Multi-tree Genetic Programming for Dynamic Tugboat Scheduling

Xinxin Xu[1,2], Fangfang Zhang[2(✉)], Yi Mei[2], Mengjie Zhang[2], Huili Gong[1], and Xiangqian Ding[1]

[1] School of Computer Science and Technology, Ocean University of China, Qingdao, China

[2] Centre for DataScience and Artificial Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

fangfang.zhang@ecs.vuw.ac.nz

Abstract. The tugboat scheduling task aims to efficiently allocate tugboat resources to assist ships entering and leaving the port in maritime transportation. Many existing scheduling methods focus on directly finding scheduling solutions. However, they are not suitable to deal with the large-scale and dynamic characteristics of maritime transportation systems due to long scheduling time. Therefore, this paper focuses on finding scheduling rules for large-scale dynamic tugboat scheduling problem (DTug-sp), including two core tasks: allocating ships to tugboats (i.e., the allocation rule) and determining the execution order of ships assigned to a particular tugboat (i.e., the order rule). To solve this problem, we propose a multi-tree genetic programming method for DTug-sp, termed as MTGP-DTsp, which uses a dual-tree encoding strategy to represent the ship allocation rule and the tugboat execution order rule, respectively. Additionally, a new crossover operator is introduced to enhance the effectiveness of the generated scheduling rules. Experimental results demonstrate that MTGP-DTsp can effectively evolve scheduling rules suitable for DTug-sp, achieving the goal of minimizing tugboat assisting time and detecting the minimum weighted average tugboat assisting time, separately.

Keywords: Dynamic tugboat scheduling problem · Dual-tree encoding strategy · Multi-tree genetic programming

1 Introduction

Throughput is a key criterion for measuring the scale and efficiency of port operations, and it is influenced by the scheduling of port terminal resources such as cranes, berths, yards and tugboats [1–6]. The rationalization of port resources is an important part in improving the effectiveness of port operations. At present, the researchers have undertaken studies of cranes [1], berths [2,5], and yards scheduling [4]. As the "first service station" for ships entering the port and the "last link" for ships leaving the port, tugboat scheduling directly affects the efficiency of port operations such as the total number of ships handled by the port [6]. Early studies have been made in the tugboat scheduling

P. García-Sánchez et al. (Eds.): EvoApplications 2025, LNCS 15613, pp. 195–210, 2025.
https://doi.org/10.1007/978-3-031-90065-5_12

problem (Tug-sp) using traditional scheduling algorithms such as integer programming, yet this approach has proven inefficient for handling Tug-sp on a large scale problems [7,8]. Additionally, most existing Tug-sp studies focus on static scheduling problems, but dynamic factors such as uncertain ship arrival times can affect tugboat operations [8–10]. Moreover, most algorithms used to solve Tug-sp problems require a large amount of computational resources, and the solutions generated are typically applicable to only a single instance, which limits their flexibility and effectiveness in varying scenarios [8]. As the problem scale increases, manually designed rules e.g., first come first served and shortest processing time, can generate solutions quickly, making it easier to handle large-scale problems and dynamic changes. However, they have limited effectiveness for tugboat scheduling.

Genetic programming (GP) is an effective approach for scheduling problems by evolving the scheduling rules [11]. GP has been widely used to learn rules for scheduling problems such as the dynamic flexible job shop scheduling problem [12–16]. A multi-tree GP (MTGP) was proposed to solve the dynamic flexible job shop scheduling problem [15]. DTug-sp is similar to the dynamic flexible job shop scheduling problem which evolves two rules for allocating jobs to machines and executing jobs on machines. Similarly, solving DTug-sp involves determining how to allocate ships to tugboats (i.e., allocation rule) and the order of executing ships on a tugboat (i.e., order rule). In this work, we construct DTug-sp as a coordinated decision-making process with two key rules. A dual-tree solution encoding (DTSE) strategy is employed for MTGP-DTsp to represent both the ship allocation rule and the tugboat execution order rule. Additionally, a crossover operator is borrowed for the DTSE strategy to promote the effectiveness of the generation of offspring in GP for the next generation.

2 Background

2.1 Tugboat Scheduling Problems

When the ship reaches the port, it normally waits at the anchorage. If a berth is available for loading and unloading operations, the ship needs to be towed by a tugboat with appropriate horsepower, facilitating the movement of ships entering and leaving the port, as illustrated in Fig. 1. There are three types of tugboat scheduling operations for ships in the channel at the port: **1) Berthing** which involves assisting the ship in reaching the berth; **2) Shifting** which entails moving the ship from one berth to another; and **3) Unberthing** which assists the ship in departing from the berth.

Currently, the tugboat scheduling problem can be classified into two categories according to the available information, i.e., either static or dynamic problem. Static Tug-sp problem generally assumes that all information is known in advance before making [7]. Dynamic Tug-sp problem (i.e., DTug-sp) considers some dynamic events which is closer to real-world applications. However, this is rarely studied. Specifically, the ship dynamic arrivals are the most common uncertain events in the real world. As time progresses, ships dynamically request

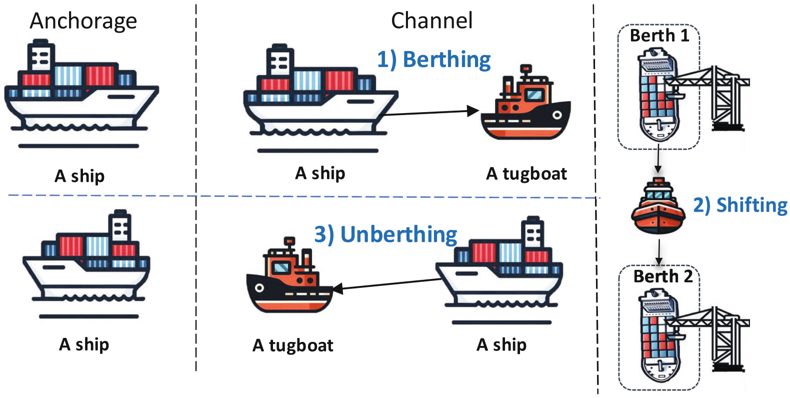


Fig. 1. An example of tugboat operations.

services, and subsequent tugboat scheduling plans need to be arranged in real-time based on ship demand. There are only a few studies focusing on DTug-sp to minimize operation costs [7,17,18]. Some methods employ heuristic algorithms to address DTug-sp. For large-scale uncertain tugboat scheduling, Li et al. [19] considered the dynamic scheduling of ship multi-tugboat berthing bases and proposed a grey wolf optimization algorithm to minimize energy consumption and total delay costs. Sun et al. [17] proposed an improved genetic algorithm based on reverse operations to solve the tugboat optimization problem, accounting for the uncertainty in cross-regional operations of tugboats at multiple terminals. Wang et al. [20] conducted tugboat scheduling on ships based on manual rules. These methods are limited by assumptions specific to certain operational conditions, which may not generalize to multiple tugboat scheduling scenarios. Additionally, relying on heuristic methods may result in suboptimal solutions under high uncertainty, and manual scheduling lacks scalability and adaptability for large-scale operations. This paper proposes to use GP approach to overcome these limitations, enhancing adaptability and optimizing scheduling rules across diverse scenarios.

2.2 GP for Scheduling Problems

GP generates multiple heuristic search schemes which uses low-level heuristics (simple rules) to generate high-level heuristics (comprehensive rules) [13]. It has been applied in various scheduling problems [14,21,22]. Many studies have shown that the production scheduling heuristic learned by GP is superior to the heuristic designed manually in literature [15,23]. However, current research has not yet considered applying GP to solve DTug-sp problems. At present, the scheduling of tugboats in ports is mostly based on the manually designed rules, such as first come first served and shortest processing time. These rules are relatively limited to fully reflect the current resource occupancy status. Therefore, we design MTGP-DTsp to search for more effective rules for tugboat scheduling.

Table 1. Model parameter definition.

Symbols	Definitions
$\mathcal{S}$	The set of arrived ships, $S = \{S_1, S_2, \ldots, S_n\}$
$\mathcal{B}$	The set of available tugboats, $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$
i	The number of the operations of a ship, $i = \{1, 2\}$
S_j	The water displacement of the j-th ship (ton)
C_s	Ship resistance coefficient
B_l^p	The horsepower value of the l-th tugboat
K	Horsepower conversion coefficient
T_0	Correction coefficient of time
o_{ji}	The operational tasks of ship j in operation i
$t_{o_{ji}}^{pro}$	The total work time of the task o_{ji}
$t_{o_{ji}}^{str}$	The operation o_{ji} starting time of ship j
w_j	Priority of ship j
N	A sufficiently large constant

3 The Proposed MTGP-DTsp Algorithm for DTug-sp

3.1 Problem Description

If more than 100 ships arrive per day, it will be considered as a large-scale problem [24]. We design three simulation scenarios for the ship arrivals, with the number of ships set at 120, 150, and 200, respectively. The notations and decision variables are provided in Table 1. The assistance of tugboats is required for ships to enter and leave port berths, and the probability of shifting from one berth to another is low. Therefore, in the construction of this model, we only consider berthing and berthing operations for the tugboat assistance required by each ship at the port. Therefore, the operation process of each ship is abstracted as two-stage operations, and the set of operations for all ships arriving at the port is set as $\{O_{11}, O_{12}, O_{21}, O_{22}, \cdots O_{n1}, O_{n2}\}$, where n is the number of ships need to be handled. Unberthing operations only occur after the completion of berthing operations. Two types of decisions need to be handled in real time. Figure 2 shows an example of the decision process of DTug-sp.

- Decision 1 ($D1$): Allocating ship's operations to the queue of tugboats. The first operation of a ship, or the second operation of a ship whose precedent operation (first operation) has been executed are the ready operations to be allocated to tugboat. The tugboat with the highest priority value will be selected to execute an operation of a ship.
- Decision 2 ($D2$): Scheduling the execution order of operations on a specific tugboat. Specifically, the operation with the highest priority value will be selected to be executed next by an idle tugboat.

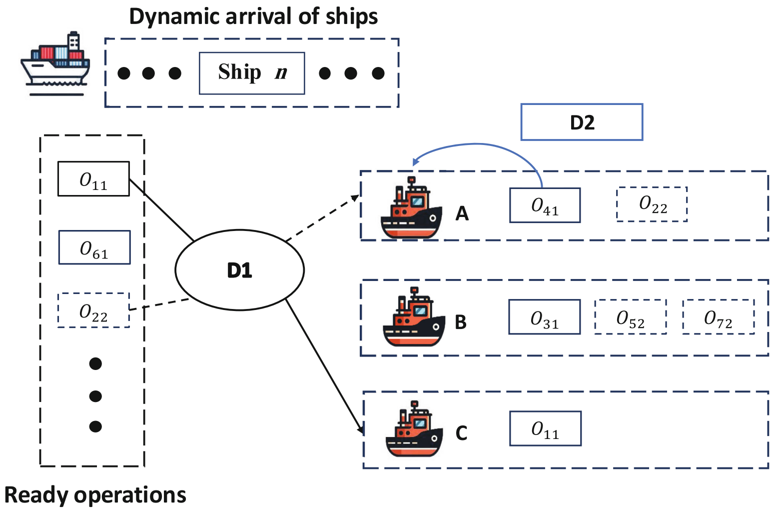


Fig. 2. An example of the decision process of DTug-sp.

In DTug-sp, the terminal has a certain number of tugboats denoted by $B = \{B_1, B_2, \ldots, B_k\}$, each with different horsepower values P, and these tugboats belong to different bases $C = \{C_1, C_2, \ldots, C_w\}$. Ships, denoted by $S = \{S_1, S_2, \ldots, S_n\}$, dynamically arrive at the port. Each ship S_j has two important features: an arrival time t_j^a and a due time t_j^d. Each operation $O_{j,i}$ is processable only by a subset $(k_{j,i})$ of tugboats. The assisting time $t_{j,i,m}^{pro}$ of tugboat operation $O_{j,i}$ depends on the selected tugboat $B_m \in B$. Once the unberthing operation is completed, all operations for that ship are considered complete. The main assumptions of DTug-sp are shown as follows:

1) For ship operations, once a tugboat operation is processed, it will be completed successfully without encountering obstacles.
2) The tugboat operation time varies for the same type of ship depending on the horsepower values of the tugboat. Once a specific tugboat is selected, the operation time of an operation of the ship is confirmed.

Different ships require different types of tugboats, and the scheduling time for berthing operations can be shortened as the horsepower value of the chosen tugboat increases. Table 2 shows the types of tugboats with different horsepower values, and Table 3 shows the available tugboat types for different ships.

3.2 Objective Functions and Constraints

This paper aims to minimize the total tugboat assisting time, and minimize the mean weighted tugboat assisting time for ships. The objective formulation is as follows:

Table 2. Types of tugboats.

Tugboat type	Bollard pull force
A	1200 tonnes
B	2600 tonnes
C	3200 tonnes
D	3400 tonnes
E	4000 tonnes
F	5000 tonnes

Table 3. The available tugboat types for different ships.

Index	Types of ships	Number tugboats required	Type of tugboats required
1	Up to 10000 tons	1 tugboat	A,B,C,D,E,F
2	10000 to 18000 tons	1 tugboat	B,C,D,E,F
3	18100 to 29900 tons	1 tugboat	C,D,E,F
4	30000 to 35000 tons	1 tugboat	D,E,F
5	35000 above tons	1 tugboat	E,F

$$F_{Total-time} = \sum\nolimits_{j=1}^{n} t_j^{com} - t_j^{str} \tag{1}$$

$$F_{Mean-weighted-time} = \frac{\sum_{j=1}^{n} w_j (t_j^{com} - t_j^{str})}{n} \tag{2}$$

where t_j^{com} is the completion time of ship S_j unberthing work, t_j^{str} is the start time of ship S_j, and w_j is the priority level of ship work weight. Different ships have different priority levels in port companies, represented by different weight values.

The formula for calculating tugboat work time based on ship tonnage and tugboat horsepower is as follows:

$$t_{o_{ji}}^{pro} = \frac{C_s \times {S_j}^{\omega}}{K \times {B_l}^{p}} \times T_0 \tag{3}$$

where C_s is the ship resistance coefficient which is 0.04, K is the horsepower conversion coefficient which is 0.3, and the standard time coefficient T_0 is set to 1 [25]. According to the Eq. (3), as the horsepower value of the selected tugboat increases, the scheduling time for tugboats to assist in berthing can be shortened.

The model is subject to the following constraints:

$$\begin{aligned} & t_{j_1,i_1}^{str} \leq t_{j_2,i_2}^{str} - t_{j_1,i_1,m}^{pro} + N \cdot (1 - x_{j_1,i_1,j_2,i_2,m}), \\ & \forall j_1, j_2 = 1, \cdots, n; \forall i_1 = 1, \cdots, q_{j1}; \\ & \forall i_2 = 1, \cdots, q_{j2}; \forall m = 1, \cdots, k \end{aligned} \tag{4}$$

$$t_{j,1}^{str} \geq t_j^{a}, \forall j = 1, \cdots, n \tag{5}$$

$$t_{j,i}^{com} = t_{j,i}^{str} + t_{j,i,m}^{pro} \cdot y_{j,i,m}, \forall_j = 1, \cdots, n; \quad \forall i = 1, \cdots, q_j; \forall m = 1, \cdots, k \tag{6}$$

$$t_{j,i+1}^{str} \geq t_{j,i}^{com}, \forall j = 1, \cdots, n; \forall i = 1, \cdots, q_j - 1 \tag{7}$$

$$\sum\nolimits_{m=1}^{k_{j,i}} y_{j,i,m} = 1, \forall j = 1, \cdots, n; \forall i = 1, \cdots, q_j \tag{8}$$

$$x_{j_1,i_1,j_2,i_2,m} = \begin{cases} 1, & \text{if } O_{j_1,i_1} \text{ is processed by } B_m \text{ before } O_{j_2,i_2}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

$$y_{j,i,m} = \begin{cases} 1, & \text{if } O_{j,i} \text{ is assigned to } B_m, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Constraint (4) specifies that tugboats can only provide assistance to one ship at a time, with N representing a sufficiently large constant. Constraint (5) denotes that the assisting of each ship's first operation can commence only after the available tugboat is released. Constraint (6) establishes the relationship between the assisting start time $t_{j,i}^{start}$ and the assisting completion time $t_{j,i}^{com}$ for the operation $O_{j,i}$. Constraint (7) states that the operation $O_{j,i+1}$ must wait for the completion of its preceding operation $O_{j,i}$ before it can be processed on the assigned tugboat. Constraint (8) indicates that an operation can be processed only on one of its candidate tugboats. Constraint (9) ensures that the unberthing operation is completed before the berthing operation. Constraint (10) indicates whether there is a tugboat serving a ship.

3.3 Framework

MTGP-DTsp aims to learn scheduling heuristics to assist the decision-making process of tugboat scheduling. This section gives a detailed description of MTGP-DTsp, including the solution representations, the fitness function, and the process of initialization and evolution. MTGP-DTsp is to automatically generate the following two rules for decision-making:

Allocation Rule (Rule 1 for D1): Select a tugboat for the berthing of ship operation.

Order Rule (Rule 2 for D2): Select the next ship operation to be executed by a tugboat from its queue.

Figure 3 shows the flowchart of MTGP-DTsp to learn heuristics rules for solving the DTug-sp. The algorithm framework for the initialization and evolution processes of MTGP-DTsp refers to work [16]. A population is initialized, where the two trees of each individual are constructed by ramp half-and-half. The evolutionary algorithm of MTGP consists of four parts: population initialization, individual evaluation, parental selection, and evolution. In MTGP-DTsp, each individual with two trees represents the heuristic rules. For individual representation, MTGP-DTsp introduces a DTSE strategy to represent the two rules of

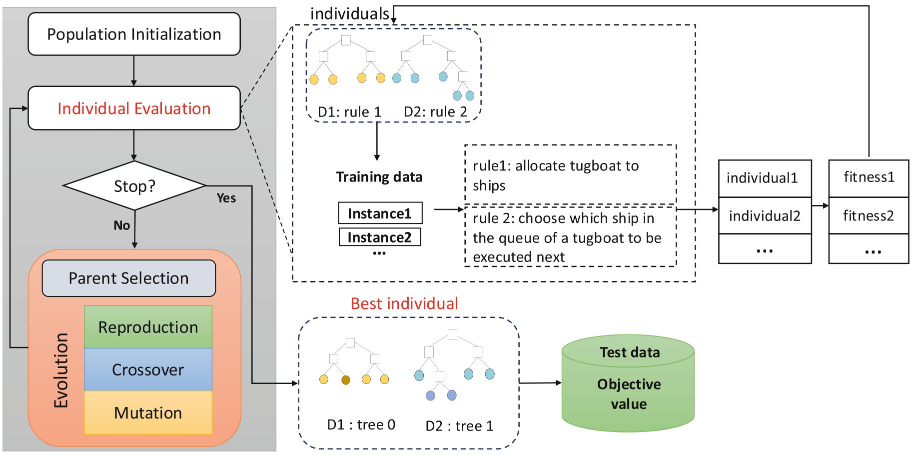


Fig. 3. The flowchart of MTGP-DTsp approach.

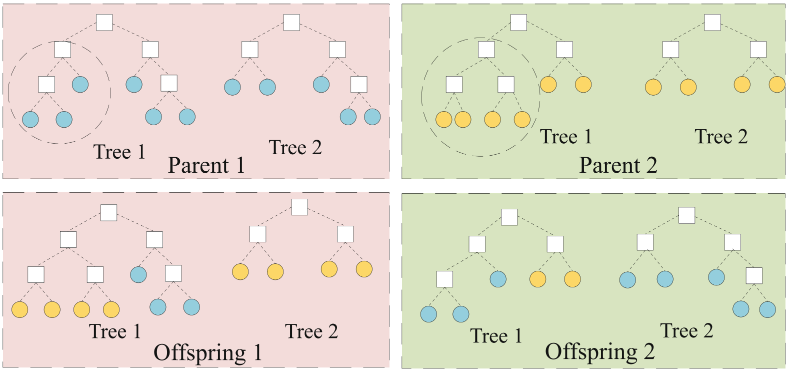


Fig. 4. An example of the crossover operator for generating offspring.

each individual. Specifically, DTSE strategy allows each individual composed of two trees, each of which corresponds to one rule. The first tree is used to determine the allocated tugboat, while the second tree is used to determine the order of tasks. The fitness of an individual depends on the cooperation of these two rules.

There are three important processes in the evolutionary process: reproduction, crossover, and mutation. To keep good individuals into the next generation, we use reproduction to copy promising individuals into the next generation. We perform crossover operations on both trees of the parents, resulting in two new individuals (each individual contains two trees with two different rules). The crossover operator is illustrated in Fig. 4. For Parent 1 and Parent 2, one same type of trees performs the crossover operator, and the other type of trees is swapped. For example, the subtrees (dashed circles) of Tree 1 in Parent 1 and Parent 2 are replaced with each other to generate Tree 1, and the other tree, tree 2 for both parents is swapped to generate the new offspring 1 and offspring

Table 4. The terminal set of MTGP-DTsp.

No.	Notation	Description
1	TWT	The waiting time of a ship operation
2	NRS	The number of operations remaining in ready work
3	W	Weight of a ship
4	NET	The median assisting time for the next tugboat operation
5	WKR	The median amount of work remaining for ships
6	OIQ	The number of ship operations in the queue
7	CTQ	The current total required assisting time of the remaining ship operations in the queue of a tugboat
8	RWT	The waiting time for tugboat in ready state
9	SET	The assisting time of a ship on a specified tugboat
10	TIS	Time in system

2. Meanwhile, we use mutation operator to increase the diversity of offspring populations, i.e., a tree is randomly selected from a parent individual, and a new subtree with the same decision type is generated to replace the selected subtree.

4 Experimental Studies

4.1 Parameter Setting

To verify the effectiveness of the rules, the simulation is configured with the number of arriving ships set to 120, 150, and 200 for different scenarios. The weights for 20%, 60%, and 20% ships are set to 1, 2, and 4, respectively. Based on the most common specifications of tugboat horsepower and ship tonnage at the terminal [8], the model includes six of tugboats with horsepower values of 1200 HP, 2600 HP, 3200 HP, 4000 HP, 5000 HP, and over 5500 HP, as well as five ship tonnage categories of 30,000 tons, 50,000 tons, 80,000 tons, 120,000 tons, and 150,000 tons. Different types of tugboats can serve different types of ships. The utilization levels are set as 0.8, 0.85, 0.9, and 0.95 in different scenarios. The ship uncertain arrival time simulates according to a Poisson process with a rate of λ. The utilization level (p) is an important indicator of the busy level of tugboats. It is expressed as $p = \lambda * \mu * P_M$, where μ is the average operation time of the tugboat. P_M is the probability that a ship is assigned to a tugboat. Each ship has two operations, then the P_M is $2/10$.

The GP requires terminal nodes and functional nodes to construct a tree representing individuals. The terminal set for MTGP-DTsp is shown in Table 4. The function set of the function nodes is generally set as $\{+, -, *, /, Max, Min\}$, the "/" operator is a protected division that returns 1 when it is divided by 0. The other parameters of MTGP-DTsp are shown in Table 5.

4.2 Design of Comparisons

To verify the effectiveness of the rules obtained by the proposed MTGP-DTsp algorithm, we compare the rule obtained by MTGP-DTsp with its two variations and the commonly used manual rules [16] as shown in Table 6. Firstly, to verify the effectiveness in optimizing the Total-time objective, DTsp-L (using the Least Work in Queue (LWQ) rule) is for only learning the allocation decision and DTsp-S (using the Shortest Processing Time (SPT) rule) is for only learning the order decision, named as DTsp-L, DTsp-S, respectively. In addition, L&S uses the manually rules, i.e., LWQ rule and SPT rule simultaneously. Secondly, to verify the effectiveness in optimizing the Mean-weighted-time objective, DTsp-E (using the early preparation time (ERT) rule) is for only learning the allocation decision and DTsp-F (using the first come first served (FCFS) rule) is for only learning the order decision the order decision, named as DTsp-E, DTsp-F, respectively. In addition, E&F uses the manually rules, i.e., ERT rule and FCFS rule simultaneously.

Table 5. Other parameter settings of MTGP-DTsp.

Parameter	Value
population size	200
Maximal depth	8
Crossover/Mutation/Reproduction	80%/15%/5%
Parent selection	Tournament selection with size 5
Elitism	2 best individuals
Number of generations	51

Table 6. Comparison of MTGP-DTsp variants and manually designed rules.

Objective	Algorithm Name	Allocation rule	Order rule
Total-time	MTGP-DTsp	GP-Rule	GP-Rule
	DTsp-L	LWQ	GP-Rule
	DTsp-S	GP-Rule	SPT
	L&S	LWQ	SPT
Mean-weighted-time	MTGP-DTsp	GP-Rule	GP-Rule
	DTsp-E	ERT	GP-Rule
	DTsp-F	GP-Rule	FCFS
	E&F	ERT	FCFS

4.3 Results and Discussions

The Quality of Learned Scheduling Rules: In this experiment, 12 scenarios are set for each objective (using 4 different utilization levels for three types of ship arrival test samples) to test the performance of MTGP-DTsp. The performance of the proposed algorithm is evaluated on the basis of the results of 30 independent runs. We compare MTGP-DTsp with the manual rules commonly used for the investigated problems. The significance level of the Wilcoxon test is 0.05, and the Wilcoxon rank sum test with Bonferroni correction is conducted between the proposed algorithm and other algorithms. In the following results, "↑" and "↓" indicate that the corresponding result is significantly better than, or worse than its counterpart, respectively.

Table 7 shows the mean and standard deviation of Total-time among four methods in 12 test scenarios over 30 runs. The smaller number represents the better performance. The bold numbers represent the best overall performance. The results show that MTGP-DTsp has a significant advantage over L&S, DTsp-S, and DTsp-L. MTGP-DTsp consistently achieves lower total times across all 12 scenarios, defined by different ship arrival rates and tugboat utilization rates. Compared to L&S, MTGP-DTsp shows a significant improvement, nearly halving the total time in all scenarios. Compared to DTsp-S, MTGP-DTsp is competitive, with significantly better performance in all scenarios. MTGP-DTsp shows a substantial advantage compared with DTsp-L, achieving much lower total times. Compared with L&S, DTsp-S and DTsp-L performs significantly better in all scenarios, which indicates that learning either of the two rules with GP can improve the performance. However, when comparing with DTsp-L, DTsp-S performs significantly better, which indicates that learning allocation rule with GP has a bigger effect on the performance improvement than learning order rule with GP.

Table 8 presents the mean and standard deviation of Mean-weighted-time for different scheduling methods (MTGP-DTsp, E&F, DTsp-E, and DTsp-F) across 12 scenarios defined by varying ship arrival rates (120, 150, and 200) and

Table 7. Mean and standard deviation of the **Total-time** for the comparison methods in 12 test scenarios over 30 runs.

Scenarios	L&S	DTsp-S	DTsp-L	MTGP-DTsp
120(0.80)	861.50(0)	454.49(3.83E+00)(↑)	857.50(1.64E+01)(↑)(↓)	**434.88(4.12E+00)**(↑)(↑)(↑)
120(0.85)	866.62(0)	455.59(1.08E+01)(↑)	858.86(3.91E+00)(↑)(↓)	**435.63(1.59E+01)**(↑)(↑)(↑)
120(0.90)	868.90(0)	448.16(2.85E+01)(↑)	861.65(1.08E+01)(↑)(↓)	**437.83(2.58E+01)**(↑)(↑)(↑)
120(0.95)	870.99(0)	465.60(8.95E+00)(↑)	868.02(3.03E+01)(↑)(↓)	**435.47(5.78E+00)**(↑)(↑)(↑)
150(0.80)	1066.23(0)	578.06(1.71E+01)(↑)	1035.83(1.90E+01)(↑)(↓)	**557.58(1.91E+01)**(↑)(↑)(↑)
150(0.85)	1072.08(0)	586.68(7.36E+00)(↑)	1042.91(7.41E+00)(↑)(↓)	**557.06(1.09E+01)**(↑)(↑)(↑)
150(0.90)	1075.00(0)	589.57(2.36E+01)(↑)	1059.57(4.87E+01)(↑)(↓)	**559.96(3.85E+01)**(↑)(↑)(↑)
150(0.95)	1077.67(0)	587.67(1.47E+00)(↑)	1059.06(1.08E+02)(↑)(↓)	**558.42(3.13E+00)**(↑)(↑)(↑)
200(0.80)	1452.99(0)	807.82(6.91E+00)(↑)	1422.88(1.80E+01)(↑)(↓)	**788.65(1.55E+01)**(↑)(↑)(↑)
200(0.85)	1456.14(0)	809.31(2.25E+01)(↑)	1433.87(1.18E+01)(↑)(↓)	**788.97(1.73E+01)**(↑)(↑)(↑)
200(0.90)	1459.62(0)	802.30(1.19E+02)(↑)	1435.18(6.91E+01)(↑)(↓)	**789.54(2.70E+00)**(↑)(↑)(↑)
200(0.95)	1463.03(0)	800.42(4.75E+00)(↑)	1438.59(8.84E+01)(↑)(↓)	**789.91(2.87E+00)**(↑)(↑)(↑)

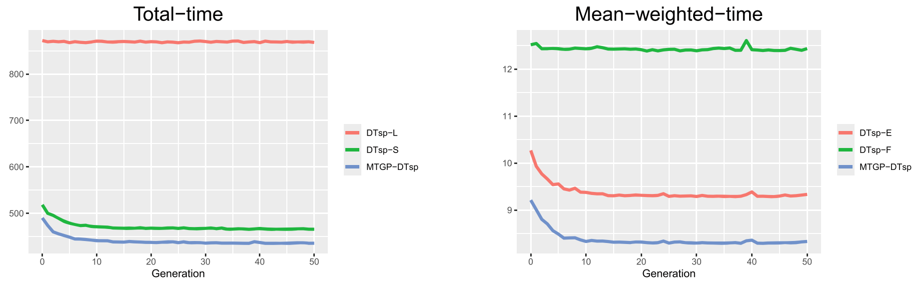
Table 8. Mean and standard deviation of the **Mean-weighted-time** of the comparison methods in 12 test scenarios over 30 runs.

Scenarios	E&F	DTsp-E	DTsp-F	MTGP-DTsp
120(0.80)	16.07(0)	9.29(6.82E−04)(↑)	12.19(9.29E−03)(↑)(↓)	**8.29(4.80E−04)**(↑)(↑)(↑)
120(0.85)	16.21(0)	9.33(1.27E−02)(↑)	12.29(1.93E−03)(↑)(↓)	**8.30(2.62E−03)**(↑)(↑)(↑)
120(0.90)	16.26(0)	9.33(6.37E−03)(↑)	12.44(2.82E−02)(↑)(↓)	**8.33(5.99E−03)**(↑)(↑)(↑)
120(0.95)	16.29(0)	9.28(3.39E−04)(↑)	12.46(1.82E−03)(↑)(↓)	**8.30(7.72E−03)**(↑)(↑)(↑)
150(0.80)	15.56(0)	9.13(1.07E−03)(↑)	11.76(1.31E−04)(↑)(↓)	**8.14(3.98E−03)**(↑)(↑)(↑)
150(0.85)	15.69(0)	9.13(7.58E−04)(↑)	11.92(1.35E−04)(↑)(↓)	**8.14(2.48E−03)**(↑)(↑)(↑)
150(0.90)	15.73(0)	9.16(5.75E−03)(↑)	12.02(1.34E−04)(↑)(↓)	**8.17(4.88E−03)**(↑)(↑)(↑)
150(0.95)	15.77(0)	9.14(3.39E−04)(↑)	12.11(6.78E−05)(↑)(↓)	**8.15(6.52E−04)**(↑)(↑)(↑)
200(0.80)	15.78(0)	9.60(8.08E−04)(↑)	12.56(2.13E−03)(↑)(↓)	**8.59(5.49E−04)**(↑)(↑)(↑)
200(0.85)	15.82(0)	9.60(1.46E−04)(↑)	12.73(9.87E−04)(↑)(↓)	**8.60(7.07E−05)**(↑)(↑)(↑)
200(0.90)	15.85(0)	9.61(1.93E−04)(↑)	12.83(9.72E−04)(↑)(↓)	**8.62(1.78E−03)**(↑)(↑)(↑)
200(0.95)	15.90(0)	9.61(1.05E−04)(↑)	12.87(2.34E−03)(↑)(↓)	**8.61(1.41E−04)**(↑)(↑)(↑)

tugboat utilization rates (0.80, 0.85, 0.90, and 0.95). MTGP-DTsp consistently achieves the lowest mean weighted time, demonstrating significant improvements over the other methods across all scenarios. This superiority is particularly evident as both arrival and utilization rates increase, where MTGP-DTsp maintains stable mean weighted time with low standard deviations, indicating robust performance. For the comparison between E&F and DTsp-E, DTsp-E performs better, with lower mean weighted time and similar or smaller standard deviations in most cases. When comparing DTsp-E and DTsp-F, DTsp-E typically performs better, achieving significantly lower mean weighted times in most scenarios. MTGP-DTsp stands out as the most effective scheduling method, optimizing mean weighted time for tugboat scheduling and demonstrating consistent effectiveness in different scenarios. E&F, on the other hand, has the worst performance. Similar to the results of optimizing Total-time, we also find that learning either rule with GP can help improve the scheduling performance, however, for optimizing Mean-weighted-time, learning the order rule has a bigger effect on the performance improvement.

In summary, both tables highlight the superior efficiency of the MTGP-DTsp method, which consistently outperforms its counterparts in various scenarios. Its ability to achieve lower total time and average weighted time prevails its effectiveness as the preferred scheduling method in tugboat operations.

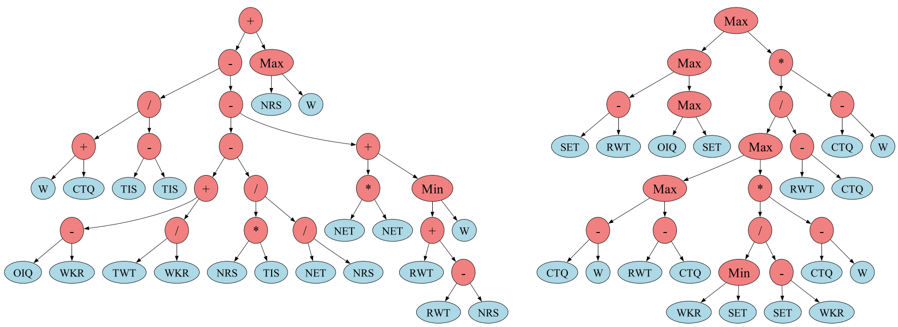
Curves of MTGP-DTsp: To explore the quality of learned rules along with generations, we select a test scenario with 120 ship arrivals and tugboat level utilization rate of 0.95 for comparison among GP related algorithms. From the curves as shown in Fig. 5, we can see that MTGP-DTsp demonstrates better convergence of learning promising rules and performs better than compared algorithms during the whole evolutionary process. We find that there is a difference in optimizing different objectives when learning only D2 decision with a fixed D1 decision compared to learning rules for D1 with fixed D2 learning. For optimizing



(a) Comparison curve of Total-time between MTGP-DTsp and DTsp-S, DTsp-L.

(b) Comparison curve of Mean-weighted - time between MTGP-DTsp and DTsp-E, DTsp-F.

Fig. 5. The average curves of a test dataset with 120 ships and the utilization rate of 0.90 according to 30 independent runs.



(a) The expression tree of allocation rule.

(b) The expression tree of order rule.

Fig. 6. An example of expression trees generated by MTGP with 120 ships at a horizontal utilization rate of 0.90.

Total-time, training rules for D1 can achieve better solutions, while for optimizing Mean-weighted-time, training rules for D2 can achieve better solutions. The reason might be that D1 and D2 decision-making stage is a key that affects DTug-sp with objective Total-time and Mean-weighted-time, respectively. This is consistent with our findings as shown in Table 7 and 8.

Insights of the Learned Scheduling Rules: To have further understanding of the behavior of the scheduling heuristics evolved by the proposed method, an evolved scheduling heuristic is selected to be analysed. Figure 6 shows two rules from the selected scheduling heuristic evolved by MTGP-DTsp in scenario (120, 0.90) for optimizing the objective of Total-time. In Fig. 6(a), the allocation rule combines nine terminals (W, CTQ, TIS, OIQ, WKR, TWT, NRS, NET and RWT) with W and NRS being the most frequently used features, indicating a focus on weights of ships and the number of operations remaining for jobs. Figure 6(b) shows the order rule, which includes six terminals (SET, RWT, OIQ, CTQ, W, and WKR), and this order rule primarily relies on the assisting time of a ship operation on a specified tugboat (SET) and the current total time of the remaining ship operations in the tugboat queue (CTQ). Analysis reveals distinct priorities at each scheduling stage: W and NRS are more crucial in stage D1, while SET and CTQ are prioritized in stage D2.

5 Conclusions

This paper aims to use MTGP-DTsp to generate effective scheduling rules for DTug-sp, with the objective of exceeding the effectiveness of manually designed scheduling rules. We propose a modular framework that integrates GP to optimize DTug-sp. Specifically, this framework addresses the limitations of conventional mathematical models and heuristic algorithms, which are often time-consuming and energy-intensive when handling large-scale scheduling tasks. By employing GP, our approach is able to automatically learn and develop robust scheduling rules that adapt to complex and dynamic scenarios, thereby reducing computational overhead and improving scalability. Experimental results prove that GP can not only learn more effective rules but also generate solutions that consistently outperform manual scheduling in terms of operational effectiveness. This advancement highlights GP's potential as a powerful tool for automating rule generation in complex scheduling environments, ultimately enhancing decision-making in real-world port scheduling.

In the future, we plan to apply MTGP-DTsp to solve multi-objective DTug-sp related to environmental sustainability, energy consumption, and company operational efficiency. It is possible that many features will have varying importance or relevance for DTug-sp in a multi-objective context. Therefore, we aim to explore GP through feature selection to identify more promising features for scheduling tasks.

References

1. Cahyono, R.T., Kenaka, S.P., Jayawardhana, B.: Simultaneous allocation and scheduling of quay cranes, yard cranes, and trucks in dynamical integrated container terminal operations. IEEE Trans. Intell. Transp. Syst. **23**, 8564–8578 (2022)
2. Zhang, D., Li, X., Wan, C., et al.: A novel hybrid deep-learning framework for medium-term container throughput forecasting: an application to China's Guangzhou, Qingdao and Shanghai hub ports. Marit Econ. Logist. **26**(1), 44–73 (2024)
3. Malekahmadi, A., Alinaghian, M., Hejazi, S.R., Assl Saidipour, M.A.: Integrated continuous berth allocation and quay crane assignment and scheduling problem with time-dependent physical constraints in container terminals. Comput. Ind. Eng. **147**, 106672 (2020)
4. Liu, W., Zhu, X., Wang, L., Zhang, Q., Tan, K.C.: Integrated scheduling of yard and rail container handling equipment and internal trucks in a multimodal port. IEEE Trans. Intel. Transp. Syst. **25**, 2987–3008 (2023)

5. Rodrigues, F., Agra, A.: Berth allocation and quay crane assignment/scheduling problem under uncertainty: a survey. Eur. J. Oper. Res. **303**, 501–524 (2022)
6. Li, K., Zhuo, Y., Luo, X.: Optimization method of fuel saving and cost reduction of tugboat main engine based on genetic algorithm. Int. J. Syst. Assur. Eng. Manag. **13**, 605–614 (2022)
7. Wei, X., Meng, Q., Lim, A., Jia, S.: Dynamic tugboat scheduling for container ports. Marit. Policy Manag. **50**(4), 492–514 (2023)
8. Abou Kasm, O., Diabat, A., Bierlaire, M.: Vessel scheduling with pilotage and tugging considerations. Transp. Res. E Logist. Transp. Rev. **148**, 102231 (2021)
9. Zhong, H., Zhang, Y., Gu, Y.: A Bi-objective green tugboat scheduling problem with the tidal port time windows. Transp. Res. Part D: Transp. Environ. **110**, 103409 (2022)
10. Wu, Y., et al.: Research on port tugboat scheduling optimization based on Genetic Algorithm. In: The International Conference on Computational Modeling, Simulation and Data Analysis. IEEE (2022)
11. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling. IEEE Trans. Evol. Comput. **28**(1), 147–167 (2023)
12. Sels, V., Gheysen, N., Vanhoucke, M.: A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. Int. J. Prod. Res. **50**, 4255–4270 (2012)
13. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. IEEE Trans. Cybernet. **51**, 1797–1811 (2020)
14. Chen, X., Bai, R., Qu, R., Dong, H.: Cooperative double-layer genetic programming hyper-heuristic for online container terminal truck dispatching. IEEE Trans. Evol. Comput. **27**, 1220–1234 (2022)
15. Zhang, F., Nguyen, S., Mei, Y., Zhang, M.: Genetic Programming for Production Scheduling: An Evolutionary Learning Approach. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-4859-5
16. Zhang, F., Mei, Y., Zhang, M.: Genetic programming with multi-tree representation for dynamic flexible job shop scheduling. In: Mitrovic, T., Xue, B., Li, X. (eds.) AI 2018. LNCS (LNAI), vol. 11320, pp. 472–484. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03991-2_43
17. Sun, C., Li, M., Chen, L., Chen, P.: Dynamic tugboat scheduling for large seaports with multiple terminals. J. Mar. Sci. Eng. **12**, 170 (2024)
18. Wei, X., Jia, S., Meng, Q., et al.: Dynamic tugboat deployment and scheduling with stochastic and time-varying service demands. Transp. Res. B Methodol. **188**, 103059 (2024)
19. Li, B., Chen, Q., Lau, Y., Dulebenets, M.A.: Tugboat scheduling with multiple berthing bases under uncertainty. J. Mar. Sci. Eng. **11**, 2180 (2023)
20. Wang, S., Zhu, M., Kaku, I., et al.: An improved discrete PSO for tugboat assignment problem under a hybrid scheduling rule in container terminal. Math. Probl. Eng., 714832 (2014)
21. MacLachlan, J., Mei, Y., Zhang, F., Zhang, M., Signal, J.: Learning emergency medical dispatch policies via genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1409–1417 (2023)
22. MacLachlan, J., Mei, Y., Zhang, F., Zhang, M.: Genetic programming for vehicle subset selection in ambulance dispatching. In: The IEEE Congress on Evolutionary Computation, pp. 1–8 (2022)

23. Huang, Z., Mei, Y., Zhang, F., Zhang, M.: Multitask linear genetic programming with shared individuals and its application to dynamic job shop scheduling. IEEE Trans. Evol. Comput. **28**(6), 1546–1560 (2024)
24. MarineTraffic (2024). Shanghai(port). https://www.vesselfinder.com/ports/CNSHG002. Accessed 17 Oct 2024
25. Nitonye, S., Adumene, S., Howells, U.U.: Numerical design and performance analysis of a tug boat propulsion system. J. Power Energy Eng. **5**(11), 80 (2017)