



Train Scheduling Problem - Phase II: A Simulation Integrated Genetic Algorithm

Özgür Yalçınkaya*

Industrial Engineering Department Dokuz Eylül University Engineering Faculty İzmir, Turkey

A B S T R A C T

In [1] a feasible timetable generator stochastic simulation modeling framework for the train scheduling problem was developed to obtain a train timetable which includes train arrival and departure times at all visited stations and calculated average train travel time for all trains in the system. In this study, the framework is integrated with a genetic algorithm (GA) in order to get an optimal or suboptimal feasible train timetable with minimum average train travel time.

Keywords: train; railway; scheduling; timetabling; rescheduling; dispatching; genetic algorithm; optimization

1. Introduction

Train scheduling (timetabling) problem is the problem of determining a timetable for a set of trains that do not violate track capacities and satisfies some operational constraints. A general train timetabling problem in the literature considers a single track linking two major stations with a number of intermediate stations in between [2].

In [1, 3 and 4] a feasible timetable generator stochastic simulation modelling framework for the problem was developed to obtain a timetable which includes train arrival and departure times at all visited stations and calculated average train travel time for all trains in the system. The inputs, random variables and output of the simulation model are exhibited in Fig. 1 in a black-box shape.

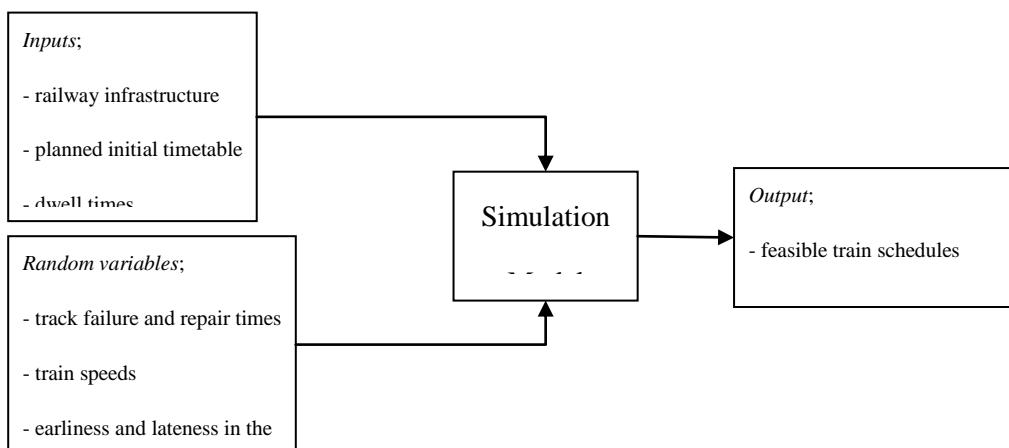


Figure 1. Black-box of the simulation mode

*Corresponding Author

Email Address: ozgur.yalcinkaya@deu.edu.tr

In this study, the previously developed framework is integrated with a GA in order to get an optimal or suboptimal feasible train timetable with minimum average train travel time. The chromosomes in the initial population of the GA are randomly generated, and the simulation model is used to evaluate the chromosomes and to find the fitness values of them. The proposed approach can be used for both the train scheduling (timetabling) problem and the train rescheduling (dispatching) problem.

The reason why we employ a GA by integrating it with simulation is that GA has been successfully adapted to solve several combinatorial optimization problems and become increasingly popular techniques among approximation techniques for finding optimal or suboptimal solutions in a reasonable time [5, 6, 7 and 8].

2. Literature review

In some related studies simulation models were integrated with GA. [9] presented a GA based approach to train dispatching problem. The objective was to minimize the total delay time. Their approach includes a model for the train dispatching on the lines with double tracks. By using this model, train dispatching can be optimized by adjusting the orders and times of trains' departure from stations. The authors also demonstrated efficiency of the method on a high speed railway via simulation. [10] presented a tool called DisTrain, dedicated to optimize railway dispatching and railway infrastructure, in order to help the dispatcher to reschedule trains if needed. The objective of the study was to minimize the number of delayed trains, GAs and branch and bound algorithm were used for solution. The following points should be regarded when using DisTrain: the dispatcher (or user) should be able to interact with the software and the proposed solutions should be dispatcher-oriented, and the number of changes from the previous schedule should be kept small, as well the complexity of the proposed solution (number of actions needed to run it). [11] focused on the railway scheduling problem and developed a constraint based deterministic simulation model with the objective of reducing the lateness of trains, and compared the results of the proposed sequentially train scheduling heuristic with those of a

GA. The first two studies [9 and 10] do not focus on the scheduling problem but rather focus on the rescheduling problem. The last one [11] develops a deterministic simulation model and integrates it with GA for dealing with the train scheduling problem. Our study differs from [11] in that we integrate GA with a stochastic simulation model not a deterministic model. Our stochastic simulation model reflects the real life stochastic nature of railway systems, since it includes stochastic track failure and repair times, train speeds between predetermined upper and lower limits, and additional unplanned delays at the stations.

Furthermore, while [10 and 11] do not contain the encoding, the most important part of a GA, the encoding used in [9] is directly dependent to the trains in the system. In addition, in their study, a chromosome describes train departure orders with real numbers. The encoding structure developed in our study does not depend on trains. Any changes in the number of trains or number of train meetings (encountering of trains that want to use the same track part but running in opposite directions), which increase the problem complexity, can be overcome easily by proposed encoding. On the other hand, solution space of a GA has two parts: feasible area and infeasible area [6]. The encoding developed here helps us to obtain chromosomes that indicate solutions in the feasible area of solution space in the GA part of our study. Thus, our encoding structure narrows the solution space by preventing us to deal with the infeasible area that will, no doubt, take a huge calculation time.

3. Simulation Integrated GA (SIMGA)

GAs are powerful and broadly applicable stochastic search and optimization approaches, and simulate the natural behaviour of biological systems. GAs have been successfully adapted to solve several combinatorial optimization problems for finding optimal or suboptimal solutions in a reasonable time. These algorithms start with an initial population that includes solutions to the problem at hand. The solutions are generally called chromosomes. Each chromosome is evaluated to obtain its fitness value. In order to create new chromosomes (children) some chromosomes (parents) from the population undergo stochastic

transformations by means of genetic operators; crossover and mutation. Crossover creates new children by combining parts from generally two parents. Mutation creates new individuals by making changes (mutating) in a single chromosome. A new population is formed by selecting chromosomes from the parent population and the children population according to a selection procedure. After several generations (a predefined iteration number), the algorithm converges the most fit chromosome, which represents an optimal or suboptimal solution to the problem at hand [12, 13, 14 and 15].

3.1. Representation

In the studied problem “which train, among the candidate trains, will be permitted to use the long track part between two consecutive real stations?” is an important question to be answered. To place the permission decision on a systematic way we used dispatching rules. We have m decision points if there are m main long track parts between the real stations. In our GA, a chromosome (a solution for the problem) in the initial population is composed of m genes and each gene is related with a decision point. That is, a dispatching rule is used to select one of the candidate trains that are waiting to use a long track.

The representation of a chromosome is given in Table 1. In this table, m denotes the number of long tracks that link the consecutive real stations on a corridor, and $(m+1)$ represents the number of the real stations. As a result a chromosome in the proposed GA has m genes. In Table 1, S_j ($j = 1, 2, \dots, m+1$) denotes the real station number and i_k ($k = 1, 2, \dots, m$) denotes the gene related with the k th decision point where m is the total number of genes in a chromosome.

Table 1. Representation of a chromosome

Decision points (relevant long tracks)	1 st	2 nd	... (m-1) th	m th
Location of the long tracks	S_1-S_2	S_2-S_3	...	$S_{m-1}-S_m$
Gene sequence of a chromosome	i_1	i_2	...	i_{m-1}

For instance, the first gene (i_1) in the chromosome denotes the dispatching rule that is used for the candidate trains waiting to use the long track part between the first (S_1) and the second (S_2) stations. A gene can take values in a range of $(i_k = 1, \dots, 6; k = 1, 2, \dots, m)$, and each gene value in the range (1,6) indicates a dispatching rule such that: 1 denotes the first come first served (FCFS) rule, 2 the last come first served (LCFS) rule, 3 the shortest current travelling time (ShrCTT) rule, 4 the longest current travelling time (LngCTT) rule, 5 the shortest remained track part (ShrRTP) rule, and 6 the longest remained track part (LngRTP) rule.

The chromosome representation used in the SimGA is flexible for both the changes in the railway infrastructure and the changes in the total number of dispatching rules. While the changes in the railway infrastructure can influence the length of a chromosome, the changes in the total number of dispatching rules can affect the value of a gene. Therefore, this chromosome representation can be adapted and used for several types of the problem, it is not matter which railway infrastructures these problems have and which dispatching rules they use. For a train schedule on a single track corridor, the problem is meeting of the trains while they are travelling in opposite directions in the system. With the encoding presented here, any changes in the number of trains or number of train meetings, which increases the problem complexity, can be overcome easily.

3.2 Initial Population and evaluation

The chromosomes in the initial population are randomly generated. After generating a chromosome, the dispatching rules in the simulation model that are used for the candidate trains waiting in the queues of the long track parts are rearranged according to the gene values of the chromosome. Then, the simulation model is run for n replications and the average train travel time value of t trains is noted to be the fitness value of the chromosome. The procedure for generating the initial population and evaluation is given in Table 2.

Table 2. Procedure for generating the initial population and evaluation

Begin SimGA
Generate INITIAL POPULATION
PS: population size
From $k = 1$ to $k = PS$
Randomly generate a chromosome
Chr(k): k^{th} chromosome in population
From $M = 1$ to $M = m$
i_M ; the gene related with M^{th} decision point
i_M = takes a random value from the set $D = \{1, 2, 3, 4, 5, 6\}$
Next M
$\text{Chr}(k) = [i_1, i_2, \dots, i_{m-1}, i_m]$
Rearrange the simulation model
From $M = 1$ to $M = m$
If $i_M = 1$ rearrange $Q(M)$ rule as FCFS
If $i_M = 2$ rearrange $Q(M)$ rule as LCFS
If $i_M = 3$ rearrange $Q(M)$ rule as ShrCTT
If $i_M = 4$ rearrange $Q(M)$ rule as LngCTT
If $i_M = 5$ rearrange $Q(M)$ rule as ShrRTP
If $i_M = 6$ rearrange $Q(M)$ rule as LngRTP
Next M
Run the rearranged simulation model
From $N = 1$ to $N = n$
Find the average train travel time of t trains for the N^{th} replication
Next N
Calculate the fitness value
Fitness; the average of n average train travel time values
Next k
Rank the initial population
Rank the initial population according to the fitness values of the chromosomes
Select best of the initial population
Record the best fitness value and related chromosome

3.3. Parent selection and crossover

The procedure for parent selection and crossover is exhibited in Table 3. At first, the parents are selected from the previous generation according to their fitness values. The best one is selected to be the first mother and the second best is selected to be the first father. Two children are obtained from the parents based on the single point crossover. Then, the third best one is selected to be the mother and the fourth best one to be the father of the new family. Two children are also attained from these parents by the single point crossover. Crossover goes on according to the predetermined crossover rate. The simulation model is rearranged for every child, the model is run for n replications, and the average train travel time value of t trains is held to be the fitness value of the child.

Table 3. Procedure for crossover

GN; generation number
maxGN; maximum generation number
From $GN = 1$ to $GN = maxGN$
CROSSOVER
CR; crossover rate
nCrs; the number of chromosomes selected for crossover
nCrs = $PS * CR$
From $k = 1$ to $k = nCrs - 1$
rChr(k): k^{th} chromosome in the ranked population of the previous generation
Select the parents
rChr(k); the mother
rChr($k+1$); the father
Implement the single point crossover and obtain two children
From $c = 1$ to $c = 2$
cChr($k,k+1,c$): c^{th} child from the parents $rChr(k)$ and $rChr(k+1)$
cChr($k,k+1,c$) = $[i_1, i_2, \dots, i_{m-1}, i_m]$
Rearrange the simulation model (Steps are in Table 2)
Run the rearranged simulation model (Steps are in Table 2)
Calculate the fitness value
Next c
$k = k + 1$
Next k

3.4. Mutation

The procedure for mutation is denoted in Table 4. At first, a child is selected randomly for mutation. Next, a gene of the child is selected randomly and this gene is changed with one of the other potential value of the gene. The new gene value is randomly selected from the set $D = \{1, 2, 3, 4, 5, 6\}$ except the current value of the gene. In this set each value indicates a dispatching rule. Then, the simulation model is rearranged, the model is run for n replications, and the average train travel time value of t trains is noted to be the fitness of the mutated child. Mutation goes on according to the predetermined mutation rate.

3.5. Termination criteria and replacement strategy

The procedure for termination and replacement strategy is denoted in Table 5. By using the termination criteria procedure, if it is desired, the SimGA can be stopped at a generation that has the same best fitness value with a predetermined number of previous generations. The replacement is made according to the elitism strategy. Elitism strategy makes a number of the best individuals at each generation survive. In order to

form the new generation, the parents, children and mutated children are ranked according to their fitness values. The newly generated population is formed by using the ranked population adhered to the prescribed population size.

Table 4. Procedure for mutation

MUTATION	
MR; mutation rate	
nMtt; the number of chromosomes selected for mutation	
nMtt = PS * MR	
From $k = 1$ to $k = nMtt$	
Select a child randomly	
cChr(k); the randomly selected child for mutation	
cChr(k) = [$i_1, i_2, \dots, i_{m-1}, i_m$]	
Select a gene of the child randomly	
i_j ; the selected gene related with j^{th} decision point, $j = 1, \dots, m$	
If $i_j = 1$ change the i_j with a random value from the set E1 = {2, 3, 4, 5, 6}	
If $i_j = 2$ change the i_j with a random value from the set E2 = {1, 3, 4, 5, 6}	
If $i_j = 3$ change the i_j with a random value from the set E3 = {1, 2, 4, 5, 6}	
If $i_j = 4$ change the i_j with a random value from the set E4 = {1, 2, 3, 5, 6}	
If $i_j = 5$ change the i_j with a random value from the set E5 = {1, 2, 3, 4, 6}	
If $i_j = 6$ change the i_j with a random value from the set E6 = {1, 2, 3, 4, 5}	
mcChr(k); the mutated child	
mcChr(k) = [$i_1, i_2, \dots, i_{m-1}, i_m$]	
Rearrange the simulation model (Steps are in Table 2)	
Run the rearranged simulation model (Steps are in Table 2)	
Calculate the fitness value	
Next k	
Rank the population	
The current population includes; individuals from the previous generation population, children and mutated child(ren)	
Select best of the population	
Record the best fitness value and related chromosome	

Table 5. Procedure for termination criteria and replacement strategy

Check TERMINATION CRITERIA	
TrmCnt; termination counter	
maxTrmCnt; the maximum termination counter	
bFtV(GN); the best fitness value of the current generation GN	
If bFtV(GN) = bFtV(GN-1)	
TrmCnt = TrmCnt + 1	
If TrmCnt < (maxTrmCnt - 1) go Next GN	
Else go Stop SimGA	
Else TrmCnt = 0	
Form the population of the next generation	
Form by using the ranked population adhered to the PS	
Next GN	
Stop SimGA	

Framework of the SimGA for the problem is briefly exhibited in Table 6.

Table 6. Framework of the SIMGA

Begin SimGA	
Generate INITIAL POPULATION (Steps are in Table 2)	
Rank the initial population	
Rank the initial population according to the fitness values of the chromosomes	
Select best of the initial population	
Record the best fitness value and related chromosome	
GN; generation number	
maxGN; maximum generation number	
From GN = 1 to GN = maxGN	
CROSSOVER (Steps are in Table 3)	
MUTATION (Steps are in Table 4)	
Rank the population	
The current population includes; individuals from the previous generation population, children and mutated child(ren)	
Select best of the population	
Record the best fitness value and related chromosome	
Check TERMINATION CRITERIA (Steps are in Table 5)	
Form the population of the next generation	
Form by using the ranked population adhered to the PS	
Next GN	
Stop SimGA	

4. Research results

Genetic operators for the hypothetic problem are as follows. Since there are nine main track parts ($= m$) between the 10 real stations in our problem, a chromosome (a solution) is composed of nine genes, and each gene is related with a decision point. As explained above, a gene can take values in a range of (1,6), and each value in this range indicates a dispatching rule. The first gene in the chromosome is the dispatching rule that is used for the candidate trains waiting to use the track between the S_1 and the S_2 , the second gene is the dispatching rule that is used for the candidate trains waiting to use the track between the S_2 and the S_3 and so on. For an instance a chromosome structure is exhibited in Table 7. In this table, the first gene value 1 means that the dispatching rule used for the candidate trains waiting to use the track between the S_1 and the S_2 is FCFS rule, the fourth gene value 2 means that the dispatching rule used for the candidate trains waiting to use the track between the S_4 and the S_5 is LCFS rule.

For this chromosome representation, there are $6^9 = 10,077,696$ possible feasible solutions for the studied problem. Since the GA presented in this study are

simulation integrated, and the simulation model is run in order to calculate the fitness value for each chromosome, we have to make more than one simulation replications. In order to make 20 ($= n$) replications for each possible feasible solution, totally 201,553,920 replications are needed to obtain fitness values of all the possible feasible solutions.

Table 7. Representation of a chromosome for the studied problem

Decision points (relevant long tracks)	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th
Location of the long tracks	S ₁ - S ₂	S ₂ - S ₃	S ₃ - S ₄	S ₄ - S ₅	S ₅ - S ₆	S ₆ - S ₇	S ₇ - S ₈	S ₈ - S ₉	S ₉ - S ₁₀
Gene sequence of a chromosome	1	3	4	2	5	1	6	2	4

For different values of the genetic operators, the population size (PS), the crossover rate (CR) and the mutation rate (MR), the *SimGA* was run, and the observed best average travel time values are recorded. The utilized values for operators are: 10, 20 and 30 for PS, 40%, 60% and 80% for CR, and 10%, 20% and 30% for MR. Since there are $3 \times 3 \times 3 = 27$ different combinations of the utilized values, these combinations will be represented by PS/CR/MR for short. The presented algorithm was run for each PS/CR/MR combination, and obtained results are displayed in Table 8.

As can be seen in Table 8, the same best average travel time value, 20742.93 seconds, is obtained by each combination. The earliest generation number, the number of the generation in which the best solution is reached, is six and appeared in 10/0.8/0.3 and 30/0.8/0.1 combinations. The latest generation number in which the best is observed is 37, and appeared in the combination 10/0.6/0.1.

The obtained results are also exhibited in Fig. 2 in order to make the comparison more easily. The circles in Fig. 2 indicate that the MR is 10% in the related combination, on the other hand the rectangles indicate that the MR is 20%, lastly the triangles mean that the MR is 30%. For instance, in this figure, the first point denotes the GN belongs to combination 10/0.4/0.1 and equals to 25.

Table 8. Result of the SIMGA

PS	CR	MR	Chromosome	Fitness value	GN
1	10	0.40	0.10 6,3,4,2,4,4,4,1,4	20742.93	25
2	10	0.40	0.20 1,3,1,2,4,4,5,6,4	20742.93	16
3	10	0.40	0.30 1,3,1,2,4,4,2,6,4	20742.93	13
4	10	0.60	0.10 6,3,4,2,2,4,4,1,4	20742.93	37
5	10	0.60	0.20 1,3,5,2,4,4,5,1,4	20742.93	27
6	10	0.60	0.30 6,3,4,2,2,5,5,1,4	20742.93	13
7	10	0.80	0.10 6,3,5,2,4,4,5,3,4	20742.93	31
8	10	0.80	0.20 1,6,5,2,4,4,5,1,4	20742.93	20
9	10	0.80	0.30 6,3,4,2,2,4,4,1,4	20742.93	6
10	20	0.40	0.10 6,3,1,2,2,4,2,6,4	20742.93	19
11	20	0.40	0.20 3,6,5,2,5,4,4,1,4	20742.93	12
12	20	0.40	0.30 6,3,1,2,5,1,4,6,4	20742.93	13
13	20	0.60	0.10 6,3,1,2,4,4,5,1,4	20742.93	17
14	20	0.60	0.20 6,3,5,2,5,1,2,1,4	20742.93	15
15	20	0.60	0.30 1,3,1,2,4,4,5,1,5	20742.93	24
16	20	0.80	0.10 6,3,5,2,2,1,2,1,4	20742.93	10
17	20	0.80	0.20 3,3,5,2,4,1,5,6,4	20742.93	17
18	20	0.80	0.30 1,3,1,2,4,5,4,6,4	20742.93	9
19	30	0.40	0.10 6,3,4,2,5,1,2,1,4	20742.93	12
20	30	0.40	0.20 6,3,4,2,4,5,2,6,4	20742.93	7
21	30	0.40	0.30 6,3,1,2,4,1,2,1,4	20742.93	7
22	30	0.60	0.10 6,3,4,2,4,1,2,1,4	20742.93	24
23	30	0.60	0.20 6,3,5,2,2,4,2,1,4	20742.93	7
24	30	0.60	0.30 6,3,5,2,4,1,2,3,4	20742.93	8
25	30	0.80	0.10 6,3,5,2,4,4,2,3,4	20742.93	6
26	30	0.80	0.20 6,3,1,2,5,4,2,3,4	20742.93	11
27	30	0.80	0.30 6,3,4,2,4,1,2,3,4	20742.93	12

GA is a member of generate-and-test algorithms. Researchers generally run their GA by different operator values to observe the effect of operators, and to find best operators combination. Therefore, the best operator values strictly depend on the problem at hand, as being in the current study.

The obtained best fitness values in generations for 10/0.8/0.3 and 30/0.8/0.1 combinations are depicted in Figure 3. The best is reached in four steps (in the 1st, 2nd, 3rd and 6th generations) in the combination 10/0.8/0.3. On the other hand an improvement is occurred in every successive generation (six steps) until the best is reached in the combination 30/0.8/0.1. In Figure 4, the results for the combination 10/0.6/0.1 are exhibited. As can be seen the best is reached in five steps (in the 1st, 10th, 13th, 34th and 37th generations).

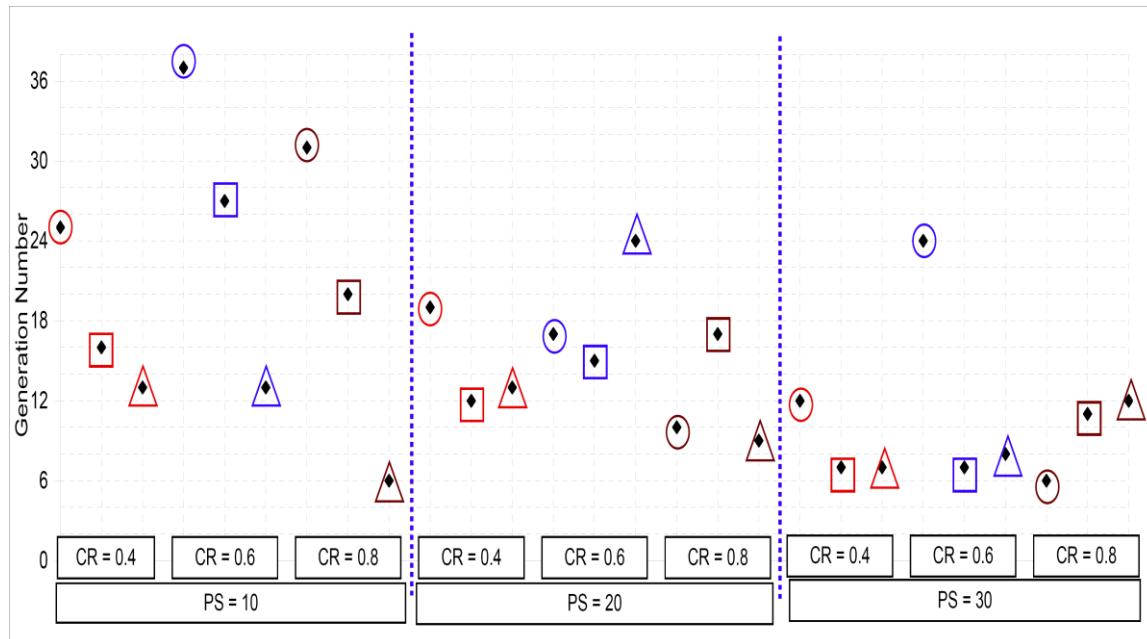


Figure 2. The best GNs of the SimGA

Fig. 3 and Fig. 4 indicate that the developed SimGA has an ability to escape from local optima which have very close fitness values to the best. In addition it can be seen that the visited local optima are different from each other. This means that the different operator values result a good exploration on the solution space of the SimGA.

In [1 and 3] after the simulation modelling framework was developed, the result of a deterministic scenario in which all inputs were deterministic and the dispatching rule was FCFS for all decision points was discussed. The deterministic scenario was run and the average train travel time was calculated as 17956 seconds. Then, the behaviour of the framework in stochastic nature on

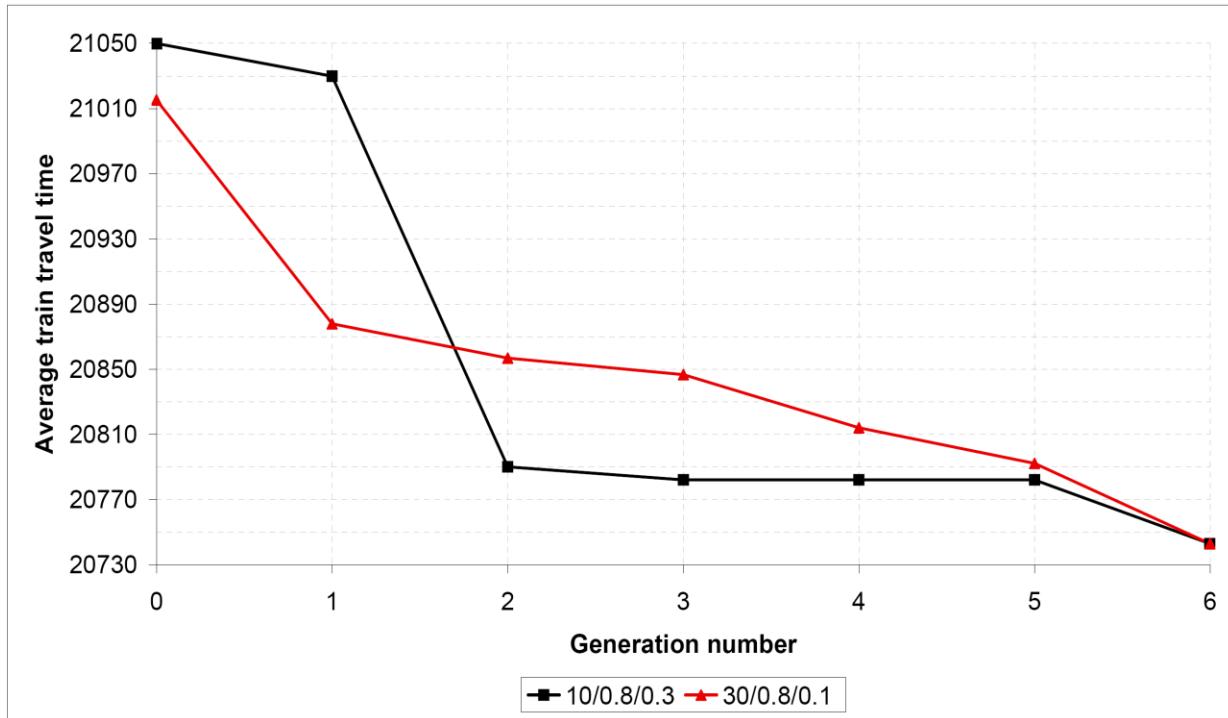


Figure 3. The best fitness values in generations for 10/0.8/0.3 and 30/0.8/0.1 combinations

one replication of the stochastic scenario was discussed, and the average train travel time was calculated as 24218 seconds.

of train meetings, which make the problem more complex, can be easily overcome. The developed encoding helps researchers to find solutions in the

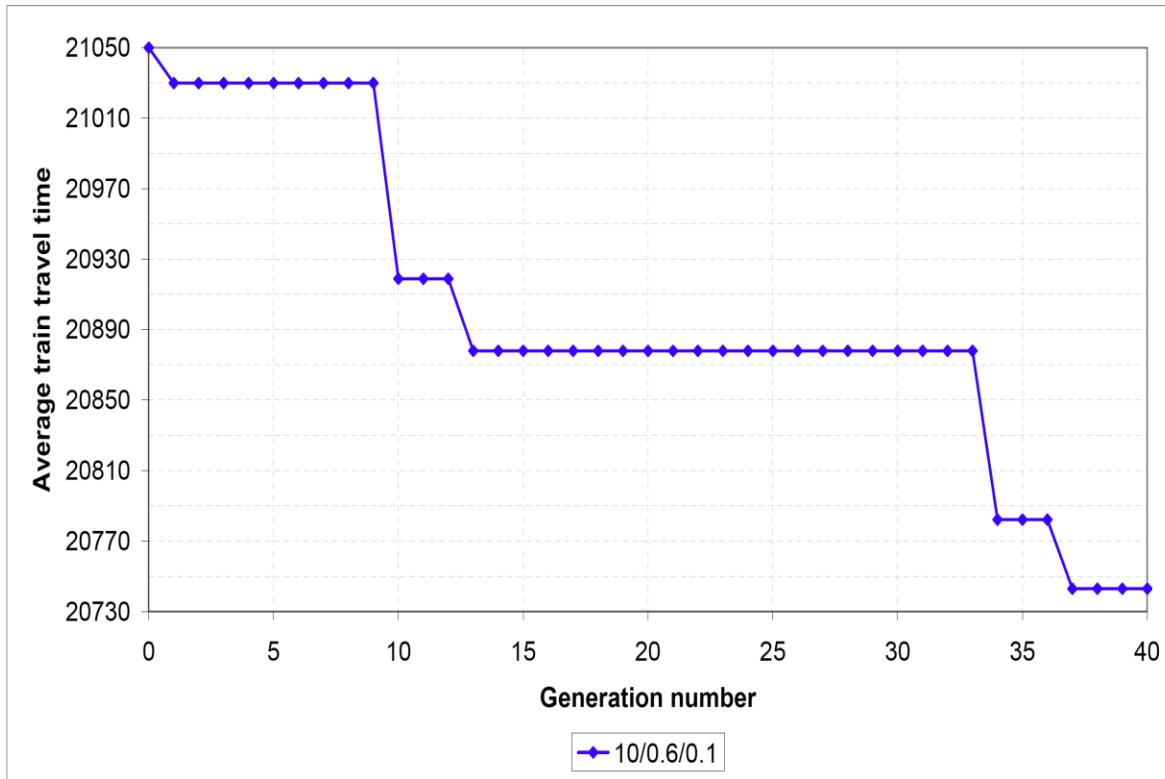


Figure 4. The best fitness values in generations for 10/0.6/0.1 combinations

In order to make comparison, in this study we replicate the stochastic simulation model 20 times with FCFS dispatching rule for all decision points. The fitness value of this chromosome is calculated as 21408 seconds by the stochastic simulation model.

While the simulation model produces 21408 seconds average train travel time, we reached 20742.93 seconds average train travel time by the SimGA. As a result, the SimGA provided 3.11% improvement on the average travel time value in a short running time.

5. Conclusion

In this study, our aim is to obtain a feasible train timetable that includes arrival and departure times of all trains at the visited stations with minimum average train travel time. By this aim, we integrated a GA with the previously developed framework [1] to get a feasible train timetable with minimum average train travel time. Using our proposed encoding structure for our GA, any changes in the number of trains or number

feasible area of GA's solution space, and consequently to reduce the solution time of the SimGA.

We see that the SimGA results 3.11% improvement on the average travel time value in a short running time. We conducted several experiments with different genetic operator values and saw that the SimGA has an ability to reach the best solution in a reasonable time. On the other hand the SimGA shows a good characteristic of easily escaping from local optimums which have fitness values very close to the best.

Acknowledgments

This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK)

References

- [1] Ö. Yalçınkaya, Train scheduling problem - Phase I: A General Simulation Modelling Framework, International Journal of Railway Research, 1 (2) (2014) 23-30.

- [2] A. Caprara, M. Fischetti, P. Toth, Modeling and solving the train timetabling problem, *Operations Research*, 50 (2002) 851-861.
- [3] O. Yalçınkaya, G. M. Bayhan, A feasible timetable generator simulation modelling framework for train scheduling problem, *Simulation Modelling Practice and Theory*, 20 (1) (2012) 124-141.
- [4] Ö. Yalçınkaya, A feasible timetable generator simulation modelling framework and simulation integrated genetic and hybrid genetic algorithms for train scheduling problem, PhD Thesis, The Graduate School of Natural and Applied Sciences of Dokuz Eylül University 2010.
- [5] M. Gen, and R. Cheng, *Genetic algorithms and engineering design*, New York: John Wiley & Sons, Inc 1997.
- [6] M. Gen, and R. Cheng, *Genetic algorithms and engineering optimization*, New York: John Wiley & Sons, Inc 2000.
- [7] M. Gen, R. Cheng, and L. Lin, *Network models and optimization: Multiobjective genetic algorithm approach*, London: Springer 2008.
- [8] X. Yu, and M. Gen, *Introduction to evolutionary algorithms*, London: Springer 2010.
- [9] L. Ping, N. Axin, J. Limin, and W. Fuzhang, Study on intelligent train dispatching, *IEEE Intelligent Transportation Systems Conference*, (2001) 949-953.
- [10] P. Rebreyend, DisTrain: A simulation tool for train dispatching, *8th International IEEE Conference on Intelligent Transportation Systems*, (2005) 801-806.
- [11] U. Geske, Railway scheduling with declarative constraint programming, *Lecture Notes in Artificial Intelligence*, 4369 (2006) 117-134.
- [12] J. H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan Press 1975.
- [13] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Massachusetts: Addison Wesley Longman, Inc 1989.
- [14] D. A. Coley, *An introduction to genetic algorithms for scientists and Engineers* (3rd ed.), Singapore: World Scientific Press 2003.
- [15] R. L. Haupt, and S. E. Haupt, *Practical genetic algorithms* (2nd ed.), New Jersey: John Wiley & Sons, Inc 2004.