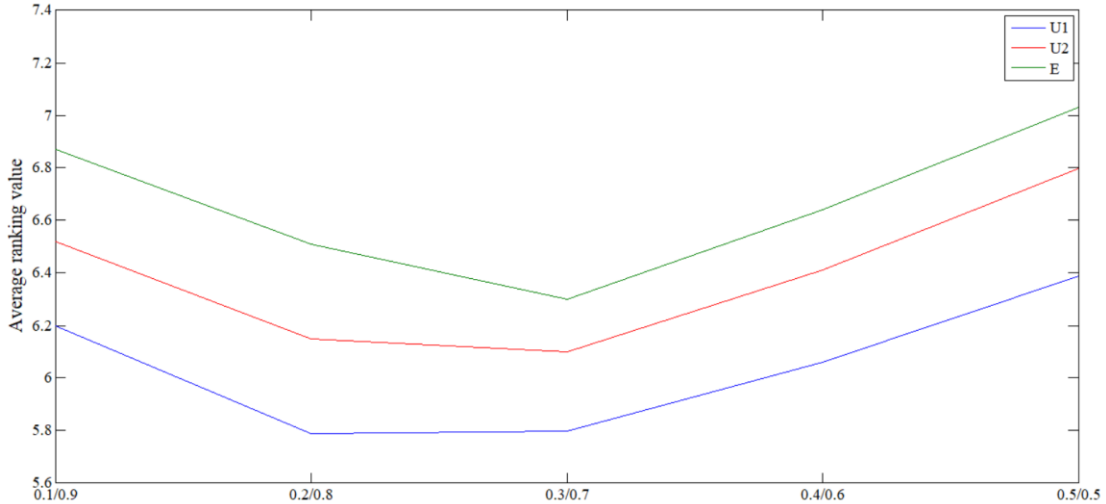


Fig.16 The average $rk_{rule,C}$ curve with different weights under three distributions

Based on the data in Table 10 and Fig.14, it can be concluded that the change of $rk_{rule,Q}$ is

similar to a parabola, and when the weight is close to f_{Q2}, the deterioration of $rk_{rule,Q}$ is more serious, that is, the sensitivity is higher. Therefore, in order to get a better $rk_{rule,Q}$, a relatively middle weight should be assigned, such as 0.3/0.7 and 0.4/0.6. From the perspective of robustness, Fig.15 shows that under different variance distributions, $rk_{rule,R}$ deteriorates with the increase of w_1, so f_{Q2} should be given a large weight for robustness requirements. Combined with these two aspects, the comprehensive performance is also shown as parabola, and has the characteristics of gentle trend under low variance. To sum up, *this part shows that 0.3/0.7 is a good weight allocation, and if the decision makers prefer robustness, 0.2/0.8 is also a very recommended weight allocation.*

6.5 The effect of evaluation method in generation stage on HH-TGP performance

As shown in Fig.2, the PR scheduling combination evolved in the selection stage depends on the non-dominated PR set obtained in the generation stage, resulting in the non-dominated evaluation method used in the generation stage affecting the HH-TGP performance under the same search operators. Therefore, similar to the experiment in Section 6.4, this section selects SPEA2 as the evaluation method in HH-TGP to explore the impact of different evaluation methods on HH-TGP. Compared with traditional PRs, the average performance ranking of 10 evolutions is shown in Table 11 (the detailed results are shown in the appendix), where HH-TGP-S represents HH-TGP with SPEA2, HH-TGP-N represents HH-TGP with NSGA-II, and the average $rk_{rule,C}$ comparison is shown in Fig.17.

Table 11 The average ranking of HH-TGP with different evaluation algorithms

Strategy	$rk_{rule,Q}$	U1		U2		B1		B2		E	
		$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$	$rk_{rule,R}$	$rk_{rule,C}$
HH-TGP-S	4.24	**8.17**	6.20	**8.68**	6.46	**8.34**	6.29	**9.03**	6.64	**9.27**	6.75
HH-TGP-N	**2.91**	8.71	**5.81**	9.30	**6.10**	8.80	**5.86**	9.05	**5.98**	9.72	**6.32**

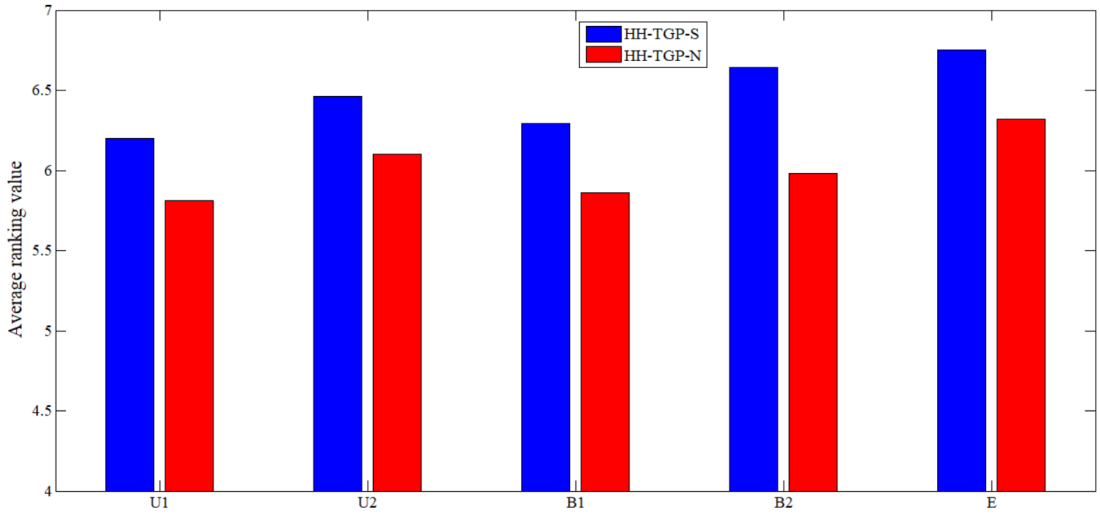


Fig.17 The average $rk_{rule,C}$ with different evaluation algorithms

The following two conclusions can be drawn from Table 11 and Fig.17. Firstly, the $rk_{rule,C}$ of HH-TGP-N is better than that of HH-TGP-S under five distributions, which shows that *when HH-TGP applied into SRCMPSP-NPI, using NSGA-II as the evaluation method is more effective than SPEA2*. Secondly, when NSGA-II is used as the evaluation method, the $rk_{rule,Q}$ of HH-TGP

increases significantly, but its $rk_{rule,R}$ decreases. Therefore, combined with the relevant conclusions in Table 9, this shows that *when a more effective non-dominated evaluation method is adopted* (i.e., a more effective non-dominated PR set is obtained), *the schedule quality and comprehensive performance of PR combination are significantly improved, but the robustness is deteriorated.*

7 Conclusion

In this study, a novel HH-TGP framework is proposed to obtain more effective scheduling strategies by decomposing the evolution into two stages of generation and selection for solving SRCMPSP-NPI, which is a stochastic multi-project scheduling problem considering two randomness of stochastic activity duration and new project insertion. With this framework, the current research is expanded from two aspects. First, the idea of GP is extended to the multi-project scheduling in stochastic environment for producing better PRs, so that it retains the advantages of PR in solving the scheduling problem, such as simplicity, rapidity, and stability, and resolves its defects of no optimization ability and problem dependence. Because of the universality of multi-project scheduling in practice and the superiority of PR scheduling in stochastic environment, this extension is very meaningful. Secondly, HH-TGP constructs a multi-state PR combination scheduling mode, that is, the whole decision process is divided into multiple states and the most suitable PRs are matched in each state by genetic evolution. This combination mode can control the dynamic change of priority to achieve better performance compared with the fixed use of PRs.

In addition to this novel two-stage framework, more specifically, there are also innovations in optimization technology to realize the evolution of generation and selection. First, in the generation stage, the attribute set in the existing literature is modified to adapt to the PR evolution under multi-project scheduling, and NSGA-II is combined to obtain a non-dominated PR, which provides an idea for multi-objective PR evolution based on non-dominated relationship. Secondly, in the selection stage, due to the idea of multi-state combination scheduling, the new state partition, coding structure and decoding method are designed to realize the normalized evolution. At the same time, the genetic operators and local search in selection evolution are also improved.

A series of experiments based on five common distributions and 1000 multi-project benchmarks are carried out to verify the effectiveness of HH-TGP. The experimental results show that HH-TGP is superior to both existing heuristics and single-stage traditional GP. Furthermore, the experimental results show the function of two stages, the influence of weights and the impact of the evaluation methods in generation stage on the performance of HH-TGP. This not only verifies the effectiveness of the two-stage architecture, but also provides the insights for decision-makers to choose the weight and the evaluation method. To sum up, we believe that this method has a great potential in SRCMPSP-NPI.

In the future work, the following parts need to be explored to further improve the practical application of HH-TGP. First, we only aim at the time level in the randomness of SRCMPSP-NPI, and the supply of resources should also change in practice, so it is necessary to establish a more complex scheduling model. Secondly, although this study relies on experience to set five states in the selection stage, the influence of some other factors on the HH-TGP performance needs to be further explored, such as the number of states and whether the ratio of each state is equal. Thirdly, this study only takes SPEA2 as a comparative example to analyse the impact of evaluation

methods on HH-TGP performance, so more advanced multi-objective evaluation methods need to be combined and analysed. Finally, because of the adoption of RB-policy, the state partition parameter of HH-TGP only considers the completion ratio. When policy can also be changed dynamically, more state parameters need to be explored to construct a more complete state representation.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (Grant number 2020YFB1712200).

References

[1] A. Pritsker, L. Watters, P. Wolfe, Multiproject scheduling with limited resources: A zero-one programming approach, Manage. Sci.16(1) (1969) 93–107.
[2] R. Pellerin, N. Perrier, F. Berthaut, A survey of hybrid metaheuristics for the resource-constrained project scheduling problem, Eur. J. Oper. Res. 280(2) (2020) 395-416.
[3] J. Blazewicz, J.K. Lenstra, A.H.G.R. Kan, Scheduling subject to resource constraints: Classification and complexity, Discret Appl. Math. 5 (1983) 11–24.
[4] S. Hartmann, D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem, Eur. J. Oper. Res. 207(1) (2010) 1–14.
[5] J.H. Payne, Management of multiple simultaneous projects: A state-of-the-art review, Int. J. Proj. Manag.13(3) (1995) 163–168.
[6] R. Van Eynde, M. Vanhoucke, Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling. J. Sched. 23(3) (2020), 301-325.
[7] F. Villafáñez, D. Poza, A. López-Paredes, J. Pajares, R. del Olmo, A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP), Soft Comput. 23(10) (2019) 3465–3479.
[8] F. Stork, Stochastic resource-constrained project scheduling (Ph.D. thesis), Technical University of Berlin (2001).
[9] V.A. Hauder, A. Beham, S. Raggl, S.N. Parragh, M. Affenzeller, Resource-constrained multi-project scheduling with activity and time flexibility, Comput. Ind. Eng. 150 (2020) 106857.
[10] P. Lamas, E. Demeulemeester, A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations, J. Sched. 19(4) (2016) 409–428.
[11] M. Brčić, M. Katić, N. Hlupić, Planning horizons based proactive rescheduling for stochastic resource-constrained project scheduling problems, Eur. J. Oper. Res. 273(1) (2019) 58-66.
[12] E. Demeulemeester, W. Herroelen, Robust project scheduling, Now Publishers Inc (2011).
[13] R.H. Möhring, F.J Radermacher, G. Weiss, Stochastic scheduling problems I—General strategies, Math. Method Oper. Res. 28(7) (1984) 193–260.
[14] R.H. Möhring, F.J. Radermacher, G. Weiss, Stochastic scheduling problems II-set strategies, Math. Method Oper. Res. 29(3) (1985) 65–104.
[15] J.H. Patterson. Project scheduling: The effects of problem structure on heuristic performance, Nav. Res. Logist. Q. 23(1) (1976) 95-123.

[16] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, R Qu, Hyper-heuristics: A survey of the state of the art, J. Oper. Res. Soc. 64(12) (2013) 1695–1724.
[17] J. Lin, Z. Wang, X. Li, A backtracking search hyper-heuristic for the distributed assembly flow-shop scheduling problem, Swarm Evol. Comput. 36 (2017) 124–135.
[18] J. Park, Y. Mei, S. Nguyen, G. Chen, M. Zhang, An investigation of ensemble combination schemes for genetic programming based hyper-heuristic approaches to dynamic job shop scheduling, Appl. Soft Comput. 63 (2018) 72–86.
[19] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Correlation coefficient based recombinative guidance for genetic programming hyper-heuristics in dynamic flexible job shop scheduling, IEEE Trans. Evol. Comput. 25(3) (2021) 552-566.
[20] H. Chen, G. Ding, J. Zhang, S. Qin, Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival, Comput. Ind. Eng. 137 (2019) 106060.
[21] S. Creemers, Minimizing the expected makespan of a project with stochastic activity durations under resource constraints, J. Sched. 18(3) (2015) 263–273.
[22] Y. Alipouri, M.H Sebt, A. Ardeshir, M.H.F Zarandi, A mixed-integer linear programming model for solving fuzzy stochastic resource constrained project scheduling problem, Oper. Res. (2020) 1-21.
[23] Z. Chen, E. Demeulemeester, S. Bai, Y. Guo, Efficient priority rules for the stochastic resource-constrained project scheduling problem, Eur. J. Oper. Res. 270(3) (2018) 957–967.
[24] R. Kolisch, Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, Eur. J. Oper. Res., 90(2) (1996) 320–333.
[25] R.L. Graham, Bounds on multiprocessing timing anomalies, SIAM J. Appl. Math.17(2) (1969) 416–429.
[26] F.J. Radermacher, Cost-dependent essential systems of ES-strategies for stochastic scheduling problems, Methods. Oper. Res. 42 (1981) 17–31.
[27] G. Igelmund, F.J Radermacher, Preselective strategies for the optimization of stochastic project networks under resource constraints, Networks. 13(1) (1983) 1–28.
[28] B. Ashtiani, R. Leus, M.B. Aryanezhad, New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing, J. Sched., 14(2) (2011) 157–171.
[29] S. Rostami, S. Creemers, R. Leus, New strategies for stochastic resource-constrained project scheduling, J. Sched. 21(3) (2018) 349–365.
[30] D Golenko-Ginzburg, A Gonik, Stochastic network project scheduling with non-consumable limited resources, Int. J. Prod. Econ. 48 (1997) (1), 29–37.
[31] Y.W. Tsai, D.D Gemmill, Using tabu search to schedule activities of stochastic resource-constrained projects, Eur. J. Oper. Res.111(1) (1998) 129–141.
[32] F. Ballestin, R. Leus, Resource-constrained project scheduling for timely project completion with stochastic activity durations, Prod. Oper. Manag. 18(4) (2009) 459–474.
[33] C. Fang, R. Kolisch, L. Wang, C. Mu, An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem, Flex. Serv. Manuf. J. 27(4) (2015) 585–605.
[34] W. Ma, Y. Che, H. Huang, H. Ke, Resource-constrained project scheduling problem with uncertain durations and renewable resources, Int. J. Mach. Learn. Cybern. 7(4) (2016) 613-621.
[35] U. Satic, P. Jacko, C. Kirkbride, Performance evaluation of scheduling policies for the dynamic

and stochastic resource-constrained multi-project scheduling problem, Int. J. Prod. Res. (2020) 1-13.
[36] K.M. Sallam, R.K. Chakrabortty, M.J. Ryan, A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems, Expert Syst. Appl. 169 (2021) 114479.
[37] T.R. Browning, A.A Yassine, Resource-constrained multi-project scheduling: Priority rule performance revisited. Int. J. Prod. Econ. 126(2) (2010) 212–228.
[38] ˜F. Villafanez, D. Poza, ´A. Lopez-Paredes, J. Pajares, R. del Olmo, A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP), Soft Comput. 23(10) (2019) 3465–3479.
[39] P.I. Adamu, H.I. Okagbue, P.E Oguntunde, A new priority rule for solving project scheduling problems, Wirel. Pers. Commun. 106(2) (2019) 681–699.
[40] X. Wang, Q. Chen, N. Mao, X. Chen, Z. Li, Proactive approach for stochastic RCMPSP based on multi-priority rule combinations, Int. J. Prod. Res. 53(4) (2015) 1098–1110.
[41] F. Rezaei, A.A. Najafi, R. Ramezanian, E. Demeulemeester, Simulation-based priority rules for the stochastic resource-constrained net present value and risk problem, Comput. Ind. Eng. 160 (2021) 107607.
[42] Y. Wang, Z. He, L.P Kerkhove, M. Vanhoucke, On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem, Comput. Ind. Eng.114 (2017) 223-234.
[43] R.K. Chakrabortty, H.F. Rahman, M.J. Ryan, Efficient priority rules for project scheduling under dynamic environments: A heuristic approach, Comput. Ind. Eng. 140 (2020) 106287.
[44] J. Branke, S. Nguyen, C.W. Pickardt, M. Zhang, Automated design of production scheduling heuristics: A review, IEEE Trans. Evol. Comput. 20(1) (2015) 110-124.
[45] K. Anagnostopoulos, G. Koulinas, Resource-constrained critical path scheduling by a GRASP-based hyperheuristic, J. Comput. Civil. Eng. 26(2) (2012) 204–213.
[46] G. Koulinas, K. Anagnostopoulos, Construction resource allocation and leveling using a threshold accepting–based hyperheuristic algorithm, J. Constr. Eng. Manage. 138(7) (2012) 854–863.
[47] G. Koulinas, L. Kotsikas, K. Anagnostopoulos, A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, Inf. Sci. 277 (2014) 680–693.
[48] J. Lin, L. Zhu, K. Gao, A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem, Expert Syst. Appl. 140 (2020) 112915.
[49] L. Zhu, J. Lin, Y.Y. Li, Z.J. Wang, (2021). A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem, Knowledge-Based Syst. 225 (2021) 107099.
[50] S. Chand, Q. Huynh, H. Singh, T. Ray, M. Wagner, On the use of genetic programming to evolve priority rules for resource constrained project scheduling problems, Inf. Sci. 432 (2018) 146–163.
[51] S. Chand, H. Singh, T. Ray, Evolving heuristics for the resource constrained project scheduling problem with dynamic resource disruptions, Swarm Evol. Comput. 44 (2019) 897–912.
[52] S. Chand, H. Singh, T. Ray, Evolving rollout-justification based heuristics for resource constrained project scheduling problems, Swarm Evol. Comput. 50 (2019) 100556.
[53] Y. Alipouri, M.H. Sebt, A. Ardeshir, W.T. Chan, Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach, J. Oper. Res. Soc. 70(3) (2019) 403-419.
[54] M. Kühn, M. Völker, T. Schmidt, (2020). An algorithm for efficient generation of customized

priority rules for production control in project manufacturing with stochastic job processing times. *Algorithms*, 13(12), 337.
[55] H. Chen, G. Ding, S. Qin, J. Zhang, A hyper-heuristic based ensemble genetic programming approach for stochastic resource constrained project scheduling problem, Expert Syst. Appl. 167 (2021) 114174.
[56] S. Luke, L. Panait, A survey and comparison of tree generation algorithms, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann San Francisco, California, USA, 2001, pp. 81-88.
[57] E. Chołodowicz, P. Orłowski, Comparison of SPEA2 and NSGA-II applied to automatic inventory control system using hypervolume indicator, Stud. Inform. Control. 26(1) (2017) 67-74.
[58] S. Luke, L. Panait, A comparison of bloat control methods for genetic programming, Evol. Comput. 14(3) (2006) 309-344.
[59] A. Sprecher, R. Kolisch, PSPLIB—a project scheduling problem library, Eur. J. Oper. Res. 96 (1996) 205–216.
[60] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, TIK-report. 103 (2001).

Appendix

This part shows the detailed evolution process of relevant hyper-heuristics under different parameters or structures in Section 6.3 and Section 6.4, in which *Per* represents the perspective of ranking, *Dis* represents the distribution and Num_{num} represents the *num*-th evolution.

The detailed comparison between HH-TGP and HH-GGP

Per	*Dis*	Strategy	Num_1	Num_2	Num_3	Num_4	Num_5	Num_6	Num_7	Num_8	Num_9	Num_{10}
$rk_{rule,Q}$	/	OPR	1.52	1.51	1.50	1.52	1.50	1.48	1.53	1.56	1.54	1.50
		HH-TGP	1.47	1.49	1.47	1.47	1.49	1.50	1.45	1.43	1.44	1.48
$rk_{rule,R}$	U1	OPR	1.54	1.61	1.56	1.53	1.53	1.55	1.58	1.52	1.55	1.54
		HH-TGP	1.43	1.37	1.42	1.45	1.46	1.43	1.40	1.45	1.43	1.44
	U2	OPR	1.56	1.64	1.58	1.55	1.57	1.55	1.57	1.56	1.54	1.57
		HH-TGP	1.43	1.35	1.41	1.44	1.42	1.44	1.41	1.43	1.45	1.42
	B1	OPR	1.58	1.60	1.57	1.51	1.53	1.54	1.57	1.51	1.55	1.55
		HH-TGP	1.41	1.38	1.42	1.47	1.45	1.44	1.41	1.47	1.43	1.44
	B2	OPR	1.52	1.58	1.55	1.53	1.55	1.52	1.52	1.53	1.51	1.56
		HH-TGP	1.47	1.40	1.45	1.46	1.44	1.47	1.47	1.45	1.48	1.42
	E	OPR	1.55	1.62	1.61	1.58	1.60	1.57	1.57	1.52	1.57	1.60
		HH-TGP	1.43	1.37	1.38	1.42	1.38	1.42	1.42	1.47	1.43	1.39
$rk_{rule,C}$	U1	OPR	1.53	1.56	1.53	1.52	1.51	1.52	1.56	1.54	1.55	1.52
		HH-TGP	1.45	1.43	1.45	1.46	1.47	1.47	1.42	1.44	1.43	1.46
	U2	OPR	1.54	1.57	1.54	1.53	1.54	1.51	1.55	1.56	1.54	1.53
		HH-TGP	1.45	1.42	1.44	1.45	1.46	1.47	1.43	1.43	1.44	1.45
	B1	OPR	1.55	1.55	1.53	1.52	1.52	1.51	1.55	1.53	1.55	1.52
		HH-TGP	1.44	1.46	1.45	1.47	1.47	1.47	1.43	1.45	1.43	1.46
	B2	OPR	1.52	1.56	1.53	1.52	1.53	1.50	1.53	1.55	1.53	1.53
		HH-TGP	1.47	1.45	1.46	1.47	1.47	1.49	1.46	1.44	1.46	1.45

	E	OPR	1.54	1.56	1.56	1.55	1.55	1.53	1.55	1.54	1.56	1.55
		HH-TGP	1.45	1.43	1.43	1.44	1.44	1.46	1.43	1.45	1.43	1.43

The detailed comparison between HH-TGP and HH-SGP

Per	*Dis*	Strategy	Num_1	Num_2	Num_3	Num_4	Num_5	Num_6	Num_7	Num_8	Num_9	Num_{10}
$rk_{rule,Q}$	/	OC	1.84	1.81	1.82	1.77	1.80	1.82	1.83	1.80	1.82	1.81
		HH-TGP	1.15	1.18	1.16	1.23	1.19	1.17	1.16	1.20	1.17	1.18
$rk_{rule,R}$	U1	OC	1.47	1.53	1.47	1.50	1.43	1.47	1.52	1.53	1.49	1.45
		HH-TGP	1.50	1.45	1.52	1.48	1.55	1.52	1.46	1.45	1.49	1.53
	U2	OC	1.47	1.50	1.43	1.51	1.43	1.44	1.47	1.51	1.46	1.44
		HH-TGP	1.52	1.49	1.56	1.48	1.55	1.55	1.52	1.48	1.52	1.55
	B1	OC	1.48	1.48	1.47	1.50	1.42	1.47	1.49	1.51	1.49	1.44
		HH-TGP	1.49	1.49	1.51	1.48	1.56	1.51	1.49	1.46	1.49	1.54
	B2	OC	1.47	1.46	1.45	1.50	1.47	1.47	1.46	1.50	1.49	1.47
		HH-TGP	1.52	1.53	1.53	1.49	1.52	1.51	1.53	1.49	1.50	1.52
	E	OC	1.46	1.47	1.46	1.53	1.45	1.46	1.49	1.50	1.53	1.49
		HH-TGP	1.53	1.52	1.53	1.46	1.54	1.53	1.50	1.49	1.46	1.50
$rk_{rule,C}$	U1	OC	1.66	1.67	1.65	1.63	1.62	1.65	1.67	1.66	1.66	1.63
		HH-TGP	1.32	1.32	1.34	1.35	1.37	1.34	1.31	1.32	1.33	1.35
	U2	OC	1.65	1.65	1.62	1.64	1.62	1.63	1.65	1.65	1.64	1.62
		HH-TGP	1.38	1.34	1.36	1.35	1.37	1.36	1.34	1.34	1.36	1.36
	B1	OC	1.66	1.65	1.65	1.63	1.61	1.65	1.66	1.66	1.65	1.63
		HH-TGP	1.32	1.34	1.34	1.36	1.37	1.34	1.33	1.33	1.33	1.36
	B2	OC	1.66	1.64	1.64	1.63	1.64	1.65	1.64	1.65	1.66	1.64
		HH-TGP	1.34	1.36	1.35	1.36	1.35	1.34	1.35	1.34	1.34	1.35
	E	OC	1.65	1.64	1.64	1.65	1.63	1.64	1.66	1.65	1.67	1.65
		HH-TGP	1.34	1.35	1.35	1.35	1.37	1.35	1.33	1.34	1.32	1.34

The detailed comparison of HH-TGP under different weights

Per	*Dis*	w_1/w_2	Num_1	Num_2	Num_3	Num_4	Num_5	Num_6	Num_7	Num_8	Num_9	Num_{10}
$rk_{rule,Q}$	/	0.1/0.9	3.73	4.19	3.99	4.03	4.10	4.17	3.78	4.51	4.45	3.82
		0.2/0.8	2.84	3.21	3.16	3.18	3.25	3.00	3.14	3.39	3.15	3.39
		0.3/0.7	2.75	2.83	3.07	3.02	2.98	2.87	2.89	2.95	2.95	2.99
		0.4/0.6	2.96	3.13	3.07	2.84	3.07	2.99	2.94	3.00	3.10	3.29
		0.5/0.5	3.06	3.55	3.14	2.98	3.39	3.08	3.17	3.51	3.21	3.46
$rk_{rule,R}$	U1	0.1/0.9	8.46	8.23	8.25	8.52	8.41	8.14	8.55	8.03	8.08	8.60
		0.2/0.8	8.55	8.44	8.64	8.49	8.44	8.59	8.46	7.98	8.27	8.40
		0.3/0.7	8.57	8.23	8.85	8.41	9.50	8.92	8.42	8.19	8.54	9.12
		0.4/0.6	9.23	9.07	8.75	8.58	9.36	9.28	8.77	9.03	9.01	9.57
		0.5/0.5	9.85	10.14	9.19	8.85	9.69	9.47	9.17	9.73	9.25	9.84
	U2	0.1/0.9	9.22	9.09	8.96	9.23	9.22	8.55	8.92	8.71	8.76	9.06
		0.2/0.8	9.02	9.12	9.57	8.95	9.21	9.20	9.43	8.53	9.21	9.02
		0.3/0.7	9.25	8.93	9.53	8.77	9.91	9.46	9.16	8.70	9.20	9.73
		0.4/0.6	10.12	9.91	9.40	9.47	9.99	9.80	9.49	9.75	9.74	10.23

		0.5/0.5	10.61	10.98	10.05	9.64	10.60	10.35	9.98	10.43	10.26	10.62
		0.1/0.9	8.56	8.41	8.35	8.58	8.62	8.23	8.50	8.00	8.12	8.68
		0.2/0.8	8.51	8.63	8.79	8.56	8.68	8.83	8.72	8.12	8.42	8.50
	B1	0.3/0.7	8.68	8.46	8.82	8.49	9.40	8.96	8.69	8.34	8.63	9.13
		0.4/0.6	9.36	9.28	8.89	8.52	9.43	9.31	8.86	9.08	9.05	9.61
		0.5/0.5	9.95	10.17	9.24	8.87	9.72	9.49	9.33	9.74	9.33	9.78
		0.1/0.9	9.52	9.48	9.15	9.41	9.44	8.99	9.14	8.94	9.09	9.44
		0.2/0.8	9.01	9.31	9.94	9.26	9.65	9.66	9.49	8.65	9.67	9.49
	B2	0.3/0.7	9.18	9.28	9.19	8.50	9.10	9.12	9.16	8.51	8.89	9.13
		0.4/0.6	9.50	9.30	9.18	9.42	9.28	9.30	9.34	9.50	9.11	9.48
		0.5/0.5	9.84	10.55	9.45	9.11	9.67	9.70	9.58	9.86	9.53	9.77
		0.1/0.9	9.95	9.88	9.66	9.77	9.93	9.20	9.56	9.41	9.54	9.62
		0.2/0.8	9.49	9.77	10.60	9.27	10.37	10.06	9.99	8.93	10.19	9.84
	E	0.3/0.7	9.82	9.88	9.84	9.26	9.99	9.81	9.63	9.63	9.23	9.64
		0.4/0.6	10.46	10.22	10.02	10.17	10.05	10.37	10.25	10.16	10.22	10.54
		0.5/0.5	11.09	11.26	10.78	10.08	10.70	10.89	10.70	10.74	10.69	11.00
		0.1/0.9	6.10	6.21	6.12	6.27	6.25	6.15	6.17	6.27	6.27	6.21
		0.2/0.8	5.70	5.82	5.90	5.84	5.84	5.80	5.80	5.68	5.71	5.89
	U1	0.3/0.7	5.66	5.53	5.96	5.71	6.24	5.90	5.65	5.57	5.75	6.06
		0.4/0.6	6.10	6.10	5.91	5.71	6.21	6.14	5.86	6.01	6.06	6.46
		0.5/0.5	6.46	6.85	6.16	5.92	6.54	6.28	6.17	6.62	6.23	6.65
		0.1/0.9	6.47	6.64	6.48	6.63	6.66	6.36	6.35	6.61	6.60	6.44
		0.2/0.8	5.93	6.16	6.36	6.06	6.23	6.10	6.28	5.96	6.18	6.21
	U2	0.3/0.7	6.00	5.88	6.30	5.89	6.45	6.17	6.03	5.82	6.07	6.36
		0.4/0.6	6.54	6.52	6.23	6.16	6.53	6.40	6.21	6.37	6.42	6.76
		0.5/0.5	6.84	7.26	6.60	6.31	6.99	6.72	6.58	6.97	6.73	7.04
		0.1/0.9	6.14	6.30	6.17	6.31	6.36	6.20	6.14	6.25	6.29	6.25
		0.2/0.8	5.68	5.92	5.98	5.87	5.96	5.91	5.93	5.75	5.79	5.96
$rk_{rule,C}$	B1	0.3/0.7	5.72	5.65	5.95	5.75	6.19	5.92	5.79	5.64	5.79	6.06
		0.4/0.6	6.16	6.21	5.98	5.68	6.25	6.15	5.90	6.04	6.08	6.45
		0.5/0.5	6.50	6.86	6.19	5.93	6.55	6.29	6.25	6.63	6.27	6.62
		0.1/0.9	6.62	6.84	6.57	6.72	6.77	6.58	6.46	6.72	6.77	6.63
		0.2/0.8	5.92	6.26	6.55	6.22	6.45	6.33	6.31	6.02	6.41	6.44
	B2	0.3/0.7	5.97	6.06	6.13	5.76	6.04	6.00	6.03	5.73	5.92	6.06
		0.4/0.6	6.23	6.22	6.12	6.13	6.17	6.14	6.14	6.25	6.11	6.39
		0.5/0.5	6.45	7.05	6.30	6.05	6.53	6.39	6.38	6.69	6.37	6.62
		0.1/0.9	6.84	7.04	6.83	6.90	7.02	6.69	6.67	6.96	7.00	6.72
		0.2/0.8	6.16	6.49	6.88	6.22	6.81	6.53	6.56	6.16	6.67	6.62
	E	0.3/0.7	6.29	6.36	6.45	6.14	6.49	6.34	6.26	6.29	6.09	6.31
		0.4/0.6	6.71	6.68	6.54	6.51	6.56	6.68	6.60	6.58	6.66	6.91
		0.5/0.5	7.07	7.41	6.96	6.53	7.04	6.99	6.94	7.13	6.95	7.23

The detailed evolution results of HH-TGP with SPEA2

Per	*Dis*	Num_1	Num_2	Num_3	Num_4	Num_5	Num_6	Num_7	Num_8	Num_9	Num_{10}
$rk_{rule,Q}$	/	4.13	4.53	4.54	4.14	4.36	4.09	4.13	4.32	4.12	4.03
$rk_{rule,R}$	U1	8.44	7.80	7.58	8.25	7.96	8.64	8.46	8.03	8.14	8.41
	U2	9.19	8.50	8.18	8.73	8.25	9.15	8.78	8.41	8.53	9.03
	B1	8.61	7.98	7.90	8.50	8.17	8.65	8.51	8.19	8.38	8.55
	B2	9.57	8.74	8.72	8.96	8.73	9.37	9.07	8.91	8.81	9.45
	E	9.86	9.23	8.62	9.12	8.64	9.80	9.51	8.91	8.91	10.05
$rk_{rule,C}$	U1	6.28	6.16	6.06	6.20	6.16	6.37	6.29	6.17	6.13	6.22
	U2	6.66	6.52	6.36	6.44	6.31	6.62	6.46	6.36	6.33	6.53
	B1	6.37	6.26	6.22	6.32	6.26	6.37	6.32	6.25	6.25	6.29
	B2	6.85	6.64	6.63	6.55	6.55	6.73	6.60	6.62	6.47	6.74
	E	6.99	6.88	6.58	6.63	6.50	6.94	6.82	6.62	6.52	7.04