Graphical Abstract

Biased Random-Key Genetic Algorithm (BRKGA) for Optimized Crew Allocation in Railway Maintenance

Fernanda Torres Vidal da Costa, Alexandre Checoli Choueiri

Highlights

Biased Random-Key Genetic Algorithm (BRKGA) for Optimized Crew Allocation in Railway Maintenance

Fernanda Torres Vidal da Costa, Alexandre Checoli Choueiri

- Research highlight 1
- Research highlight 2

Biased Random-Key Genetic Algorithm (BRKGA) for Optimized Crew Allocation in Railway Maintenance

Fernanda Torres Vidal da Costa, Alexandre Checoli Choueiri

[a]*Federal University of Paraná (UFPR), Department of Production Engineering, Curitiba, , Paraná, Brazil*

Abstract

Railway infrastructure plays a crucial role in large-scale countries due to its high capacity for transporting heavy loads. In Brazil, although freight costs are competitive and the railway share of total cargo transport is projected to reach 40% by the end of 2035, road transport still predominates. In this context, efficient maintenance of the permanent way is essential to ensure safety and operational performance, making the optimization of resource allocation a central challenge. This paper proposes an optimization model for railway maintenance planning, focusing on the efficient allocation of crews within available time windows, aiming to minimize operational impact and crew travel distances while maximizing the impact of repaired defects. The problem is solved using the Biased Random-Key Genetic Algorithm (BRKGA), a metaheuristic that stands out for operating independently from the decoding process. Applied to five instances representing real railway scenarios, the BRKGA achieved significantly superior performance compared to Simulated Annealing (SA) and Classic Local Search (CLS). Quantitative results show that BRKGA achieved an average cost reduction of 32% compared to SA and 38% compared to CLS, validating it as an efficient and flexible technique for solving complex resource allocation problems in railway systems.

Keywords: Biased Random-Key Genetic Algorithm; BRKGA; Crew Allocation;Railway Maintenance Planning

---

1. Introduction

The five largest countries by land area, in order, are Russia, Canada, China, United States and Brazil. In this context, for countries with continental dimensions, the railway mode stands out as one of the most efficient due to its high capacity to transport large volumes and heavy loads over long distances. This makes it vital to the economic activity of these nations [1]. This relevance is reflected in the ranking of the world's largest railway networks: among the five countries with the most extensive rail networks, four are also among the largest in territory. The United States leads the list, followed by China, Russia and Canada, which occupy the second, third, and fifth positions, respectively [2].

However, Brazil is left out of this scenario, as it remains highly dependent on the road transport mode. According to the National Transport Confederation (CNT), in February 2019, more than 60% of cargo transportation in Brazil was carried out by highway, while 20% used railways [3]. This scenario contrasts with the ideal model proposed by [4], which suggests a cargo distribution in Brazil of 40% by rail, 40% by waterways and only 20% by road. This balance would help reduce the dependence on highways, making Brazilian logistics more efficient and competitive.

Nevertheless, according to the National Association of Railway Transport (ANTF), in 2023, Brazil already demonstrated competitiveness in railway freight costs compared to other countries. The average cost in the country was $ 1.90 per tonne-kilometer useful (TKU) —that is, the cost to transport one tonne of cargo over a distance of one kilometer— ranking behind only Russia ($0.84 per TKU) and China ($1.59 per TKU) [5]. Furthermore, according to the National Logistics Plan (PNL) 2025, in Brazil, it is estimated that by the end of 2025, approximately 31% of the total cargo transport will be consist of the railway mode, and projections indicate that this share may reach around 40% by 2035 [6]. Thus, the country presents a strategic opportunity for railway expansion, which, in addition to strengthening the logistics infrastructure, also boosts the economy.

In this expansion process, maintaining railway infrastructure becomes essential to ensure both transport efficiency and safety. According to the accident report prepared by the National Land Transport Agency (ANTT) in 2014, between 2006 and 2013, 37.94% of railway accidents in Brazil were caused by structural, geometric, or construction-related issues on the permanent way [7]. This highlights that track maintenance should be considered a priority in operations. Based on the 2022 Yearbook of the National Confederation of Transport (CNT), approximately 50% of the investments made by the railway concessionaires in 2016 were allocated to the superstructure and infrastructure of the railways [8]. Given such significant investments, it is crucial that maintenance resources are used in the most efficient and optimized way possible.

Given this scenario, this article proposes an optimization model for railway maintenance planning, focusing on the efficient allocation of resources and the optimal use of available time windows for corrective activities. The methodology aims to minimize the operational impacts caused by track defects, improve maintenance planning, and contribute to the efficiency of the railway system,being solved using the *Biased Random-Key Genetic Algorithm* (BRKGA) proposed by [9].

The paper is organized as follows: Section 2 presents the literature review that covers studies related to the topic of this research. Section 3 provides a detailed definition of the problem, describing both its operational context and the mathematical formulation used to model it. Section 4 presents the construction of the decoder, which is later combined with the Biased Random-Key Genetic Algorithm (BRKGA). Section 5 discusses the computational results obtained from the application of the algorithm to real railway scenarios. Finally, Section 6 presents the main conclusions and suggestions for future research.

2. Literature review

In this article, the studies in the literature related to the topic are organized into four categories: those focusing exclusively on train scheduling; those integrating maintenance planning with railway traffic management; those emphasizing maintenance planning with a focus on team allocation; and those providing the foundation for the use of genetic algorithms and random-key methods, serving as the basis for the approach proposed in this work.

2.1. Train scheduling problems

In [10], a Lagrangian relaxation-based approach is proposed to generate optimized train timetables, considering track capacity constraints and different types of services. In [11], a stochastic optimization model is presented for allocating time supplements and buffer times in timetables, aiming to increase robustness against operational disturbances. As described by [12], the problem of inserting freight trains into passenger-dominated railway networks is addressed using an integer linear programming formulation and a Lagrangian heuristic to assign feasible timetables close to the ideal ones. As proposed in [13], a collaborative optimization model is developed for stop planning and train scheduling in high-speed railways, aiming to minimize total dwelling time and departure delays, using a multi-objective mixed integer linear programming formulation. As shown in [14], a bi-objective mixed integer linear programming model is developed to reschedule timetables in high-speed rail corridors, minimizing travel time for new trains and schedule adjustments for existing ones, while considering station capacity and acceleration/deceleration times.

2.2. Integration problems between maintenance planning and train scheduling.

Recent studies have focused on the integration of maintenance planning and train scheduling. In [15], a joint optimization model is proposed for maintenance scheduling and train timetabling in double-track railway networks, taking into account that upstream and downstream trains operate independently and that each maintenance task must be continuous and indivisible. A heuristic algorithm based on Lagrangian relaxation is used to handle the model's complexity and generate feasible solutions for real-world railway networks. Following this line, [16] focuses on the Dutch context, where passenger demand is expected to double by 2050. The study develops a Mixed Integer Linear Programming (MILP) model to allocate maintenance projects in a way that minimizes passenger delays. It also proposes a relaxation of so-called event requests—requests made by operators to block specific time windows for maintenance—with the aim of better balancing operational needs and major event schedules. Expanding on this perspective, [17] addresses the simultaneous integration of three key components: train scheduling, platform assignment, and maintenance planning in a high-speed railway network. The model uses a binary integer formulation to minimize operational costs and delays, employing a heuristic approach that dynamically manages available time windows for train operations.

In a similar vein, [18] proposes minimizing the impact of maintenance activities on train circulation by allocating maintenance windows during naturally occurring gaps in the train schedule. To achieve this, a simulated annealing technique is employed, incorporating constraints such as the earliest start and latest end times for maintenance teams' working hours. Meanwhile, [19] focuses on the efficient allocation of rolling stock (trains and locomotives), integrating both circulation and maintenance considerations. The study presents an exact optimization model alongside a hybrid heuristic approach to enhance fleet utilization. The results demonstrate significant improvements in time efficiency and productivity compared to manual planning methods. Finally, [20] develops a space-time network-based approach to jointly represent maintenance activities and train operations. The problem is decomposed into two subproblems: the allocation of major maintenance tasks, and the integrated optimization of maintenance windows with train scheduling. Both subproblems are addressed using customized heuristic algorithms. Practical case studies from the Chinese railway network demonstrate the effectiveness of the proposed methods.

2.3. Maintenance planning focused on crew allocation

These studies address maintenance planning with an emphasis on crew allocation. In [21], a real-world case in Denmark is analyzed, where a new signaling system requires quick responses to failures, placing pressure on the planning system. In this context, a decentralized approach is proposed, in which crews begin their shifts from their homes rather than from a single central location. The study introduces a hyper-heuristic framework that allocates tasks based on two main criteria: balancing the workload among team members and minimizing the distance between tasks within each subregion, thus promoting agility in response times. In [9], the so-called 'curfew planning problem' is addressed, which involves defining an annual schedule for maintenance projects that require full track closures (curfews). Each crew has a specific skill set, and the schedule aims to minimize disruption across the railway network throughout the year. To solve the problem, the authors propose four iterative optimization algorithms, applied to real data from a North American railway operator, achieving promising computational results within minutes. Meanwhile, [22] proposes a time-space network model to handle the allocation of crews to maintenance projects distributed across an extensive railway network. The objective is to minimize both crew travel costs and the operational impact of maintenance activities, while respecting constraints such as exclusivity, precedence, and time windows. An iterative heuristic method is developed to solve the large-scale problem, and the results show a 66.8% reduction in total costs compared to current industry practices, along with the elimination of constraint violations.

2.4. Random Keys and Genetic Algorithms

Introduced by [23], genetic algorithms use the so-called *chromosomes* to represent solutions, which are evaluated and subjected to selection and recombination processes inspired by evolutionary mechanisms of biology. Over time, this approach has become established as an efficient optimization technique, applicable to a wide variety of problems [24].

In this context, the representation of the solution of a problem, the chromosome, serves as the bridge between the **phenotype** (the set of all possible solutions) and the **genotype** (the computational, encoded form through which these solutions are manipulated and processed by the algorithm)[25]. However, in situations where the solution is complex or the direct representation is too restrictive, it becomes necessary to employ indirect encodings. In such cases, a decoder is used to transform the genotype (the encoded representation) into the phenotype (the decoded solution) [26], as illustrated in Figure 1.

Figure 1: Application of the decoder

Later, [27] proposed the encoding of solutions through random keys, in which each solution is represented by a vector of random real numbers. This strategy introduced greater flexibility to genetic algorithms, as it allowed the manipulation to be carried out solely on the solution representation, independently of the decoder. Subsequently, [28] presented a new class of heuristics, called *Biased Random-Key Genetic Algorithms* (BRKGA), in which the genetic method operates fully decoupled from the problem's decoding, manipulating exclusively the chromosome defined by the random keys throughout the flow depicted in Figure 2 [28].

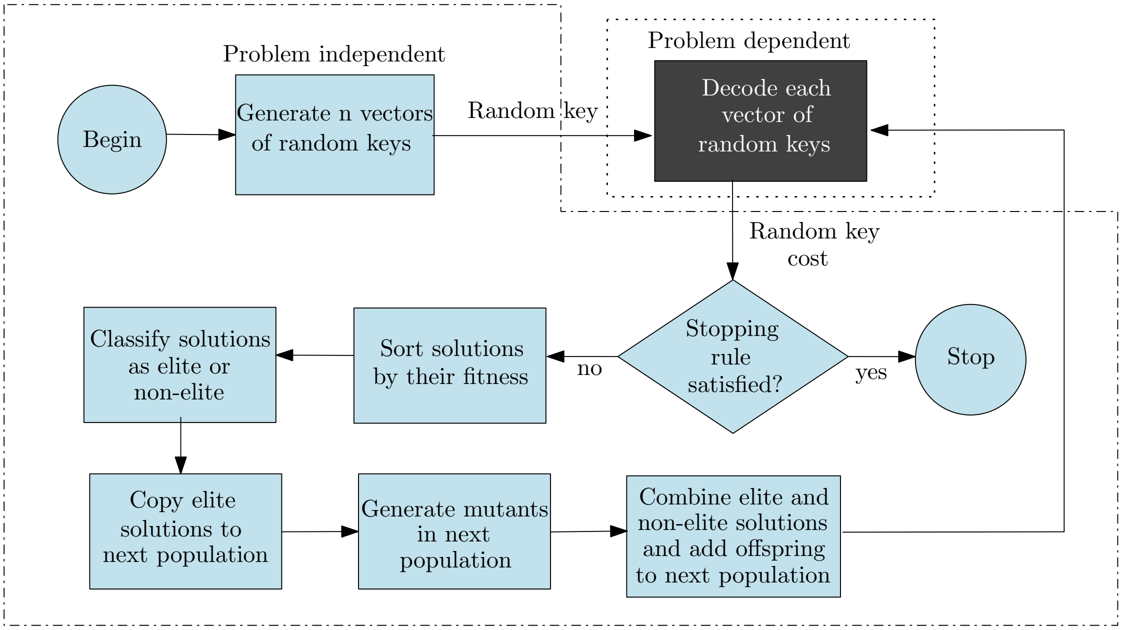

Figure 2: Flowchart of a BRKGA

In this work, the BRKGA framework will be applied to the problem addressed in the following section, leveraging its ability to manipulate solution representations independently of the decoding process. This approach allows for a flexible and efficient exploration of the solution space, making it particularly suitable for complex optimization problems such as the one considered in this study.

3. Problem definition

This section presents the operational context of a logistics company operating a railway network divided into *corridors* and *block sections*. The main factors impacting operational performance include track defects, speed restrictions, and the work of maintenance crews. These elements, together with the available maintenance windows, characterize the resource allocation problem under investigation.

Section 3.1 explains the problem in a descriptive manner, providing all the necessary context. Section 3.2, in turn, defines the problem in mathematical terms — not as a complete MIP model, but through the components required to fully understand the cost function used in the GA.

3.1. Context

A railway network is divided into portions of its total length, called *corridors* **(C)**. Within these corridors, there is an even smaller division, called a *block section* **(BS)**, which represents shorter segments of the track. Figure 3 depicts how corridors are subdivided into block sections.

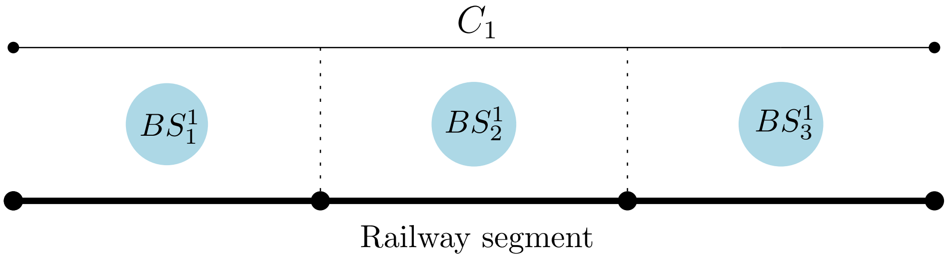

Figure 3: Track segments

A crucial factor for measuring operational performance is the average time a train takes to travel from the beginning to the end of the corridor. The higher the train's speed, the shorter this time will be and, consequently, the better the performance.

However, the presence of **track defects**, caused by equipment degradation, adverse weather conditions, or railway accidents, imposes **speed restrictions** on trains for safety reasons. The greater the number of defects, the higher the speed reduction and therefore, more compromised becomes the **safety level** of the railway.

Each defect identified on the track has an **impact related to its severity**. This impact is calculated based on the train's deceleration curve when passing entirely over the defect and is represented by a distance measure, as presented in Figure 4. This corresponds to the distance traveled from the point where the train departs from its normal speed until it resumes that condition. The greater the distance indicated by the deceleration curve, the higher the severity of the defect. Another relevant variable is the impact resulting from the **number of days** the defect remains active without repair. The greater the number of active days, the higher the safety risks and the greater the severity attributed to the defect.

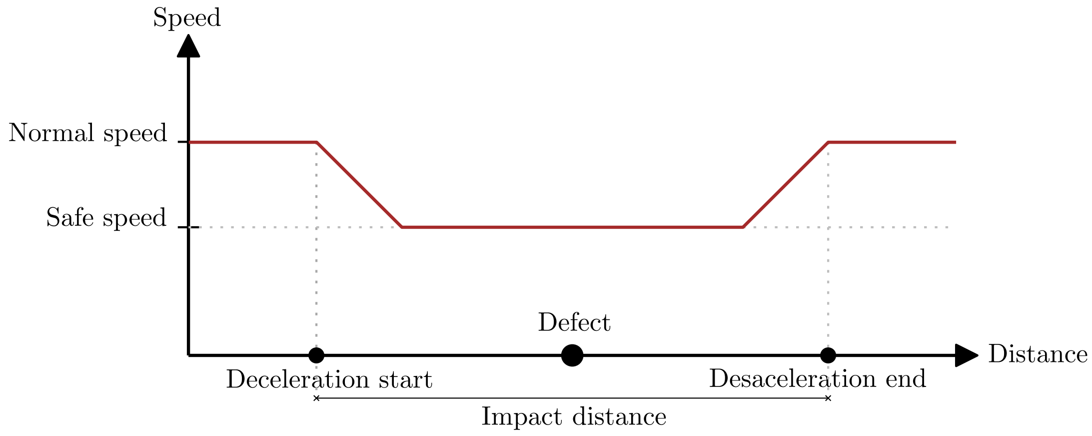

Figure 4: Deceleration curve

There are also two additional pieces of information related to each defect: the **address**, which indicates the geographical location of the defect within the Corridor, and the **MH** (man-hours required for the repair, considering a single worker). For example, a defect that requires 2MH can be repaired in 2 hours by one worker or in 30 minutes by four workers.

To address these defects, there are **maintenance crews** responsible for track repairs. Each crew is assigned to a **specific corridor** and has a fixed number of workers. In addition, each crew has a base located at a strategic point within the corridor, allowing them to respond more efficiently to nearby defects. A corridor may have one or more crews operating simultaneously, with no two crews working on the same defect at the same time.

For maintenance crews to operate, it is necessary to respect the **intervals** between the passage of two consecutive trains. These intervals are predefined for each Block Section (BS), specifying on which days and for how long they will be available. This defines the time windows during which maintenance can be safely performed without interfering with railway traffic.

The objective of the problem is to allocate maintenance teams to defects in such a way that the maintenance windows are respected and no team is assigned to more than one defect at the same time. The method should aim to maximize the total impact of the defects corrected while minimizing the distance traveled by the maintenance teams.

Figure 5 shows a chart representing an example of a solution obtained for the problem. It illustrates the allocation of maintenance crews to the selected defects, where defects 12, 15, 10, 31, and 23 were repaired by crews 1, 2, 3, and 4, as part of the obtained solution. The horizontal axis indicates time in hours, while the vertical axis represents the maintenance crews. Therefore, the length of each bar in the chart reflects the total repair time associated with each defect.

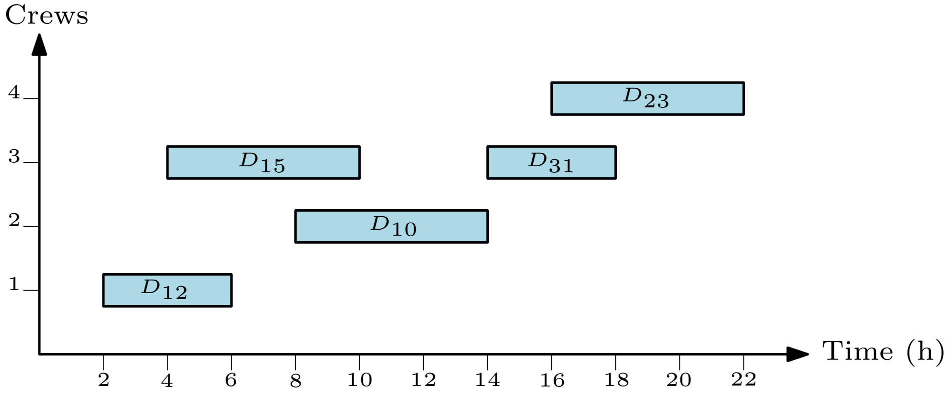

Figure 5: Solution example

3.2. Formal definition

Assume that **each corridor** has the following set of indices, parameters, and variables:

Sets and Indices:

1. BS : Set of block Sections.
2. D : Set of defects.
3. T : Set of crews.
4. TW_s : Set of available maintenance time windows for block section $s, s \in BS$.

Parameters:

1. MH_d : Man-hours required to repair defect d with a single worker, $\forall d \in \{1, \ldots, D\}$.
2. N_t : Number of workers in crew t, $\forall t \in \{1, \ldots, T\}$.
3. TW_s^b : Set of start times of availability window for the Block Section $s, s \in BS$.

4. TW_s^e : Set of end times of availability window for the Block Section $s, s \in BS$.
5. A_{td} : Distance between the base of the crew t and the defect d, $d \in \{1, \ldots, D\}$, $t \in \{1, \ldots, T\}$.
6. $D_{dd'}$: Distance between defect d and defect d', $\forall d, d' \in D$.
7. i_d : Impact of defect d, expressed in kilometers (km) and obtained from the train deceleration curve (Figure 4), $\forall d \in \{1, \ldots, D\}$.
8. K_d : Fraction of i_d, corresponding to the additional impact added for each day that defect d remained active until it was repaired, $\forall d \in \{1, \ldots, D\}$.

Variables:

1. TM_d^b : Start time of maintenance activity to repair defect d, $TM_b^d \in Z^+$, $d \in \{1, \ldots, D\}$.
2. TM_d^e : End time of maintenance activity to repair defect d, $TM_e^d \in Z^+$, $d \in \{1, \ldots, D\}$.
3. TD_d : Total number of days that defect d remained active without being repaired, $d \in \{1, \ldots, D\}$.
4. $z_{tdd'}$: Assumes value 1 if crew t consecutively repair defect d and d', and 0 otherwise, $z_{tdd'} \in \{0, 1\}$
5. x_d : Assumes value 1 if defect d has been repaired, and 0 if that defect remains active, $x_d \in \{0, 1\}$

The objective function of the problem comprises two groups of components: those associated with the impact of the repaired defects, and those associated with the total distance traveled by the maintenance crews.

Components related to the impact:

$$P_1 = \sum_{d \in D} i_d \, x_d \tag{1}$$

$$P_2 = \sum_{d \in D} K_d \, i_d \, TD_d \, x_d \tag{2}$$

Component $P1$ represents the total impact of all repaired defects, which should be **maximized** since it reflects the removal of the most critical defects from the track. Meanwhile, component $P2$ captures the additional impact related to the number of days each defect remained active. This value should be **minimized**, as smaller values indicate that critical defects were repaired sooner. Both components are measured in kilometers (km), in accordance with the definition of i_d.

Components related to the total distance traveled:

$$Y_1 = \sum_{t \in T} \sum_{d'' \in D} A_{td''}\, x_{d''} \tag{3}$$

$$Y_2 = \sum_{t \in T} \sum_{d \in D} \sum_{d' \in D} D_{dd'}\, z_{tdd'} \tag{4}$$

Let d'' denote the first defect repaired by crew t. The first term represents the total distance traveled by all crews from their respective base locations to d''. The second term corresponds to the total distance between consecutive defects repaired by the same crew. Both measures are intended to be **minimized**, indicating that maintenance teams should travel the shortest possible distances. Component P_3, in turn, defines the total distance expressed in kilometer (km) covered by all crews during the repair activities:

$$\mathrm{P3} = Y_1 + Y_2 \tag{5}$$

Given that components $P1$, $P2$, and $P3$ represent different physical contexts of the problem, the direct sum of these terms in the calculation of the objective function cost would not be consistent. To balance the contribution of each component to the total cost, a weighting system (α_i) was adopted, formulated as a convex combination of the objective function components, as follows:

$$\sum_{\forall i} \alpha_i = 1 \tag{6}$$

$$\alpha_i \in (0, 1) \tag{7}$$

The objective function is then:

$$\min \quad F = -\alpha_1 P1 + \alpha_2 P2 + \alpha_3 P3 \tag{8}$$

To prioritize railway track safety, greater weights were assigned to the variables related to impact ($\alpha_1 = \alpha_2 = 0.4$, $\alpha_3 = 0.2$). Therefore, the objective of the problem is to determine a set of elements $E_i = (d, t, TM_d^b, TM_d^e)$ in such a way that the objective function (F) is minimized.

4. Solution method

As previously mentioned, the random keys method was used to represent the solution indirectly. The resulting vector has a size equal to the sum of the total number of defects (D), blocking sections (BS), and maintenance crews (T), as illustrated by Figure 6:

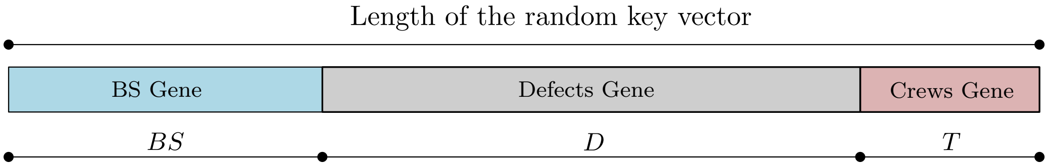

Figure 6: Random keys vector

4.1. Decoder

The decoder is applied **individually** to each corridor. As illustrated in Figure 8, this process consists of four levels of iteration. At level 1, a block section (BS) is selected; at level 2, a defect belonging to that section is chosen. Level 3 determines the earliest available time window for maintenance execution, and level 4 selects the crew responsible for the repair. Levels 1, 2, and 4 follow the order defined by the random key, while time windows (level 3) are considered in ascending order of their starting times. Once this set of elements is defined, an allocation function is applied to verify whether the defect can be repaired by the selected crew within the corresponding time window.

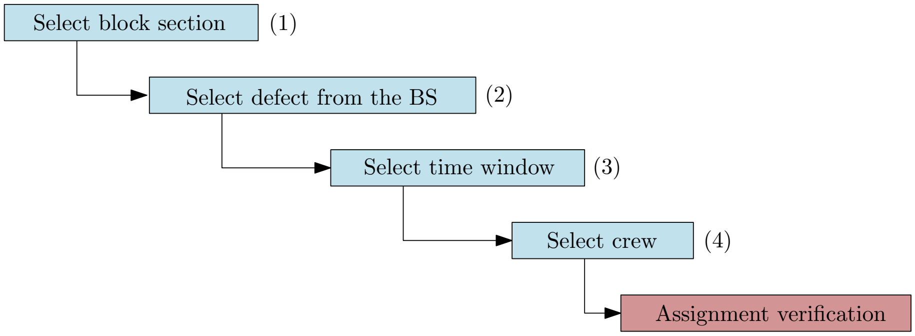

Figure 7: Levels of iteration

A defect remains unallocated after all possible combinations of crews and time windows have been tested by the allocation function. The decoding process is completed once all defects in the global list — that is, all defects belonging to the corridor — have been evaluated. A more detailed representation of the decoder is presented in the flowchart of Figure 8.

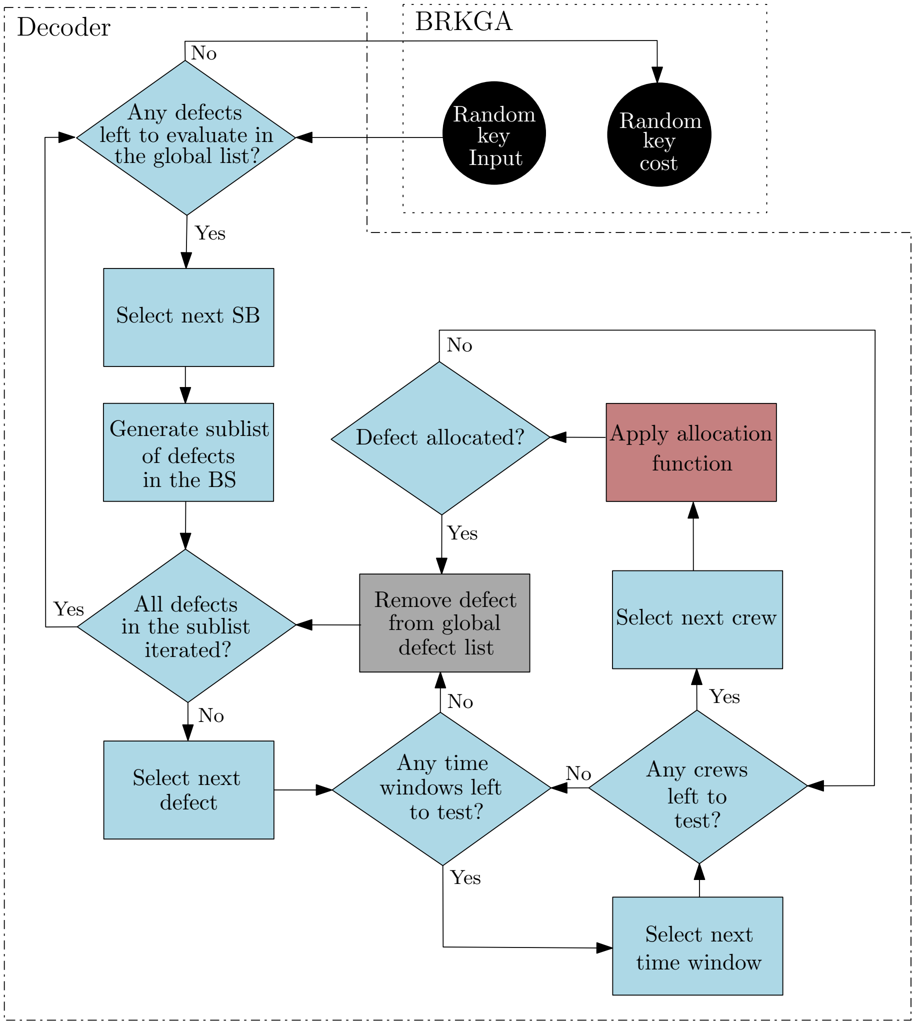

Figure 8: Flowchart of decoder

4.2. Allocation Function

The allocation function is responsible for assessing the feasibility of assigning a maintenance crew to a given defect, considering the operational constraints imposed by the maintenance windows and the crew's temporal availability. Based on the provided parameters, the function calculates the total time required to perform the activity and verifies whether this allocation is feasible within the available time window.

The function receives four main parameters:

1. t : Select crew, $t \in \{1, \ldots, T\}$.
2. d : Select defect, $d \in \{1, \ldots, D\}$.
3. twb_i : The i-eth element of the set of maintenance window start times (TW_s^b) for the block section s where the defect is located, $s \in \{1, \ldots, BS\}$.
4. twe_i : The i-eth element of the set of maintenance window end times (TW_s^e) for the block section s where the defect is located, $s \in \{1, \ldots, BS\}$.

Considering also that, for crew t, the following sets are established:

1. D_t : Defects repaired by crew t, where $D_t \subset D$ and $t \in \{1, \ldots, T\}$.
2. R_t^e : Set of end times of repair activities assigned to crew t, where the variable $TM_d^e \in R_t^e$, and $d \in D_t$.
3. R_t^b : Set of start times of repair activities assigned to crew t, where the variable $TM_d^b \in R_t^b$, and $d \in D_t$.

Using these parameters and sets, the allocation function computes the following elements:

1. **Current time** (c_t) : The last moment when crew t was occupied performing a repair, obtained as the maximum value of the set R_t^e. From this time onward, the crew is available for new assignments.

2. **Available repair time** (a_t) : The total duration of the maintenance window:
$$a_t = twe_i - twb_i \tag{9}$$

3. **Repair time** (r_t) : Ratio between the effort required to repair the defect (MH_d) and the number of workers in the crew (N_t):
$$r_t = \frac{MH_d}{N_t} \tag{10}$$

4. **Travel distance** (d_p) : Distance between the current location of crew t and defect d. If the defect is the first assignment of the crew, this distance is measured from the crew's base (A_{td}); otherwise, it corresponds to the distance between the last repaired defect and the new one ($D_{dd'}$).

5. **Travel time (t_t)** : Ratio between the travel distance (d_p) and the average travel speed of the crew (v_t):

$$t_t = \frac{d_p}{v_t} \tag{11}$$

6. **Waiting time (w_t)** : Any idle period between the crew's arrival at the defect and the beginning of the maintenance window, when the arrival occurs before twb_i.

7. **Final time (f_t)** : The final moment of the repair activity, obtained as the sum of the current time, travel time, repair time, and waiting time:

$$f_t = c_t + (t_t + r_t + w_t) \tag{12}$$

A defect is assigned to a crew if the final time is less than the end of the maintenance window and the repair time is less than the available time for the repair, that is:

$$f_t < twe_i \quad \text{and} \quad r_t < a_t \tag{13}$$

When these conditions are satisfied, the defect d is allocated to set Dt, and the sets R_b^t and R_e^t are updated with the respective start and end times of the repair activity. Figure 9 visually represents how these variables are interpreted by the allocation function. In the example, three crews are considered for the repair of the same defect, with t_1 being the only one that satisfies both constraints established in (13). It can be observed that t_3 does not meet the restriction $r_t < a_t$, while t_2 violates the condition $f_t < tew_i$.

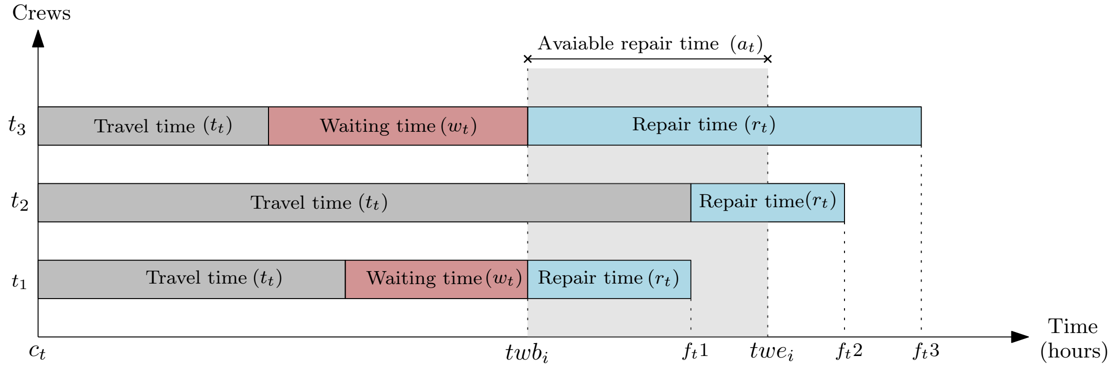

Figure 9: Allocation function illustration

Thus, the allocation function acts as the decision core of the decoder, ensuring that the solutions generated by the random-key structure are feasible and comply with the operational constraints of the railway system.

5. Results

The BRKGA, combined with the decoder, was applied to five different instances representing real railway scenarios. The number of items included in the main parameters — block sections (BS), defects (D) and crews (T)— for each instance is presented below:

Instance	BS	D	T
0	3	31	4
1	10	47	6
2	17	47	5
3	6	27	8
4	15	44	6

Table 1: Instances

In addition to the BRKGA, two additional optimization methods based on local search were applied for comparison purposes: **Classic Local Search (CLS)** and **Simulated Annealing (SA)**. Both methods were implemented using the SWAP neighborhood.

Local search stands as one of the earliest generic algorithmic templates for solving optimization problems in an approximate manner. It iteratively intensifies a neighborhood until no further improvement is possible, thus reaching a local optimum.

On the other hand, the Simulated Annealing (SA) technique follows a principle inspired by the physical process of metal cooling. Unlike SWAP, it may accept worse solutions with a probability controlled by the system's "temperature." As the temperature decreases, the likelihood of accepting inferior solutions gradually diminishes, allowing the algorithm to escape local optima and converge toward near-optimal or global solutions [29].

There was no fine-grained parameter tuning for the algorithms. They were used as follows:

- **Classic Local Search (CLS)**

The neighborhood considered was based on the SWAP operation. At each step of the search, two elements of the solution are exchanged, generating a neighboring solution. After evaluating this solution, if it yields a lower value for the objective function, it is immediately adopted as the new current solution, following the *first improvement* selection criterion. All possible combinations between pairs of positions (i, j) are examined. Thus, considering n as the number of elements in the key, the number (N) of neighbors tested is:

$$N = \frac{n(n-1)}{2} \tag{14}$$

In this case, the algorithm does not return to re-evaluate the pairs of positions that were already tested before the update, but instead continues the process from the new solution.

- **Simulated Anneling (SA)**

The neighborhood was also defined through SWAP operations; however, for the Simulated Annealing, five distinct combinations of positions (i, j) are randomly generated, producing five neighbors from the initial solution. The neighbor selection follows the *best improvement* criterion: among the five generated neighbors, the one with the best objective function value is chosen as the candidate neighbor, even if its objective value is worse than that of the current solution. This candidate neighbor is then evaluated according to the Simulated Annealing acceptance rule to determine whether it will be adopted as the next solution. The parameters used in the Simulated Annealing were:

- **Initial Temperature:** $T_0 = 1000$
- **Final Temperature:** $T_f = 40$
- **Cooling Schedule:** Temperature decay given by

  $$T_{k+1} = \alpha \cdot T_k, \quad \alpha = 0.99$$

  where T_k is the temperature at iteration k.

- **Neighbor Acceptance:** Let Δ be the difference in objective function values between a neighbor and the current solution. The acceptance probability is given by

  $$P_{\text{accept}} = e^{-\frac{\Delta}{T}}$$

  and a random number $r \in [0, 1]$ is generated. The neighbor is accepted if $r < P_{\text{accept}}$. At higher temperatures, worse solutions have a higher probability of being accepted, allowing the algorithm to explore the solution space more freely; as the temperature decreases, this probability diminishes, guiding the search toward local improvement.

- **BRKGA**

Default parameters of the BRKGA implementation (`BrkgaMpIpr`) were defined as follows:

- **Population size:** 1000
- **Elite percentage:** 0.05
- **Mutants percentage:** 0.25
- **Number of elite parents:** 3
- **Total parents:** 12
- **Bias type:** LINEAR
- **Number of independent populations:** 5
- **PR number of pairs:** 0
- **PR minimum distance:** 0
- **PR type:** PERMUTATION
- **PR selection:** BESTSOLUTION
- **Alpha block size:** 1
- **PR percentage:** 0.0
- **Exchange interval:** 10
- **Number of exchange individuals:** 50
- **Reset interval:** 75

The results of the objective function obtained by the three methods across the five instances are presented below.

Method	**0**	**1**	**2**	**3**	**4**	**Total**
BRKGA	0.27	83.80	84.20	41.42	97.52	306.78
SA	0.84	88.35	134.12	72.17	159.02	454.50
CLS	4.32	103.11	148.99	68.15	176.47	501.04

Table 2: Results

The processing times, in seconds, for each method applied are presented in the following.

Method	**0**	**1**	**2**	**3**	**4**	**Average time (s)**
BRKGA	44.20	84.70	76.90	63.30	64.00	66.62
SA	2.40	7.50	5.30	4.30	4.80	4.92
CLS	0.6	1.8	2.4	0.8	1.6	1.44

Table 3: Results

The results show that the BRKGA exhibited consistent performance compared to the other methods, achieving an average cost reduction of 32% relative to Simulated Annealing (SA) and 38% relative to Classic Local Search (CLS). Although the BRKGA was slower, with an average runtime of 66.62 seconds per instance (compared to 4.92 seconds for SA and 1.44 seconds for CLS), it produced higher-quality solutions. Neither SA nor CLS were able to improve the solutions previously obtained by the BRKGA when applied as additional local search steps. Thus, the BRKGA stands out as a promising genetic technique for allocation problems, exploring the solution space independently of the decoding process and enabling optimization with good performance.

6. Conclusion

The results obtained demonstrate that the BRKGA proved to be a highly promising approach for the proposed allocation problem, achieving remarkable results and superior performance compared to the other methods tested, even in the presence of some limitations identified during development.

These limitations, however, represent opportunities for improvement that could further enhance the efficiency of the algorithm. First, there is a need for a deeper understanding of the BRKGA parameters, as limited knowledge about their tuning may have led to the adoption of suboptimal values, restricting part of its performance. In addition, the allocation function used does not consider intermediate insertions between already allocated defects, always prioritizing the latest available time — which limits the exploitation of potential time windows. Finally, it was observed that the same random key may generate identical solutions, since defects are directly dependent on the blocking sections (BSs), which reduces population diversity and, consequently, the algorithm's exploratory potential.

Despite these restrictions, the BRKGA presented highly satisfactory results, demonstrating its optimization capability. Additionally, it is important to highlight that the entire work was developed using Python and a public and freely available implementation of the BRKGA, which means that anyone can reproduce the experiments carried out, adapt the code to their needs, and apply the solution in different railway contexts or even in other sectors. In this way, overcoming or mitigating the identified limitations will further enhance the quality of the solutions, highlighting BRKGA as a versatile, accessible, and effective alternative for the optimization of complex allocation systems.

References

[1] Heriberto Brito Pereira, Cleudson Silva Santos, Roselia de Sousa Alves, Jessica Batista Abreu, Carlos Renato Brandão, and Angelo Teixeira Santos Junior. Análise empírica do setor ferroviário de cargas do brasil. *OBSERVATÓRIO DE LA ECONOMÍA LATINOAMERICANA*, 23(2):e8881–e8881, 2025.

[2] International Union of Railways (UIC). Railway statistics synopsis. Statistical bulletin, International Union of Railways (UIC), 2024. Accessed: 2025-04-08.

[3] Confederação Nacional do Transporte (CNT). Boletim estatístico - 2019. Technical report, Confederação Nacional do Transporte, 2019. Accessed: 2025-04-08.

[4] Raphael Padula. Transportes - fundamentos e propostas para o brasil. *Brasília: Confea*, 2008.

[5] Associação Nacional de Transporte Ferroviário (ANTF). Por que o brasil vai bem de trem. `https://www.antf.org.br/por-que-o-brasil-vai-bem-de-trem/`, 2023. Institutional information accessed 2025-04-08.

[6] Empresa de Planejamento e Logística (EPL). Plano nacional de logística - pnl 2025. Government report, Ministério da Infraestrutura e Empresa de Planejamento e Logística (EPL), 2018. Accessed: 2025-04-08.

[7] Agência Nacional de Transportes Terrestres (ANTT). Relatório de acidentes 2006 a 2013. Government report, Agência Nacional de Transportes Terrestres, 2014. Accessed: 2025-04-08.

[8] Confederação Nacional do Transporte (CNT). Evolução dos principais investimentos das concessionárias por tipo de investimento - 2006 a 2016. Relatório técnico, Confederação Nacional do Transporte, 2022. Accessed: 2025-04-08.

[9] S Boğ, Ashish K Nemani, and Ravindra K Ahuja. Iterative algorithms for the curfew planning problem. *Journal of the Operational Research Society*, 62(4):593–607, 2011.

[10] Ulf Brännlund, Per Olov Lindberg, Andreas Nou, and J-E Nilsson. Railway timetabling using lagrangian relaxation. *Transportation science*, 32(4):358–369, 1998.

[11] Leo Kroon, Gábor Maróti, Mathijn Retel Helmrich, Michiel Vromans, and Rommert Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553–570, 2008.

[12] Valentina Cacchiani, Alberto Caprara, and Paolo Toth. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, 44(2):215–231, 2010.

[13] Lixing Yang, Jianguo Qi, Shukai Li, and Yuan Gao. Collaborative optimization for train scheduling and train stop planning on high-speed railways. *Omega*, 64:57–76, 2016.

[14] Yuan Gao, Leo Kroon, Lixing Yang, and Ziyou Gao. Three-stage optimization method for the problem of scheduling additional trains on a high-speed rail corridor. *Omega*, 80:175–191, 2018.

[15] Chuntian Zhang, Yuan Gao, Lixing Yang, Ziyou Gao, and Jianguo Qi. Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using lagrangian relaxation. *Transportation Research Part B: Methodological*, 134:64–92, 2020.

[16] YR de Weert, Konstantinos Gkiotsalitis, and EC van Berkum. Improving the scheduling of railway maintenance projects by minimizing passenger delays subject to event requests of railway operators. *Computers & Operations Research*, 165:106580, 2024.

[17] Qin Zhang, Richard Martin Lusby, Pan Shang, and Xiaoning Zhu. A heuristic approach to integrate train timetabling, platforming, and railway network maintenance scheduling decisions. *Transportation Research Part B: Methodological*, 158:210–238, 2022.

[18] Paulo MS Bueno, Plínio RS Vilela, Luciano M Christofoletti, and Anderson P Vieira. Optimizing railway track maintenance scheduling to minimize circulation impacts. In *ASME/IEEE Joint Rail Conference*, volume 58523, page V001T04A001. American Society of Mechanical Engineers, 2019.

[19] Yung-Cheng Lai, Dow-Chung Fan, and Kwei-Long Huang. Optimizing rolling stock assignment and maintenance plan for passenger railway operations. *Computers & Industrial Engineering*, 85:284–295, 2015.

[20] Haonan Yang, Shaoquan Ni, Haoyang Huo, Yuhua Yang, Su Liu, Qingpeng Zhang, and Dingjun Chen. Variable neighborhood search and alternating direction method of multipliers for integrated optimization of maintenance windows and train timetables. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[21] Shahrzad M Pour, John H Drake, and Edmund K Burke. A choice function hyper-heuristic framework for the allocation of maintenance tasks in danish railways. *Computers & Operations Research*, 93:15–26, 2018.

[22] Fan Peng, Seungmo Kang, Xiaopeng Li, Yanfeng Ouyang, Kamalesh Somani, and Dharma Acharya. A heuristic approach to the railroad track maintenance scheduling problem. *Computer-Aided Civil and Infrastructure Engineering*, 26(2):129–145, 2011.

[23] John H Holland. Adaptation in natural and artificial systems. *University of Michigan Press google schola*, 2:29–41, 1975.

[24] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[25] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.

[26] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[27] James C Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160, 1994.

[28] José Fernando Gonçalves and Mauricio GC Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

[29] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.