The Journal of Supercomputing (2024) 80:2128–2163
https://doi.org/10.1007/s11227-023-05529-0

A stochastic multi-objective optimization method for railways scheduling: a NSGA-II-based hybrid approach

Massoud Seifpour[1] · **Seyyed Amir Asghari**[2] · **Mostafa Ghobaei-Arani**[1]

Accepted: 28 June 2023 / Published online: 27 July 2023

Abstract

Optimizing resource utilization and train scheduling is essential to satisfy passengers and reduce operating costs. This study develops the train schedule under scenario-oriented stochastic conditions. The proposed approach is a multi-objective mathematical-based mixed integer linear programming (MILP) approach; the objective is to minimize the average passenger expectation and the total number of train operation cycles. The non-dominated sorting genetic algorithm (NSGA-II) has been developed with multi-crossover and multi-mutation operators, then hybrid with simulating annealing (SA) operator (NSGA-II-SA). The model with four meta-heuristic algorithms has been technically analyzed. In a case study, the train schedule in the double-track rail network of the Tehran–Mashhad railway of Iran has been compared with the golden point. Experimental results show that a proposed approach can suitably fit the problem considering important metrics with an improvement of %7.34 and %6.89 for the average passenger waiting time and the total number of train operation cycles, respectively.

Keywords Multi-objective optimization · Train schedule · Meta-heuristic · NSGA-II · Simulated annealing

✉ Seyyed Amir Asghari
asghari@khu.ac.ir

Massoud Seifpour
massoudseifpour@Gmail.com

Mostafa Ghobaei-Arani
mo.ghobaei@iau.ac.ir

[1] Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, Iran

[2] Electrical and Computer Engineering Department, Kharazmi University, Tehran, Iran

1 Introduction

The railway has always been considered by the people and supported by governments as an efficient transportation system. In addition to critical features such as reliability, comfort, and safety, its essential feature of economic fuel consumption, which also significantly reduces greenhouse gas emissions, has made it a green transportation system. Due to the increasing demand for travel and freight by rail network, having efficient computer-based models in response to challenges and complexities such as train scheduling or rescheduling, passenger waiting time, and arrival time is inevitable.

Improving scheduling and providing passenger satisfaction with increasing travel demand in different seasons conflicts with the available resource limitation. On the one hand, increasing the number of trains leads to decreasing the waiting time and increasing passenger satisfaction. In contrast, operating costs have increased, and traffic management, locomotive, and crew schedules have become more complex. Train schedule depends on many conflicting factors and parameters, such as resource constrains, continuous parameter changes, and various passenger requests. Different models, such as MILP, are utilized to solve these conflicts. [1, 2]. Train scheduling computation usually requires several hours or even a few days of utilizing computational units, high-performance hardware resources, and high-processing units. Due to contingency rescheduling and time constrains, train scheduling with meta-heuristic algorithms is more effective [3].

NSGA-II is generally utilized to solve various scientific problems such as non-classified multi-objective optimization or multi-dominance characteristics with five features, fast crowded distance, well diversity, well convergence, fast non-dominated rank, and exclusive observant approach [4]. As a stochastic algorithm, NSGA-II comprises six steps: initialization, sorting, crowding distance, selection, genetic operators, and recombination to transfer the population to the next generation. The population is processed and handed over to the next step at each step to get enough maturity for the final mutation [5]. Despite its numerous advantages, NSGA-II suffers from some drawbacks, such as a non-dominated sorting process on 2N size, the probability of convergence constrain due to crowded comparison, the possibility of restrictions on the uniform distribution of space in some problems, and the possibility of producing a similar population. Creating a new generation makes identification much more complicated [6, 7].

The SA algorithm is a meta-heuristic method that provides near-optimal solutions for big combined optimization problems by sampling high-temperature solids and gradually decreasing the temperature [8]. In searching the solution space, SA tries not to get stuck in the trap of local optimum points with a neighborhood function and to find new solutions close to global optimum points [9]. The SA algorithm in search of optimal points generates and evaluates a random point in the neighborhood of the current response and the specified temperature T. Repeating the process of reducing T and searching for new points continues until the desired result is achieved. The probability of accepting new searched answers will be generous at high temperatures and greedy as the temperature decreases.

The advantages of the SA algorithm are the possibility of theoretical proof and ease of implementation with appropriate efficiency [10]. Despite the numerous advantages of the SA algorithm, this algorithm also has some disadvantages, including the low convergence speed [11–13].

Based on the abovementioned discussion, each NSGA-II and SA algorithm has some disadvantages. To tackle these disadvantages, the mentioned algorithms are combined as a hybrid form to solve the problem, and to utilize a random guided hybrid, search method, a homogeneous set of more optimal solutions is created on the Pareto front [14, 15]. Each chromosome of the population calculates its value based on a fitness function, and using mutation and crossover actions, new solutions are improved [16]. One of the advantages of hybrid optimization is finding new solutions that each algorithm alone cannot find near the global optimum. Also, creating a balance between exploration and exploitation in the best possible way is another advantage of hybrid optimization in terms of accuracy and execution time [17, 18].

Multi-object Evolutionary Algorithm with Hybrid Sampling Strategy (MOEA-HSS) is a multi-objective meta-heuristic algorithm with a hybrid elitist sampling strategy that takes selective solutions using the vector genetic method from the edge vector evaluated genetic algorithm (VEGA) and central area Pareto dominating and dominated relationship-based fitness function (PDDR_FF) of Pareto front. Sampling for each objective function is done at the maximum size of half of the original population, utterly independent of other objective functions, and based on a proportionality function. They converge toward the central area of the edge of the Pareto front [4, 6].

Multi-object Evolutionary Algorithm Based on Decomposition (MOEA/D) has a pleasing efficiency in solving the multi-objective optimization problem. This algorithm is more effective than other evolutionary algorithms in the dominant Pareto space for solving a large optimization problem. One of the advantages of decomposing the multi-objective optimization problem into a few simple scalar optimization problems and computational complexity is less than evolutionary algorithms. However, one of the disadvantages of this algorithm is the low speed of convergence and the lack of population diversity compared to other evolutionary algorithms [19, 20].

A new metaheuristic algorithm is recently being developed and gained attention in the research community. The Multi-Objective Grey Wolf Optimizer (MOGWO) is an algorithm for optimization problems with multiple objectives [21]. The MOGWO is the developed version of the Grey Wolf Optimizer (GWO) algorithm [22]. The algorithm simulates the leadership hierarchic structure and hunting manner of grey wolves in nature. Four types of grey wolves, namely alpha, beta, delta, and omega, are utilized for simulating the leadership hierarchical structure. The optimization in three steps is implemented including search hunting, encircling, and attacking prey [23].

A fixed-size archive is integrated with the GWO algorithm for storing and retrieving non-dominated solutions. The optimal responses are compared with the individuals' archives in the iteration process, and then, the non-dominant answer is replaced by a random individual from the archives [23]. The MOGWO algorithm can present very competitive results compared to other meta-heuristics algorithms for solving multi-objective problems [21, 24].

The NSGA-II and SA are still widely studied and used in the literature on optimization problems because they have different strengths and limitations. NSGA-II has an excellent performance at exploring the search space and finding solutions quickly [25]. On the other hand, SA is good at escaping the trap of local optimum [8]. Hybridization of NSGA-II and SA is more efficient and effective by overcoming their limitations, combining their strengths, and using them simultaneously to solve optimization problems [16]. This hybridization has various advantages including:

1. Improved global optimization: Both NSGA-II and SA algorithms are performing well in finding global optima. Simultaneous use of both algorithms increases the chance of global optimization.
2. Enhanced exploration and exploitation: NSGA-II is good at exploring the search space, while SA is excellent at exploiting the search space hybridization can lead to a better balance between exploration and exploitation.
3. Faster convergence: NSGA-II is known to converge slowly toward the optimal solution, while SA converges faster. The hybrid approach has faster convergence and better performance.
4. NSGA-II and SA can be customized for specific optimization problems [26–28].

In this article, to improve the quality of the answers to the multi-objective optimization problem of train schedules, the NSGA-II algorithm is hybrid with the SA operator as the main framework, and multi-crossover and multi-mutation methods are used to produce a faster and better next-generation. To prove the superiority of NSGA-II-SA solutions, three other multi-objective optimization algorithms, namely NSGA-II, MOEA-HSS, and MOEA/D, are implemented, and an experiment and Pareto front analysis is performed. In another approach, by selecting one solution from NSGA-II-SA, a comparison will be made with the real train scheduling data.

This study's contributions are listed as the following:

- A developed mathematical scheduling model is proposed to reduce the waiting time of passengers and the number of train scheduling cycles under conditions of uncertainty.
- The meta-heuristic NSGA_II algorithm is a hybrid with an SA operator, which has a good convergence speed, suitable variety, and is entirely uniform.
- Four algorithms solve the proposed model, and the Pareto front solutions are analyzed with several metrics.
- In a case study, using the decomposition approach and selecting the gold point, the superiority of the hybrid NSGA-II-SA algorithm response is proven in terms of train scheduling with real data.

The rest of this article is written as follows. Section 2 refers to recently published papers in train scheduling and meta-heuristic multi-objective algorithms. Section 3 refers to the presentation of a mathematical model to formulate a multi-objective function for train scheduling. Section 4 refers to the detailed description of the hybrid NSGA-II algorithm with the SA operator for the proposed model. Section 5 discusses the numerical experiments, and the comparison of the results of four algorithms in the Pareto front space and the comparison of the results of the NSGA-II-SA algorithm are compared with the real values of railway scheduling. Finally, Sect. 7 refers to conclusions and suggestions for future studies in train scheduling by meta-heuristic algorithms.

2 Literature overview

Serval studies in research and academic centers regarding improving train scheduling, passenger satisfaction, and reducing operational costs are being carried out. This section examines the optimization problems articles and the use of meta-heuristic algorithms in transportation, especially rail, in recent years.

The paper [29] presented a multi-objective optimization of train schedules, passenger traffic control, and stop patterns. A mixed integer nonlinear programming (MINLP) model is applied to balance operating costs and service level performance. The robust optimization model is used to balance solution robustness and scenario-based model robustness. Finally, this research demonstrates the accuracy of the proposed model with numerical experiments according to the Beijing metro line data set.

The paper [30] investigated train scheduling problems considering energy consumption efficiency and operation cycle on urban railways. The aim is to solve the problem of the train operating cycle, the use of regenerative braking energy, and the energy saving of the traction motor. This paper utilized a particle swarm algorithm for optimizing train scheduling and operating cycle. Finally, the numerical experiments indicate reduced operating costs compared with the initial train scheduling on Guangzhou Subway Line 9.

The paper [31] presented a novel approach to optimizing train scheduling and rolling stock operating cycles based on dynamic passenger demand and transport capacity. Two integer linear programming models have optimized for total cost and time–space by flexible coupling/splitting activities. A diving heuristic algorithm optimizes train scheduling and rolling stock operating cycle. Finally, this research demonstrates the efficiency of the proposed methods and algorithm with numerical experiments compatible with the Shanghai Subway line real data set.

The paper [32] the train timetable developed based on passenger transport priority and minimum cargo transfer delay using an objective function. An analysis of multi-modal scheduling solutions and demand setting investigated in a simulation. The results demonstrate that obtained the train schedule increases service quality and reduces freight demand.

The paper [33] developed an optimization multi-module model by utilizing the accessible route in the primary train schedule. This model solves a repeatable framework of the Lagrangian relaxation by the branch-and-bound algorithm. It aims to generate the train schedule by resolving constrains and daily operational conflicts. Finally, this paper shows the validity of the proposed approach with the INFORMS RAS 2016 (Railway Applications Section) real dataset.

The paper [34] investigated multi-objective optimization problems of train operation with station classification and passenger arrival rate. The multi-layer programming model is provided with train stop optimization to minimize operation costs and travel costs. The proposed model is implemented with a meta-heuristic SA algorithm and sequential average method. Finally, this research discusses the effectiveness of the results using sensitivity analysis on the operation cost and the travel cost reduction with actual data of Shanghai Metro.

The paper [35] investigated the problem of optimizing train scheduling based on dynamic passenger demand on the double-track railway in Turkey, and a linear programming model has been formulated to minimize the average passenger waiting time. The model investigates resource limitations, train capacity, and fleet size of passengers remaining from the first train. Finally, this research discusses the sensitivity analysis between passenger satisfaction and operating costs by comparing railway traffic scheduling.

The paper [1] presented a multi-objective optimization of train scheduling and a robust stop schedule by uncertainty. The aim is to find in order robust solutions to improve passenger satisfaction under uncertainty, and a MILP model has been provided based on the light robustness technique for passengers in each pair of origin and destination stations for each train. Finally, this research discusses the effectiveness of the nominal passenger demand reduction with real-life data of the Wuhan–Guangzhou high-speed railway.

The paper [16] studied minimizing the total transportation costs, the allocation of resources, and risks of disruption in the hub of the Turkish postal transport network. This research combines meta-heuristic SA and Tabu Search (TS) [36] algorithms to improve the solution accuracy and computational time. The result is compared with other meta-heuristic algorithms.

The paper [6] presented a meta-heuristic algorithm by parallel processing with hybrid sampling and mutation-based learning for computation speed improvement. A multi-objective optimization model is solved with the MOEA-HSS and several meta-heuristic algorithms. The model aims to minimize the average passenger waiting time and reduce the average train circulation. This research shows that the MOEA-HSS algorithm needs less average execution time and outperforms the NSGA-II algorithm, and Power Pareto Evolutionary Algorithm (SPEA2) [37].

The paper [38] studied the scheduling problem of uncertain processing with a discrete scenario set. This research combines the NSGA-II algorithm and the SA algorithm to minimize the average execution time and the execution time of the worst under scenarios. The studied results show the developed algorithm performs better than the other four meta-heuristic algorithms defined in the paper. Table 1 includes the related works in optimization to transportation, problem model, model type, objective function, and problem-solving approach.

Table 1 The related works in the optimization to solve transport network and solution approach

References	Result	Problem model	Problem type	Model type	Objective function	Solution approach
Hu et al. [29]	Train scheduling	Collaborative optimization	Uncertainty	MINLP	Service level & Operating cost	LR
Zhou et al. [30]	Train scheduling & Circulation planning	Collaborative optimization	Certainty	MINLP	Operation cost & rolling stocks energy	particle swarm
Pan and Yang [31]	Train timetable & Rolling stock Plan	Optimization	Certainty	MILP	Total generalized cost & sub-problems	Diving heuristic
Hörsting [32]	Train schedule	Optimization	Stochastic	MILP	Passenger transport & Cargo delay	Simulation model
Wang et al. [33]	Train scheduling	Optimization	Certainty	STN	Space–time network & free routing	LR & BB
Bucak and Demirel [35]	Timetable	Optimization	Uncertainty	NLP	Average passenger waiting	Branch & bound GAMS
Sangaiah et al. [16]	Process time	Model presentation	Certainty	NLP	Transportation cost	Meta-heuristic SA & TS
Cacchiani et al. [1]	Timetable	Multi-objective optimization	Stochastic	MILP	Timetable & Stop Planning patterns	GAMS
Nitisiri et al. [6]	Timetable & Process time	Multi-objective optimization	Certainty	NLP	Train cycle& Passenger waiting	Meta-heuristic MOEA-HSS-LM
Li et al. [39]	Time table	Multi-objective optimization	Certainty	MILP	Total travel time & disutility	Meat-heuristic SA
Zhang et al. [40]	Time table	optimization	Certainty	MINLP	Total travel time	Adaptive lager Neighborhood Search
Wang et al. [38]	Scheduling	Bi-criteria robust	Uncertainly fuzzy	MILP	Mean make span & Worst-case make span	Meta-heuristic NSGA-II & SA
Wang et al. [41]	Traffic management	Multi-objective optimization	Certainty	MINLP	Train scheduling & rolling stock	Iterative
This study	Time table	Multi-objective optimization	Stochastic	MILP	Average waiting time & Number of the Operating cycles	Meat-heuristic NSGA-II, MOEA/D, MOEA-HSS, NSGA-II-SA

*Symbols description Table 1 *NLP* Nonlinear programming, *MINLP* Mixed integer nonlinear programming, *MILP* Mixed integer linear programming, *BB* Branch-and-bound, *STN* Space–time network

3 Problem definition

In this section, mathematical formulas for the railways scheduling model, parameters, constrains, and variables in uncertainty conditions are presented [6]. In Fig. 1, a modeling diagram is illustrated for a double-track railways network, where the route has permission for a single direction. The network consists of 2S stations, with stations 1 to S for the departure route and stations S + 1 to 2S for the return path. Trains movement are designed not to overtake each other along the route or station, and a safe headway has guaranteed for two consecutive trains. The operating period P continues with trains' movement from the first station until the final station S and then comes back from the S + 1 station to the 2S station. As a result, they complete an operation cycle c.

Other assumptions are as follows: (1) The capacity of the train and the station is not taken into consideration; (2) no passenger is left behind at the train station; (3) only one train stops at each station simultaneously; and (4) scheduling is presented for one period. Table 2 describes the indicators, parameters, and decision variables of the proposed model.

The objective functions (1) and (2) are defined to minimize the average passenger waiting time at the stations and the total number of train operation cycles during an active period. Increasing the number of trains operation can enhance passenger satisfaction but lead to higher train operating costs. Based on the above description, the objective functions and problem constrains are stated in constrains (3)–(14) as follows:

The objective function (1) corresponds to the minimized average waiting time of the passenger, p_{ui} is a parameter including the probability for passenger arrival rate at station i under scenario u. The $a_{ui}(p)$ is a parameter showing the number of passengers arriving at station i during period p under scenario u. The $t_{ui}^{PA}(p)$ is another parameter to determine the passenger arrival time at Station i under scenario u. The t_{ucki}^{D} is a decision variable that includes the arrival time $\mathrm{t}_{\mathrm{ucki}}^{\mathrm{A}}$, stop time at the station $\mathrm{t}_{\mathrm{ucki}}^{\mathrm{S}}$, and train turnaround time $t_u{}^T$. The turnaround time is only considered at the first and last stations.

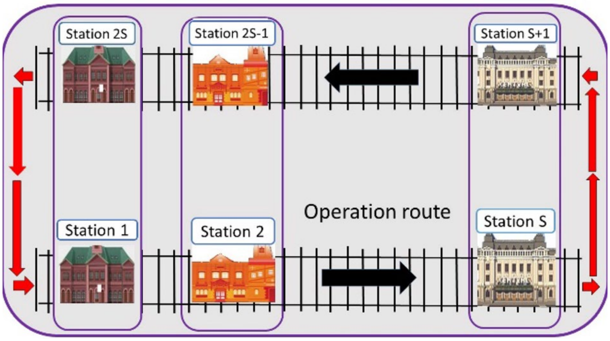

Fig. 1 Schematic double-track railway network

Table 2 Indexes, parameters, and decision parameters

Indexes	Description
I	Station index, $i = \{1,2,3,\ldots,2S\}$
K	Train index, $k = \{1,2,3,\ldots,N\}$
C	Operating cycle for the train, $c = \{ 1,2,3,\ldots,C\}$
P	Operating period, $p \in \left[P_{\text{start}}, P_{\text{end}}\right]$
U	Scenario $u = \{$spring, summer, fall, winter$\}$
Problem parameters	**Description**
$a_{iu}(p)$	The number of passengers arriving at station i during period p under scenario u
$t_{iu}^{PA}(p)$	Passenger arrival time at station i period p under scenario u
$t_u{}^T$	A turnaround time needed for a train at the first and last stations under scenario u
$h_u{}^{min}, h_u{}^{max}$	The maximum and minimum headway between two consecutive trains under scenario u
$t_{iu}^{Rmin}, t_{iu}^{Rmax}$	The maximum and minimum traveling time between stations i and i + 1 under scenario u
p_u	probabilities for the passenger arrival rate at station *i* under scenario u
M	A large positive number
Decision variable	**Description**
t_{ckiu}^{A}	The arrival time of train k in cycle *c* at station *i* under scenario *u*
t_{ckiu}^{D}	The departure time of train k in cycle *c* from station *i* under scenario *u*
t_{ckiu}^{R}	The travel time of train k in cycle *c* between station *i* and *i* + 1 under scenario *u*
t_{ckiu}^{S}	The dwelling time of train k in cycle *c* at station *i* under scenario *u*
t_{ckiu}^{W}	The waiting time for passenger boarding train k in cycle *c* at station *i* under scenario *u*
$x_{ucki}(p)$	= 1, if train k in cycle *c* arrives at station *i* early enough for passenger at period *p* under scenario *u* 0 =, otherwise

$$\min Z_1 = \sum_{p=P_{start}}^{P_{end}} \sum_{u=1}^{U} \sum_{c=1}^{C} \sum_{k=1}^{N} \sum_{i=1}^{2S} (p_u a_{ui}(p)(t_{ucki}^{D} - t_{ui}^{PA}(p))) / \sum_{u=1}^{U} \sum_{i=1}^{2S} p_u a_{ui}(p) \tag{1}$$

The objective function (2) corresponds to minimizing the total number of operating cycles during the operating period. The $\mathrm{x}_{\mathrm{ucki}}(\mathrm{p})$ is a binary decision variable to control the possibility of passengers getting on the trains. If $\mathrm{x}_{\mathrm{ucki}}(\mathrm{p}) = 1$, it means that the passenger has enough time to get on the train.

$$Min\, Z_2 = \sum_{p=P_{start}}^{P_{end}} \sum_{u=1}^{U} \sum_{k=1}^{N} \sum_{i=1}^{2S} p_u x_{uki}(p) \tag{2}$$

Constrain (3) describes the train's arrival time to station $i + 1$, which includes the departure time $\mathrm{t}_{\mathrm{ucki}}^{\mathrm{D}}$ from the previous station i and the travel time $\mathrm{t}_{\mathrm{ucki}}^{\mathrm{R}}$ between two stations i, and $i + 1$.

$$\mathrm{t}^{\mathrm{A}}_{\mathrm{ucki}+1} = \mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}} + \mathrm{t}^{\mathrm{R}}_{\mathrm{ucki}}, \quad \forall\, \mathrm{c}, \mathrm{k}, \mathrm{i} \ \geq 2\,, \mathrm{u} \tag{3}$$

Constrain (4) shows the departure time of the train at station *i*, which includes the arrival time $\mathrm{t}^{\mathrm{A}}_{\mathrm{ucki}}$ and stopping time $\mathrm{t}^{\mathrm{S}}_{\mathrm{ucki}}$ of the train at station *i* the turnaround time $\mathrm{t_u}^{\mathrm{T}}$ is only considered at the first and last stations.

$$\mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}} = \mathrm{t}^{\mathrm{A}}_{\mathrm{ucki}} + \mathrm{t}^{\mathrm{S}}_{\mathrm{ucki}} + \mathrm{t}^{\mathrm{T}}_{\mathrm{u}}, \quad \forall\, \mathrm{u}, \mathrm{c}, \mathrm{k}, \mathrm{i} \tag{4}$$

Constrain (5) is defined to control of the train traveling time between station *i* and station $i+1$. The $\mathrm{t}^{\mathrm{Rmax}}_{\mathrm{ui}}$ and $\mathrm{t}^{\mathrm{Rmin}}_{\mathrm{ui}}$ are parameters to control the maximum and minimum traveling time between stations *i* and $i+1$.

$$\mathrm{t}^{\mathrm{Rmin}}_{\mathrm{iu}} \leq \mathrm{t}^{\mathrm{R}}_{\mathrm{ucki}} \leq \mathrm{t}^{\mathrm{Rmax}}_{\mathrm{iu}}, \quad \forall\, \mathrm{u}, \mathrm{c}, \mathrm{k}, \mathrm{i} \tag{5}$$

Constrains (6) and (7) describe safe headway time for two consecutive trains. The $h^{\max}_{u}$ and $h^{\min}_{u}$ are parameters for controlling the maximum and minimum headway between two trains.

$$\mathrm{h}^{\min}_{u} \leq \mathrm{t}^{\mathrm{D}}_{\mathrm{ck}+1,\mathrm{iu}} - \mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}} \leq \mathrm{h}^{\max}_{u}, \quad \forall\, \mathrm{u}, \mathrm{c}, \mathrm{k}, \mathrm{i} \tag{6}$$

$$\mathrm{h}^{\min}_{\mathrm{u}} \leq \mathrm{t}^{\mathrm{A}}_{\mathrm{uck}+1,\mathrm{i}} - \mathrm{t}^{\mathrm{A}}_{\mathrm{ucki}} \leq \mathrm{h}^{\max}_{\mathrm{u}}, \quad \forall\, \mathrm{u}, \mathrm{c}, \mathrm{k}, \mathrm{i} \tag{7}$$

Constrain (8) describes passenger waiting time $\mathrm{t}^{\mathrm{W}}_{\mathrm{ucki}}$ at the station. It must always be greater or equal to the train departure time. *M* is a parameter with a high positive value. When the train departure time $\mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}}$ is earlier than the passenger arrival time $t^{PA}_{i}(p)$, in this case, the passenger waiting time $\mathrm{t}^{\mathrm{W}}_{\mathrm{ucki}}$ will be zero.

$$\mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}} - \mathrm{t}^{\mathrm{PA}}_{\mathrm{iu}}(\mathrm{p}) - Mx_{ucki}(p) \leq \mathrm{t}^{\mathrm{W}}_{\mathrm{ucki}}, \quad \forall\, \mathrm{c}, \mathrm{k}, \mathrm{i}, \mathrm{u}, p \in \left[\mathrm{P}_{\mathrm{start}}, \mathrm{P}_{\mathrm{end}}\right] \tag{8}$$

Constrain (9) controls the train turnaround at the station.

$$\mathrm{t}^{\mathrm{A}}_{\mathrm{ck},\,\mathrm{S}+1} - \mathrm{t}^{\mathrm{D}}_{\mathrm{ck}+1,\mathrm{S}} \geq 0, \quad \forall\, \mathrm{c}, \mathrm{k} \tag{9}$$

Constrains (10) and (11) are defined to impose the value of positive variables.

$$\mathrm{t}^{\mathrm{A}}_{\mathrm{ucki}}, \mathrm{t}^{\mathrm{D}}_{\mathrm{ucki}}, \mathrm{t}^{\mathrm{R}}_{\mathrm{ucki}}, \mathrm{t}^{\mathrm{S}}_{\mathrm{ucki}}, \mathrm{t}^{\mathrm{W}}_{\mathrm{ucki}} \geq 0, \quad \forall\, \mathrm{u}, \mathrm{c}, \mathrm{k}, \mathrm{i} \tag{10}$$

$$\mathrm{x}_{\mathrm{ucki}}(\mathrm{p}) \in \{0, 1\}, \quad \forall\, \mathrm{c}, \mathrm{k}, \mathrm{i}, \mathrm{p} \in \left[\mathrm{P}_{\mathrm{start}}, \mathrm{P}_{\mathrm{end}}\right] \tag{11}$$

Constrains (12) and (13) calculate the traveling time between stations and headway between two consecutive trains.

$$t^{R}_{ucki} = t^{Rmin}_{iu} + \left(t^{Rmax}_{iu} - t^{Rmin}_{iu}\right), \quad \forall\, u, c, k, i \tag{12}$$

$$h_c = h^{min}_{u} + \left(h^{max}_{u} - h^{min}_{u}\right), \quad \forall\, u, c, k, i = \{1, S+1, 2S+1, \left(C_{max}\right)S+1\} \tag{13}$$

In constrain (14), the P_U is a parameter to represent the probability of the number of passengers entering stations in each season. It is calculated using the Poisson distribution with λ_u as the mean number of events.

$$P_u \sim Poisson(\lambda_u)\ ,\forall\ u = \{spring, summer, fall, winter\} \tag{14}$$

4 Proposed solution

In recent years, meta-heuristic algorithms and one fitness function have been given much attention in solving engineering optimization problems due to their impressive efficiency [42]. This section describes the steps of developing the classic NSGA-II algorithm using multiple crossover and mutation operators, then the hybrid of NSGA-II with the SA operator. Finally, the proposed train scheduling model has been implemented using the developed hybrid algorithm.

4.1 Chromosome and population initialization

In algorithm 1, a random initial population is created and initialized based on input variables, namely the number of stations S, train cycles C, and scenario U (Fig. 2). First, the individual structure is created based on the problem's parameters and index (line 1). Then, the population matrix of the problem is produced in the required number based on U, C, and S (line 2). The main body includes three nested loops that initialize the population matrix (lines 3–5). Each individual will randomly have a value of 0 or 1 in the population matrix (line 6) [6].

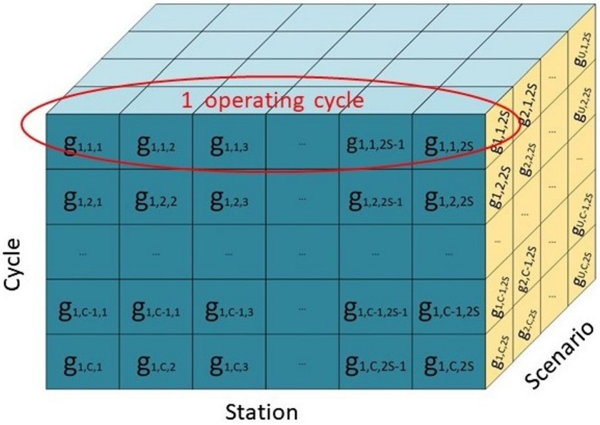

Fig. 2 Chromosome representation

Algorithm1. *Create population routine*

```
Input: number of station S, number of cycle C, number of scenario U, number of population npop
Output: randomly population P
1:    Create empty individual chromosome structure P
2:    Create population matrix P    /* to the number of population
3:    for u=1 to U do
4:      for c=1 to C do
5:        for s =1 to 2*S do
6:          copy random [0,1]  to $g_{ucs}$    /*  of population npop
7:        end for s
8:      end for c
9:    end for u
10:   end    /*create population matrix $P_{ucs}$
```

4.2 Crossover process

The crossover operator is the critical leading force in finding new and optimal solutions and offers excellent solutions sometimes. Uniform and reduced surrogate operators are simultaneously utilized to improve the creation of the next generation [43].

4.2.1 Uniform crossover operator

As illustrated in Fig. 3, the classical crossover operator usually applies to chromosomes' predetermined location (s) in one or two fixed positions. In algorithm 2, the uniform operator does not need to determine the intersection point(s) in advance. The crossover process extends equally to the entire chromosome length. In this method, one binary string is randomly created with the same size as the chromosome (line 3). The exchange process of parents' genes will be performed to produce two offspring depending on a binary string value (lines 6, 7) [43].

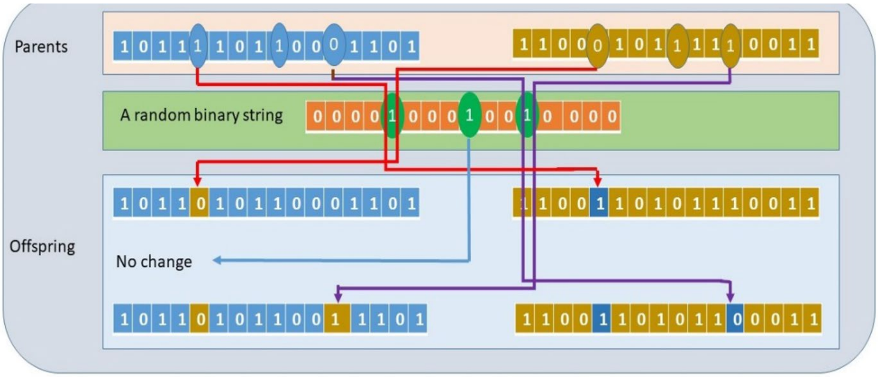

Fig. 3 Uniform crossover operator

Algorithm 2. *Uniform crossover routine*

```
Input: set of two parents P link parent1, parent2
Output: set of two offspring Q link offspring1, offspring2
    l: Binary string, b: binary
    offspring1, offspring2, l= null;
    l= create Randomly Binary String equal size parent1
    for i = 1 to size l do
        b = Position String $l_i$
        if b = 1 & (gene i of parent1 and parent2 is not equal)
            update offspring1, offspring2 with change gene i of parent1, parent2
        end if
    end for i
end
```

4.2.2 Reduced surrogate crossover operator

In Fig. 4, the crossover operator may create a poor-quality population or analogous in some situations. The reduced surrogate operator will likely improve the production of the poor-quality population to an acceptable extent.

In algorithm 3, operations are avoided on chromosomes with similar genes. As a result, it will reduce the number of unnecessary executions of the fitness function and execution time. The parents' genes have been compared to a similar position, and differences are stored in a string (line 3). If the parents have at least one different gene, one place will be chosen randomly (lines 4, 5). The exchanging of genes will be done to produce two new offspring in the specified position (lines 6) [43].

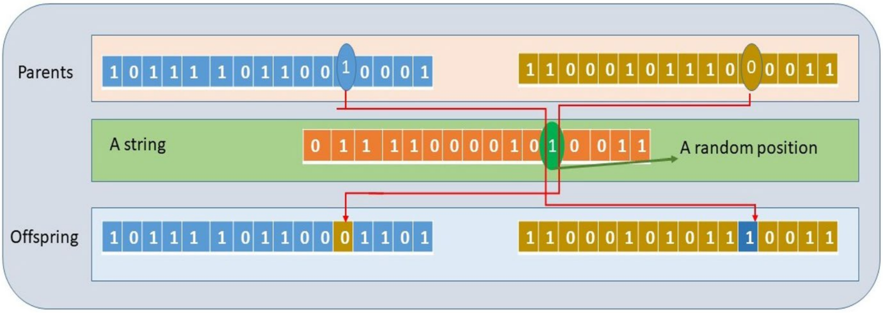

Fig. 4 Reduced surrogate crossover operator

Algorithm 3. *Reduced surrogate crossover routine*

```
Input: the set of two parents P link parent1, parent2
Output: set of two offspring Q link offspring1, offspring2
l: binary string
offspring1, offspring2, l = null;
l = Save Deferent Position gene parent1, parent2
if l ≠ null
    k = select random a position of l
    update offspring1, offspring2 of Q with change gene k of parent1, parent2
end if
end
```

4.2.3 Multi-crossover operator main body

In algorithm 4, the multi-crossover operators' routine is provided as the replace surrogate and uniform operator. The crossover operator can be applied to a population or a percentage of the entire population according to the problem condition (lines 2, 3). The two parents have been randomly selected among the population (line 4). The reduced surrogate and uniform crossover routines are executed according to the value of a random binary number. This approach executes routines randomly and equally (lines 5–9).

Algorithm 4. *Multi crossover routine*

```
Input: population P, number of population npop
Output: offspring Q        /* include neighborhoods
1:   Q= empty individual chromosome structure
2:   pc= probability crossover* npop     /* crossover probability = 0.8
3:   for i = 1 to pc/2 do
4:       select random parent1, parent2 of P
5:       b= random [0, 1]
6:       if b= 0
7:           update Q by reduced surrogate crossover routine
8:       else
9:           update Q by uniform Crossover routine
10:  end for i
11:  end
```

4.3 Mutation process

The main effect of the mutation operator compared to the crossover operator is to preserve the diversity of the population and increase the probability of being close to the global optimum [43]. The probability of producing a diverse and elite population increases with the simultaneous use of several mutation operators with various characteristics. In the following, two replacement and insertion operators are presented to improve the production efficiency of the new generation [44].

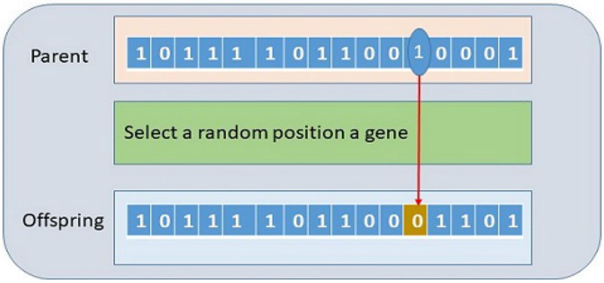

Fig. 5 Replacement mutation operator

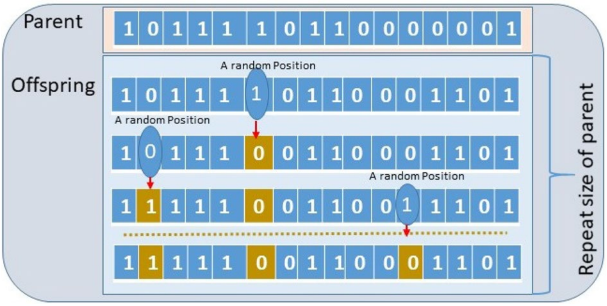

Fig. 6 Insertion mutation operator

4.3.1 Replacement mutation operator

In algorithm 5, the independent random replacement operator is presented to increase the offspring's diversity among the parents' population [44]. A gene from the parent is randomly selected during the replacement process (line 2). The operation of producing new offspring will be done by changing the value of the gene (line 3) (Fig. 5).

```
Algorithm 5. Replacement mutation routine
Input: an individual of population link parent1
Output: an individual of R link offspring1
1:      offspring1= parent1
2:      k= Select the random position of a gene of parent1
3:      change gene k of offspring1
4:      end
```

4.3.2 Insertion mutation operator

In algorithm 6, the insertion operator changes one or more genes from one chromosome to increase the ability to produce a new generation of offspring (Fig. 6). In the insertion process, several changeable genes are randomly selected from the parent chromosome based on the mutation rate and the size of the genes [44]. In the initialization, one offspring is placed equal to one parent (line 1). The number of parent genes is calculated to create the insertion process loop (lines 2, 3). A new offspring is created by randomly changing the gene position (Lines 4–5).

Algorithm 6. *Insertion mutation routine*

```
Input: an individual of population link parent1
Output: an individual of R link offspring1
1:     offspring1= parent1
2:     n= number of genes the parent1
3:     for i= 1 to n do
4:         k= Select the random position of a gene of offspring1
5:         change gene k of offspring1
6:     end for i
7:  end
```

4.3.3 Multi-mutation main body operator

In algorithm 7, a multi-mutation method with replacement and insertion operator is presented to improve mutation efficiency in offspring production. An empty structure is created to store the new neighbors (line 1). The offspring is produced according to the ratio of the population and the percent probability of mutation (lines 2, 3). The mutation operator is performed only on one individual (line 4). The reduced surrogate and uniform crossover routines are executed by a binary variable random and equally (lines 6–9).

Algorithm 7. *Multi mutation routine*

```
Input: population P, number of population npop
Output: offspring R   /* includes neighbors
1:     R= empty individual chromosome structure
2:     pm = probability mutation *npop    /* probability of mutation 0.2
3:     for i = 1 to pm do
4:         select randomly parent1 of P
5:         b= Random [0, 1]
6:         if b = 0
7:             update R by replacement mutation routine
8:         else
9:             update R by insertion mutation routine
10:    end for i
11:    end
```

4.4 Simulated annealing (SA) process

Algorithm 8 describes the competition between the parent population and the candidate neighbors. A chromosome of the candidate neighbors competes with one chromosome of the parent population based on the SA rule. The number of iterations of the main loop is equal to the size of the population (line 1). The chromosome of the candidate neighbors is selected with more competency than the parent chromosomes by competition (line 2), and the neighbor replaces the parent (line 3). Otherwise, the Δf is calculated as the difference between the value of the function neighbor and the parent (line 5). The candidate neighbor has a chance to replace the parent; if the value of $e^{-\Delta f}/_{T}$ is more prominent than a random number in the range {0, 1} (lines 6, 7), the candidate neighbor is selected and replaces the parent (line 8).

```
Algorithm 8. Simulated annealing rule routine
Input: population P, offspring S, number of population npop, temperature T
Output: next-generation population P
1:    for i = 1 to npop do
2:        if  $f(S_i) \geq f(P_i)$        /* offspring S_i better than parent p_i
3:            replace S_i with P_i
4:        else
5:            $\Delta f = f(S_i) - f(P_i)$
6:            p= random {0, 1}
7:            if p $\leq$ exponential $(-\Delta f/T)$     /* fortuity again to be select
8:                replace S_i with P_i
9:            end if
10:       end if
11:   end for i
12: end
```

4.5 Evaluated process for solving the railway scheduling

Figure 7 shows the train movement graph schematically with decision variables and problem parameters for traffic management and control. It is the primary management tool in the railway dispatch centers for train planning, rolling stock movement, train rescheduling accident management, and all event control. It can also help better understand algorithm 9.

In algorithm 9, the train timetable and functions Z_1 and Z_2 create for individuals under four scenarios. This algorithm presents train scheduling on each station for each chromosome by the limitations of constrains (3)–(14). The algorithm starts with four main nested loops by the number of scenarios, operating cycles, several trains, and stations (lines 1–4). The train travel time t^R_{ucki} is calculated as the difference between

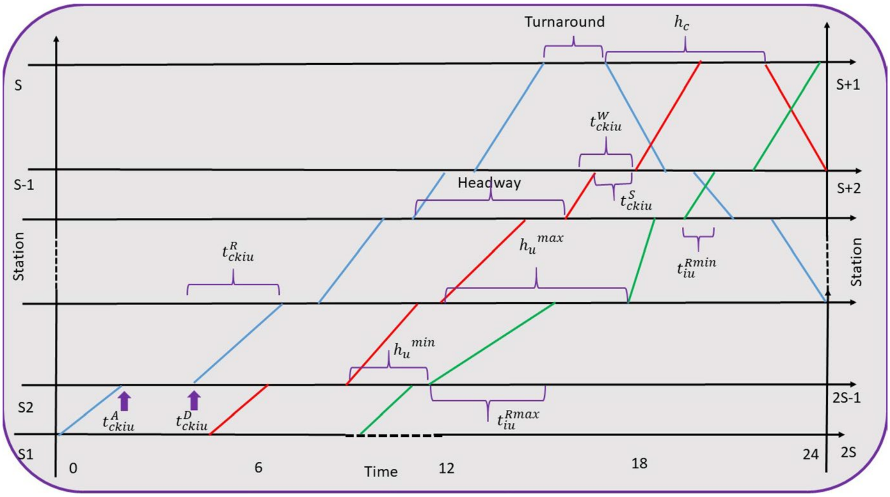

Fig. 7 An overview of the train schedule graph

max and min travel time and the train travel time between two stations S_i, and S_{i+1} (line 5 and used parameters $t_{ui}^{Rmax}, t_{ui}^{Rmin}$)). The headway turnaround time h_c has been calculated as the difference between the maximum and minimum headway parameters $h_u^{max} - h_u^{min}$ and minimum headway h_u^{min} between two consecutive trains (line 6). Suppose the train needs a turnaround, the train departure time t_{ckiu}^D is replaced with headway turnaround time h_c and train arrival time $t_{c-1.k.2Su}^A$ at the last station (lines 7, 8). Otherwise, the train arrival $t_{ck.i+1.u}^A$ and departure times t_{ckiu}^D are calculated at two consecutive stations S_i, and S_{i+1} (lines 10, 11). The train arrival time t_{ucki}^A and departure time t_{ucki}^D must have the minimum headway h_u^{min} between two consecutive trains to guarantee safety essentials (lines 13–16). If the passenger arrival time parameter $t_{ui}^{PA}(p)$ is longer than the train departure time t_{ucki}^D then calculate the value of function Z_1 (line 18–20 and used parameters$a_{ui}(p), p_u, M$). Calculation of the total average passenger waiting time is done in three steps. Firstly, the passengers' waiting times are calculated in line 21, secondly, the total number of passengers is calculated in line 29, and the value of the total average passengers' waiting times (Z_1) is calculated in line 32.

Algorithm 9. *Train arrival and departure time*

```
Input: population pop, problem data definition
Output: train arrival and departure time
        /* initialize temporary variables, NumP
1:  for u = 1 to 4 do
2:    for c = 1 to C do
3:      for k = 1 to N do
4:        for i =1 to 2S do
5:          $pop.t_{ucki}^R = t_{ui}^{Rmin} + pop.(t_{ui}^{Rmax} - t_{ju}^{Rmin})$ /* Calculate train travel time
6:          $h_c = h_u^{min} + pop.(h_u^{max} - h_u^{min});$   /* Headway turnaround time
7:          if (i mod 2S) = 1 then                 /* Check first or last station
8:            $pop.t_{ucki}^D = pop.t_{uc-1.k.2S}^A + h_c;$          /* Need turnaround
9:          else
10:           $pop.t_{uck.i+1}^A = t_{ucki}^D + t_{ucki}^R;$       /* Train arrival time
11:           $pop.t_{ucki}^D = pop.t_{ucki}^A + pop.t_{ucki}^S;$  /* Train departure  time
12:         end i
13:             if $pop.t_{uc,k-1.i+1}^A$ - $pop.t_{ucki}^A + h_u^{min} < 0$ then    /* Check safety train arrival time
14:               $pop.t_{ucki}^A = pop.t_{ucki}^A + (pop.t_{uc.k-1.i+1}^A - pop.t_{ucki}^A + h_u^{min});$
15:         if $pop.t_{uc,k-1.i+1}^D$ - $pop.t_{ucki}^D + h_u^{min} < 0$ then    /* Check safety train departure time
16:           $pop.t_{ucki}^D = pop.t_{ucki}^D + (pop.t_{uc.k-1.i+1}^D - pop.t_{ucki}^D + h_u^{min});$
17:         end if
18:           if $a_{iu}(p) > 0$ then                  /* The number of passengers arriving at the station
19:             if $pop.t_{ucki}^D - t_{ui}^{PA}(p) < t_{ckiu}^W - Mx_{ucki}(p) <$ then /*Passenger arrival time at the station must be early the train
20:               $pop.Z_2 = pop.Z_2 + p_u * x_{ucki}(p)$          /* Calculates the function Z2
21:             $pop.Z_1 = pop.Z_1 + p_u * a_{ui}(p)*(pop.t_{ucki}^D$ - $pop.t_{ui}^{PA})$  /* Calculate  Passengers' waiting times
22:           end if
23:       end for i
24:     end for k
25:   end for c
26: end for u
27:   for u = 1 to u
28:     for i= 1 to 2S
29:       $NumP = NumP + p_u * a_{ui}(p)$
30:     end i
31:   end u
32:   $pop.Z_1 = pop.Z_1 / NumP$                     /* Calculates the function Z1
33: output train timetable & functions Z1 and Z2
34: end
```

4.6 Hybrid NSGA-II and SA generate Pareto solution and railway timetable

Figure 8 illustrates the steps of the hybrid NSGA-II algorithm and SA operator, which provides a big schematic view for a good understanding of algorithm 10.

In algorithm 10, NSGA-II-SA will be initialized by the railway's data set, variables, and parameters in (3)–(14) (number of population, iteration, sub-iteration, temperature, Alpha …) (lines 1–3). The initial population is randomly created (line 4) and is evaluated by the fitness function (line 5). The non-dominated sorting and calculating crowding distance will be done on population by NSGA-II rule (lines 6–7, 16–17). The sorting operation will be done based on crowding distance; this approach ensures sorting the initial population based on ranking [11] (line 8, 18). With nested loops, the hybridization of NSGA-II and SA has been utilized to produce the next generation based on NSGA-II rules. The dominant generation is chosen with competition and more competency based on SA rules [45] (lines 9, 10). The multi-crossover and multi-mutation operators are utilized to generate for evaluating the population by the fitness function (lines 11–14). Offspring population S'(t) has merged with population multi-crossover *Q*(*t*) and multi-mutation *R*(*t*) (line 15). Superior neighbors *S*' (*t*) are inserted into *S*"(*t*) the size of the parent population *P*(*t*) (line 19).

The elite generation has been chosen among the neighbors *S*''(*t*) and the parent population *P*(*t*) by the SA rule (line 20) [38]. The dominant population will be stored as the best solution in the Pareto front (line 21). The temperature reduction is continued with a fixed and predetermined (Like Alpha) coefficient until the final

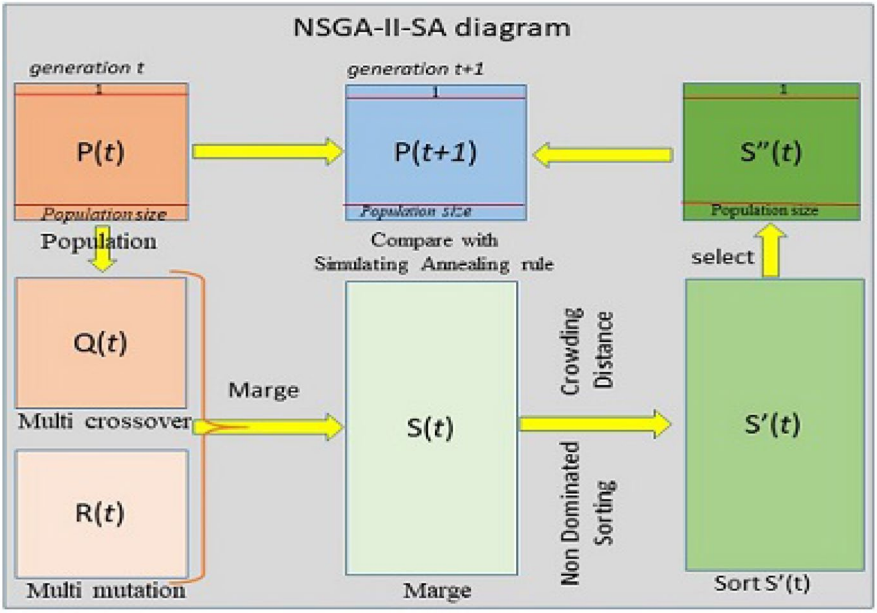

Fig. 8 Hybrid NSGA-II and SA

result is obtained (line 22). Finally, the timetable has been extracted from the Pareto front population for the subsequent analysis (lines 26).

Algorithm 10. *NSGA-II-SA generates Pareto solution and railway timetable*

```
Input: problem Railway dataset
Output: the best Pareto front, Train timetable
1:  set problem data definition
2:  set the number of population npop, population P, max iteration, sub iteration, crossover ratio, mutation ratio,
3:  set temperature T, Alpha
4:  initialization population P(t) by create population routine
5:  evaluate fitness function P(t) by train arrival and departure time routine
6:  non-dominated sorting P(t)
7:  calculate crowding distance P(t)
8:  sort P(t) first base on crowding distance and second base on rank
9:  for t =1 to max iteration do
10:     for subt =1 to sub iteration do
11:         create neighbors Q(t) by multi crossover routine
12:         evaluate fitness function Q(t) by train arrival and departure time routine
13:         create neighbors R(t) by multi mutation routine
14:         evaluate fitness function R(t) by train arrival and departure time routine
15:         marge offspring's S(t) form Q(t) and R(t)
16:         non-dominate sorting S′(t)
17:         calculate crowding distance S′(t)
18:         sort S′(t) first base on crowding distance and second base on rank
19:         insert to S′′(t) form S'(t) from 1 to number of population
20:         select P(t) from compare P(t) and S′′(t) with Simulated annealing rule routine
21:     end for t   /* loop sub iteration
22:     store the best Pareto solution
23:     T =Alpha*T
24: end for subt   /* loop iteration
25: calculate Euclidean distances of the Pareto front population from the ideal point (0, 0)
26: Extract the timetable of the Pareto front
```

5 Discussion

In this section, the proposed model has been implemented by numerical calculation and evaluated and validated.

This model is implemented and executed with the developed NSGA-II-SA algorithm and four algorithms NSGA-II, MOEA-HSS, MOEA/D, and MOGWO.

Several measures are discussed to evaluate the efficiency of the algorithms:

(1) Three prominent metrics are utilized to evaluate the performance of the algorithms (discussed in 5.1).
(2) The minimum distance of the Pareto front set is utilized to choose the gold point from the ideal point in Euclidean two-dimensional distances (discussed in 5.2).
(3) The compromised distance of the Pareto front set is utilized to analyze the minimum distance from the ideal point in Euclidean n-dimensional distances (discussed in 5.4).
(4) The proposed algorithm has evaluated the improvement of real train operations. The main target is to evaluate the average passenger waiting time and the total

number of train operations on one cycle in line Tehran–Mashhad of Iran (discussed in 6).

All algorithms' execution parameters are mentioned in Table 3. Noteworthy, these algorithms with different conditions and parameters are compared with each other.

The Taguchi method is one of the best statistical methodologies, which uses finding the minimum number of experiments to tune the parameters of the algorithms [46]. It is utilized to find the optimum values of the algorithm's parameters mentioned in Table 3, which is implemented by Minitab software V16.2.4.4. Two approaches are provided to analyze the experimental results and select the most appropriate parameters: 1. standard method, which calculates the effect of factors, and variance analysis and 2. Signal-to-noise-ratio (SNR), which calculates dispersion relative to a specific value [47]. In this article, the standard method is utilized to analyze the experiments.

5.1 Algorithm evaluation by metrics

In the research communities, there is no comprehensive agreement about the metrics for evaluating multi-objective optimization algorithms [48]. Researchers are more inclined to metrics such as Diversity, Spacing, Mean Ideal Distance, and Pareto front [49]. Convergence speed and exploration–exploitation balance are critical to a role in local optimum points search [50]. To maintain exploration efficiency in hybrid algorithms, the widely utilized number of neighborhoods is five [38].

Table 4 shows the statistical results of several Pareto fronts, metrics of diversity, spacing, and MID. The NSGA-II-SA, MOEA-D, MOEA_HSS, and MOGWO hybrid algorithms improve the diversity and convergence more than the NSGA-II algorithm.

Diversity Diversity shows the feasible solutions in the Pareto front, which is diversity is an essential factor for the solution search in the Pareto front. Diversity allows the genetic algorithms to explore multiple regions of the solution space and avoid getting stuck in the local optimum. The genetic algorithm may converge too quickly to suboptimal solutions with a population that has low diversity [51]. The NSGA-II has the low performance of other algorithms by a value of 0.2879. The MOEA-D, MOEA-HSS, and MOGWA have values of 0.4987, 0.4361, and 0.4797 in the same range, respectively. The NSGA-II-SA has a high diversity with a value of 0.5341. The NSGA-II-SA has more diversity to generate elite offspring and better convergence than other algorithms.

$$\begin{aligned}\mathbf{Diversity}_{\mathrm{NSGA-II}} &< \mathbf{Diversity}_{\mathrm{MOEA-HSS}} < \mathbf{Diversity}_{\mathrm{MOGWA}}\\ &< \mathbf{Diversity}_{\mathrm{MOEA-D}} < \mathbf{Diversity}_{\mathrm{NSGA-II-SA}}\end{aligned}$$

Spacing The spacing metric is an excellent indicator to evaluate the approximation uniformity and the distribution quality of solutions in the Pareto optimal set. The distance indicator measures the degree of deviation toward getting stuck in local optima. An algorithm with a lower spacing metric has a better performance than others. [49].According to the results of the three algorithms, NSGA-II, MODEA-D,

Table 3 List of algorithm parameters

Population size	25	Sub-iteration*	5
Max iterations	50	Temperature*	2.5e^{6}
Crossover ratio	0.8	Number of neighborhoods*	5
Mutation ratio	0.2	Alpha*	0.9

*SA algorithm parameters

Table 4 the algorithm evaluation metrics

	Pareto front set	Diversity	Spacing	MID (0,0)
NSGA-II	44	0.2879	0.6658	5.7623
MOEA-D	67	0.4987	0.6234	3.2476
MOEA-HSS	64	0.4361	0.7872	3.1181
NSGA-II-SA	59	0.5341	0.5927	2.8321
MOGWO	66	0.4797	0.6432	3.0265

and MOGWA have values of 0.6658, 0.6234, and 0.6432 in a similar range, respectively. They have indicated almost the same performance. The NSGA-II-SA has the highest spacing metric of 0.5927 the MOEA-HSS has the least amount spacing metric of 0.7872, which have a difference acted more than 24%. Their results represent significant differences in spacing matric.

The distance indicator measures the degree of deviation toward getting stuck in local optima.

$$\mathbf{Spacing}_{\mathrm{MOEA-HSS}} > \mathbf{Spacing}_{\mathrm{NSGA-II}} > \mathbf{Spacing}_{\mathrm{MOGWA}} > \mathbf{Spacing}_{\mathrm{MOEA-D}} > \mathbf{Spacing}_{\mathrm{NSGA-II-SA}}$$

MID The Mean Ideal Distance is the size of how far each Pareto front individual is from the ideal point in the objective space. MID is calculated as the average distance between each solution on the Pareto front and the ideal solution, which allows for a quick assessment of how well-suited the solutions are toward satisfying all objectives simultaneously. The main goal in multi-objective optimization problems is to achieve the lowest MID possible so that Pareto front solutions successfully equilibrate all objectives as closely as possible [52]. The results of the MID show that the NSGA-II-SA has the highest MID metric of 2.8321, and the NSGA-II-SA has the least amount spacing metric of 5.7623, which have a difference acted more than 24%. The MOEA-D, MOEA-HSS, and MOGWA have values of 3.2476, 3.1181, and 3.3265 in the analogous range, respectively, which has the same act.

$$\mathbf{MID}_{\mathrm{NSGA-II}} > \mathbf{MID}_{\mathrm{MOEA-D}} > \mathbf{MID}_{\mathrm{MOGWA}} > \mathbf{MID}_{\mathrm{MOEA-HSS}} > \mathbf{MID}_{\mathrm{NSGA-II-SA}}$$

5.2 Pareto front data normalization

The proposed model is a train scheduling problem with two objective functions and discrete variables. The numerical amounts of the objective functions (Z_1, Z_2) differ considerably in numerical value. The result of objective functions (Z_1, Z_2) has normalized to have a good understanding, better display, analysis, and create a balance between the amounts [44]. In formulas (15)–(18), the variables z_k^{max} and z_k^{min} are maximum and minimum for each objective function, respectively, n is the number of the Pareto front population, and k is the number of the objective function [53].

$$\theta = \left\{\theta_1, \theta_2, \dots, \theta_n\right\} \to \text{the pareto front set}\, n \geq 2 \tag{15}$$

$$Z_k(\theta) = \left\{z_k\left(\theta_i\right)\right\} \quad k = 1, 2\,\text{and}\, i = 1, 2, \dots, n \tag{16}$$

$$z_k^{\max} = \max Z_k(\theta)\,\text{and}\, z_k^{\min} = \min Z_k(\theta) \quad k = 1, 2 \tag{17}$$

$$\bar{z}_k\left(\theta_i\right) = \frac{z_k\left(\theta_i\right) - z_k^{min}}{z_k^{max} - z_k^{min}} 0 \leq \bar{z}_k\left(\theta_i\right) \leq 1\; k = 1, 2\; i = 1, 2, \dots, n \tag{18}$$

5.3 Gold point on Pareto front set

Multiple Attribute Decision Making (MADM) and the Gold Point method are various approaches to decision-making. The MADM approach is utilized for decisions with several inconsistent attributes. The attributes include recognizing, analyzing, and evaluating conflicting alternatives with different criteria for the decision-maker [54]. The MADM approach includes allocating weights criteria and detecting the best alternative based on each criterion alternative [55]. The gold point approach is for a specific MADM method that selects one optimal solution from several alternatives. In summary, the MADM is an approach to decision-making with several attributes and criteria. The gold point is a specific technique of the MADM that chooses the optimal decision among several alternatives.

The results of traditional multi-objective optimization problems include several local optimal points [9]. The railway operators need to select one program of the local optimum points. Therefore, the single-objective optimization problems method can select one local optimal point between the results of a multi-objective optimization problem. The gold programming of the decomposition methods is an approach to select a locally optimal point [56]. In the first step, the point (0, 0) is chosen as the ideal point with the lowest value for the objective functions. In the next step, the value of Euclidean norm (2- norm) for each local optimum point relative to the ideal point is calculated by formula (19). The third step is to select an individual with the lowest Euclidean norm with formula (20).

Finally, the gold point GP is chosen as an optimal solution among optimal solutions [12], and the train schedule is extracted for five algorithms.

$$Euclidean\,norm\,(2-\ norm)\ :\ \overline{\boldsymbol{Z}}\left(\theta\right)_2 = \sqrt{\sum_{i=1}^{n} \overline{z}\left(\theta_i\right)^2} \tag{19}$$

$$GP = \min \overline{\boldsymbol{Z}}(\theta)_2 \tag{20}$$

In Fig. 9, the classical two-dimensional representation scatters negative correlation plot is drawn with normalized Pareto front data, the gold point, and the Euclidean distance (Ed) for five algorithms. A scatter plot helps visually understand the relationship between objective functions. The middle points in the Pareto front are usually closer to the ideal solution, and the population of the Pareto front near the gold point has the highest diversity and the lowest crowding distance [51, 57, 58].

5.4 Algorithms analysis

Multi-Criteria Decision Making (MCDM) is used for problems with multiple criteria [59], which (TOPSIS) and analytic hierarchy process (AHP) are two different methods of MCDM. In the AHP method, a pairwise comparison matrix has been utilized to calculate the weights of each criterion and alternative, which is determined based on subjective factors, relative priorities, and experts' opinions [60]. The AHP method has some weaknesses including relative value of options, large computational resource requirement, more time, mental nature, more pairwise comparisons, and application of tastes by experts [61, 62]. As another MCDM approach, the TOPSIS model is to rank the solutions according to relative closeness to the ideal points. Therefore, the best alternative should have less distance from the ideal positive point and most distance from the ideal negative point [63]. The primary difference between TOPSIS and AHP is that TOPSIS is based on finding the best alternative among all solutions, whereas AHP selects the best based on the importance of criteria.

It is a fact that passenger satisfaction will be improved by reducing the waiting time and increasing the number of trains, which result in increasing the train operating costs. On the other hand, passenger dissatisfaction will be increased by reducing the number of trains, which results in decreasing train operating costs. These types of challenges are permanent challenges in daily railway operations. This subsection analyzes the compromise solution by the TOPSIS method. The compromise results are shown in Table 5.

Figures 10, 11 and 12 show a big picture of the compromise between problem objective functions with different weights of TOPSIS. One normal matrix or weighted scale is created to assign weight to each criterion. The criteria weights are determined by expertise and range from zero to one, with a total sum of one. In this study, the weights are chosen for the objective functions with equal values (0.3, 0.7), (0.5, 0.5), and (0.7, 0.3).

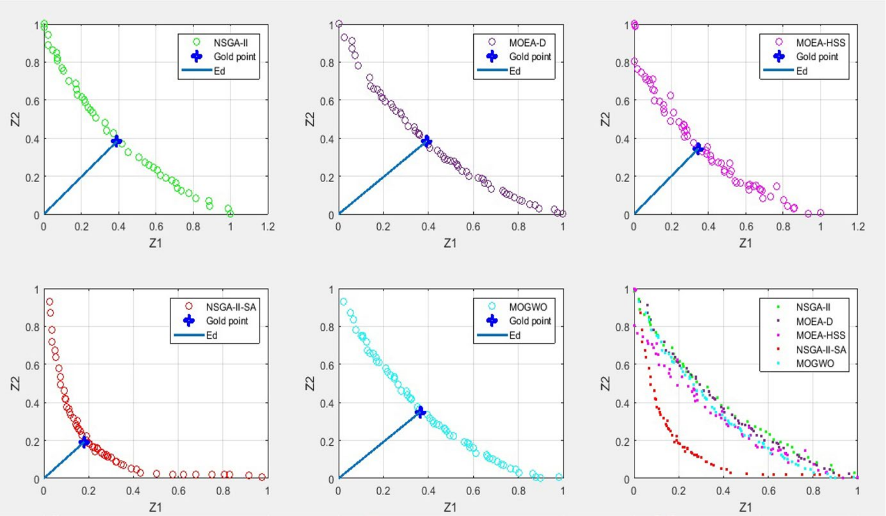

Fig. 9 The scatter negative correlation plot of five algorithms

In Fig. 10, the MOEA-D and MOGWA algorithms have a more orientation to the negative ideal point than others. In addition, the NSGA-II, NSGA-II-SA, and MOEA-HSS algorithms have a more inclination to the positive ideal point. The algorithm MOGWA has a more interquartile range IQR and diagonal to a negative ideal solution than other algorithms. The MOGWA and MOEA_D have a more tendency to the negative ideal point, respectively.

In Fig. 11, all algorithms have solutions with a relatively symmetrical distribution, but the MOGWO tends to a negative ideal solution and negative skewness. The best compromise has happened with minimum IQR by NSGA-II-SA.

In Fig. 12, the algorithm NSGA-II has a more considerable interquartile range IQR, diagonal to a negative ideal solution, and negative skewness than other algorithms. The algorithm MOEA-HSS has widespread IQR, balance IQR, and a little negative skewness. The algorithm NSGA-II-SA has widespread and balanced IQR and more positive skewness. MOEA-D and MOGWA algorithms have almost similar performance.

The algorithms may choose different alternatives to the optimal solution by different parameters. The NSGA-II-SA has obtained the best-compromised solution than other algorithms with values (0.5, 0.5).

6 Numerical experiment

In this section, through numerical calculation and evaluation, the experimental results of the proposed model will be presented. The model is implemented and executed utilizing the developed NSGA-II-SA algorithm and four algorithms, namely

Table 5 the compromise solution by algorithms

Weight	Algorithm	Z_1	Z_2
(0.7, 0.3)	NSGA-II	9.64	64
	MOEA/D	9.84	63
	MOEA-HSS	10.01	61
	NSGA-II-SA	9.27	60
	MOGWO	9.96	63
(0.5, 0.5)	NSGA-II	10.49	56
	MOEA/D	10.43	56
	MOEA-HSS	10.30	55
	NSGA-II-SA	10.09	54
	MOGWO	10.42	55
(0.3, 0.7)	NSGA-II	11.43	53
	MOEA/D	11.13	54
	MOEA-HSS	11.29	53
	NSGA-II-SA	10.61	51
	MOGWO	11.53	54

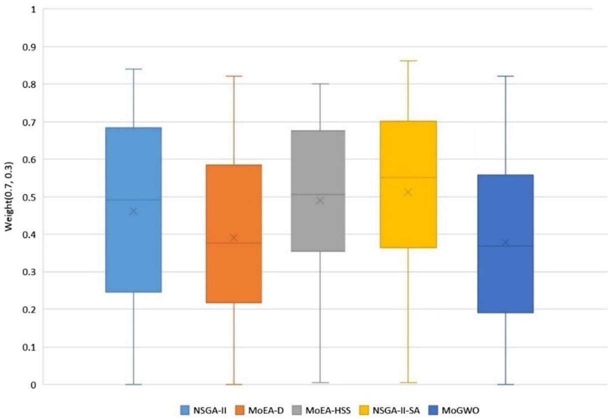

Fig. 10 The compromised solutions of algorithms for weight = (0.7, 0.3)

NSGA-II, MOEA-HSS, MOEA/D, and MOGWO with the simulation environment specification presented in Table 6.

6.1 Case study

The case study has been analyzed by comparing the actual train schedule of the Iranian railway and an optimal solution with the NSGA-II-SA algorithm with the lowest Ed (mentioned in 5.3) to evaluate the proposed model by a real dataset. It should

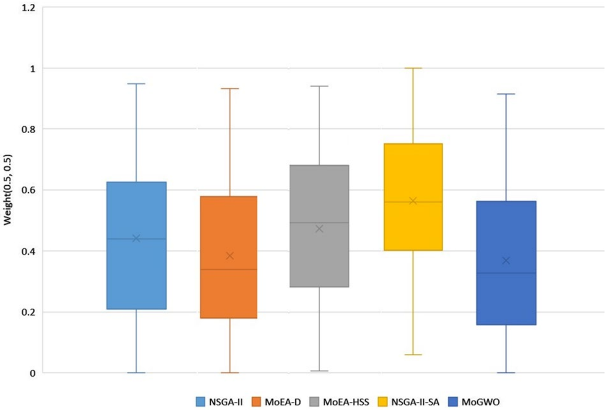

Fig. 11 The compromised solutions of algorithms for weight = (0.5, 0.5)

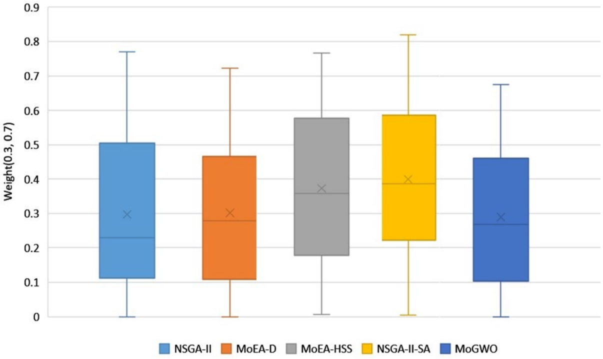

Fig. 12 The compromised solutions of algorithms for weight = (0.3, 0.7)

be noted that real datasets are the same in all algorithms. Figure 13 shows the double-track with a high-traffic railway route of the Tehran–Mashhad, Iran. This part of the double-track railway network is 924 km long, has 50 stations, and has 58 trains' movements in 24 h.

Figure 14, shows the scheduling graph of high-traffic passenger trains on the Tehran–Mashhad route for 24 h.

Train traffic management needs to be more efficient in dispatch, avoiding the cause of congestion and operation complexity. The train schedule is run from 00:30 AM to 5:00 PM on the outbound route and 9:30 AM to 12:00 PM on the return. The train dispatch density causes passenger presence crowding in stations in the limited time of 24 h. The turnaround interval is reduced between two successive trains due to crowding and traffic of turning trains at the first and last stations. As a result, the time interval must be decreased between two sequence trains. This approach causes operation density and increases traffic management risk.

Figure 15 shows the train schedule graph by the NSGA-II-SA algorithm on the same day and with the same data. Trains are scheduled to run from 00:00 AM to 10:30 PM on the outbound route and from 00:30 AM to 11:00 PM on the return route on one cycle. According to constrain (5), the difference between the maximum and minimum time traveling between two stations i, $i+1$ is proportional to the dispatch of trains during a 24-h cycle. Constrain (6) and (7) control the difference between the maximum and minimum safe distance between two consecutive trains h_u^{min}, h_u^{max}. The distribution of passengers $a_{iu}(p)$ spread in the station, service management, and the time required for train rotation t_u^T are expanded on the first and last station. This algorithm has improved traffic management risk, congestion, and operation density of trains in dispatch.

6.2 Scenarios analysis of the proposed deterministic-based approach

The genetic algorithm (GA) is one of the population-based stochastic algorithms, with a randomly initialized population, random crossover operator, and random mutation operator. The SA is a stochastic algorithm with a global search that tries to escape the trap of local optima points [26]. First, consider the uncertainty of passenger demands on the model. Suppose the uncertain passenger demands occur with known probability under seasons. Passenger arrivals follow a Poisson distribution, where the arrival intensity λ_u (14) of each passenger arriving is considered the same at variant stations, at different hours of the day, in each season [75]. In this case study, the probability of the number of passengers entering the station in scenarios U (spring, summer, fall, and winter) is considered to be 0.65, 0.85, 0.55, and 0.45, respectively.

Table 7 illustrates the normalized values of the first and second objective functions, Average Passenger Waiting time (APW), and the Total Number of Trains (TNT) in four scenarios. The highest and lowest amount of function Z_1 occurred in the algorithm the NSGA-II and NSGA-II-SA with APW = 15.3 and APW = 10.09 under scenario summer and winter, respectively. The highest and lowest amount of function Z_2 is in the algorithm the MODE-HSS and NSGA-II-SA with TNT = 57 and TNT = 49 under scenario spring and winter, respectively. The best optima solution is in the algorithm NSGA-II-SA with APW = 10.09 and TNT = 54 under scenario summer. The worst case is in the algorithm MOEA-HSS with APW = 14.52 and TNT = 57 under scenario spring. The difference between the most optimal

Table 6 Simulation environment specification

Hardware specification	11th Gen of Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz RAM 20.0 GB
Operating system	Windows 10
Simulation environment	MAT Lab R2020a programming language
Algorithms' number of runs	20 times for each algorithm

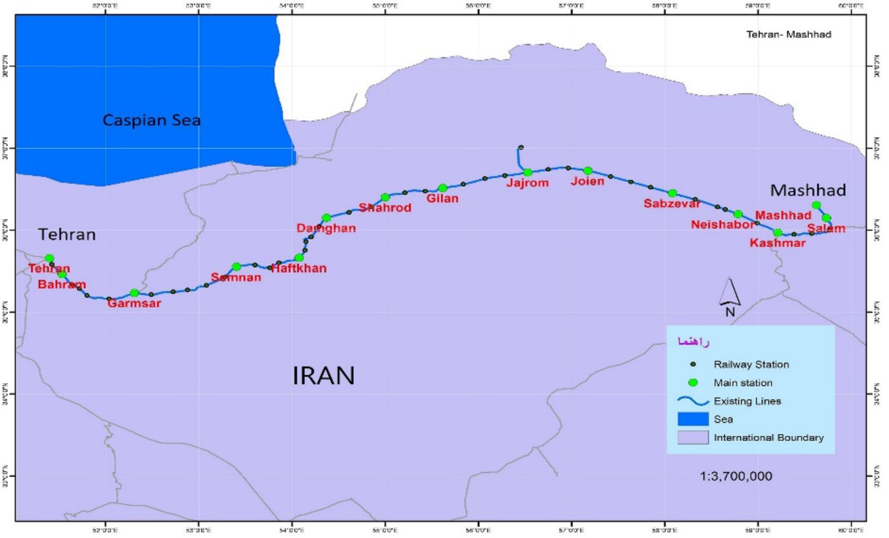

Fig. 13 Part of the map of the Iran railway [64]

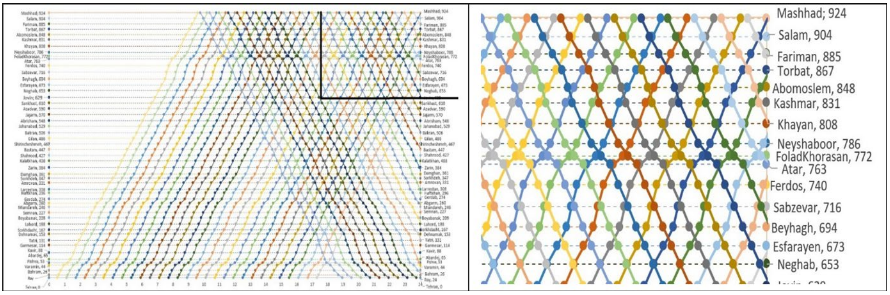

Fig. 14 Real scheduling graph Tehran–Mashhad route

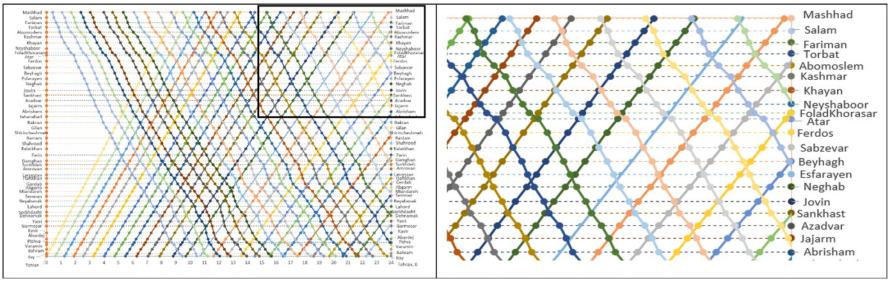

Fig. 15 Modeling graph NSGA-II-SA Tehran–Mashhad route

Table 7 the algorithms result based on four scenarios (spring, summer, fall, winter)

Scenario/Algorithm	Uspring			Usummer			Ufall			Uwinter		
	(Z_1, Z_2)	APW	TNT	(Z_1, Z_2)	APW	TNT	(Z_1, Z_2)	APW	TNT	(Z_1, Z_2)	APW	TNT
NSGA-II	(0.521, 0.3740)	13.56	55	(0.391, 0.383)	10.49	56	(0.507, 0.354)	13.58	54	(0.574, 0.367)	15.3	53
MOEA_D	(0.473, 0.341)	12.51	55	(0.405, 0.350)	10.49	56	(0.510, 0.335)	13.28	54	(0.579, 0.329)	15.1	53
MOEA-HSS	(0.369, 0.335)	14.52	57	(0.342, 0.343)	10.3	55	(0.433, 0.329)	13.08	53	(0.493, 0.315)	14.9	51
NSGA-II-SA	(0.200, 0.185)	11.46	52	(0.181, 0.192)	10.09	54	(0.223, 0.182)	12.51	51	(0.260, 0.172)	14.6	49
MOGWA	(0.425, 0.349)	13.33	54	(0.352, 0.359)	10.41	55	(0.447, 0.344)	13.19	53	(0.486, 0.331)	14.9	51

solution and the worst case in terms of the average passenger waiting time and the total number of trains is 43.9% and 5.55%, respectively.

Figure 16 shows the average travel time of the NSGA-II-SA algorithm on gold point (GP) and the actual train scheduling between stations for all trains on one route. The average passenger waiting time has been calculated for the NSGA-II-SA algorithm and the real train scheduling 10.09 and 10.90 min, respectively. This figure shows that the average passenger waiting time decreases, and passengers will experience an average waiting time of 7.35% decrease based on function (1).

Table 8 includes the result of the algorithms and the actual train schedule of the Tehran–Mashhad trains.

This table is regulated based on a GP and Ed of the Pareto front and includes the average passenger waiting time and the total number of trains in an operating cycle. The percentage improvement has been compared to five algorithms with the real train schedule. Finally, the NSGA-II-SA algorithm has improved the average passenger travel time and the number of train operations on one cycle by 7.34% and 6.89%, respectively.

7 Conclusions and future research

Train schedule is a stochastic multi-objective optimization problem with many constrains and parameters, by which achieving the ideal solution is not practical in a short time due to passenger satisfaction. Passenger satisfaction will be improved by reducing the waiting time and increasing the number of trains, which result in increasing the train operating costs. In contrast, passenger dissatisfaction will be increased by reducing the number of trains, decreasing train operating costs. These types of challenges are permanent challenges in daily railway operations. To alleviate these challenges, a stochastic multi-objective optimization model for train schedules in Tehran–Mashhad, Iran, route has been developed. NSGA-II meta-heuristic algorithm was developed, and as a hybrid model, its combination with the simulated annealing (SA)

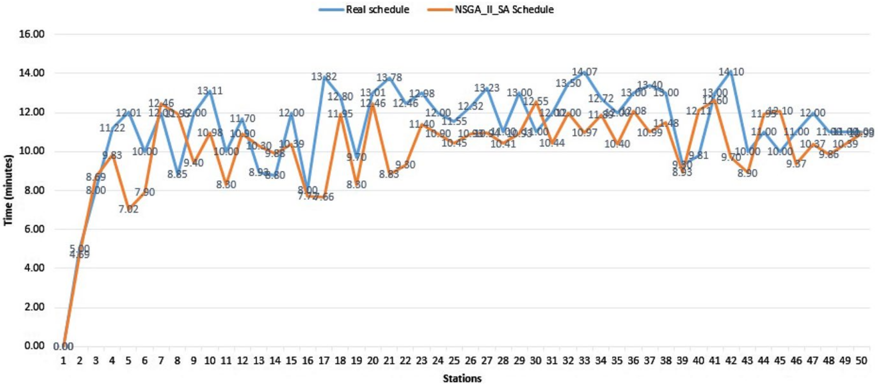

Fig. 16 Average travel time between stations on one route

Table 8 Comparison of algorithms and real scheduling one cycle

	GP (Z_1, Z_2) normalized	Ed normalized	Average passenger waiting time (One cycle)	Improvement	Total number of trains (one cycle)	Improvement
NSGA-II	(0.391, 0.383)	0.548	10.49	%3.72	56	%3.44
MOEA-D	(0.405, 0.350)	0.536	10.49	%3.72	56	%3.44
MOEA-HSS	(0.342, 0.343)	0.484	10.30	%5.45	55	%5.17
NSGA-II-SA	(0.181, 0.192)	0.264	10.09	%7.34	54	%6.89
MOGWO	(0.398, 0.340)	0.501	10.43	%4.31	56	%3.44
Real scheduling	(–,–)	–	10.90	–	58	–

algorithm has been presented. Then, the proposed model has been solved simultaneously with five algorithms, namely NSGA-II, MOEA-HSS, MOEA-D, MOGWA, and NSGA-II-SA multi-objective meta-heuristic algorithms, and compared based on several metrics. After, the TOPSIS method has been utilized to analyze the compromised solutions on the Pareto front set. In the case study, the gold programming of the decomposition method was utilized to select only one optimal solution among all Pareto front optimal solutions. Following, the improvement percentage has been calculated for the actual operation. The main goal is to study NSGA-II-SA algorithm efficiency as the optimal solution superior for reducing the average passenger waiting time and the total number of train cycles in one operational period.

For future research, it is suggested to combine multi-objective meta-heuristic algorithms with artificial intelligence, machine learning techniques, or game theory. The following changes can be applied in the objective functions according to the train operation requirements. The parameters include passenger satisfaction, the probability of accidents, delays caused by a locomotive or wagon deficiency, train speed and weight, fuel consumption reduction, rescheduling, and moving blocks. In the case study, the results of the multi-objective optimal problem can be solved by several single-objective optimization methods until the best compromise solution is gained with the real operations railway during the different 24 h.

Author contributions The proposed approach is a multi-objective mathematical-based Mixed Integer Linear Programming (MILP) approach; the objective is to minimize the average passenger expectation and the total number of train operation cycles. The Non-dominated Sorting Genetic Algorithm (NSGA-II) has been developed with multi-crossover and multi-mutation operators, then hybrid with Simulating Annealing (SA) operator (NSGA-II-SA). The model with four meta-heuristic algorithms has been technically analyzed. In a case study, the train schedule in the double-track rail network of the Tehran–Mashhad railway of Iran has been compared with the golden point. Experimental results show that a proposed approach can suitably fit the problem considering important metrics with an improvement of %7.34 and %6.89 for the average passenger waiting time and the total number of train operation cycles, respectively.

Funding Not applicable.

Data availability https://github.com/seifpour/Massoud/commit/5bcb66177b7669ded404e9f446eed0231e06739f.

Declarations

Conflict of interest Not applicable.

Ethical approval I hereby declare that this paper represents the author’s work and that it has not been submitted, in whole or in part, to any journal.

References

1. Cacchiani V, Qi J, Yang L (2020) Robust optimization models for integrated train stop planning and timetabling with passenger demand uncertainty. Transp Res Part B: Methodol 136:1–29
2. Tian Q, Wang H (2022) Optimization of preventive maintenance schedule of subway train components based on a game model from the perspective of failure risk. Sustain Cities Soc 81:103819

3. Wang X, Chen G, Xu S (2022) Bi-objective green supply chain network design under disruption risk through an extended NSGA-II algorithm. Clean Logist Supply Chain 3:100025
4. Rahimi I et al (2022) Scheduling by NSGA-II: review and bibliometric analysis. Processes 10(1):98
5. Yusoff Y, Ngadiman MS, Zain AM (2011) Overview of NSGA-II for optimizing machining process parameters. Procedia Eng 15:3978–3983
6. Nitisiri K, Gen M, Ohwada H (2019) A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling. Comput Ind Eng 130:381–394
7. Deng W et al (2022) An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. Inf Sci 585:441–453
8. Wu C-C et al (2021) Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times. Swarm Evol Comput 60:100765
9. Türk S et al (2021) Interval type-2 fuzzy sets improved by Simulated Annealing for locating the electric charging stations. Inf Sci 547:641–666
10. Guilmeau T, Chouzenoux E, Elvira V (2021) Simulated annealing: a review and a new scheme. In: 2021 IEEE Statistical Signal Processing Workshop (SSP). IEEE
11. Rajamani M, Rajesh R, Willjuice Iruthayarajan M (2021) Design and experimental validation of PID controller for buck converter: a multi-objective evolutionary algorithms based approach. IETE J Res 1–12
12. Blasco X et al (2008) A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. Inf Sci 178(20):3908–3924
13. Venkateswaran C et al (2022) Application of simulated annealing in various field. Mater Charact 1(1):01–08
14. Tombak GI et al (2022) Simulated annealing assisted NSGA-III-based multi-objective analog IC sizing tool. Integration 85:48–56
15. Tai XY et al (2022) Multi-objective optimisation with hybrid machine learning strategy for complex catalytic processes. Energy and AI 7:100134
16. Sangaiah AK, Khanduzi R (2022) Tabu search with simulated annealing for solving a location–protection–disruption in hub network. Appl Soft Comput 114:108056
17. Meselhi M et al (2022) A decomposition approach for large-scale non-separable optimization problems. Appl Soft Comput 115:108168
18. Elreedy D, Atiya AF, Shaheen SI (2021) Novel pricing strategies for revenue maximization and demand learning using an exploration–exploitation framework. Soft Comput 25(17):11711–11733
19. Wang W-X et al (2020) An improved MOEA/D algorithm with an adaptive evolutionary strategy. Inf Sci 539:1–15
20. Petchrompo S et al (2022) A review of Pareto pruning methods for multi-objective optimization. Comput Ind Eng 167:108022
21. Mirjalili S et al (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl 47:106–119
22. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
23. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
24. Sadeghi AH et al. (2023) Grey wolf optimizer and whale optimization algorithm for stochastic inventory management of reusable products in a two-level supply chain. IEEE Access
25. Chaudhari P et al (2022) Comparison of NSGA-III with NSGA-II for multi objective optimization of adiabatic styrene reactor. Mater Today: Proc 57:1509–1514
26. Goudarzi S et al (2016) Comparison between hybridized algorithm of GA–SA and ABC, GA, DE and PSO for vertical-handover in heterogeneous wireless networks. Sādhanā 41:727–753
27. Heng S et al (2022) How to solve combinatorial optimization problems using real quantum machines: a recent survey. IEEE Access 10:120106–120121
28. Salhi S, Thompson J (2022) An overview of heuristics and metaheuristics. The Palgrave Handbook of Operations Research. Palgrave Macmillan, Cham, pp 353–403
29. Hu Y et al (2023) Robust metro train scheduling integrated with skip-stop pattern and passenger flow control strategy under uncertain passenger demands. Comput Oper Res 151:106116
30. Zhou W et al (2023) Collaborative optimization of energy-efficient train schedule and train circulation plan for urban rail. Energy 263:125599

31. Pan H, Yang L, Liang Z (2023) Demand-oriented integration optimization of train timetabling and rolling stock circulation planning with flexible train compositions: a column-generation-based approach. Eur J Oper Res 305(1):184–206
32. Hörsting L, Cleophas C (2023) Scheduling shared passenger and freight transport on a fixed infrastructure. Eur J Oper Res 306(3):1158–1169
33. Wang E et al (2023) Joint optimization of train scheduling and routing in a coupled multi-resolution space–time railway network. Transp Res Part C: Emerg Technol 147:103994
34. Tang L, Xu X (2022) Optimization for operation scheme of express and local trains in suburban rail transit lines based on station classification and bi-level programming. J Rail Transp Plan Manag 21:100283
35. Bucak S, Demirel T (2022) Train timetabling for a double-track urban rail transit line under dynamic passenger demand. Comput Ind Eng 163:107858
36. Cai L et al (2022) A hybrid adaptive large neighborhood search and tabu search algorithm for the electric vehicle relocation problem. Comput Ind Eng 167:108005
37. Biswas A et al (2022) A study of multi-objective restricted multi-item fixed charge transportation problem considering different types of demands. Appl Soft Comput 118:108501
38. Wang B et al (2018) A NSGA-II algorithm hybridizing local simulated-annealing operators for a bi-criteria robust job-shop scheduling problem under scenarios. IEEE Trans Fuzzy Syst 27(5):1075–1084
39. Li D et al (2019) Trade-off between efficiency and fairness in timetabling on a single urban rail transit line under time-dependent demand condition. Transportmetrica B: Transp Dyn 7:1203–1231
40. Zhang T, Li D, Qiao Y (2018) Comprehensive optimization of urban rail transit timetable by minimizing total travel times under time-dependent passenger demand and congested conditions. Appl Math Model 58:421–446
41. Wang Y et al (2018) Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line. Transp Res Part B: Methodol 118:193–227
42. Abd Elaziz M et al (2022) Advanced metaheuristic techniques for mechanical design problems. Arch Comput Methods Eng 29(1):695–716
43. Xue Y et al (2021) Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. Knowl-Based Syst 227:107218
44. Maskooki A, Deb K, Kallio M (2022) A customized genetic algorithm for bi-objective routing in a dynamic network. Eur J Oper Res 297(2):615–629
45. Zhi H, Liu S (2019) Face recognition based on genetic algorithm. J Vis Commun Image Represent 58:495–502
46. Chen W-H et al (2022) A comprehensive review of thermoelectric generation optimization by statistical approach: Taguchi method, analysis of variance (ANOVA), and response surface methodology (RSM). Renew Sustain Energy Rev 169:112917
47. Ahmad MN et al (2022) Application of Taguchi method to optimize the parameter of fused deposition modeling (FDM) using oil palm fiber reinforced thermoplastic composites. Polymers 14(11):2140
48. Riquelme N, Von Lücken C, Baran B (2015) Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI). IEEE
49. Audet C et al (2021) Performance indicators in multiobjective optimization. Eur J Oper Res 292(2):397–422
50. Hashim FA et al (2022) Honey Badger Algorithm: new metaheuristic algorithm for solving optimization problems. Math Comput Simul 192:84–110
51. Osuna-Enciso V, Cuevas E, Castañeda BM (2022) A diversity metric for population-based metaheuristic algorithms. Inf Sci 586:192–208
52. Behnamian J, Memar Dezfooli S, Asgari H (2021) A scatter search algorithm with a novel solution representation for flexible open shop scheduling: a multi-objective optimization. J Supercomput 77(11):13115–13138
53. Gautier NJD, Manzanares Filho N, da Silva Ramirez ER (2022) Multi-objective optimization algorithm assisted by metamodels with applications in aerodynamics problems. Appl Soft Comput 117:108409
54. Baumann M et al (2019) A review of multi-criteria decision making approaches for evaluating energy storage systems for grid applications. Renew Sustain Energy Rev 107:516–534
55. Liu P et al (2020) A multiple attribute decision making three-way model for intuitionistic fuzzy numbers. Int J Approximate Reasoning 119:177–203

56. Tang H, Haynes R, Houzeaux G (2021) A review of domain decomposition methods for simulation of fluid flows: concepts, algorithms, and applications. Arch Comput Methods Eng 28(3):841–873
57. Zheng W, Doerr B (2022) Better approximation guarantees for the NSGA-II by using the current crowding distance. arXiv preprint arXiv:2203.02693
58. Smith S et al (2022) Multiobjective optimization and Pareto front visualization techniques applied to normal conducting rf accelerating structures. Phys Rev Accel Beams 25(6):062002
59. Tian G et al. (2023) A survey of multi-criteria decision-making techniques for green logistics and low-carbon transportation systems. Environ Sci Pollut Res 1–23
60. Barrios MAO et al (2016) An AHP-topsis integrated model for selecting the most appropriate tomography equipment. Int J Inf Technol Decis Mak 15(04):861–885
61. Dung NB et al (2022) The role of factors affecting flood hazard zoning using analytical hierarchy process: a review. Earth Syst Environ 6(3):697–713
62. Kalita K et al (2023) Parametric optimization of non-traditional machining processes using multi-criteria decision making techniques: Literature review and future directions. Multiscale Multidiscip Model Exp Des 6(1):1–40
63. Dymova L, Sevastjanov P, Tikhonenko A (2013) A direct interval extension of TOPSIS method. Expert Syst Appl 40(12):4841–4847
64. Railway I (2022) Iran RailWay. Available from: http://www.rai.ir

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.