

ORIGINAL RESEARCH

Open Access

# A population-based algorithm for the railroad blocking problem

Masoud Yaghini\*, Masoud Seyedabadi and Mohamad M Khoshraftar

## Abstract

Railroad blocking problem (RBP) is one of the problems that need an important decision in freight railroads. The objective of solving this problem is to minimize the costs of delivering all commodities by deciding which inter-terminal blocks to build and by specifying the assignment of commodities to these blocks, while observing limits on the number and cumulative volume of the blocks assembled at each terminal. RBP is an NP-hard combinatorial optimization problem with billions of decision variables. To solve the real-life RBP, developing a metaheuristic algorithm is necessary. In this paper, for the first time, a new genetic algorithm-based solution method, which is a population-based algorithm, is proposed to solve the RBP. To evaluate the efficiency and the quality of solutions of the proposed algorithm, several simulated test problems are used. The quality and computational time of the generated solutions for the test problems with the proposed genetic algorithm are compared with the solutions of the CPLEX software. The results show high efficiency and effectiveness of the proposed algorithm.

**Keywords:** Railroad blocking problem, Genetic algorithm, Budget design problem

## Background

Railroad blocking problem (RBP) is an important subject for freight railroad companies because its solution can reduce some extra costs. To reach a destination, a load on its route may traverse on many classification yards from its origin to its destination. In these places, the arriving load, which is usually composed of different individual shipments, is regrouped to be carried by departing trains. However, because the reclassification process is difficult and capital is intensive, this classification always produces some delay in the movement of these loads. Therefore, to avoid loads from being regrouped at every classification yard they pass through, it is better to group various shipments together to form a block. A block may contain individual shipments whose origin or destination may or may not be the origin or destination of the whole block. These shipments are regrouped only after reaching the final destination of the block.

The RBP is a network budget design problem (BDP). In the BDP, the flow cost of a set of commodities through a network is minimized while observing budget

constraints on the fixed costs of the links used. BDP is related to several other problems, such as the fixed-charge network design problem and the multicommodity flow problem. The fixed-charge network design problem is different because the fixed costs for the arcs appear in the objective function. This problem is known as nondeterministic polynomial time-hard (NP-hard) combinatorial optimization problem (Magnanti and Wong 1984; Minoux 1989). No polynomial-time optimization techniques are known to solve the NP-hard problem. The multicommodity flow problem is a BDP with fixed binary arc selection variables. The interested reader of the network design problem should consult the surveys by Magnanti and Wong (1984), Minoux (1989), and the annotated bibliography of Balakrishnan et al. (1997), Crainic (2000, 2006, 2009).

We can define the RBP as follows. To reduce the intermediate handlings of shipments as they travel over the railroad network, a set of shipments is classified (or grouped together) at a railroad yard to create a group of cars called block. The aim of the RBP is to choose which blocks to put together at each yard and to determine the sequences of blocks that deliver each shipment to minimize the total miles that the shipments travel and

\* Correspondence: yaghini@iust.ac.ir  
School of Railway Engineering, Iran University of Science and Technology,  
Tehran 16846-13114, Iran

their intermediate handlings (or reclassifications). In fact, the RBP aims to identify the classification plan for all shipments at all yards in the network to minimize the total shipment cost, i.e., to create a blocking plan which dictates which blocks should be built at each individual yard and which traffic should be assigned to each block as they travel from origins to destinations.

The RBP in real-life is a very large-scale problem with billions of decision variables. To solve it, we need to design the underlying blocking network, which is a virtual network that is overlaid on the physical network, and to route different commodities, where each set of railcars with the same origin-destination pair of nodes defines a separate commodity on the blocking network to minimize the system-wide transportation costs. Multicommodity flow and network-design problems are among the most difficult combinatorial problems in research operations. When they comprise a few hundred design variables, we try solving them optimally. However, the real-life RBP often contain over a million design variables and hundreds of billions of flow variables. Modeling and solving them, using commercial software, is nearly impossible.

A pioneer effort to place railroad modeling within a hierarchical structure is provided in the work of Assad (1980). The author offers a three-tiered hierarchy of models, including strategic, tactical, and operational levels. Differences in these levels are about planning horizon, required investment, and modeling methods (Assad 1980). RBP is one of the main problems that need a decision in rail transportation planning, which is made in the tactical level of railroad operation.

Central to railroad operations is the operating plan that dictates the movement of shipments (railcar loads), crews, and locomotives over the railroad's network. Each railroad company has a service design department that is responsible for creating operating plans to enable efficient movement of shipments (Figure 1; Ahuja et al. 2007).

The blocking plan determines how to aggregate a large number of shipments into blocks of shipments as they travel from origins to destinations. Train scheduling

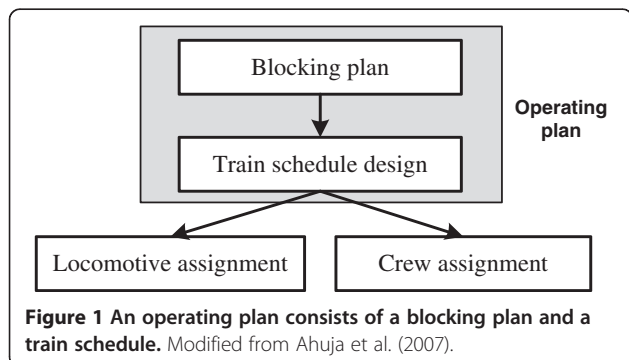
consists of designing train routes, days of operation, timings, and routing of blocks on trains to minimize the total system wide cost; this cost includes car hire, crew, and locomotive costs. An operating plan dictates the flow of three important railroad assets: crews, locomotives, and railcars. A well-designed operating plan can reduce the costs of railroads significantly (Ahuja et al. 2007).

Greedy heuristic methods produce solutions that are far from optimal and/or cannot handle real-life complexities of the blocking problem. As a result, no railroad company uses any optimization-based approach to determine blocking plans, but they rely on manual decision-making processes to make and change these plans. This manual process is very time intensive and does not save substantially on transportation costs. Consequently, we believe that there are major opportunities for cost saving using an optimization-based approach that can determine near-optimal blocking plans.

In this research, we used a population-based metaheuristic (p-metaheuristics) for solving RBP. The p-metaheuristics share many common concepts. They could be viewed as an iterative improvement in a population of solutions. First, the population is initialized; a new population of solutions is then generated. Finally, this new population is integrated into the current one, using some selection procedures. The search process is stopped when a given condition is satisfied (stopping criterion) (Talbi 2009).

We use a mathematical model for RBP wherein its decision variables identify a blocking scheme and a commodity assignment to this block. For the first time, we proposed a new genetic algorithm-based solution method for the model, and several genetic search operators are used to find optimal solutions. To implement the proposed algorithm, we utilized the Java programming language (Oracle Corporation, Redwood City, CA, USA), and to validate the proposed algorithm, we made several blocking problems randomly and compared the proposed algorithm results with the results of the CPLEX software (IBM Corporation, Armonk, NY, USA) for these problems. The evaluating of results revealed the effectiveness and efficiency of the proposed solution method for RBP.

The outline of this paper is as follows. In the 'Literature review' subsection, related work is reviewed. 'The railroad blocking problem' subsection introduces the railroad blocking problem. Two mathematical models that include nod-arc and path-based formulations are presented in 'The Mathematical formulation' subsection of the 'Methods' section. In 'The proposed algorithm' subsection, the proposed algorithm is described. A small example is solved with the proposed algorithm in the 'An example' subsection. The next two sections presented are 'Results and discussion' and 'Conclusions', respectively.



**Figure 1** An operating plan consists of a blocking plan and a train schedule. Modified from Ahuja et al. (2007).

## Literature review

Previous surveys by Cordeau et al. (1998) and Crainic (2000, 2002) reviewed optimization models for rail transportation.

One of the first models for blocking belongs to Bodin et al. (1980) who suggested a nonlinear mixed integer programming formulation of the problem. The model is a multicommodity flow problem with additional side constraints that simultaneously determine the optimal blocking strategies for all the classification yards in a railroad system.

Assad (1983) proposed a solution approach for a problem defined on a line network. The cars are received at the first yard in arbitrary order and must be separated as they proceed along the line to allow each successive yard to extract the traffic destined for it.

Van Dyke (1986, 1988) described an interactive, heuristic approach that attempts to improve an existing blocking plan by solving a series of shortest-path problems on a network whose arcs represent available blocks.

Keaton (1989) proposed a nonlinear mixed integer program (MIP) model and a heuristic method based on Lagrangian relaxation for the combined problem of blocking, train routing, and makeup. In a subsequent paper by Keaton (1992), pure strategy constraints for blocking and maximum transit times for each origin-destination pair are also considered.

Huntley et al. (1995) developed a simulated annealing approach, and Gorman (1998) proposed a hybrid tabu search and genetic algorithm for constructing operating plans including blocking, train connections, and block-to-train assignments.

One of the significant researches on RBP has been done by Newton (1996) and Newton et al. (1998). They modeled the RBP as a network design model and formulated it as an MIP. They developed column generation and branch-and-price algorithms to solve it. Barnhart et al. (2000) used the same formulation as Newton et al. (1998) and proposed the Lagrangian relaxation technique to decompose the problem into two subproblems. Their approaches focused on determining a near-optimal solution. However, the running times are not scalable as the problem increases in size; this prohibits their use in practice.

Ahuja et al. (2007) developed an algorithm using a technique known as very large-scale neighborhood (VLSN) search that is able to solve the problem to near optimality using 1 to 2 h of computer time on a standard workstation computer. Jha et al. (2008) presented two formulations for the block-to-train assignment problem: an arc-based formulation and a path-based formulation. The latter is generally smaller than the former, and it can better handle practical constraints. They also proposed exact and heuristic algorithms based on the path-based formulation.

Zhu et al. (2009) attempted to combine blocking, makeup, and empty car distribution in an integrated dynamic model. To do this, they built a two-layer network, one for block flows and another for car flows. The objective function includes two fixed costs on open service and block, and a variable cost on all blocks. The model has three types of capacity constraints on maximum cars in each train, maximum car on each block, and maximum block being built in each yard at each time. They used a tabu search metaheuristic to solve the proposed model. Yang et al. (2010) proposed a freight transportation model with mixed uncertainty of randomness and fuzziness. Table 1 represents the well-known research in this field.

## The railroad blocking problem

In this section, we describe the RBP and rail operations in general. In the RBP, the physical rail network (the railroad terminals and tracks) is already defined. The blocking network to be constructed is a virtual network that is overlaid on the physical network. The blocks are virtual express arcs which a commodity may take to have uninterrupted service between two terminals that are not necessarily connected by a physical link (Newton 1996).

On railroads, a typical general merchandise shipment may pass through many terminals on its route from origin to destination (OD). At these terminals, the incoming traffic is reclassified (sorted and grouped together) to be placed on outgoing trains. On average, each reclassification results in a 1-day delay for the shipment. In addition, the classification process is labor and capital intensive since many workers and large quantities of equipment are needed to sort the traffic, and construction and maintenance of large terminals are necessary to handle the sorting task. To prevent shipments from being reclassified at every terminal they pass through, several shipments may be grouped together to form a block. A block has associated with it an OD pair, which may or may not be the OD pair of any of the individual cars contained in the block. Once a shipment is placed in a block, it is not reclassified until it reaches the destination of that block (Newton 1996). The railroads develop the blocking plans that dictate which blocks should be built at each individual terminal and which traffic should be assigned to each block.

The objective of the RBP is to develop a feasible blocking plan which, in conjunction with other operating policies, minimizes the total cost of delivering the commodities. These costs are usually broken down into car-handling costs associated with handling (or blocking) a car and car-mile costs associated with the movement of a car. To solve the RBP, we must decide which blocks to include in the blocking plan and which blocks to use to

**Table 1 The prominent railroad car blocking research**

Authors	Problem type	Planning horizon	Objective function	Model structure	Solution approach
Bodin et al. (1980)	Blocking	Tactical	Minimum operating and delay costs	Nonlinear MIP	Heuristic
Assad (1983)	Blocking	Operational	Minimum total classification	Shortest path	Dynamic programming
Van Dyke (1986)	Blocking	Tactical	Minimum operating costs	Shortest path	Heuristic
Newton (1996)	Blocking	Tactical	Minimum operating costs	NDP with node budget	Dantzig-Wolfe decomposition
Barnhart et al. (2000)	Blocking	Tactical	Minimum operating costs	NDP with node budget	Lagrangian relaxation
Ahuja et al. (2007)	Blocking	Tactical	Minimum operating costs	Linear MIP	VLSN

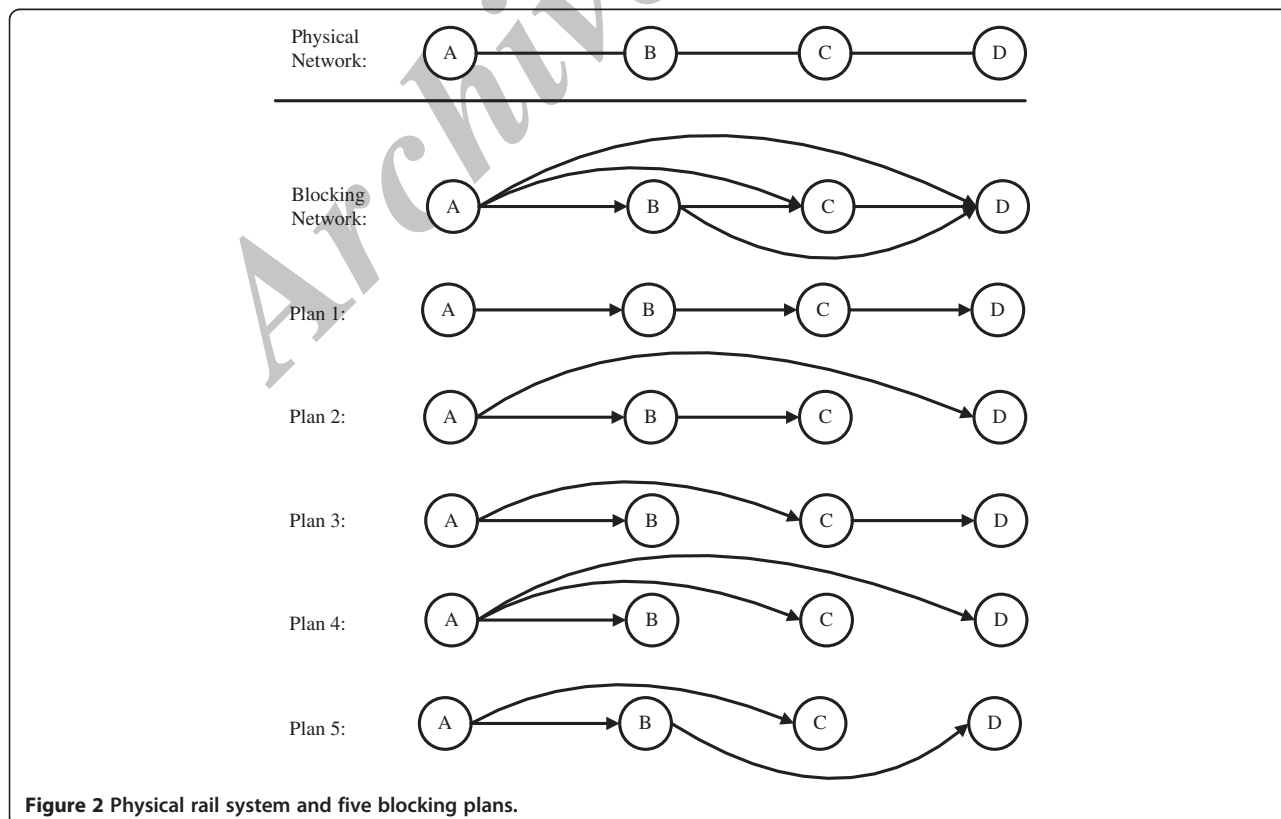
deliver each commodity (Newton 1996). An example of a rail system with four terminals is shown in Figure 2.

We define commodities as origin-destination pairs of terminals. We assume that the only commodities are  $A \rightarrow B$  with 100 cars,  $A \rightarrow C$  with 80 cars, and  $A \rightarrow D$  with 90 cars. Terminal A has a volume capacity of 270 cars, which is its minimum since 270 cars originate there. Terminals B and C can each block 90 cars. Terminal A can build two blocks, whereas each of the other terminals can only build one block.

At each terminal, limited classification resources restrict the number of cars (or car volume), which can be classified, and the number of blocks. In terms of our graph, the car volume constraints would be the upper bounds on the flow through each node, and the number of block constraints would be limits on the out-degree of each node. One performance metric of a blocking

plan is the total number of classifications necessary to deliver all commodities. We consider five different blocking plans for this example.

With blocking plan 1, at terminal A, all three commodities are sorted to move to terminal B. The resulting string (block) of cars may have the commodities intermingled. At terminal B, commodity  $A \rightarrow B$  has reached its destination and leaves the system. Commodities  $A \rightarrow C$  and  $A \rightarrow D$  are blocked to be move to terminal C. At terminal C, commodity  $A \rightarrow C$  has reached its destination and leaves the system, and commodity  $A \rightarrow D$  is blocked to move to terminal D. Using this blocking plan, the 100 cars for commodity  $A \rightarrow B$  use one block, (A, B), the 80 cars for commodity  $A \rightarrow C$  use two blocks, (A, B) and (B, C), and the 90 cars for commodity  $A \rightarrow D$  use three blocks, (A, B), (B, C), (C, D). Therefore, for blocking plan 1, there are 530 classifications; however,



**Figure 2 Physical rail system and five blocking plans.**

this plan is not feasible because it requires blocking 170 cars at terminal B that has a maximum volume of 90 cars.

Blocking plan 2 provides a direct block from terminal A to terminal D. Cars which travel in this block will still move along the physical track through terminals B and C, but since they are presorted for terminal D, they do not require classification resources at B or C. To deliver the three commodities using this blocking plan requires 350 classifications, since the 90 cars of commodity A → D now require only one block, (A, D). This blocking plan is feasible since it observes the limits on number and total volume of the blocks built at each terminal.

Similar calculations show that blocking plans 3, 4, and 5 require 360, 270, and 360 classifications, respectively. However, blocking plan 4 is infeasible since it requires three blocks for terminal A, which has a maximum capacity of two blocks. Commodities are always carrying directly to their destination, as in blocking plan 4.

Therefore, blocking plan 2 with 350 handlings is optimal with respect to the total number of classifications required. However, if the number of cars which could be blocked at B is reduced to less than 80 (or equivalently, the volume of commodity A → C were more than 80), then blocking plan 2 would be infeasible and blocking Plan 3 with 360 handlings would be optimal.

## Methods

### The mathematical formulation

The RBP can be modeled as a network design problem where the nodes represent the railroad terminals, and the arcs represent potential blocks (Newton 1996). In this section, we present a node-arc MIP for the RBP. The parameters and variables used in this model are as follows:

- Parameters

- $G = (N, A)$ : the graph with terminal set  $N$  and potential blocks set  $A$ .
- $K$ : the set of all commodities  $k$  designated by an origin–destination pair of nodes.
- $v^k$ : the volume of commodity  $k$ .
- $orig(k)$ : the origin terminal for commodity  $k$ .
- $dest(k)$ : the destination terminal for commodity  $k$ .
- $orig(a)$ : the origin of potential block  $a$ .
- $dest(a)$ : the destination of potential block  $a$ .
- $u_a$ : the capacity of potential block  $a$ .
- $c_a$ : the per unit cost of flow on arc  $a$ .
- $B(i)$ : the number of blocks which may originated at terminal  $i$ .
- $V(i)$ : the volume which may be classified at terminal  $i$ .

- Variables

- $x_a^k = 1$  if the commodity  $k$  is flowing on block  $a$ ; 0 otherwise.
- $y_a^k : y_a = 1$  if block  $a$  is included in the blocking network; 0 otherwise.

- Node-arc formulation

$$\min \sum_{k \in K} \sum_{a \in A} c_a v^k x_a^k \quad (1)$$

$$\text{st.} \quad \sum_{a \in A} x_a^k - \sum_{a \in A} x_a^k \quad (2)$$

$$\begin{aligned} \text{orig}(a) = i \quad \text{dest}(a) = i \\ = \begin{cases} 1 & \text{orig}(k) = i \\ -1 & \text{dest}(k) = i \forall i \in N, k \in K \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

$$\sum_{k \in K} v^k x_a^k \leq u_a y_a \quad \forall a \in A \quad (3)$$

$$\sum_{a \in A} y_a \leq B(i) \quad \forall i \in N \quad (4)$$

$$\begin{aligned} \text{orig}(a) = i \\ \sum_{k \in K} \sum_{a \in A} v^k x_a^k \leq V(i) \quad \forall i \in N \end{aligned} \quad (5)$$

$$y_a \in \{0, 1\} \quad \forall a \in A \quad (6)$$

$$x_a^k \in \{0, 1\} \quad \forall a \in A, k \in K \quad (7)$$

The objective is to minimize the sum of the costs of delivering each commodity using the blocking network formed by blocks for which  $y_a = 1$ . For each terminal, constraints (2) are balance equations for the flow of each commodity. For each potential block, constraints (3) prevent flow on blocks, which are not built and enforce the upper bound  $u_a$  on flow for blocks, which are built. The constraints (4) enforce the terminal limit  $B(i)$  for the sum of the blocks which leave the terminal. The constraints (5) model the volume of cars, which may be classified at each terminal.

We also present the path-based formulation for the RBP. The parameters and variables used in this model are as follows:

- Parameters  $Q(k)$ :

- the set of legal paths for commodity  $k$ .
- $orig(k)$ : the origin node for commodity  $k$ .
- $dest(k)$ : the destination node for commodity  $k$ .
- $PC_q^k$ : the path cost for flowing one unit of commodity  $k$  on path  $q$ .

$\delta_a^q$ : the binary design parameter;  $\delta_a^q = 1$  if  $q$ th path contain arc (block)  $a$  0 otherwise.

- Variables  $y_a$ :  
 the binary design variable for including arc (block)  $a$ :  $y_a = 1$  if arc  $a$  is selected; 0 otherwise.  
 $f_q^k$ : the binary flow variable;  $f_q^k = 1$  if commodity  $k$  flows on path  $q$  0 otherwise.

The path-based mathematical model for RBP is explained as flowing equations:

- Path based formulation

$$\min \sum_{k \in Q} \sum_{q \in Q(k)} PC_q^k v^k f_q^k \quad (8)$$

st.

$$\sum_{k \in Q} \sum_{q \in Q(k)} f_q^k v^k \delta_a^q - u_a y_a \leq 0 \forall a \in A \quad (9)$$

$$\sum_{q \in Q(k)} f_q^k = 1 \forall a \in A \quad (10)$$

$$\sum_{a \in A} y_a \leq B(i) \forall a \in A \quad (11)$$

$orig(a) = i$

$$\sum_{k \in K} \sum_{q \in Q(k)} \sum_{a \in A} v^k f_q^k \delta_a^q \leq V(i) \forall i \in N \quad (12)$$

$orig(a) = i$

$$f_q^k \in \{0, 1\} \forall q \in Q(k), k \in K \quad (13)$$

$$y_a \in \{0, 1\} \forall a \in A \quad (14)$$

Equation 8 is the objective function of this model; it minimizes total costs of delivering each commodity on paths because classification must be done on all commodities at its origin and destination. In this model, classification cost is not considered at that place for each commodity. Constraints (9) ensure that flow is only allowed on arcs (blocks) included in the network. Constraints (10) ensure that all of each commodity is delivered. Constraints (11) limit the number of blocks, which may be built at each node, which is modeled by the node-budget constraints. Constraints (12) limit the volume of cars, which may be classified at each terminal.

### The proposed algorithm

This section introduces the proposed p-metaheuristic algorithm for solving the RBP. The proposed solution method is based on genetic algorithms (GAs). GAs are one of the most famous p-metaheuristic algorithms

which have been developed by J. Holland in the 1970s (University of Michigan, USA) to understand the adaptive processes of natural systems; Then, they have been applied to optimization and machine learning in the 1980s. A GA usually applies a crossover operator to two solutions that plays a major role, plus a mutation operator that randomly modifies the individual contents to promote diversity. GA uses a probabilistic selection that is originally the proportional selection. The replacement (survivor selection) is generational, i.e., the parents are replaced systematically by the offsprings. The crossover operator is based on the  $n$ -point or uniform crossover while the mutation is bit flipping. A fixed probability  $p_m$  ( $p_c$ ) is applied to the mutation (crossover) operator (Talbi 2009). GAs have several advantages. Instead of single solution optimization, GAs use a population of solution simultaneously, so the probability of getting stuck in local minima is prevented; in comparison with other p-metaheuristic methods, GAs have high-quality solutions and good convergence speed.

### General structure of the proposed algorithm

The proposed algorithm has several stages. At the first stage, solutions (feasible and infeasible) are encoded into the chromosome. The idea of this stage is simulation and good representation of all possible solutions for the next stage. Encoding solutions into chromosomes is important to other stages. In the proposed algorithm, to find solutions in a reasonable time, each possible solution is encoded into strings of integer numbers (chromosome). To construct the initial population, each possible solution is created randomly, but at least one feasible solution is added to the initial population. There are two approaches to find a feasible initial solution. If the car blocking problem is implemented for a railway which has a specified blocking plan for each O-D pairs, this blocking plan could be used as initial feasible solution. However, if a car blocking problem is implemented for a railway without a specified blocking plan for each O-D pairs, it needs a heuristic algorithm to initialize feasible solutions. To present a feasible solution, one must consider two problem constraints (i.e., number of block and car volume). Figure 3 represents the process of this heuristic.

To do a comparison among chromosomes, we use a fitness function. A fitness value is assigned to each chromosome based on this function. This stage of the proposed algorithm for solving the problem is extremely important. Because it establishes a measure to indicate superiority or inferiority of each possible solution, the measure could attribute a real number to each possible solution; this number represents its degree of dominance, e.g., suppose  $f$  is a fitness function, and  $X_1$  and  $X_2$  are two chromosomes of a population and  $f(X_1) = 3$  and

- Step 1. O-D Pair Selection:** If there are unselected O-D pairs, choose an unselected O-D pairs randomly, otherwise go to Step 7;
- Step 2. Path Selection:** Select one legal path of the O-D pairs, randomly;
- Step 3. Assignment:** Among all possible commodities, assign those which could be delivered through this selected path;
- Step 4. Update:** Update number of block and car volume for the selected O-D pairs and its intermediate station;
- Step 5. Feasibility:** If this path is a feasible path then go to Step 1, if not, go to Step 2 and select another legal path. Otherwise if all path are infeasible go to Step 6;
- Step 6. Cancellation:** Unselect all O-D pair which this origin or destination is a part of it and go to Step 1;
- Step 7. End:** the end of method.

**Figure 3** The heuristic steps for finding a feasible solution.

$f(X_2) = 5$ , then  $X_2$  is better than  $X_1$ , or it has a greater fitness value.

After evaluating the fitness value of each chromosome of the initial population, the proposed GA randomly chooses two chromosomes according to the roulette wheel method; the chromosomes with greater fitness value are selected as the parents. In constructing a new generation of chromosomes, crossover and mutation

**Algorithm:** the GA algorithm for solving RBP

**Begin**

**Read** data;

**Create** paths for all commodities;

**Create** initial population;

**Identify**  $K$  best feasible solution;

**Repeat**

**Do** parent selection;

**Do** recombination;

**Do** mutation;

**Replace** current population with new population;

**Evaluate** fitness of chromosomes;

**Find** feasible solutions;

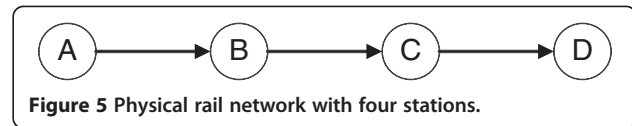
**Update**  $K$  best feasible solutions based on new population;

**Replace**  $K$  best feasible solutions in the new population;

**Until** termination condition is met;

**End**

**Figure 4** The pseudocode for the proposed algorithm.

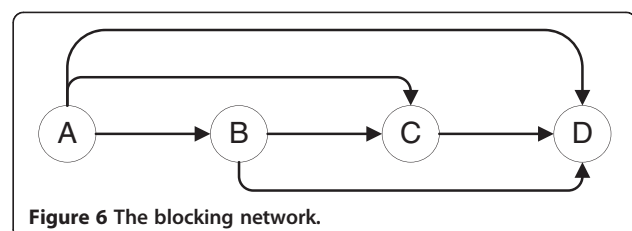


**Figure 5** Physical rail network with four stations.

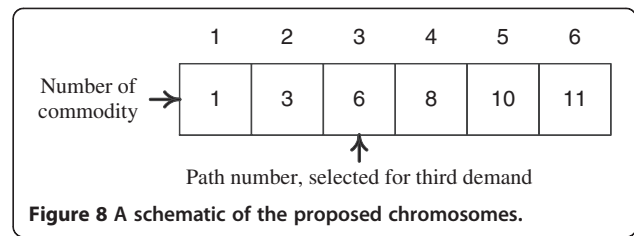
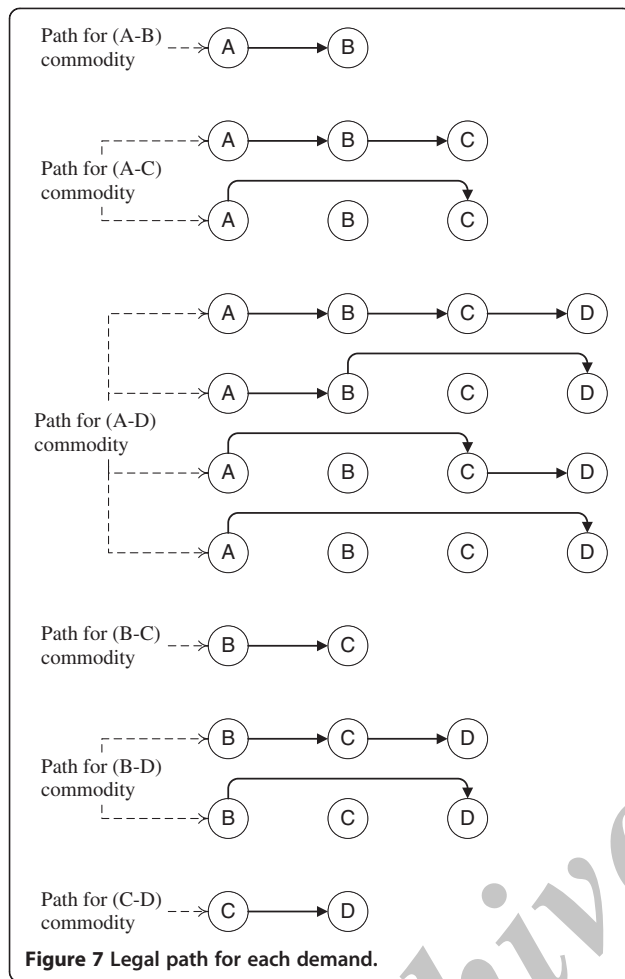
operators are utilized. The crossover operator first selects two parents by roulette wheel selection method and chooses a place on each parent as crossover point randomly. To make the first (second) child, the algorithm swaps the content of the left (right) side of each parent with another one. The mutation operator selects a chromosome from the population randomly and then selects one point on the chromosome randomly, and the content of its gene is changed according to the mutation probability ( $p_m$ ); this process is continued for all selected chromosome gene. After producing a new generation of chromosomes, the next generation is identified. In the stage of producing the next generation, the proposed algorithm keeps  $k$  as the best feasible solution of the current generation to include in the next generation. If  $m$  is the number of initial population, in order to create the next generation, the  $m$  and  $k$  chromosomes of new generation are initially selected randomly, and then,  $k$  is selected as the best feasible solution of the current population. Their union makes the next generation. The algorithm checks two termination conditions simultaneously: (1) if the iteration number does exceed  $N$  and (2) if after  $L$  iterations of the algorithm, the best solution so far does not improve. Figure 4 is a pseudocode for the proposed algorithm.

#### Parameter tuning

The metaheuristics are sensitive to the value of their parameters. The parameters are the configurable components of a metaheuristic algorithm. Parameter tuning may allow a larger flexibility and robustness to the algorithm but requires a careful initialization. Those parameters may have a great influence on the efficiency and effectiveness of the search. It is not obvious to define *a priori* which parameter setting should be used. The optimal values for the parameters depend mainly on the problem, and even the instance to deal with, and on the search time that the user wants to spend in solving the



**Figure 6** The blocking network.



problem. A universally optimal parameter value set for a given metaheuristic does not exist.

One main step of this research is fine parameter tuning of algorithm. In the proposed algorithm, there are three main parameters which include the number of individuals of a population, crossover, and mutation rate. We use a trial-and-error procedure to obtain locally optimal values for the parameters. To tune parameters, we generated three instances with different sizes. For tuning, only one parameter changed at that time, while all other parameters didn't change. Proper values of the parameters were then determined through running the algorithm 10 times over different values of the parameters and through calculating the average of the goal function for these 10 runs. The criteria for modifying parameters are the quality of solutions and time spent to find them. The final value for the parameter is as follows: the population size is  $6 \times$  the number of commodity, cross over rate is 0.7, and mutation rate is  $(1/\text{the number of commodity})$ .

#### An example

In the proposed method, each gene represents one commodity, and the value of this gene is the legal

**Table 2** Comparison results of the proposed algorithm with CPLEX

Name	Problem		CPLEX		GA algorithm		Percentage error
	Number of terminals	Number of commodities	Objective value	CPU time (s)	Objective value	CPU time (ms)	
SP1	5	3	5640	1000	5640	15	0%
SP2	5	5	11400	1000	11400	16	0%
SP3	5	10	20880	2000	20880	46	0%
SP4	10	5	11100	3000	11100	93	0%
SP5	10	10	23940	3000	23940	140	0%
SP6	10	31	60720	5000	60720	951	0%
SP7	20	105	8586	14000	8656	1,717	1%
SP8	20	195	13676	15000	13898	4,041	1.6%
SP9	30	133	10298	18000	10672	4,150	3.6%
SP10	30	292	21604	23000	22770	16,099	4.9%
SP11	50	220	15922	38000	16702	15,803	4.8%
SP12	50	521	39700	110000	39804	81,242	0.3%



blocking path number (on the blocking network) which this commodity can select. To determine this legal path for each commodity, a comprehensive preprocess is needed. In the preprocess procedure, the legal paths for each commodity, cost of each path, number and type of block on each path are determined.

Figure 5 represents a physical rail network with four stations and six demands ( $A \rightarrow B$ ,  $A \rightarrow C$ ,  $A \rightarrow D$ ,  $B \rightarrow C$ ,  $B \rightarrow D$ , and  $C \rightarrow D$ ). Demand  $A \rightarrow B$  is named (1),... , and demand  $C \rightarrow D$  is named (6), respectively.

On the first step, the blocking network is made with respect to the physical rail network and demand. Figure 6 represents the blocking network for the presented physical rail network. Each arc on this network represents a legal block from each station.

The next step is to compose legal paths of each demand as shown in Figure 7, and then delivering cost of cars on a legal path, which contains the shipment cost of cars on the physical path and reclassification cost at the intermediate terminal (except origins and destinations), is computed.

According to this legal path, the chromosomes of the proposed algorithm are created. Figure 8 represents a schematic of these chromosomes that we used to solve the problem.

To do a comparison among chromosomes, a fitness function is used as Equation 15, which  $X_i$  is a chromosome of the population,  $K$  is set of commodity,  $c_k^{X_i}$  is the cost of selected path at  $k$ th gene of  $X_i$  chromosome, and  $v^k$  is the volume of  $k$ th commodity:

$$f(X_i) = \sum_{k \in K} v^k c_k^{X_i} \quad (15)$$

Note that the proposed algorithm has a possibility to produce infeasible solutions (chromosomes). Hence, after producing an infeasible chromosome, a penalty term is considered; this term is added to the fitness value of infeasible solutions as an extra cost.

To create a new generation, we select some chromosomes as parents to produce children. This selection must give greater chances not only to chromosomes with lower cost, but also to chromosomes with greater cost because they may have some genes with high quality. To do this, we implemented the roulette wheel method. Suppose a circular plane that is able to circle around its central point. On this plane, consider some sections; the number of these section is equal to the number of the population, and the area of this section has a direct relationship with the selection probability of an individual chromosome. Beside this plane is a fixed point; the plane spins around its central point to stop. A section that is in front of this fixed point is selected

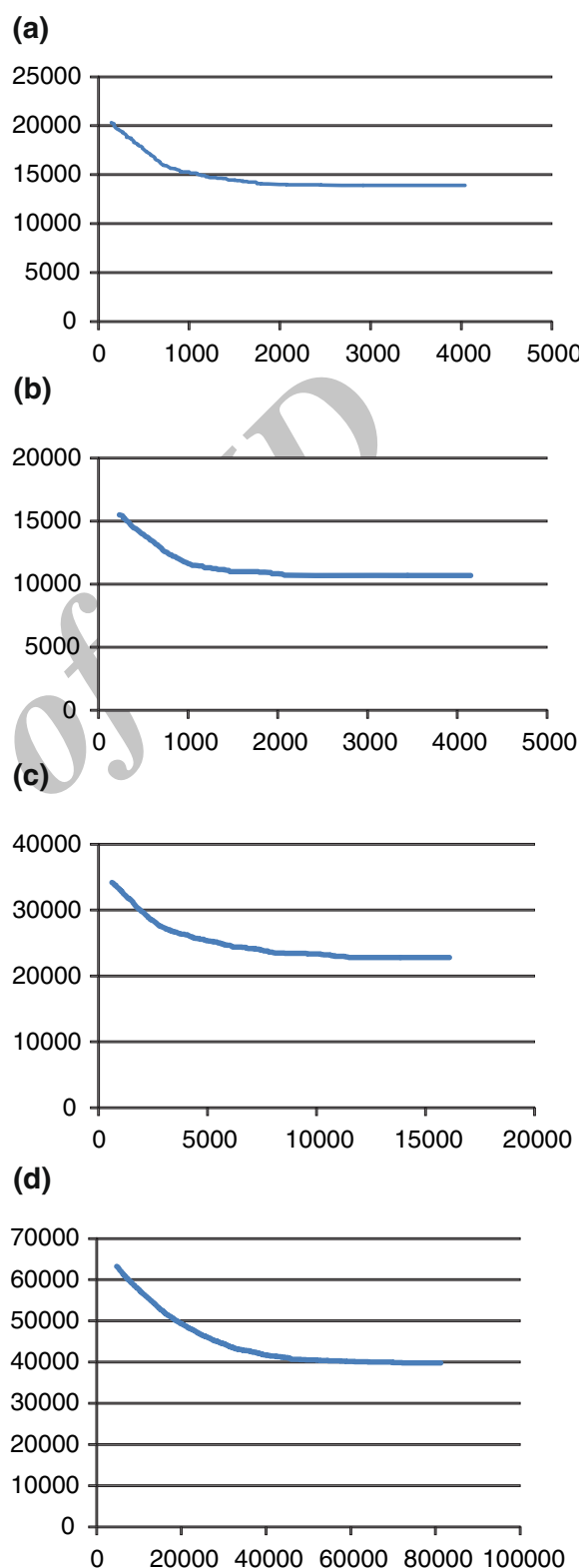


Figure 9 Time-cost graphs for (a) SP8, (b) SP9,(c) SP10, and (d) SP12.

when the plane stops spinning. To simulate the roulette wheel action, according to Equation 16, each chromosome of the current generation initially gets a number between 0 to 1.  $X_i$  is a chromosome;  $m$  is the number of population, and  $\text{sumfitness} = \sum_{i=1}^m f(x_i)$ .

$$(1 - (f(X_i)/\text{sumfitness})/(m - 1)). \quad (16)$$

A random number between 0 to 1 named  $r$  is selected, and  $n$  is selected as Equation 17.

$$\sum_{n=1}^{i=1} (1 - (f(X_i)/\text{sumfitness})/(m - 1)) \leq r \leq \sum_{m=1}^{i=1} (1 - (f(X_i)/\text{sumfitness})/(m - 1)). \quad (17)$$

After the parent selection, we use a one point crossover to produce a new generation. After selecting two parents from  $[1, L - 1]$  interval, this operator creates a random number (with uniform distribution) named  $k$ ;  $L$  is the length of the chromosome. The gene at  $k + 1, k + 2, \dots, L - 1$ , and  $L$  from the first parent is replaced by the corresponding gene at the second parent to create two new strings. After the crossover action, the mutation operator changes some genes of the chromosome. The mutation operator on chromosome  $X_i$  first creates a random number between 0 and 1; if this number is less than the mutation probability, then the algorithm changes the content of this gene. To change the value of gene  $k$ th of the chromosome, from interval  $[1, |Q(k)|]$ , an integer number is initially created randomly ( $|Q(k)|$  is the number of legal path for commodity  $k$ ). The algorithm then swaps the generated number with the previous one at gene  $k$ th of the chromosome.

## Results and discussion

The proposed algorithm is implemented with the use of the Java programming language, and a personal computer with Intel® Pentium® processor (2M cache, 2 GHz; Santa Clara, CA, USA), 32-bit Windows 7 Ultimate operating system (Microsoft Corporation, Redmond, WA, USA), and 2-GB installed memory (RAM) is used to achieve all of the results.

Capacity to classification operations in each terminal, commodity volume, commodities' origin and destination, and physical location of each terminal were all under consideration. Through the process, the main issue was to create a problem similar to real-world conditions as possible. The generated problems were solved using the proposed algorithm, which were implemented with Java programming language and CPLEX 11.1 software, separately. The CPLEX software is designed for large and complex problems. It is capable to solve linear

programming, quadratic programming, and integer programming problems. Each one of the simulated problems is solved using the proposed algorithm 10 times. Table 2 lists the average of the objective values and the average of CPU time spent for finding the solutions by the proposed algorithm.

The last column shows the error percentages. As illustrated in Table 2, the proposed algorithm found optimum solutions for problems SP1 to SP12 in a short time. The algorithm has superior performance than the CPLEX software with respect to time spent to generate the solutions. The error percentage for the first six of these problems is 0%, and for the six remaining is approximately near 0%.

Figure 9a,b,c,d represents convergence behavior of the proposed algorithm for some test problems to find optimal solution during the successive iteration. In this figure, (1) time was measured in one thousandth second (millisecond), and (2) the solid line represents the GA solutions.

The proposed algorithm checks two termination conditions simultaneously. In the first termination condition, if after 50 iterations there is not any improvement in the best-so-far solution, the algorithm stops. In the second termination condition, if the number of iterations exceeds 5,000 iterations, the algorithm stops.

## Conclusions

This paper, for the first time, introduced an approach based on genetic algorithm metaheuristic to solve the railroad blocking problem in a short reasonable time and high accuracy. The seven simulated problems were solved using the CPLEX software and the proposed algorithm, the solution accuracy, and the CPU time spent to find the solutions used to make comparison. The results show high efficiency and effectiveness of the proposed algorithms. It is possible to make significant savings in time and costs of operations through applying solutions produced by the algorithm. Finding feasible solutions for the algorithm is a challenging task with respect to great solution space. Future research on this paper could have two directions; one direction is to develop a heuristic algorithm to find initial feasible solution to speed up convergence, and the other direction is to use a hybrid genetic algorithm and ant colony optimization to enable the algorithm for solving more large-scale problems and decrease solving time of algorithm.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

MY has proposed the genetic algorithm for the railroad blocking problem. MS has worked on the modeling and the literature review; he also performed numerical experiments. MMK has generated the test problems and implemented the proposed genetic algorithm by Java programming language. All authors read and approved the final manuscript.

Received: 1 December 2010 Accepted: 3 March 2012  
Published: 25 July 2012

## References

- Ahuja RK, Jha KC, Liu J (2007) Solving real-life railroad blocking problems. *Interfaces* 37(5):404–419
- Assad AA (1980) Modeling of rail networks: toward a routing/makeup mode. *Transp Res* 14B:101–114
- Assad AA (1983) Analysis of rail classification policies. *Infor* 21:293–314
- Balakrishnan A, Magnanti TL, Mirchandani P (1997) Network design. In: Maffioli F, Martello S (eds) Dell 'Amico M. Annotated Bibliographies in Combinatorial Optimization. John Wiley, New York, pp 311–334
- Barnhart C, Jin H, Vance PH (2000) Railroad blocking: a network design application. *Oper Res* 48:603–614
- Bodin LD, Golden BL, Schuster AD, Romig W (1980) A model for the blocking of trains. *Transp Res* 14B:115–120
- Cordeau JF, Toth P, Vigo D (1998) A survey of optimization models for train routing and scheduling. *Transp Sci* 32:380–404
- Crainic TG (2000) Service network design in freight transportation. *Eur J Oper Res* 122(2):272–288
- Crainic TG (2002) A survey of optimization models for long-haul freight transportation. In: Hall RW (ed) *Handbook of Transportation Science*, 2nd edn. Kluwer, Boston
- Crainic TG (2006) Long-haul freight transportation. *International Series in Operations Research & Management Science* 56:451–516
- Crainic TG (2009) Service design models for rail intermodal transportation. *Lecture Notes in Economics and Mathematical Systems* 619(1):53–67
- Gorman MF (1998) An application of genetic and tabu searches to the freight railroad operating plan problem. *Ann Oper Res* 78:51–69
- Huntley CL, Brown DE, Sappington DE, Markowics BP (1995) Freight routing and scheduling at CSX transportation. *Interfaces* 25(3):58–71
- Jha KC, Ahuja RK, Guvenc S (2008) New approaches for solving the block-to-train assignment problem. *Networks* 51(1):48–62
- Keaton MH (1989) Designing optimal railroad operating plans: lagrangian relaxation and heuristic approaches. *Transp Res* 23B:415–431
- Keaton MH (1992) Designing railroad operating plans: a dual adjustment method for implementing lagrangian relaxation. *Transp Sci* 26:263–279
- Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. *Transp Sci* 18:1–55
- Minoux M (1989) Network synthesis and optimum network design problems: models, solution methods and applications. *Networks* 19:313–360
- Newton HN (1996) Network design under budget constraints with application to the railroad blocking problem. Auburn University, USA, Ph.D. dissertation
- Newton HN, Barnhart C, Vance PH (1998) Constructing railroad blocking plans to minimize handling costs. *Transp Sci* 32:330–345
- Talbi EG (2009) *Metaheuristics: from Design to Implementation*. Wiley, New Jersey
- Van Dyke CD (1986) The automated blocking model: a practical approach to freight railroad blocking plan development. *Transportation Research Forum* 27:116–121
- Van Dyke CD (1988) Dynamic management of railroad blocking plans. *Transportation Research Forum* 29:149–152
- Yang L, Gao Z, Li K (2010) Railway freight transportation planning with mixed uncertainty of randomness and fuzziness. *Appl Soft Comput* 11(1):778–792
- Zhu E, Crainic TG, Gendreau M (2009) Integrated Service Network Design for Freight Transportation. CIRRELT: Interuniversity Research Centre on Enterprise Networks. Logistics and Transportation, Montreal, Canada, Working Paper

doi:10.1186/2251-712X-8-8

**Cite this article as:** Yaghini et al.: A population-based algorithm for the railroad blocking problem. *Journal of Industrial Engineering International* 2012 **8**:8.