DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A FEASIBLE TIMETABLE GENERATOR SIMULATION MODELLING FRAMEWORK AND SIMULATION INTEGRATED GENETIC AND HYBRID GENETIC ALGORITHMS FOR TRAIN SCHEDULING PROBLEM

by

Özgür YALÇINKAYA

October, 2010

İZMİR

A FEASIBLE TIMETABLE GENERATOR SIMULATION MODELLING FRAMEWORK AND SIMULATION INTEGRATED GENETIC AND HYBRID GENETIC ALGORITHMS FOR TRAIN SCHEDULING PROBLEM

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Industrial Engineering, Industrial Engineering Program

by
Özgür YALÇINKAYA

October, 2010
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A FEASIBLE TIMETABLE GENERATOR SIMULATION MODELLING FRAMEWORK AND SIMULATION INTEGRATED GENETIC AND HYBRID GENETIC ALGORITHMS FOR TRAIN SCHEDULING PROBLEM"** completed by **ÖZGÜR YALÇINKAYA** under supervision of **PROF.DR. G. MİRAÇ BAYHAN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

....

Prof.Dr. G. Miraç BAYHAN

Supervisor

....

Prof.Dr. A. Nihat BADEM

Thesis Committee Member

....

Assist.Prof.Dr. Güleser KALAYCI DEMİR

Thesis Committee Member

....

Examining Committee Member

....

Examining Committee Member

Prof.Dr. Mustafa SABUNCU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First of all, I would like to express my deepest thanks and gratitude to my supervisor Prof.Dr. G. Miraç BAYHAN for her guidance, support, encouragement and valuable advice throughout the progress of this PhD dissertation. No doubt, without her valuable effort, it would be much more difficult for me to reach my targets.

I would like to express my sincere thanks to precious members of my thesis committee Prof.Dr. A. Nihat BADEM and Assist.Prof.Dr. Güleser KALAYCI DEMİR for their consciousness expanding comments and valuable suggestions throughout the progress of this dissertation.

This study has been supported by the TÜBİTAK- BİDEB in the scope of “2211- National Ph.D. Scholarship Programme”.

I would like to thank my friends Assist.Prof.Dr. Güzin ÖZDAĞOĞLU and Research Assist. Mehmet Ali ILGIN for their support and friendship.

I would like to express my thanks to all the professors and colleagues in the Industrial Engineering Department of Dokuz Eylül University for their supports, encouragements and understandings. I also would like to thank all my companions in Chamber of Mechanical Engineers for their friendship and supports.

Finally, I would like to express my special thanks and deep appreciation to my wife Esen for her love, companion, endless support, understanding and patience, and to my father Hasan, mother Nebahat and elder sister Sevda for their love, trust, and endless supports in my whole life.

Özgür YALÇINKAYA

A FEASIBLE TIMETABLE GENERATOR SIMULATION MODELLING FRAMEWORK AND SIMULATION INTEGRATED GENETIC AND HYBRID GENETIC ALGORITHMS FOR TRAIN SCHEDULING PROBLEM

ABSTRACT

An important problem in management of railway systems is train scheduling problem (TrnSchPrb). This is the problem of determining a timetable for a set of trains that does not violate track capacities and satisfies some operational constraints. In this thesis, a feasible timetable generator stochastic simulation modelling framework is developed. The objective is to obtain a feasible train timetable for all trains in the system. The feasible train timetable includes train arrival and departure times at all visited stations with calculated average train travel time. In addition to obtaining a feasible timetable, hybrid algorithms are developed with the objective of minimizing the average train travel time. The first hybrid is obtained by integrating simulation and genetic algorithm (GA), and the other three hybrids are obtained by embedding each of three local search algorithms in simulation integrated GA. The simulation modelling framework developed in this thesis is implemented for a TrnSchPrb based on an infrastructure which was inspired by a real railway line system with single track corridor. The set of feasible train timetables found by simulation forms the initial solution space of the developed hybrid GAs. These hybrid GAs are run for getting a feasible train timetable with optimum average train travel time. The optimum average train travel times found by the hybrid GAs are compared, and the results are discussed. Although this thesis focuses on train scheduling/timetabling problem, the developed simulation integrated framework can also be used for train rescheduling/dispatching problem if this framework can be fed by real time data. Since the developed simulation model includes stochastic events, and this model can easily cope with the disturbances occur in the railway system.

Keywords: Train, Scheduling, Timetabling, Rescheduling, Optimization, Simulation Integrated GA, Simulation Integrated Hybrid GA.

TREN ÇİZELGELEME PROBLEMİ İÇİN BİR OLURLU TARİFE ÜRETİCİ BENZETİM MODELLEME YAPISI VE BENZETİMLE BÜTÜNLEŞİK GENETİK VE MELEZ GENETİK ALGORİTMALAR

ÖZ

Demiryolu sistemlerinin yönetiminde önemli bir problem tren çizelgeleme problemidir (TrnÇzgPrb). Bu bir küme tren için ray kapasitelerini ihlal etmeyen ve bazı eylemsel kısıtları tatmin eden bir tarife belirleme problemidir. Bu tezde, bir olurlu tarife üretici stokastik benzetim modelleme yapısı geliştirilmiştir. Amaç sistemdeki tüm trenler için bir olurlu tren tarifesi elde etmektir. Olurlu tren tarifesi hesaplanmış ortalama tren seyahat süresi ile birlikte tüm ziyaret edilen istasyonlar için tren geliş ve hareket zamanlarını içerir. Bir olurlu tarife elde etmenin yanında, ortalama tren seyahat süresini minimize etmek amacıyla melez algoritmalar geliştirilmiştir. İlk melez, benzetim ve genetik algoritma (GA) bütünleştirilerek elde edilmiştir, diğer üç melez ise üç yerel arama algoritmasından her birinin benzetimle bütünleşik GA içerisine gömülmesiyle elde edilmiştir. Bu tezde geliştirilen benzetim modelleme yapısı gerçek bir demiryolu hat sisteminden esinlenmiş altyapı tabanlı bir tek ray koridorlu TrnÇzgPrb için uygulanmıştır. Benzetim tarafından bulunan olurlu tren tarifeleri kümesi geliştirilen melez GA'ların başlangıç çözüm alanını oluşturmaktadır. Bu melez GA'lar eniyi ortalama tren seyahat süresiyle birlikte bir olurlu tren tarifesi elde etmek için çalıştırılmıştır. Melez GA'lar tarafından bulunan en iyi ortalama tren seyahat süreleri karşılaştırılmış ve sonuçlar tartışılmıştır. Bu tez tren çizelgeleme/tarife oluşturma problemine odaklandığı halde, geliştirilen benzetimle bütünleşik yapı, eğer gerçek zamanlı veriler ile beslenebilirse, aynı zamanda yeniden tren çizelgeleme/sevk etme problemi için de kullanılabilir. Çünkü geliştirilen benzetim modeli stokastik olaylar içermektedir ve bu model demiryolu sisteminde meydana gelen bozulmalarla kolaylıkla baş edebilir.

Anahtar sözcükler: Tren, Çizelgeleme, Tarife oluşturma, Yeniden çizelgeleme, Eniyileme, Benzetimle Bütünleşik GA, Benzetimle Bütünleşik Melez GA.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

In this chapter, the background, motivation and objectives of this study are stated, and the organization of this dissertation is outlined.

1.1 Background and Motivation

Management of railway systems is increasingly becoming an important issue of transport systems. Several reasons motivate better usage and planning of the rail infrastructures where track resources are limited due to greater traffic densities. One of the important reasons is that in many European countries railways are being transformed into more liberalized and privatized companies, which are expected to compete on a more profit oriented basis. Another reason is that the rail transport system is subject to increasing pressure by governments and social interest groups to improve its overall efficiency and quality of service for passengers/customers. In addition, the strategic character of the sector is highlighted in view of ecological impacts and national policies aiming at spilling freight/passenger traffic from roads to rails. Also, the ratio of passenger transportation in an urban area is increasing in favour of the rail transport systems.

One of the important problems in management of railway systems is train scheduling problem (*TrnSchPrb*). This is the problem of determining a timetable for a set of trains that does not violate track capacities and satisfies some operational constraints. Several variations of the problem can be considered, mainly depending on the objective function to be optimized, decision variables, constraints and on the complexity of the relevant railway network. Several names are given to the problem widely using three-word phrases; beginning with *train* or *railway* words and going on with one of the words; *scheduling*, *rescheduling*, *planning*, *timetabling*, *dispatching*, and *pathing*, and ending with *problem* word with a few exceptions.

A general *TrnSchPrb* considers a single one way track linking two major stations with a number of intermediate stations in between. We assume that $S = \{1, \ldots, s\}$ represents the set of stations, numbered according to the order in which they appear along the rail line. In particular, 1 and s denote the initial and final station, respectively. Analogously, we assume that $T = \{1, \ldots, t\}$ denotes the set of trains which are candidate to be run in a given time horizon. For each train $j \in T$, a starting station f_j and an ending station l_j ($l_j > f_j$) are given. Let $S^j = \{f_j, \ldots, l_j\} \subseteq S$ be the ordered set of stations visited by train j. A timetable defines, for each train $j \in T$, the arrival and departure times for the stations f_j, f_j+1, $\ldots l_j-1$, l_j. The running time of train j in the timetable is the time elapsed between origin to destination station of the train (Caprara, Fischetti & Toth, 2002). This general *TrnSchPrb* can be more sophisticated by adding some real life behaviour of rail systems or relaxing some assumptions made related with the railway system under consideration.

The *TrnSchPrb* has been studied by researchers and so far many efforts have been spent to solve the problem. The first scientific article was published in 1966, and till now more than one hundred articles have been published. In early years, due to the limitations of computers' abilities and the complexity of the problem, the problem was relaxed by unrealistic assumptions and generally deterministic models were studied. Depending on the increasing computer capabilities more realistic models were developed, and optimization methods were integrated with the modelling structures. The researchers tried to develop fast solution generator algorithms, and these efforts are increasingly going on. Although simulation for modelling has been used in some articles, none of them includes a comprehensive framework. This has been motivation for us to develop a feasible timetable generator simulation modelling framework. Another motivation for us to develop simulation integrated genetic algorithms (GAs) is that GAs have been successfully adapted to solve several combinatorial optimization problems and have become increasingly popular techniques among approximation techniques for finding optimal or near optimal solutions in a reasonable time (Gen & Cheng, 1997; Gen & Cheng, 2000; Gen, Cheng & Lin, 2008; Yu & Gen, 2010). Although a few article studied integration of simulation with GAs to solve the *TrnSchPrb*, they did not handle the problem comprehensively. In addition, since GA provides flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem (Gen et al., 2008) we develop simulation integrated hybrid GAs.

1.2 Research Objectives

In this thesis, the *TrnSchPrb* is studied with three main objectives;

- To review the relevant literature.
- To develop a feasible solution generator model for the problem.
- To develop an optimization algorithm(s) for the problem.

In order to meet the objectives;

- The studies on *TrnSchPrb* are reviewed through 1966-2009 and classified according to the problem type, railway infrastructure, objective(s), developed model structure(s) and solution approach(es).
- A feasible timetable generator stochastic simulation modelling framework is developed for obtaining a feasible train timetable for all trains in a railway system. This framework includes train arrival and departure times for all stations visited by each train and calculated average train travel time.
- A simulation integrated GA, and three local search embedded simulation integrated hybrid GA are developed to obtain a feasible train timetable with optimized average train travel time.

The contributions can be summarized as follows;

- During literature review, we have confronted with some survey papers in which the *TrnSchPrb* was considered briefly since these papers focused on commonly studied rail transportation problems. On the other hand, there exist some short survey papers which involve only a few popular articles. Our literature review includes a lot of studies which focused on the *TrnSchPrb* and published in years between 1966 and 2009, 140 papers along 44 years. Our literature review exhibits the evolution of the related researches over 44 years.

- A general stochastic simulation modelling framework is developed and depicted step by step in order to guide to researchers who aim to develop a simulation model of railway transportation systems. By using this framework all the railway transportation systems can be modelled with only problem/infrastructure specific modifications and feasible solutions can be easily obtained. In order to avoid a deadlock, a general *Blockage Preventive Algorithm* is developed and embedded into the simulation model.
- In the studies of Ping, Axin, Limin & Fuzhang (2001), Rebreyend (2005), and Geske (2006) a simulation model is developed and integrated with GA. The first two studies are focused on rescheduling/dispatching problem that is not the main scope of the thesis. In the last one scheduling/timetabling problem is considered and a deterministic simulation model is developed. To the best of our knowledge our study is the first one which integrates stochastic simulation model with GA and also with hybrid GAs to deal with train scheduling /timetabling problem.
- In the study of Ping, Axin, Limin & Fuzhang (2001), the used encoding is directly dependent to the trains in the system. The articles of Rebreyend (2005) and Geske (2006) do not contain the encoding, one of the most important parts of GAs. The developed encoding structure in this thesis is not dependent to trains which are the causes of problems (conflicts). Thus, the encoding provides to obtain feasible chromosome structures in GA part of our study.
- Another contribution is that three local search embedded GAs are integrated with the simulation model. To the best of our knowledge our study is the first one that employs simulation integrated hybrid GAs to solve the *TrnSchPrb*.

1.3 Organization of the Thesis

The organization of this thesis is as follows.

In chapter two, a comprehensive literature review of the studies on the *TrnSchPrb* that have appeared 1966-2009 is given. Chapter three comprises an overview of GA. In chapter four, first a hypothetic *TrnSchPrb* is introduced, than the proposed simulation modelling framework is explained step by step, and finally the simulation model results are discussed. Four approaches; a simulation integrated GA and three local search embedded simulation integrated GAs, to the *TrnSchPrb* are presented and tested on our hypothetic *TrnSchPrb* in chapter five. The objective of using these four approaches is to obtain a feasible train timetable with optimized average train travel time. Concluding remarks and future research directions are listed in the last chapter.

CHAPTER TWO
LITERATURE REVIEW ON TRAIN SCHEDULING PROBLEM

We reviewed 140 papers on the *TrnSchPrb* published in 1966-2009 and then classified in Table A.10 in Appendix in chronological order, and also discussed more than 70 papers we reviewed.

The studies in the relevant literature can be classified into two main groups; *scheduling (timetabling)* and *rescheduling (dispatching)*. The studies in the former group aim at achieving a train timetable with arrival and departure times of all trains at the visited stations in the system. These studies generally begin with a planned infeasible initial (draft) timetable with many conflicts. After these conflicts were solved a feasible train timetable is composed, and the train operating authority runs the trains according to the timetable. The studies in the latter group reschedule the trains after disturbances. These studies generally begin with a planned feasible timetable with no conflicts. During the implementation of the feasible timetable, it is possible to be encountered various problems. Of course, these problems prevent to obey the feasible timetable. At this point, the timetable is needed to be revised, that is the trains must be rescheduled. The rescheduling is temporary, depends on real time information and the equipment to gather data from the whole system. The goal is to regulate the system temporary in order to implement the train schedule/timetable. For rescheduling real time data and the equipment that will gather data from the whole system are needed.

We classified the papers we examined under four headings; review, scheduling /timetabling, rescheduling/dispatching, and scheduling/timetabling and rescheduling /dispatching.

2.1 Review Papers

There exist a few review papers in the literature (Assad, 1980; Bussieck, Winter & Zimmermann, 1997; Cordeau, Toth & Vigo, 1998; Newman, Nozick & Yano, 2002; Caprara, Kroon, Monaci, Peeters & Toth, 2007). In these papers, some railway optimization problems are considered, and the *TrnSchPrb* is regarded in only one section, and no one concentrates only on the *TrnSchPrb*.

The first review paper, Assad (1980), is related with the mathematical models for rail transportation, and considers two objectives; the first is to collect and categorize rail modelling efforts, and the second is to position the related literature in the context of other transportation models and provide an introduction to this field for nonspecialists. The role of each model class is discussed in relation to its function and its position within the total planning activity of a railroad. After years, Bussieck et al. (1997) consider the development and the usage of mathematical programming methods in public rail transport planning. The authors focus on some aspects of the planning process, and on some planning results which lead to more comprehensive planning and optimization of railroad network systems. They discussed in particular the computation of the line plans, train schedules, and schedules of rolling stock. In another review paper, Cordeau et al. (1998) present a comprehensive survey of optimization models for the most commonly studied rail transportation problems. For each group of problems, they propose a classification of models and describe important class characteristics in terms of model structure and algorithmic aspects. The review concentrates on routing and scheduling problems since these problems form the most important portion of the planning activities performed on the railways. The routing models concern the operating policies for freight transportation and railcar fleet management, whereas scheduling models address the dispatching of trains and the assignment of locomotives and cars. On the other hand, Newman et al. (2002) review optimization problems in the rail industry such as infrastructure (track and siding) planning and track maintenance; sizing of fleets of locomotives and railcars; locomotive, railcar, and container repositioning; train scheduling; freight routing; meet-pass planning; and timetable construction. These problems are specific to the rail industry with technological or cost considerations. A recent review by Caprara et al. (2007) is related with the operational planning problems such as line planning, timetabling, platforming, rolling stock circulation, shunting, and crew planning problems in passenger transportation in Europe.

2.2 Papers on Scheduling/Timetabling

These papers deal with to prepare a train timetable that includes arrival and departure times of all trains at visited stations. In these papers, first an initial (draft) timetable with conflicts is planned, then these conflicts are solved and a feasible train timetable is obtained. In the following subsections, we classify these papers according to the developed model structure such as mathematical model, simulation model, and the others.

2.2.1 Mathematical Models

In the preliminary scheduling (timetabling) articles (Frank, 1966; Salzborn, 1969; Nemhauser, 1969; Amit & Goldfarb, 1971; Szpigel, 1973; Cury, Gomide & Mendes, 1979, and Cury, Gomide & Mendes, 1980) mathematical models are used.

Frank (1966) studied the capacity for one way traffic and for certain regular systems of two way traffic with priority at the nodes for trains going in one direction. This problem is called *railway planning problem*. The cycle times of the trains and the number of trains needed to accomplish the transports for different systems was also studied. Two cases were considered; in the first case only one train was allowed to wait at every inner node, and in the second case more than one train was allowed to wait at every inner node. The objective was to find the optimum traffic system that maximizes the traffic capacity. Some of the prominent characteristics of the study can be itemized as follows. There can be at most one train on a track; the nodes had space for an unlimited number of trains; the trains' speed was fixed for both directions; and the trains can not overtake each other. The system infrastructure is a single tracked line with double tracked stations and this infrastructure has been widely used in the literature so far. Salzborn (1969) developed a method in order to construct timetables for a suburban railway line without branches. It was shown that such timetables were largely determined by stop schedules. Two criteria for stop schedules was considered; the number of intermediate passenger stops and the number of carriage miles. A mathematical formulation was presented and dynamic programming was used for solution. The objective was to find stop schedules with minimum number of carriage miles or with minimum number of passenger stops. Nemhauser (1969) developed a model for finding a jointly optimal schedule of local and express transportation service operates between an origin point and a termination point. The objective was to find a schedule that yields maximum total profit which was the sum of the profits over all scheduled trains, and the model was solved by dynamic programming. Amit & Goldfarb (1971) studied on timetable problem of railways, in which the objective was to minimize the overall passage time of trains and a one train at a time based heuristic algorithm was developed for solution. Many studies indicate that Szpigel (1973) is the first author studies on the *TrnSchPrb*. The problem was to determine the best crossing and overtaking (meet/pass) locations with a given routes and departure times of the trains on a single track railway. The objective was to minimize the weighted average of train travel times. The mathematical model built in this study was solved by dynamic programming. Cury et al. (1979) and Cury et al. (1980) presented a methodology developed for the automatic generation of optimal schedules for a metro line. An analytical model was created to represent the behaviour of the system which includes train and passenger movements. Based on the model characteristics, the goal coordination method was utilized to produce the optimal reference schedule by considering comfort levels for passengers, the number of trains in the line, and the performance of the trains. The objective was to minimize a total cost function of average delay, headways and passengers.

Some of the mostly citied articles such as Mees (1991), Jovanovic & Harker (1989, 1991b), Odijk (1996) also used mathematical models. Mees (1991) presented an approximate algorithm to find feasible solutions for railway scheduling problem with a single track network. The objective of the study was to minimize the total cost of running trains on arcs. A shortest path algorithm was developed to solve the integer linear programming model constructed for the problem. Jovanovic & Harker (1989 and 1991b) presented an overview of a decision support model, SChedule ANalysis (SCAN), for the tactical scheduling of freight railroad traffic which was designed to support the weekly or monthly scheduling of rail operations. The purpose of SCAN was to help in the design of robust (reliable) train schedules, not to provide an optimal schedule. This decision support model starts with given train schedules and evaluates their feasibility. If these schedules are found to be infeasible, the decision support system offers automatic procedures to modify the given schedules until these schedules become feasible. Three algorithms are incorporated within the SCAN system; the first one evaluates the feasibility of a given set of schedules over a given lane, the second one modifies the infeasible schedules until feasibility is achieved, and the last one estimates a measure of reliability of a given set of schedules. Odijk (1996) discussed the usage of a particular mathematical model to construct periodic railway timetables. In the model, departure and arrival times of trains are related pair wise on a clock by means of periodic time window constraints, and a solution to a set of such constraints constitutes a periodic timetable. A cut generation algorithm is presented to solve the problem. This algorithm is terminated in a finite number of iterations result in a feasible timetable structure.

After the preliminary articles discussed above, a series of articles - Cai & Goh, 1994; Cai, Goh & Mees, 1998; Carey, 1994a; Carey, 1994b; Carey & Lockwood, 1995 – related with each other were published. Cai & Goh (1994) concerned with the problem of scheduling trains on a single track railway where the trains were allowed to cross only at one of the passing loops. The objective was to minimize the total cost due to stopping and waiting. A one train at a time base heuristic algorithm was developed to solve the related integer programming model. Cai et al. (1998) described a heuristic algorithm for train scheduling problem and had shown that it can produce schedules for a single track system. Although the algorithm was demonstrated on artificial examples, a more complex version of it was installed on a real railway system. It was the aim of the paper to construct an algorithm that extends the model of Cai & Goh (1994) to a greater generality, to include most of the practical constraints whilst retaining the essential characteristic of the greedy heuristic approach, namely the ability to compute a good feasible solution quickly. Two of important extensions were; firstly the algorithm allowed a train to start from any position (not necessarily from a node or terminus) at any time instant, and secondly the algorithm possessed a capability to schedule, if needed, physical backups (reverse) of some trains. The objective was the same as Cai & Goh (1994).

Carey & Lockwood (1995) set out a model, algorithms and strategies for the train pathing problem for a single line, which was the problem of assigning trains to available track (lines, platforms, etc) in a rail network so as to minimize train delays or delay costs. They proposed a solution heuristic and strategies analogous to those which had enabled expert train pathers to plan large scale complex rail systems by traditional manual graphical methods. They set out a basic train pathing problem as a mathematical programming model and then decomposed it into a sequence of similar subproblems, each representing pathing a single train while holding fixed the sequence order of all already pathed trains. They did not intended to provide a ready to implement train pathing system, rather it was a research contribution to developing suitable basic models and algorithms and demonstrating that these can be resolved in acceptable times. Carey (1994a) set out a detailed mathematical programming model for train pathing and planning by extending the basic single line pathing model introduced in Carey & Lockwood (1995). The author allowed trains to choice a line, station platform, and route, and to make it tractable when solving the mathematical programming model, decomposed the mathematical model into a sequence of simpler mathematical programming subproblems. Each of these models corresponds to pathing a single train while temporarily holding fixed the sequence order but not the timings of all other already pathed trains. The basic strategy was to path trains one at a time, until all trains are pathed once, and if necessary iteratively repath trains until an acceptable solution was found. The objective was to minimize cost associated with arrival times, departure times, trip times on links and dwell times at stations. In Carey (1994a), and Carey & Lockwood (1995), it is assumed that each rail line has two or more tracks and each is dedicated to traffic in one direction (one way tracks), Carey (1994b) showed how to adapt and extend the model and the algorithms presented in Carey (1994a) and Carey & Lockwood (1995) to handle trains on single line two way tracks.

The metaheuristics have attracted the authors who deal with the solution of the mathematical models developed for train scheduling. Nachtigall & Voget (1996) considered the compilation of timetables for periodic served railway networks and focused on railway synchronization. The calculation of timetables with minimal waiting time for passengers who were changing trains (used more than one train for a trip) was modelled by a periodic network optimization problem. They developed a mathematical model and presented a GA which was combined with a greedy heuristic and a local improvement procedure. Higgins, Kozan & Ferreira (1997) developed a mathematical model and applied a local search heuristic (LSH), GAs, tabu search (TS), and two hybrid algorithms to train scheduling problem. The purpose of a single line train scheduling model in their paper was to resolve the train conflicts (overtaking as well as crossing) at the sidings in such a way so as to achieve the minimization of total weighted travel time objective. Total weighted travel time was the total travel time from origin to destination (including conflict delays) for all trains, weighted by train priority. The LSH was based upon repeatedly replacing a current solution with an improved neighbouring solution. The main aim for applying the possible moves was to try to improve the train schedule in terms of reducing total weighted travel time. In GA each gene in the solution was a conflict with three attributes; the train delayed, the train with right of way, and the track segment where the conflict occurs. When two train schedule solutions (parents) in the population were selected to mate, genes from both solutions were used to make two offspring and a single point crossover was used for the train scheduling problem so as to keep the number of infeasible child train schedule solutions at a minimum. As the LSH methods, the general TS heuristic was based on transforming a current solution to one of the neighbouring. The proposed first hybrid algorithm (HA1) consists of applying the LSH to the best five percent of the population, after the crossover operator was performed. HA1 uses the LSH as a new genetic operator. The second hybrid algorithm (HA2) incorporates the advantages of TS into the crossover operator for conducting a search for suitable parents and crossover points. Brännlund, Lindberg, Nou & Nilsson (1998) presented an optimization approach for the timetabling problem of a railway company. The objective was to schedule a set of trains to obtain a profit maximizing timetable, while not violating track capacity constraints. The authors constructed a very large integer programming model, and used a Lagrangian relaxation solution approach, in which the track capacity constraints are relaxed and assigned prices. Thus, the problem was separated into shortest path programs for each physical train.

In recent years, the authors have spent efforts to optimize multi objectives. Chang et al. (2000) developed a multi objective linear programming model for the optimal allocation of passenger train services on an intercity high speed rail line without branches. Minimizing the operator's total operating cost and minimizing the passenger's total travel time loss were the two planning objectives of the model. The operator's total operating cost was defined to be the sum of the fixed and variable operating costs over all train trips that were required to meet the travel demand. The passenger's total travel time loss was defined as the sum of the time losses for stopping at intermediate stations for all the passengers served by all the train trips. For a given many-to-many travel demand and a specified operating capacity, the model was solved by a fuzzy mathematical programming approach to determine the best compromise train service plan, including the train stop schedule plan, service frequency, and fleet size. Ghoseiri, Szidarovszky & Asgharpour (2004) developed a multi objective optimization model for passenger train scheduling problem on a railway network with single and multiple tracks, as well as multiple platforms with different train capacities. The lowering the fuel consumption cost, the measure of satisfaction of the railway company, was regarded as a criterion of efficiency, and shortening the total passenger time, the passenger satisfaction criterion, was regarded as a criterion of effectiveness. The solution of the problem consists of two steps; at first the Pareto frontier is determined, and then based on the obtained Pareto frontier detailed multi objective optimization is performed. Zhou & Zhong (2005) concerned with a double track train scheduling problem for planning applications with multiple objectives on a high speed passenger rail line in an existing network. The problem was to minimize both the expected waiting times for high speed trains and the total travel times of high speed and medium speed trains. By applying two practical priority rules, the problem with the second criterion was decomposed and formulated as a series of multi mode resource constrained project scheduling problems. The high speed trains always take priority over medium speed trains rule was used to determine the priorities between different types of trains. The earlier a train enters a station, the earlier it will leave the station rule was used to specify the priorities between the same types of trains. Liebchen (2008) concentrated on periodic railway timetabling problem related with a subway network. The minimization of the weighted sum of passenger waiting times and the minimization of the number of trains that was required to operate the timetable were the objectives of the study. In a recent study, Lee & Chen (2009) proposed an optimization oriented four step heuristic to solve a set of train paths and a timetable for a train system. The heuristic uses a simple rule to generate an initial feasible solution, and then improves the solution iteratively. Each iteration attempts to improve the current solution by altering the order the train services travel from station to station, assigning the services to tracks within the stations, determining the order the services pass through the stations, and finally solving to obtain a timetable. According to the quality of the timetable, the solution is accepted or rejected with a threshold accepting rule. The objectives are to minimize the sum of weights of tracks assigned to all services at all stations and to minimize the sum of the difference between the services' scheduled departure time and the target departure time.

Recently three study series have appeared. The first serial includes four articles; Caprara, Fischetti, Guida, Monaci, Sacco & Toth (2001), Caprara, Fischetti & Toth (2002), Caprara, Monaci, Toth & Guida (2006) and Cacchiani, Caprara & Toth (2008). The second one consists of Peeters & Kroon (2001) and Kroon & Peeters (2003), and the last one contains Zhou & Zhong (2007) and Castillo, Gallego, Ureña & Coronado (2009). Caprara et al. (2001) and Caprara et al. (2002) concentrate on train timetabling problem relevant to a single, one way track linking two major stations with a number of intermediate stations between them. The railway networks typically contain few important lines, called also corridors, connecting major stations. On these corridors, made of two independent one way tracks carrying traffic in opposite directions, track resource is limited by great traffic densities. Once the timetable for the trains on the corridors is determined, it is relatively easy to find a convenient timetable for the trains on the other lines of the network. A graph theoretic formulation is proposed for the problem using a directed multigraph in which nodes correspond to departures or arrivals at a certain station at a given time instant. This formulation is used to derive an integer linear programming model that is relaxed in a Lagrangian way and the relaxation is embedded within a heuristic algorithm. The objective is to maximize sum of the profits of the scheduled trains. An ideal timetable is assigned to each train. An ideal timetable, which would be the most desirable timetable for the train, however, may be modified to satisfy the track capacity constraints. In particular, it is allowed to slow down each train with respect to its ideal timetable and/or to increase the stopping time interval at the stations. The final solution of the problem is called the actual timetable. Caprara et al. (2006) extend the train timetabling problem, considered by Caprara et al. (2002), by taking into account additional real world constraints, the manual block signalling constraints, the station capacities constraints, the prescribed timetable for a subset of the train constraints, and the maintenance operations constraints. On the other hand, Cacchiani et al. (2008) propose heuristic and exact algorithms for the periodic and nonperiodic train timetabling problem on a corridor to maximize the sum of the profits of the scheduled trains. The heuristic and the exact algorithms are based on the solution of the relaxation of an integer linear programming formulation in which each variable corresponds to a full timetable for a train. This approach is in contrast with previous approaches proposed by Caprara et al. (2001), Caprara et al. (2002) and Caprara et al. (2006) so that these authors considered the same problem, and used integer linear programming formulations in which each variable was associated with a departure and/or arrival of a train at a specific station in a specific time instant. Peeters & Kroon (2001) propose an optimization approach to the cyclic railway timetabling problem. This approach enables one to search for an optimal timetable and to make the necessary changes to an infeasible instance, by allowing a penalized violation of the constraints. The authors use a mixed integer programming formulation for the problem, where the integer variables correspond to cycles in the graph. The objective is to minimize halting and transfer times. In addition, Kroon & Peeters (2003) describe how variable trip times can be embedded into an existing cyclic railway timetabling model for the periodic event scheduling problem. Thereby they provide an extension of the existing model presented in Peeters & Kroon

(2001). Because in the existing model it was assumed that the trip times of all trains on all tracks of the railway network were fixed and known a priori which may be too restrictive in practice. Since the extended model has the same general structure as the original model, the developed solution methods are applied to the extended model. The study of Kroon & Peeters (2003) only deals with the planning process of generating an appropriate feasible timetable where the trip times may be varied in order to obtain a feasible timetable. However, the paper do not consider real time traffic control of railway operations, and the variable trip times should therefore not be interpreted as a tool to deal with disturbances that occur during the operation of a railway timetable. Zhou & Zhong (2007) focus on single track and propose a generalized resource constrained project scheduling formulation for train timetabling problem. In this study, segment and station headway capacities are considered as limited resources, and a branch and bound solution procedure is presented to obtain feasible schedules. The developed algorithm chronologically adds precedence relation constraints between conflicting trains to eliminate conflicts, and the resulting subproblems are solved by the longest path algorithm to determine the earliest start times for each train in different segments. The authors adapt three approaches to reduce the solution space. First, a Lagrangian relaxation based lower bound rule issued to dualize the segment and station entering headway capacity constraints. Second, an exact lower bound rule is used to estimate the least train delay for resolving the remaining crossing conflicts in a partial schedule. Third, a tight upper bound is constructed by a beam search heuristic method. The objective is to minimize the total train travel time, the sum of the free running time and additional delay. Castillo et al. (2009) use an optimization method to solve train timetabling problem for a single tracked bidirectional line, similar to the one presented by Zhou & Zhong (2007) but more complex, and discuss the problem of sensitivity analysis. A three stage method is proposed to deal with the problem and a sequential combination of objective functions is used for solution. In fact, the proposed method sequentially minimizes the maximum travel time for single trains, allocates trains to circulate as soon as possible, and minimizes the total station dwell time of all the trains, i.e., the model can be considered as a sequential multi stage approach.

2.2.2 Simulation Models

In a few papers a simulation model was developed for train scheduling (timetabling) problem. To our knowledge, Wong & Rosser (1978) is the first study in the literature that developed a simulation model for train scheduling (timetabling) problem. The output of the simulation model comprises a pictorial representation of the pattern of train movements as well as detailed statistics for each train. The problem is to determine where a crossing or overtaking should be allowed to occur, and the objective is to minimize the sum of weighted costs of delaying trains at passing loops where the weights chosen reflect the importance of each type of train. To improve the system performance, train starting times are varied, and one train at a time heuristic iterative procedure is used for improvements. Petersen & Taylor (1982) presented a state space description for the problem of moving trains over a line, and an algebraic description of the relationships that must be hold for feasibility and safety considerations was given. The line blockage problem at high traffic intensities was discussed under conditions that ensure the blockage not to occur. The objective of the study is to minimize the terminating times of the trains. Geske (2006) focused on railway scheduling problem and developed a constraint based deterministic simulation model with the objective of reducing the lateness of trains. Selecting alternative paths in stations was an optimization task to reduce lateness and to find a conflict free solution. The results of the proposed sequentially train scheduling heuristic was compared with those of a GA.

2.2.3 Other Models

Salim & Cai (1997) proposed a GA for solving a simplified train scheduling problem in a mineral transport railway system. The problem under consideration involves moving a number of trains carrying mineral deposits across a long haul railway line with both single and double tracks in either direction. The problem was modelled to minimize environmental impacts in mineral transportation. The objective function to be minimized in the scheduling model is related to the costs of stopping and waiting for trains travelling on the railway line during a span of time.

Isaai & Singh (2000) developed a heuristic algorithm for predictive scheduling of passenger trains on a single track railway with some double track parts by using an object oriented methodology. The heuristic tends to minimize the total waiting time of the trains concerned. The model has three modules; the initialization module that deals with the creation of computational structures and data inputs for the problem, the scheduling module that generates a feasible solution using a heuristic, and the evaluation module that computes the quality of the solution. Real predictive schedules that were manually generated by train planning experts are used to evaluate the model's outputs. Kwan & Mistry (2003) reported on an evolutionary approach for the automatic generation of planning train timetables at the early stages. The timetables produced at early stage (planning timetables) were used as the basis for planning and negotiations. After iterations of refinements and detailed conflict resolution, the planning timetables would eventually be evolved into the final operational timetables. The authors concerned with the automatic generation of planning timetables, and explored how train timetabling problem could be substructured. The problem was decomposed into modules such as the departure times, the scheduled run times and the resource options. The advantages of such decomposition are the independent representation of interacting subcomponents and the independent evolution of these subcomponents. The objective function of the study is to minimize the weighted sum of violations expressed in time units.

Carey & Carville (2003) considered the problem of train platforming or scheduling for large, busy, complex train stations which are the key components of the busy passenger rail networks, and are the location of most train conflicts. Train schedule for a large busy station ensures that there are no conflicts among the trains by guaranteeing that each train is allowed at least its minimum required headways, dwell time, turnaround time and trip time. In the heuristic approach, which is similar to train planners using manual methods, the authors considered one train at a time, detected and resolved all the conflicts for that train before considering the next train. The objective is to minimize the cost of deviations from the desired times, platforms or lines for each train. There are a set of three costs; the time adjustment costs, the platform desirability costs, and the platform obstruction costs. Carey & Crawford

(2007) developed heuristic algorithms to assist in finding and resolving the conflicts in draft train schedules. They employed an algorithm developed by Carey & Carville (2003) for scheduling trains at a single station and extended this algorithm to obtain algorithms for multiple stations on a rail corridor. In the first algorithm of Carey & Carville (2007) when the current train conflicted with any other train, the conflict was resolved by adjusting the times of the current train. In the second algorithm (a new procedure) of Carey & Crawford (2007), conflicts were resolved by adjusting the times of either train, depending on which requires the smaller adjustments or smaller costs or penalties. Carey & Crawford (2007) applied the new procedure, adjusting the times of the current train or the other trains, to resolve conflicts between trains at station exits and conflicts between trains on lines between stations. In the third algorithm of Carey & Crawford (2007) they extend the procedure also to resolving conflicts between trains using the same platform. All these three algorithms resolved all conflicts, but the second gave much better solutions than the first, and the third gave better solutions than the second algorithm. Salido, Abril, Barber, Ingolotti, Tormos & Lova (2007) proposed to distribute the railway scheduling problem into a set of sub problems as independent as possible. Their goal was to model the railway scheduling problem as constraint satisfaction problems (CSPs) and solve it using constraint programming techniques. However, due to the huge number of variables and constraints that this problem generates, a distributed model was developed to distribute the resultant CSP into semi-independent sub problems such as the solution can be found. The first way to distribute the problem was carried out by means of a graph partitioning software called METIS. The second model was based on distributing the original railway problem by means of train type. The third model was based on distributing the original railway problem by means of contiguous stations. The objective in the study was to minimize the journey time of all trains. Tormos, Lova, Barber, Ingolotti, Abril & Salido (2008) focused on the application of evolutionary algorithms to solve train timetabling problem. The problem considered implied the optimization of trains on a railway line that was occupied (or not) by other trains with fixed timetables. The timetable for the new trains was obtained with a GA that included a guided process to build the initial population. The objective was to minimize the average delay of the new trains. Liu &

Kozan (2009) modelled train scheduling problem as a blocking parallel machine job shop scheduling problem. Firstly, a parallel machine job shop scheduling problem was solved by an improved shifting bottleneck procedure algorithm without considering blocking conditions. Inspired by the proposed shifting bottleneck procedure algorithm, feasibility satisfaction procedure algorithm was developed to solve and analyze the blocking parallel machine job shop scheduling problem by an alternative graph model. The objective was to minimize the makespan.

2.3 Papers on Rescheduling/Dispatching

These studies deal with the rescheduling of trains after disturbances, and at first begin with a planned feasible timetable that contains no conflicts. While implementing the feasible timetable, it is not surprise to have problems, which prevents to obey the feasible timetable. At that time the timetable is needed to be repaired, the trains must be rescheduled. Since the repairs depend on real time information and temporary, rescheduling is a temporary solution, the goal is to regulate the system in order to implement the train schedule/timetable. For rescheduling real time data are needed, and the equipments that can gather data from the whole system must be set up. The first article (Sauder & Westerman, 1983) on the rescheduling/dispatching was published 17 years later than the first one (Frank, 1966) on the scheduling/timetabling problem.

2.3.1 Mathematical and Simulation Models

To our knowledge, Sauder & Westerman (1983) is the first paper dealing with the rescheduling problem. The authors developed a minicomputer based information system with online optimal route planning capability to assist dispatchers. The routing plan was revised automatically as conditions changed. The potential for an online planning algorithm laid in considering all feasible future train meets throughout the territory and advising the dispatcher of that combination which would minimize total train delay. The first attempt to model the process evaluated feasible train routes with a decomposition approach incorporated a shortest path algorithm and a linear programming formulation. Although the optimal were obtainable, since convergence time was excessive suboptimal solutions were obtained. After that, the method was subsequently replaced with a branch and bound technique enumerating all feasible meet locations and this approach insured optimal results. The objective in the study was to minimize total train delay. In the study, only the delay within the limits controlled by the dispatcher was included and only the delay that the dispatcher's planning would influence was considered. In another study, Şahin (1999) dealt with inter train conflicts (meet/pass) problem which occurred when two opposing trains move on a single track section between neighbouring meet points, or if a faster train caught a slower one moving in the same direction. The objective was to minimize the sum of deviation of the expected arrival times of trains from their scheduled times within a prescribed time horizon. In his study, firstly, a zero one mixed integer programming model was built in order to have an optimal solution. After that, he analyzed dispatchers' decision process in inter train conflict resolutions and developed a linear programming model of this decision process that produces same results with dispatchers' preferences. In model building he assumed that the train dispatcher uses a utility function of weighted attributes in order to model his/her choice behaviour. Then, he developed a heuristic algorithm for rescheduling trains by modifying existing meet/pass plans in conflicting situations in a single track railway. The heuristic algorithm was developed in order to obtain better conflict solutions than train dispatchers and optimal or near optimal solutions in reasonable length of time. He compared three solution methods; the optimal solution of mixed integer programming, the dispatcher's solution and the heuristic's solution. The comparison criteria were total waiting times and computation time. As a result, the heuristic gave better solution than dispatcher's, and also performed almost as well as the optimal solution method in selecting the better conflicting train to stop.

2.3.2 Mathematical Models

Mills, Perkins & Pudney (1991) described a dynamic rescheduling algorithm for scheduling future train movements with the objective of minimizing the overall cost of train lateness and energy consumption. The dynamic rescheduling system calculated the location and time for each cross or overtakes and determined which of the trains involved in the cross should take the siding, so it was used to determine the arrival and departure times for each train at each station. Kraay, Harker & Chen (1991) presented a mathematical programming model for the pacing problem and described alternative solution algorithms for this model. The purpose of the pacing model, which included velocity as a decision variable, was to define a good operating policy for the dispatcher. A train dispatcher can improve the operations of a rail line by pacing trains over a territory, namely to permit trains to travel at less than maximum velocity to minimize fuel consumption while maintaining a given level of performance. Kraay & Harker (1995) presented a model for the optimization of freight trains schedules that was intended to be used as part of a real time control system. The goal of the model was to provide a link between strategic schedules and line dispatching or computer assisted dispatching models by providing starting and ending times for each line while taking into account overall performance of all the trains across the rail network. They described the model and associated algorithm for the real time scheduling of trains over the entire rail network. The time based objective function was to be minimized has three components; the first term was related with the deviation of arrival and departure times to the stations, the second term was a penalty term for a train violating the 12 hour rule (crews legally changed every 12 hours), and the third term was the cost of a block missing a scheduled connection. Higgins, Kozan & Ferreira (1996) designed a model to be used as a decision support tool for train dispatchers to schedule trains in real time in an optimal way and as a planning tool to evaluate the impacts of timetable changes, as well as rail road infrastructure changes on train arrival times and train delays. The objective was to minimize the cost function includes fuel consumption and train delays, with an assumption that the cost of tardiness has a higher priority than fuel costs. Adenso-Diaz, Gonzalez & Gonzalez-Torre (1999) presented the experience of designing and implementing a system for rescheduling the services of a regional network. They tried to obtain a new schedule, which was the most similar as possible to the original one that had generated manually by the marketing department according to customers needs, when an unplanned event had occurred. The process of exploring the solutions space in order to select the best evaluated solutions was carried out by means of a backtracking algorithm that was a depth first search based branch and bound implicit enumeration procedure. The evaluation of the quality of each solution obtained was made on the basis of the priority of each service, the passengers transported and the delays that these passengers had to suffer. The best results were offered to the traffic controller so that, using what-if tools, he/she may choose the alternative that he/she considers the most adequate from among these. The objective of the study was to maximize the number of passenger transported and the model was a mixed integer programming model.

In recent years, Semet & Schoenauer (2005) concentrated on the particular problem of local reconstruction of the schedule following a small perturbation, seeking minimization of the total accumulated delay by adapting times of departure and arrival for each train and allocation of resources (tracks, routing nodes). They described a permutation based evolutionary algorithm that relied on a heuristic to gradually reconstruct the schedule by inserting trains one after the other following the permutation. This algorithm was hybridized with mixed integer programming tool CPLEX; the evolutionary part was used to quickly obtain a good but suboptimal solution and this intermediate solution was refined using CPLEX. Once the population had converged, its best individual was fed to CPLEX as a starting point. The goal of the optimization procedure was to minimize the total accumulated delay, i.e., for all trains at all nodes, the difference between the actual time of arrival and the theoretical one. Semet & Schoenauer (2006) described an inoculation procedure which enhanced an evolutionary algorithm for train rescheduling problem. The procedure consisted in building the initial population around a precomputed solution based on problem related information available beforehand. The optimization was performed by adapting times of departure and arrival, as well as allocation of tracks, for each train at each station. This was achieved by a permutation based evolutionary algorithm that relied on a heuristic scheduler inserted trains one after another. One difficulty was that; not all the individuals were feasible schedules. The goal of the optimization procedure was to minimize the total accumulated delay, i.e., for all trains at all nodes, the difference between the actual time of arrival and the theoretical one. Törnquist & Persson (2007) presented an optimization approach to the problem of rescheduling railway traffic in an n-tracked network after a disturbance had been occurred. They developed a mixed integer linear programming model and used branch and bound algorithm for solution. There are two alternative objective functions; the first one is to minimize the total final delay of the traffic, i.e., the sum of the final delays when trains arrive at their final destination, or rather the last stop considered within the rescheduling time horizon, and the second one is to minimize the total cost associated with delays when trains arrive at their final destination (or last stop considered).

2.3.3 Simulation Models

The number of articles that used simulation model for the rescheduling /dispatching is much more than the articles for the scheduling/timetabling.

Kraft (1987) presented a deterministic algorithm for train dispatching problem. A probability model of train delay was derived to show how dispatching decisions can be made, and by using a random number generator speed fluctuations were introduced into the simulation model. The objective of the study was to minimize the weighted average of the train delays. A branch and bound based combinatorial train dispatching algorithm was developed for solution and its performance was compared with a local optimization technique. Iyer & Ghosh (1991 and 1995) introduced a distributed decision making algorithm for railway networks (DARYN), wherein the overall decision process was analyzed and distributed onto every natural entity of the system; the trains and the stations. The decision process for every train was executed by an onboard processor that negotiated, dynamically and progressively, for temporary ownership of the tracks with the respective station controlling the tracks, through explicit processor to processor communication primitives. This processor then computed its own route utilizing the results of its negotiation, its knowledge of the track layout of the entire system, and its evaluation of the cost function. Every station’s decision process was also executed by a dedicated processor that, in addition, maintained absolute control over a given set of tracks and participated in the negotiation with the trains. Cheng (1998a) proposed a hybrid method of the network based simulation and the event driven simulation for resolving resource conflicts in train traffic rescheduling, in where resolving resource conflicts was; to decide which train should use the shared resources first. The objective of the study was to minimize the total delay of trains. There were two kinds of trains running on the same railway lines; the long distance trains run with a high speed and had less stops at stations, and the local trains stop nearly at almost every station and used a normal speed.

Ping, Axin, Limin & Fuzhang (2001) presented a GA based solution to train dispatching in which an individual describes the trains departure order. At first a model for the train dispatching on the lines with double tracks was established, which can optimize train dispatching by adjusting the orders and times of trains' departure from stations. Then the efficiency of the method was demonstrated via simulation on a high speed railway. The objective was to minimize total delay time. Rebreyend (2005) presented a tool called DisTrain, dedicated to optimize railway dispatching and railway infrastructure, in order to help the dispatcher to reschedule trains if needed. There were some important points; the dispatcher (or user) should be able to interact with the software and the proposed solutions should be dispatcher's oriented, and the number of changes from the previous schedule should be keep small, as well the complexity of the proposed solution (number of actions needed to run it). The objective of the study was to minimize the number of delayed trains, GAs and branch and bound algorithm were used for solution.

Flamini & Pacciarelli (2008) addressed a scheduling problem arising in the real time management of a metro rail terminus. It consisted in routing incoming trains through the station and scheduling their departures with the objective of optimizing punctuality and regularity of the train service. The terminus was divided into blocks of different lengths in where a block being a track segment between two signals. Within the station a signal may turn into two colours; red or green. A red signal indicated that the subsequent block was not available, occupied by another train. A green signal indicated that the subsequent block section was empty and available. Two different objective functions were considered in lexicographical order; the first one was the minimization of the sum of total tardiness plus total earliness for all trains with respect to the off line timetable, and the second objective function was the minimization of the difference between the off line headway and the actual headway for all pairs of consecutive trains leaving the station. The problem was solved in two steps; at first a heuristic built a feasible solution by considering the first objective function, and then the regularity was optimized without deteriorating the first objective function. In a recent study, Luethi, Medeossi, & Nash (2009) investigated a critical problem faced by railways that was how to increase capacity without investing heavily in infrastructure and impacting on schedule reliability. One way of increasing capacity was to reduce the buffer time added to timetables that was used to reduce the impact of train delays on overall network reliability. The performance of the two loop approach increased when the rail network was strategically divided into bottleneck areas; areas operating at or near their capacity limit, condensation zone and non bottleneck areas; compensation zone. The trains should be operated at their maximum allowed speeds and with very small buffer times in condensation zones. The objective was to minimize the total delay of all the trains.

2.3.4 Other Models

Khan, Zhang, Jun & Li (2006) presented an application of GA to solve problem with the aim to minimize delays at the intermediate and final train stations. The term delay describes the deviation of trains from its scheduled departure and arrival times. There may be infeasible child individuals that were replaced with one of their parents. Mazzarello & Ottaviani (2007) introduced the architecture, the approach and the current implementation of an advanced Traffic Management System (TMS) able to optimize traffic fluency in large railway networks equipped with either fixed or moving block signalling systems. They concentrated on the core modules of the TMS architecture, which were responsible for automatic local traffic optimization and control, respectively named Conflict Detection and Resolution (CDR) and Speed Profile Generator (SPG). The CDR was responsible for automatic real time train scheduling and routing, and applied a model based on the alternative graph formulation. The SPG was responsible for plan execution. Operating strictly connected to CDR, SPG computed an optimal speed profile for each train, in order to make the CDR plan being executed in a safe and energy saving manner. The objective was to minimize delays, acting both on train precedence relations at conflict points and on train routings.

D'Ariano, Pranzo & Hansen (2007) introduced a variable speed dispatching system that can control the railway traffic in a regional network. They focused on the real time optimization of train scheduling and speed coordination. The proposed model took into account simultaneously all trains and aims at minimizing the maximum delay due to conflicts. The railway network was composed of block sections separated by signals. The signals controlled the train traffic on the routes and imposed safe space distance headway. A block section was a track segment between two main signals, and a signal aspect may be red, yellow, or green. A red signal aspect indicated that the subsequent block section was either out of service or occupied by another train, on the other hand a yellow signal aspect indicated that the subsequent block section was empty, but the following block section was occupied by another train. A green signal aspect indicated that the next two block sections were empty. A train was allowed to enter the next block section if the signal aspect was either green or yellow, but the train required deceleration and stopping before the next signal if the signal aspect remained red.

In a recent study, Cheng & Yang (2009) aimed to transform a train dispatcher's expertise into a useful knowledge rule. They adopted the fuzzy Petri Net to formulate the decision processes based on the train dispatching rule in the case of abnormality, in order to obtain any possible dispatching options. The dispatching decision rules, factors, and possible options when perturbation happens were collected via expert interviews and literature reviews. The fuzzy membership function of individual dispatching factors derived the correspondent fuzzy value and incorporated it in the fuzzy Petri Net approach to simulate appropriate dispatching options under various abnormal circumstances such as; centralized traffic control system failure, automatic train protection failure, and locomotive failure. Dispatch decision factors were; train type, uncompleted distance, train connection, track layout, average stopping time at the platform, current train delay, passenger balance, and travelling type. The dispatching options were; overtaking, backed trains, added trains, and combined trains. The objective of the study was to minimize the total passenger delay.

2.4 Articles on Both Scheduling/Timetabling and Rescheduling/Dispatching

Komaya & Fukuda (1991) are the first authors who dealt with not only scheduling /timetabling but also rescheduling/dispatching in the same paper. They proposed a problem solving architecture for knowledge based integration of simulation and scheduling, and described two knowledge based systems for railway scheduling; DIAPLAN and ESTRAC-III. The objective was to minimize total delay time that was the sum of the delays of each train at each station. In order to emulate experts' problem solving processes the architecture included four components; partial simulation, basic command, tactical knowledge, and strategic knowledge. The partial simulation and the basic command were employed for simulation of train movements in a subsystem, the tactical knowledge was employed for local scheduling, and the strategic knowledge was employed to manage the order of integrating simulation with scheduling in order to solve subproblems. DIAPLAN and ESTRAC-III were designed to support experts in planning and restoration of railway systems, respectively. DIAPLAN was able to prepare a complete timetable from given initial conditions for a train, and ESTRAC-III could prepare a rescheduling plan in the case of disturbed train traffic.

Medanic & Dorfman (2002c) concentrated on modelling a single line and presented an approach, which was called travel advance strategy (TAS), which was based on a discrete event model of a railway line. The discrete event formulation removed the complexity of the scheduling problem and allowed one to obtain suboptimal time efficient and energy efficient schedules. The energy related system wide performance was measured by the sum of all energy costs. The system wide time related performance measures were; the total time-to-clear-the-track criterion that was the time interval from the instant the first train on the schedule leaves its point of departure till the instant the last train on the schedule arrives at its destination, the total delay criterion that was the total delay of all trains, and the maximal delay criterion that was the maximum delay of any individual train. In Dorfman & Medanic (2004) TAS concept proposed in Medanic & Dorfman (2002c) was extended to scheduling trains in a railway networks. Also, a capacity check was introduced that prevents deadlock from occurring due to the developed train schedule. The rules developed for the TAS for a single railway line and the three time related performance criteria were appropriately extended to the network situation. Also, in this study extensions to the strategy were developed for networks with double track sections and with variable train priorities. The TAS was a service discipline at each meet/pass point as to which train in the vicinity of another (i.e., on adjacent sections of the line) should continue to travel, and which trains should be stopped at a meet/pass point. The TAS can be used to quickly develop schedules for perturbed cases; a change in a particular departure time, a velocity modification in some section for some train, an existence of lateness in scheduled train, an addition of new train into a given schedule. The TAS did not generate mathematically optimal solutions, but developed suboptimal schedules that closely approach the optimal in practical situations.

Chang & Chung (2005) proposed train operation model that considered not only the flexibility of train regulation, or train rescheduling problem, but also the objectives of timetabling process. There were two stages in the proposed operating mechanism. In the first stage, they used the historical passenger flow provided by the railway company to construct an optimized train timetable, called “planned timetable”. Since the occurrence of an unpredicted event might disrupt the planned timetable, in the second stage, they developed a rescheduling process. A GA was applied to solve the problem. The objective function was to minimize the average travel time of passengers and to maximize the utilization of the trains.

2.5 Discussion on the Literature Review

Although many papers indicate the study of Szpigel (1973) to be the first study on the *TrnSchPrb*, we see that the article of Frank (1966) is the first one in this area. An interesting point is that although the first study was published many years ago, it is still going on to be cited by recent papers such as Cheng & Yang (2009), Yang et al. (2009), Castillo et al. (2009) and Lee & Chen (2009).

There exist a few review papers where the authors concentrate on the problem in only one section. Although there have been more than one hundred articles, published so far, deals with the *TrnSchPrb*, to the best of our knowledge, none of these papers have directly focused on the problem.

The preliminary articles in the relevant area were focused on the scheduling and used mathematical models. On the other hand, in a few papers simulation models were constructed for the scheduling problem. The first article which focused on the rescheduling problem was published 17 years later than the first one on the scheduling problem. While the mathematical models were generally used for scheduling, simulation models were often built for rescheduling.

The metaheuristics were employed by the researchers in the relevant area after 1990s, multi objectives were optimized after 2000s. We confronted with only a few articles on both the scheduling and the rescheduling. They were published in 1990s.

Although the current study focuses on the scheduling/timetabling problem, the simulation integrated framework developed can also be used for the rescheduling /dispatching problem if this framework can be fed by real time data. The current study can be located among a few studies which comprise simulation modelling for scheduling/timetabling.

In the studies of Ping et al. (2001), Rebreyend (2005), and Geske (2006) a simulation model was developed and integrated with a GA. The first two studies are on the rescheduling that is not the main scope of the current study. In the last one the scheduling problem was considered and a deterministic simulation model was built. To the best of our knowledge our study is the first one which integrates a stochastic simulation model with GA and also with hybrid GAs to deal with the scheduling problem.

2.6 The First Studies in the Relevant Literature

As a result of our literature review, the studies which are the first in their area are listed in Table A.10 in Appendix in chronological order. Some of them are given below.

- Frank (1966) is the first author among the others who studied a single tracked line with double tracked stations. This infrastructure has been widely used in the scheduling literature.
- Salzborn (1969) is the first author who considered passenger dimension, headway and dwell time.
- The first study where overtaking, passenger demand, capacity of train concepts, and different train types were employed belongs to Nemhauser (1969).
- One train at a time based algorithm was presented firstly by Amit & Goldfarb (1971). The technical data (speed limits, acceleration time, slope and curves), motive power units and crews concepts were also firstly placed in this study.
- The first study that developed a branch and bound algorithm is Szpigel (1973).
- The first study that developed a simulation model for the problem is Wong & Rosser (1978).
- The first review paper belongs to Assad (1980).
- The first study that mentioned the problem of line blockage and also took the train length into account is Petersen & Taylor (1982).
- The first study that dealt with the rescheduling (dispatching) problem is Sauder & Westerman (1983), and this is also the first study that developed a shortest path algorithm and considered a network instead of a single line.
- The first study that used a rule based solution approach is Araya, Abe & Fukumori (1983).
- The first study that included train velocity as a decision variable in the model is Kraay et al. (1991).
- The first study that dealt with not only the scheduling (timetabling) problem but also the rescheduling (dispatching) problem in the same study is Komaya & Fukuda (1991).

- The first study that used a constraint based solution approach is Chiang & Hau (1993).
- The first study that integrated tabu search and simulated annealing metaheuristics to its solution approach is Chiang & Hau (1995).
- The first study assumed that the trains can follow each other on a track segment with a minimum headway is Higgins et al. (1996).
- The first study that focused on the periodic *TrnSchPrb* is Odijk (1996).
- The first study that developed a GA for solution is Nachtigall & Voget (1996).
- The first study that described an algorithm possessed a capability to schedule physical backups (reverse) of some trains if it is needed is Cai et al. (1998).
- The first study that included cancelling a train as an option instead of scheduling all the trains is Brännlund et al. (1998), and also Lagrangian relaxation solution approach is used in the first time in this study.
- The first study that had more than one objective (multi objective) to be optimized simultaneously is Chang et al. (2000), and also this is the first study that included fuzzy concept in its solution approach.
- Petri Net approach was firstly used for solution of the *TrnSchPrb* by Fay (2000).
- A graph model was firstly evolved by Caprara et al. (2001).
- The first study that mentioned the attraction of the service for the passengers is Peeters & Kroon (2001).
- The first study that took the junctions into account is Semet & Schoenauer (2005).
- The first studies that used ant colony heuristic in order to solve the problem are Ghoseiri & Morshedsolouk (2006) and Su & Huang (2006), which were published in the same year.
- The first study that permitted real time decision on alternative train routes is Mazzarello & Ottaviani (2007).
- The first study that mentioned condensation zone and compensation zone concepts is Caimi et al. (2007).
- The first study that concentrated on only a terminus part of a track is Flamini & Pacciarelli (2008).

CHAPTER THREE
AN OVERVIEW OF GENETIC ALGORITHMS

GAs are powerful and broadly applicable stochastic search and optimization approaches, and simulate the natural behaviour of biological systems. Holland (1975) introduced the developed fundamental ideas of GAs and then GAs became a popular method by the study of David Goldberg (Goldberg, 1989), one of Holland's students, who solved a difficult problem involving the control of gas pipeline transmission by using GA.

GAs have been successfully adapted to solve several combinatorial optimization problems for finding optimal or near optimal solutions in a reasonable time (Gen & Cheng, 1997; Gen & Cheng, 2000; Gen et al., 2008; Yu & Gen, 2010). A typical GA might consist of the followings (Coley, 2003);

- A number, or population, of guesses of the solution to the problem.
- A way of calculating how good or bad the individual solutions within the population are.
- A method for mixing fragments of the better solutions to form new, on average even better solutions.
- A mutation operator to avoid permanent loss of diversity within the solutions.

Goldberg (1989) introduced the differences of GAs from traditional optimization techniques in four ways (Gen & Cheng, 1997);

- GAs work with a coding of the parameter set, not the parameters themselves.
- GAs search from a population of points, not a single point.
- GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules.

GAs maintain a population of chromosomes (individuals), each of them represents a solution to the problem at hand. Each chromosome is evaluated to give measure of its fitness. In order to create new chromosomes some chromosomes (parents) from the population undergo stochastic transformations by means of genetic operators; crossover and mutation. Crossover creates new chromosomes (children, offspring) by combining parts (mating) from generally two parents. Mutation creates new individuals by making changes (mutating) in a single chromosome. A new population is formed by selecting individuals 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 individual, which represents an optimal or suboptimal solution to the problem at hand (Goldberg, 1989; Gen & Cheng, 1997; Gen & Cheng, 2000; Coley, 2003; Haupt & Haupt, 2004; Gen et al., 2008; Yu & Gen, 2010).

Haupt & Haupt (2004) mentioned the below advantages of GAs;

- Optimize with continuous or discrete variables,
- Do not require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deal with a large number of variables,
- Are well suited for parallel computers,
- Optimize variables with extremely complex cost surfaces (they can jump out of a local minimum),
- Provide a list of optimum variables, not just a single solution,
- May encode the variables so that the optimization is done with the encoded variables, and
- Work with numerically generated data, experimental data, or analytical functions.

GAs have received considerable attention regarding their potential as a novel optimization technique. There are three major advantages when applying GA to optimization problems (Gen et al., 2008);

- *Adaptability*: GA does not have much mathematical requirement regarding about the optimization problems. Due to the evolutionary nature, GA will search for solutions without regard to the specific inner workings of the problem. GA can handle any kind of objective functions and any kind of constraints, i.e., linear or nonlinear, defined on discrete, continuous or mixed search spaces.

- *Robustness*: The use of evolution operators makes GA very effective in performing a global search (in probability), while most conventional heuristics usually perform a local search. It has been proved by many studies that GA is more efficient and more robust in locating optimal solution and reducing computational effort than other conventional heuristics.
- *Flexibility*: GA provides flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem.

3.1 GA Vocabulary

Because GA is rooted in both natural genetics and computer science, the terminologies used in GA literatures are a mixture of the natural and the artificial. In a biological organism, the structure that encodes the prescription that specifies how the organism is to be constructed is called a *chromosome*. One or more chromosomes may be required to specify the complete organism. The complete set of chromosomes is called a *genotype*, and the resulting organism is called a *phenotype*. Each chromosome comprises a number of individual structures called *genes*. Each gene encodes a particular feature of the organism, and the location, or *locus*, of the gene within the chromosome structure, determines what particular characteristic the gene represents. At a particular locus, a gene may encode one of several different values of the particular characteristic it represents. The different values of a gene are called *alleles*. The correspondence of GA terms and optimization terms is summarized in Table 3.1 (Gen & Cheng, 1997; Gen et al., 2008).

Table 3.1 Explanation of GA terms

GAs	**Explanation**
Chromosome (string, individual)	Solution (Coding)
Genes (Bits)	Part of the Solution
Locus	Position of Gene
Alleles	Values of Gene
Phenotype	Decoded Solution
Genotype	Encoded Solution

GAs have a number of components and operators that must be specified in order to define a particular GA. The most important are given below.

3.2 Components of GAs

Representation (encoding of chromosomes). The first step in defining a GA is to link the “real world" to the “GA world", that is to set up a bridge between the original problem context and the problem solving space where evolution will take place. Objects forming possible solutions within the original problem context are referred to as phenotypes, their encoding, the individuals within the GA, are called genotypes. The first design step is commonly called representation, as it amounts to specifying a mapping from the phenotypes onto a set of genotypes that are said to represent these phenotypes. It is important to understand that the phenotype space can be very different from the genotype space, and that the whole evolutionary search takes place in the genotype space (Eiben & Smith, 2003).

Binary encoding (Figure 3.2) is the most common one, mainly because the first research of GA used this type of encoding and because of its relative simplicity. In binary encoding, every chromosome is a string of bits 0 or 1. Permutation encoding (Figure 3.3) can be used in ordering problems, such as travelling salesman problem or task ordering problem. In permutation encoding, every chromosome is a string of numbers that represent a position in a sequence. Direct value encoding (Figure 3.4) can be used in problems where some more complicated values such as real numbers are used. In the value encoding, every chromosome consists some values that can be anything connected to the problem.

Chromosome A	1	0	1	1	0	0	1	1	0
Chromosome B	1	1	1	1	1	0	0	0	1

Figure 3.2 Examples for binary encoding

Chromosome A	1	5	3	2	6	4	7	9	8
Chromosome B	8	5	6	7	2	3	1	4	9

Figure 3.3 Examples for permutation encoding

Chromosome A	1.23	5.32	3.45	2.47	6.32	4.63	7.12	9.01	8.25
Chromosome B	b	h	a	g	e	d	i	f	c
Chromosome C	1	5	6	3	4	2	1	4	6

Figure 3.4 Examples for direct value encoding

Population. The role of the population is to hold (the representation of) possible solutions. A population is a multiset (a set where multiple copies of an element are possible) of genotypes. Given a representation, defining a population can be as simple as specifying how many individuals are in it, that is, setting the *population size* (Eiben & Smith, 2003). The initial population is usually generated randomly. There are also other alternatives. One of them is to carry out a series of initializations for each individual and then pick the highest performing values. Another alternative is to locate approximate solutions by using other methods (i.e., simulated annealing, tabu search) and to start the algorithm from such points (Coley, 2003).

Evaluation. Each chromosome is evaluated and assigned a fitness value after the creation of an initial population. The fitness evaluation is a black box for the GA this may be achieved by a mathematical function, a simulation model, or a human expert that decides the quality of a chromosome.

Parent selection. The role of parent selection or mating selection is to distinguish among individuals based on their quality, in particular, to allow the better individuals to become parents of the next generation. An individual is a parent if it has been selected to undergo variation in order to create offspring (Eiben & Smith, 2003).

Crossover. It is the main genetic operator that operates on two chromosomes at a time and generates offspring by combining both chromosomes' features. A simple way to achieve crossover would be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut point with the segment of the other parent to the right of the cut point (Gen et al., 2008). For two individuals selected to cross over, we assign a point between 1 and $l-1$ randomly, where l is the length of the chromosome. This means generating a random integer in the range [1, $l-1$]. The genes after the point are changed between parents and the resulting chromosomes are offspring. This operator is called *single point crossover* exhibited in Figure 3.5 (Yu & Gen, 2010).

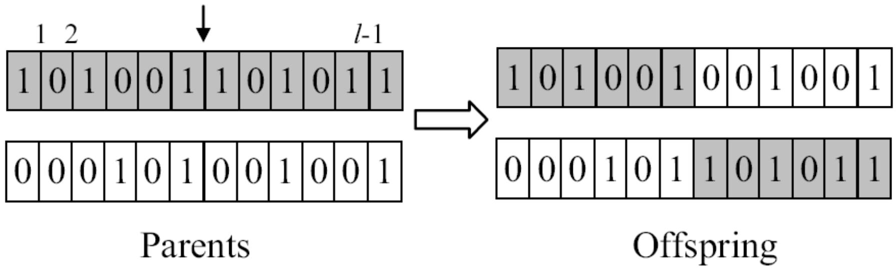


Figure 3.5 Single point crossover

The *crossover rate* is defined as the probability of the number of offspring produced in each generation to the population size. This rate controls the expected number of chromosomes to undergo the crossover operation. A higher crossover rate allows exploration of more of the solution space, and reduces the chances of settling for a false optimum; but if this rate is too high, it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space (Gen et al., 2008).

Mutation. It is a background operator which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes. In GA, mutation serves the crucial role of either (*a*) replacing the genes lost from the population during the selection process so that they can be tried in a new context or (*b*) providing the genes that were not present in the initial population (Gen & Cheng, 1997; Gen et al., 2008). An illustration is given in Figure 3.6, in where the j^{th} gene is changed from 1 to 0 randomly (Yu & Gen, 2010).

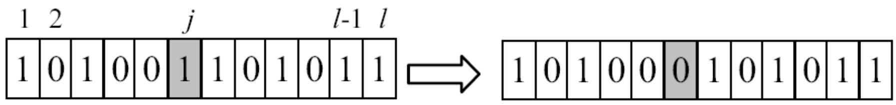


Figure 3.6 An illustration of mutation

The *mutation rate* controls the probability with which new genes are introduced into the population for trial. If it is too low, many genes that would have been useful are never tried out, while if it is too high, there will be much random perturbation, the offspring will start losing their resemblance to the parents, and the algorithm will lose the ability to learn from the history of the search (Gen et al., 2008).

A *replacement strategy* is required in order to form new generation. This strategy determines which chromosomes stay in population and which are replaced by offsprings, generated by crossover or mutation. The individuals of the new generation may be (*a*) individuals from the current generation, (*b*) offspring product of crossover or (*c*) individuals who underwent mutation. One of the commonly used replacement strategy is *elitism*, which makes survive some number of the best individuals at each generation, hence guaranteeing that the final population contains the best solution ever found (Gen & Cheng, 1997; Gen & Cheng, 2000; Coley, 2003; Haupt & Haupt, 2004).

Termination criteria. Unlike other search methods that terminate when a local optimum is reached, GAs are stochastic search methods that could in principle run forever. In practice, a termination criterion is needed, common approaches are; to set a limit on the number of fitness evaluations or the computer clock time, or to track the population's diversity and stop when this falls below a preset threshold (Gen & Cheng, 1997; Gen & Cheng, 2000; Coley, 2003; Haupt & Haupt, 2004).

The flowchart of the used GA in this study is exhibited in Figure 3.1.

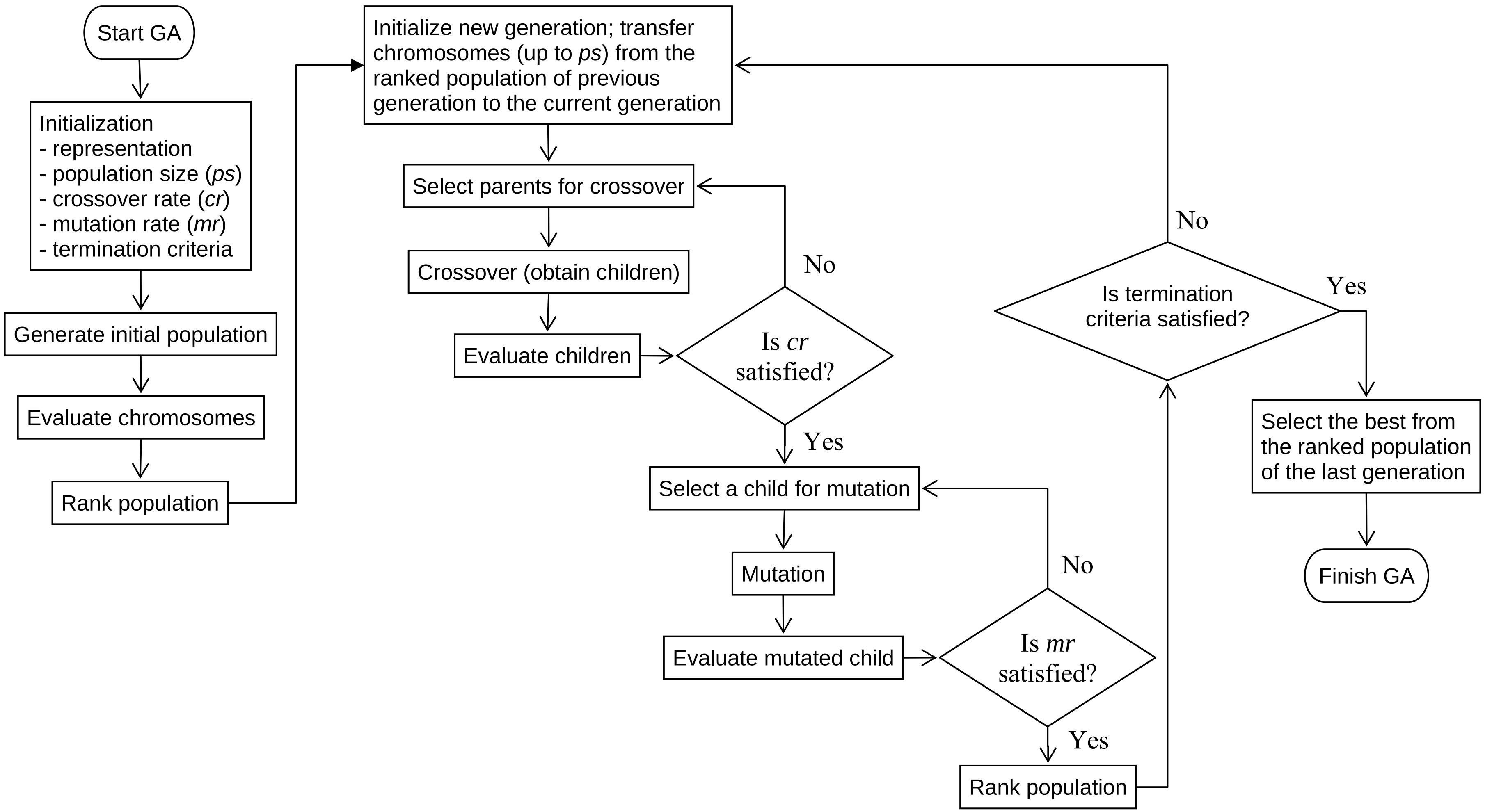


Figure 3.1 Flowchart of a GA

3.3 Hybrid GAs

GA have proved to be a versatile and effective approach for solving optimization problems. Nevertheless, there are many situations in which the simple GA does not perform particularly well, and various methods of hybridization have been proposed. One of most common forms of hybrid GA is to incorporate local optimization as an add on extra to the conventional GA loop. With the hybrid approach, local optimization is applied to each newly generated offspring to move it to a local optimum before injecting it into the population. GA is used to perform global exploration among a population while heuristic methods are used to perform local exploitation around chromosomes (Gen et al., 2008).

A hybrid GA combines the power of the GA with the speed of a local optimizer, in where GA finds the region of the optimum, and then the local optimizer takes over to find the better. Hybrid GA can take one of the following forms;

- Running a GA until it slows down, then letting a local optimizer take over. Hopefully the GA is very close to the global minimum.
- Seeding the GA population with some local minima found from random starting points in the population.
- Every so many iterations, running a local optimizer on the best solution or the best few solutions and adding the resulting chromosomes to the population. (Haupt & Haupt, 2004)

CHAPTER FOUR

A FEASIBLE TIMETABLE GENERATOR SIMULATION MODELLING FRAMEWORK FOR TRAIN SCHEDULING PROBLEM

In this chapter, a feasible timetable generator simulation modelling framework for the *TrnSchPrb* is given. The objective is to obtain a feasible train timetable for all trains in the system. The feasible train timetable includes train arrival and departure times at all visited stations with calculated average train travel time. This chapter involves three subsections. In the first subsection, a hypothetic *TrnSchPrb* is introduced. In the next section, the simulation modelling framework is developed and applied on the hypothetic *TrnSchPrb.* In the last subsection, the results are discussed.

4.1 A Hypothetic Train Scheduling Problem

The proposed simulation modelling framework is implemented on a hypothetic *TrnSchPrb*. The infrastructure in the problem has a line structure inspired by a real railway line system, and has a planned initial timetable with arrival and departure times of trains only at two end stations of the infrastructure.

4.1.1 Railway Line Description

The railway line, which is inspired by a real line, is a single track corridor as analogous to many lines in the literature and in real railway systems. The line-station diagram of the single track corridor and the infrastructure of stations are shown in Figure 4.1. There are 10 real stations on the single track corridor that are labelled as Si (i = 1, 2, …, 10) from the east to the west.

The single track corridor has two terminuses, TS1 and TS10. TS1 at the east point and TS10 at the west point indicate the beginning and the finishing points of the single track corridor. As it is seen in Table 4.1, the total track length from the TS1 to the TS10 is 286270 meters. Since all the real stations have 200 meters platform for boarding and alighting events, and there are 10 real stations on the corridor, the whole length of the corridor is 288270 meters.

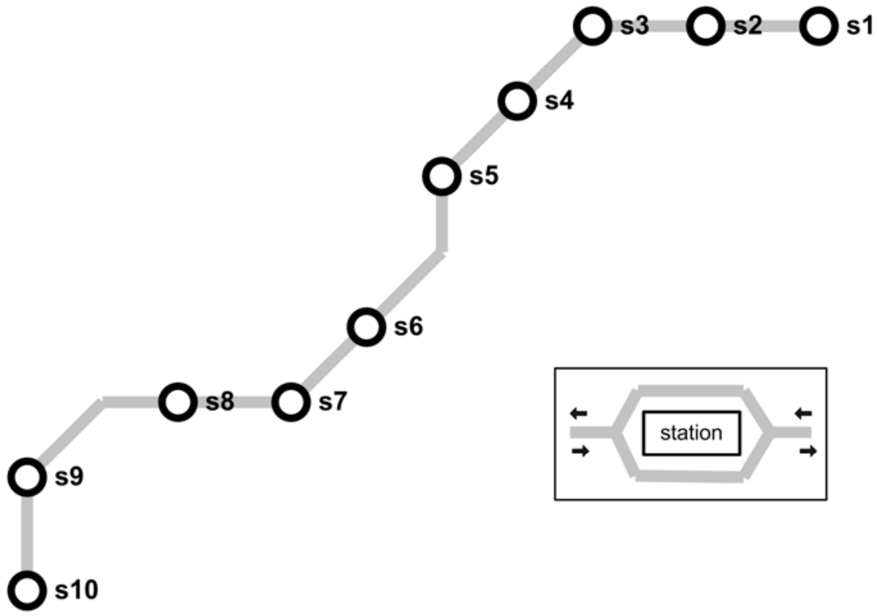


Figure 4.1Line-station diagram of the single track corridor

Table 4.1 Track lengths between the real stations

To From	TS1 (*East*)	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	TS10 (*West*)
TS1 (*East*)	0	500	28070	60170	88400	125210	170060	197060	214460	243560	285770	286270
S1	500	0	27570	59670	87900	124710	169560	196560	213960	243060	285270	285770
S2	28070	27570	0	32100	60330	97140	141990	168990	186390	215490	257700	258200
S3	60170	59670	32100	0	28230	65040	109890	136890	154290	183390	225600	226100
S4	88400	87900	60330	28230	0	36810	81660	108660	126060	155160	197370	197870
S5	125210	124710	97140	65040	36810	0	44850	71850	89250	118350	160560	161060
S6	170060	169560	141990	109890	81660	44850	0	27000	44400	73500	115710	116210
S7	197060	196560	168990	136890	108660	71850	27000	0	17400	46500	88710	89210
S8	214460	213960	186390	154290	126060	89250	44400	17400	0	29100	71310	71810
S9	243560	243060	215490	183390	155160	118350	73500	46500	29100	0	42210	42710
S10	285770	285270	257700	225600	197370	160560	115710	88710	71310	42210	0	500
TS10 (*West*)	286270	285770	258200	226100	197870	161060	116210	89210	71810	42710	500	0

4.1.2 Planned Initial Train Timetable

The train arrival and departure times at two end stations are given in Table 4.2, where WB*i* (*i* = 1, 2, …, 10) indicates a *westbound* train that begins its trip from the S1 (on the *east* of the corridor) and plans to finish at the S10 (on the *west* of the corridor), and EB*i* (*i* = 1, 2, …, 10) indicates an *eastbound* train which departures from the S10 (on the *west* of the corridor) and arrives to the S1 (on the *east* of the corridor). The planned initial timetable is formed with the assumption that there will be 20 trains running in a day, 10 of them are the WB trains and the other 10 are the EB trains.

Table 4.2 Planned initial train timetable

Station	**Train**	**Arrival Time**	**Departure Time**
S1	WB1	00:00	00:10
	WB2	02:00	02:10
	WB3	04:00	04:10
	WB4	06:00	06:10
	WB5	08:00	08:10
	WB6	10:00	10:10
	WB7	12:00	12:10
	WB8	14:00	14:10
	WB9	16:00	16:10
	WB10	18:00	18:10
S10	EB1	00:00	00:10
	EB2	02:00	02:10
	EB3	04:00	04:10
	EB4	06:00	06:10
	EB5	08:00	08:10
	EB6	10:00	10:10
	EB7	12:00	12:10
	EB8	14:00	14:10
	EB9	16:00	16:10
	EB10	18:00	18:10

4.2 A Feasible Timetable Generator Simulation Model

A feasible timetable generator simulation model is developed by using ARENA 10.0 discrete event simulation software in a modular manner. First, the railway corridor with links, intersections and the stations is modelled, and track failures and repairs are included. Then, train movement logic on the corridor is modelled. We use a rule for track allocation to candidate trains that are the trains waiting at neighbour stations of the track to use it. Fixed train speeds are relaxed, and additional unplanned delays at the stations are inserted. The number of trains in the system is increased and randomness is added to the planned initial train timetable. As a last step, animation of the system is developed.

Some assumptions are made during modelling phase of the simulation model. It must be noted that many our assumptions are the fundamental assumptions made by the existing studies.

Some of the assumptions made for our simulation model are;

- The unit for length and time is meter and second, respectively.
- It takes 32 seconds for trains to reach the real stations S1 and S10 from park area, then the trains wait 568 seconds at these stations, i.e., they spend totally 600 seconds (10 minutes) as a dwell time. First trips are planned to begin at 00:10:00 o'clock. But due to additional unplanned delays at the stations lateness may occur.
- Time spent for reaching to a terminus (TS1 or TS10) from the park area is negligible.
- The WB trains' departure station is the S1 and destination is the S10, and the EB trains' departure station is the S10 and destination is the S1.
- There will be 20 trains running in a day, 10 of them are the WB and the other 10 are the EB trains.
- All the trains are the same type.
- Passengers are ignored at this level of the model.
- There is a time interval (headway) between two consecutive trains at a station, which have the same trip direction, in order to have a safe trip.
- The train lengths are 50 meters.
- Earliness and lateness time in the planned initial train timetable, due to some uncontrollable events that occur out side of the corridor, is uniformly distributed between -900 and +900 seconds.

4.2.1 Railway Corridor Modelling

The detailed line-station diagram of the corridor is denoted in Figures 4.2(a)-4.2(c). The letter “E” indicates the east and the letter “W” indicates the west directions.

The railway corridor is a union of intersections and links, and modelled via *Networks Element* of the ARENA. The links are the track parts on which train traverses during its trip from a station to another neighbour station. The links are numbered from 1 to 111, and shown as lines in Figures 4.2(a)-4.2(c), and are modelled via *Links Element* of the ARENA.

The intersections, the connection points of the links, are numbered from 1 to 102 and shown in lozenge shape in Figures 4.2(a)-4.2(c). The big lozenge shapes denote the intersections related to the real stations, and have lengths in meter. For instance, the big lozenge shape numbered as 2 is related to the first part of the S2 and connects the link 9 with link 10. The small lozenge shapes denote the intersections that are only used for connecting the links and dummy stations that are located on the tracks between the real stations to keep a train wait during the repairing of a track failure, and have no length. For instance, the small lozenge shape numbered as 22 connects the link 1 with link 2 and link 4. The intersections numbered from 63 to 102 are related to the park areas where the empty trains can park. The intersections are modelled via *Intersections Element* of the ARENA.

The stations are locations where a train can stop for boarding and alighting events, for parking or for waiting until a failure is accomplished. The real stations, Si (i = 1, 2, …, 10), are interrelated with 2 intersections, that is, the real stations have capacity of two, at most two trains can locate on a real station at the same time. The dummy stations, dSij (j = 1 for i = 7; j = 1, 2 for i = 1, 2, 3, 6, 8; j = 1, 2, 3 for i = 4, 5, 9), are located on the tracks between the real stations to keep a train wait during the repairing of a failure, if the failure occurs while a train is traversing between the real stations. The stations are modelled via *Stations Element* of the ARENA.

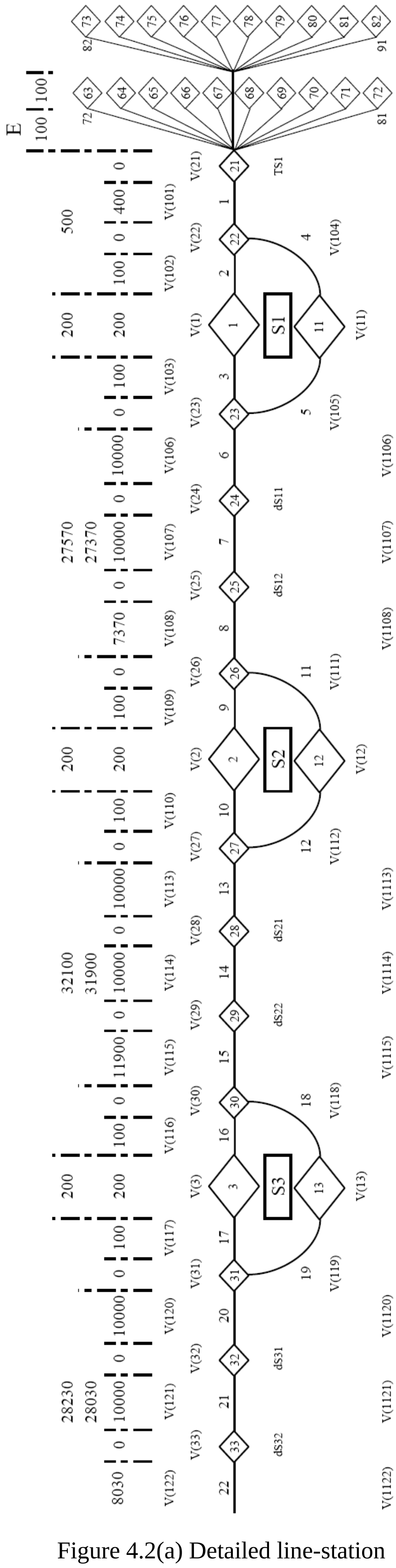


Figure 4.2(a) Detailed line-station diagram of the corridor from S1 to S3

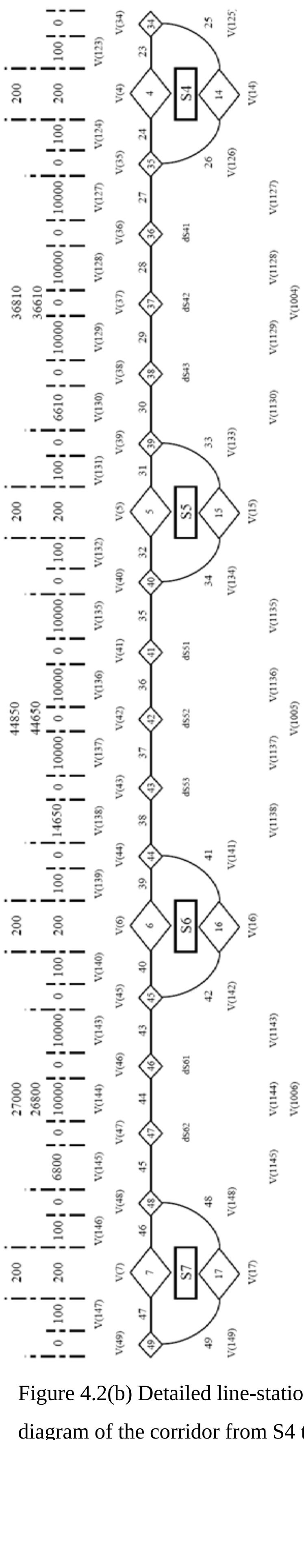


Figure 4.2(b) Detailed line-station diagram of the corridor from S4 to S7

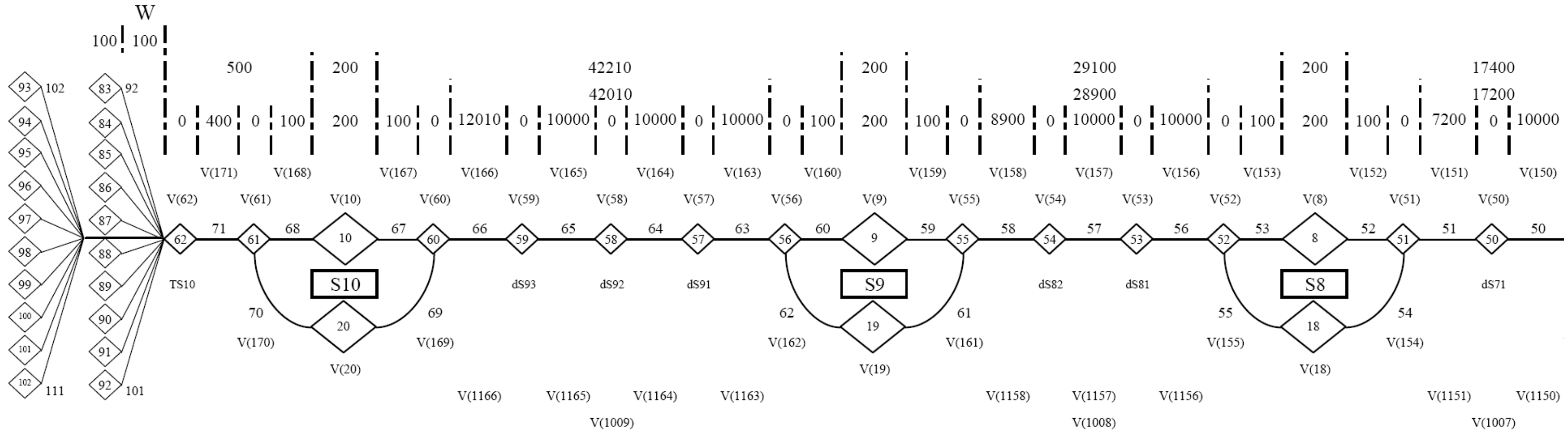


Figure 4.2(c) Detailed line-station diagram of the corridor from S8 to S10

The links and the intersections related to the corridor are shown in Table 4.3, and the stations related to the corridor are depicted in Table 4.4. In *Description* column of the Table 4.3, a brief explanation is given to define the corresponding link and intersection. In this column, while describing an intersection the dummy station related with the described intersection is given.

Table 4.3 Links and intersections related to the corridor

	Name	**Description**
Link number		
i = 1, …, 111	Lnk(i)	Link (*i*) in the railway network that is the track part trains traverse on
Intersection number		
i = 1, ..., 10	Int(i)_S(i)_1	Intersection (*i*) interrelated with first part of station (*i*)
i = 11, ..., 20	Int(i)_S(i-10)_2	Intersection (*i*) interrelated with second part of station (*i* -10)
21	Int21_TS1	Intersection interrelated with TS1
i = 22, 23, 26, 27, 30, 31, 34, 35, 39, 40, 44, 45, 48, 49, 51, 52, 55, 56, 60, 61	Int(i)	Intersection (*i*) in the railway network that is connecting neighbour links
24	Int24_dS11	dS11
25	Int25_dS12	dS12
28	Int28_dS21	dS21
29	Int29_dS22	dS22
32	Int32_dS31	dS31
33	Int33_dS32	dS32
36	Int36_dS41	dS41
37	Int37_dS42	dS42
38	Int38_dS43	dS43
41	Int41_dS51	dS51
42	Int42_dS52	dS52
43	Int43_dS53	dS53
46	Int46_dS61	dS61
47	Int47_dS62	dS62
50	Int50_dS71	dS71
53	Int53_dS81	dS81
54	Int54_dS82	dS82
57	Int57_dS91	dS91
58	Int58_dS92	dS92
59	Int59_dS93	dS93
62	Int62_TS10	Intersection interrelated with TS10
i = 63, ..., 102	Int(i)_park	Intersection (*i*) interrelated with park area

Table 4.4 Stations related to the corridor

	Name	**Description**
Station number		
i = 1, ..., 10	S(i)_1	First part of real station (*i*)
i = 11, ..., 20	S(i-10)_2	Second part of real station (*i* -10)
21	TS1	Terminus TS1
24	dS11	Dummy station
25	dS12	Dummy station
28	dS21	Dummy station
29	dS22	Dummy station
32	dS31	Dummy station
33	dS32	Dummy station
36	dS41	Dummy station
37	dS42	Dummy station
38	dS43	Dummy station
41	dS51	Dummy station
42	dS52	Dummy station
43	dS53	Dummy station
46	dS61	Dummy station
47	dS62	Dummy station
50	dS71	Dummy station
53	dS81	Dummy station
54	dS82	Dummy station
57	dS91	Dummy station
58	dS92	Dummy station
59	dS93	Dummy station
62	TS10	Terminus TS10
i = 63, ..., 102	Sta(i)_park	Station (*i*) interrelated with park area

In the simulation model the links, the intersections and the track failures are controlled via variables. For instance, the link 2 is controlled by a variable that has number 102 in *Variables Element* of the simulation model and demonstrated by V(102) in Figure 4.2(a). The variables related to the corridor model are given in Table 4.5. In this table, *Description* column defines the dummy station related with the intersection that is controlled by the corresponding variable.

Assumptions related to the railway corridor part of the simulation model are;

- The railway system is a single track line, a corridor.
- The traffic on tracks is bidirectional, two way.
- All the real stations have 200 meters platforms for boarding and alighting events.

Table 4.5 Variables related to the corridor

	Name	**Description**
Variable number		
i = 1, ..., 10	V(i)_S(i)_1_Int(i)	Variable (*i*) controls intersection (*i*) interrelated with first part of station (*i*)
i = 11, ..., 20	V(i)_S(i-10)_2_Int(i)	Variable (*i*) controls intersection (*i*) interrelated with second part of station (*i* -10)
21	V21_TS1_Int21	Variable controls intersection 21 interrelated with TS1
i = 22, 23, 26, 27, 30, 31, 34, 35, 39, 40, 44, 45, 48, 49, 51, 52, 55, 56, 60, 61	Var(i)_Int(i)	Variable (*i*) controls intersection (*i*)
24	V24_dS11_Int24	dS11
25	V25_dS12_Int25	dS12
28	V28_dS21_Int28	dS21
29	V29_dS22_Int29	dS22
32	V32_dS31_Int32	dS31
33	V33_dS32_Int33	dS32
36	V36_dS41_Int36	dS41
37	V37_dS42_Int37	dS42
38	V38_dS43_Int38	dS43
41	V41_dS51_Int41	dS51
42	V42_dS52_Int42	dS52
43	V43_dS53_Int43	dS53
46	V46_dS61_Int46	dS61
47	V47_dS62_Int47	dS62
50	V50_dS71_Int50	dS71
53	V53_dS81_Int53	dS81
54	V54_dS82_Int54	dS82
57	V57_dS91_Int57	dS91
58	V58_dS92_Int58	dS92
59	V59_dS93_Int59	dS93
62	V62_TS10_Int42	Variable controls intersection 62 interrelated with TS10
i = 101, ..., 171	V(i)_Lnk(i-100)	Variable (*i*) controls link (*i* -100)

- There are 10 real stations and 20 dummy stations on the corridor, that is, the corridor is 288270 meter long.
- The terminuses (TS1 and TS10) have infinitive train capacity.
- The terminus TS1 is located on the east point, and the terminus TS10 is located on the west point of the corridor.
- Every middle real station has capacity of two trains, that is, there will be at most two trains at a real station at the same time.

- Every dummy station has capacity of one train, that is, there will be at most one train at a dummy station at a specific time.
- There are 100 meter length park areas near the terminuses. These park areas are used by a train, which finished its trip, while leaving the corridor, or waiting to enter the corridor.
- Distance between these park areas and terminuses are 100 meters.

4.2.2 Track Failure Modelling

Track failure is an event that prevents a train to occupy the impaired track for a trip. The train can use the track after it is repaired. Distributions for failure times and repair times for the failed tracks are depicted in Table 4.6.

Table 4.6 Failure times and repair times distributions

Variable number	Location	Length (meter)	Rank	Ratio	1/Ratio	Failure time distribution	Repair time distribution
1001	S1-S2	27370	3	1.591	0.628	Expo (54296)	Expo (2864)
1002	S2-S3	31900	6	1.855	0.539	Expo (46586)	Expo (3338)
1003	S3-S4	28030	4	1.630	0.614	Expo (53017)	Expo (2933)
1004	S4-S5	36610	7	2.128	0.470	Expo (40592)	Expo (3831)
1005	S5-S6	44650	9	2.596	0.385	Expo (33283)	Expo (4673)
1006	S6-S7	26800	2	1.558	0.642	Expo (55451)	Expo (2805)
1007	S7-S8	17200	1	1.000	1.000	Expo (86400)	Expo (1800)
1008	S8-S9	28900	5	1.680	0.595	Expo (51421)	Expo (3024)
1009	S9-S10	42010	8	2.442	0.409	Expo (35374)	Expo (4396)

In the first three columns, variable number, location and length of the tracks are given. These tracks are ranked according to their lengths. The shortest one has rank 1 and lies between the S7 and the S8, and is selected to be *base track*. The longest one has rank 9 and lies between the S5 and the S6. The ratios are obtained by dividing lengths of the tracks to the length of the *base track*.

It is assumed that failure time of the *base track* is distributed exponentially with a mean of 86400 seconds (24 hours), that is, it is expected to observe one failure for the *base track* in a day. Failure times of the other tracks are also assumed to be distributed exponentially with a mean of 86400 seconds (24 hours) times 1/ratio.

That is, they have expected values inverse ratio to their lengths, namely the longer the track part the more frequently the track failure occurrence. For instance, it is assumed that failure time of the longest track is distributed exponentially with a mean of 33283 seconds (9.25 hours), that is, it is expected to observe more than two (24 / 9.25 = 2.6) failures in a day for the longest track. The repair time distribution is assumed to be exponential with a mean of 1800 seconds (0.5 hours) for the *base track*, and with a mean of 1800 seconds (0.5 hours) times ratio value for the other tracks.

We divided the long tracks into smaller parts (links) and located the dummy stations between the links as depicted in Figures 4.2(a)-4.2(c). After obtaining failures for tracks according to the distributions shown in Table 4.6, these failures are transferred to the links with the probabilities exhibited in Table 4.7.

In the simulation model, the track failures are controlled via variables. If a failure occurs in a track part, trains are prevented to use this part until it is repaired. The variables related to the track failure model are exhibited in Table 4.8.

The line-station diagram of the track between the S5 and the S6 is given in Figure A.1 in Appendix, and the SIMAN View of the failure model logic related to the track between the S5 and the S6 is given in Table A.1 in Appendix. The flowcharts for *track failure event* and *track repair event* are exhibited in Figure 4.3 and Figure 4.4 respectively.

Assumptions related to the track failure part of the simulation model are;

- There will be track failures that will stop traffic on the related track. The failure and repair times distributions are given in Table 4.6.
- The failure time of the *base track* is distributed exponentially with a mean of 86400 seconds (24 hours), that is, it is expected to observe one failure for the *base track* in a day.
- Failure times for other tracks are also exponentially distributed with a mean of 86400 seconds (24 hours) times 1/ratio value. That is, failure times of other

tracks have expected values inverse ratio to their lengths, namely the longer the track part the more frequently the track failure occurrence.

- The repair time distribution is exponential with 1800 seconds (0.5 hours) mean for the *base track*.
- Repair times for other tracks are also exponentially distributed with a mean of 1800 seconds (0.5 hours) times related ratio value.
- After failures for the tracks were created according to the probability distributions shown in Table 4.6, these failures are transferred to the links due to the probabilities given in Table 4.7.

Table 4.7 Failure probabilities of the links

Variable number of track	Location of track	Length of track (meter)	Link number	Variable number	Length (meter)	Probability (%)	Location of link
1001	S1-S2	27370	6	1106	10000	33	S1-dS11
			7	1107	10000	33	dS11-dS12
			8	1108	7370	34	dS12-S2
1002	S2-S3	31900	13	1113	10000	33	S2-dS21
			14	1114	10000	33	dS21-dS22
			15	1115	11900	34	dS22-S3
1003	S3-S4	28030	20	1120	10000	33	S3-dS31
			21	1121	10000	33	dS31-dS32
			22	1122	8030	34	dS32-S4
1004	S4-S5	36610	27	1127	10000	25	S4-dS41
			28	1128	10000	25	dS41-dS42
			29	1129	10000	25	dS42-dS43
			30	1130	6610	25	dS43-S5
1005	S5-S6	44650	35	1135	10000	25	S5-dS51
			36	1136	10000	25	dS51-dS52
			37	1137	10000	25	dS52-dS53
			38	1138	14650	25	dS53-S6
1006	S6-S7	26800	43	1143	10000	33	S6-dS61
			44	1144	10000	33	dS61-dS62
			45	1145	6800	34	dS62-S7
1007	S7-S8	17200	50	1150	10000	50	S7-dS71
			51	1151	7200	50	dS71-S8
1008	S8-S9	28900	56	1156	10000	33	S8-dS81
			57	1157	10000	33	dS81-dS82
			58	1158	8900	34	dS82-S9
1009	S9-S10	42010	63	1163	10000	25	S9-dS91
			64	1164	10000	25	dS91-dS92
			65	1165	10000	25	dS92-dS93
			66	1166	12010	25	dS93-S10

- If a track failure happens while a train is traversing on this track and if the next station is a dummy one, train goes to next dummy station and a check is made if the failure is on that train's destination direction or not. If the failure is on its destination side, the train waits until failure is repaired, else the train goes on its trip.

Table 4.8 Variables related to the track failure model

	Name	**Description**
Variable number		
i = 1001, …, 1009	V(i)_fS(i-1000)	Variable (*i*) controls track failure between S(*i*) and S(*i* +1), default value is 0, takes value 1 when there is a failure, and after repair again takes value 0
i = 1106, 1107, 1108, 1113, 1114, 1115, 1120, 1121, 1122, 1127, 1128, 1129, 1130, 1135, 1136, 1137, 1138, 1143, 1144, 1145, 1150, 1151, 1156, 1157, 1158, 1163, 1164, 1165, 1166	V(i)_fLnk(i-1100)	Variable (*i*) controls track failure interrelated with link (*i* -1100), default value is 0, takes value 1 when there is a failure, and after repair again takes value 0

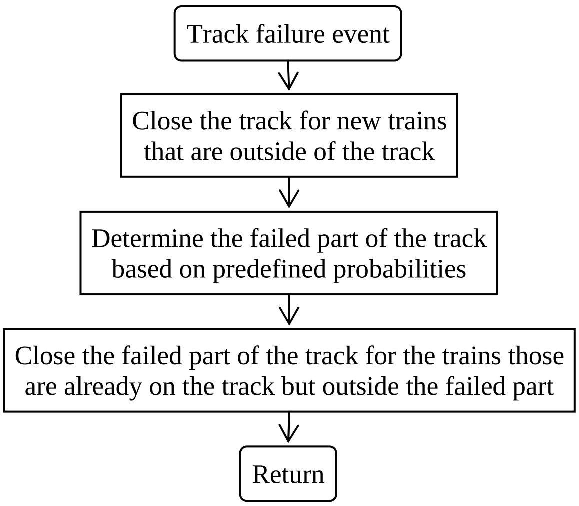


Figure 4.3 Flowchart for track failure event

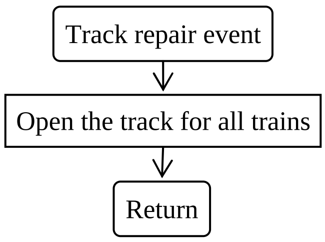


Figure 4.4 Flowchart for track repair event

4.2.3 Train Movement Modelling

The train movement modelling logic in the simulation model is explained on some parts of the corridor.

4.2.3.1 Train Movement Logic from Park Area to a Real Station via a Terminus

Assume that a train is moving from the park area located at the east to reach the S1 via the TS1. In Figure A.2 in Appendix, the line-station diagram of the TS1 and its neighbourhood is given. The SIMAN Views of the train movement logic from the park area to the TS1 is depicted in Table A.2 in Appendix, and the train movement logic at the TS1 is denoted in Table A.3 in Appendix. The flowcharts for *train movement from the park area event* and *train arrival to the terminus event* are given in Figures 4.5 and 4.6 respectively.

4.2.3.2 Train Movement Logic at a Real Station

Assume that a train is just arrived to the first part of the real station S5 that is interrelated with the intersection 5 in the simulation model. The line-station diagram of the S5 and its neighbourhood were given in Figure 4.2(b). The SIMAN View of the train movement logic at the first part of the S5 is shown in Table A.4 in Appendix. The flowcharts for *train arrival to the real station event* and *train departure from the real station event* are exhibited in Figure 4.7 and 4.8 respectively.

4.2.3.3 Train Movement Logic at a Dummy Station

We assume that a train is just arrived to the dummy station dS51. The line-station diagram of the dS51 and its neighbourhood were depicted in Figure 4.2(b). The SIMAN View of the train movement logic at the dummy station is shown in Table A.5 in Appendicx. The flowcharts for *train arrival to the dummy station event* and *train departure from the dummy station event* are exhibited in Figure 4.9 and 4.10 respectively.

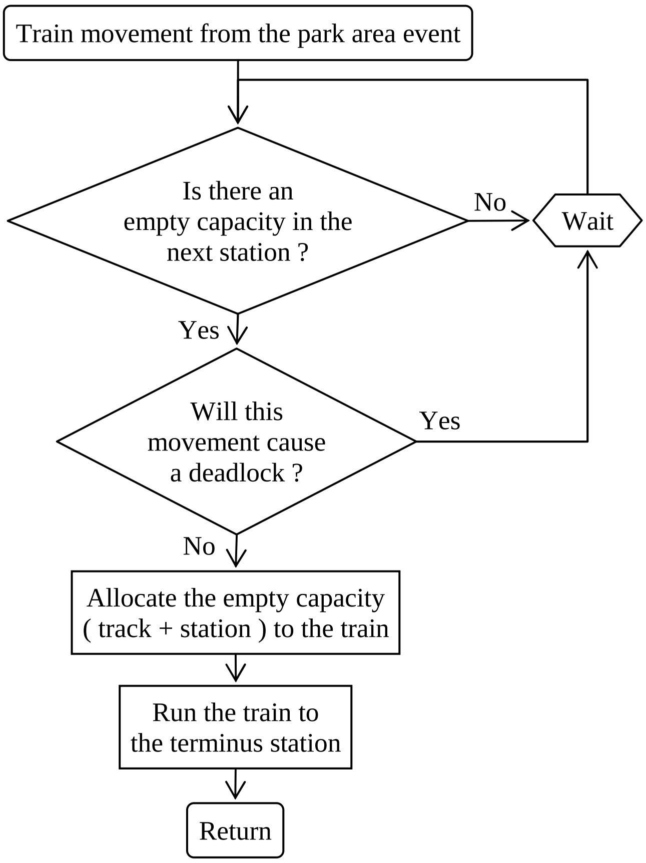


Figure 4.5 Flowchart for train movement from the park area event

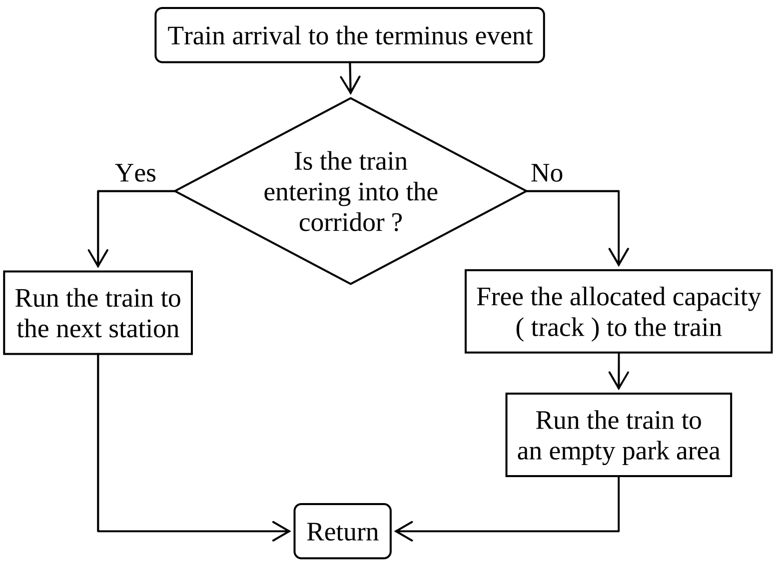


Figure 4.6 Flowchart for train arrival to the terminus event

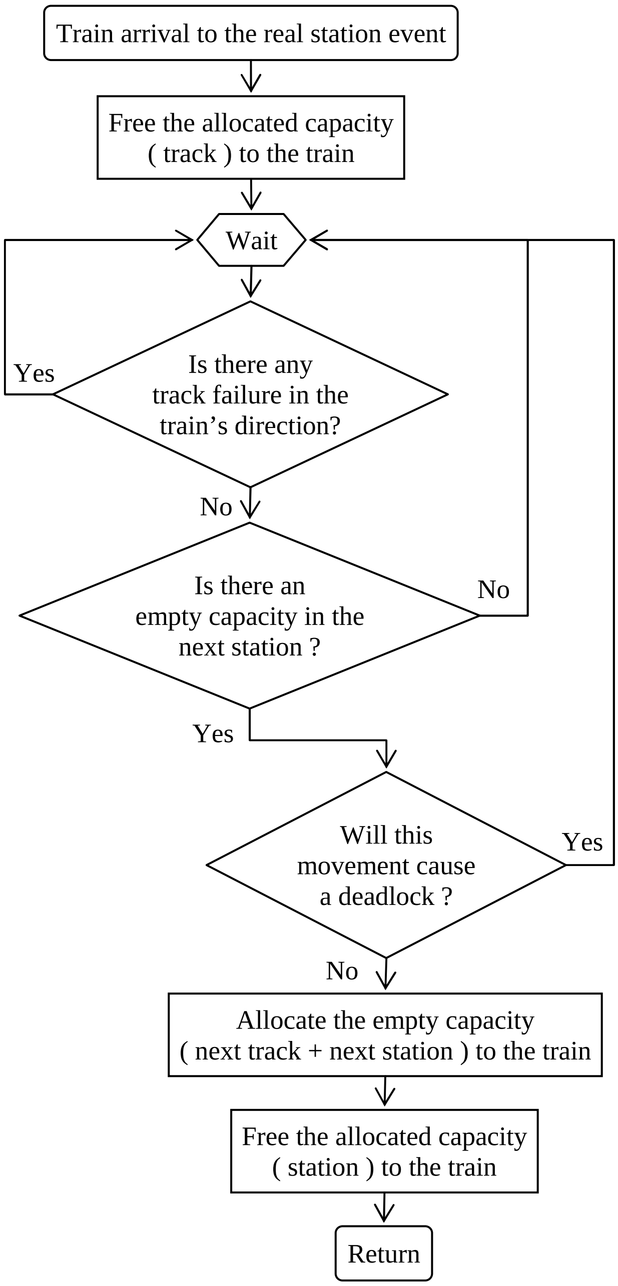


Figure 4.7 Flowchart for train arrival to the real station event

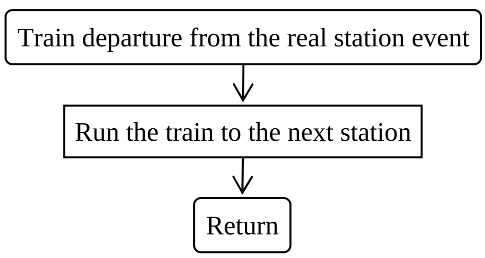


Figure 4.8 Flowchart for train departure from a real station event

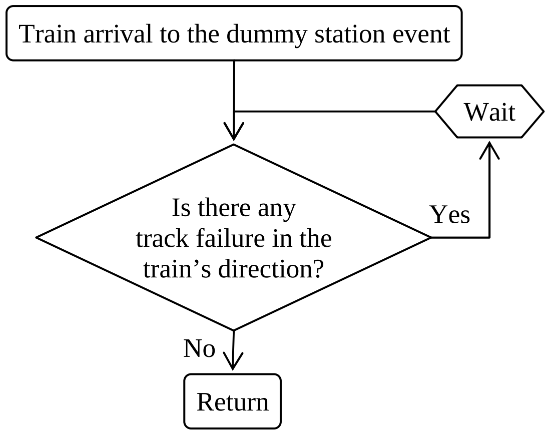


Figure 4.9 Flowchart for train arrival to the dummy station event

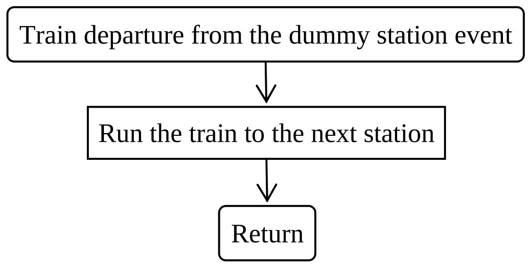


Figure 4.10 Flowchart for train departure from the dummy station event

Assumptions related to the train movement part of the simulation model are as follows.

- Trains' speeds are uniformly distributed over an interval (90 kilometres/hour, 110 kilometres/hour).
- Dwell times for each station are 600 seconds (10 minutes). That is each train will stop at least 600 seconds at the all stations for boarding and alighting events.
- To represent unplanned delays at a station, a time, it is assumed that delay time is exponentially distributed with a mean of 90 seconds. Delay time is added to the dwell times. Due to this unplanned delay, overtaking is possible.
- Each train stops at real stations except terminuses.
- A train stops at a dummy station if there is a failure in a track placed in front of that train.

- Track occupying decision is taken at the real stations based on the answers given the following questions; Are the links and intersections suitable? Does a track failure exist? Does this decision cause a deadlock?
- First come first served (FCFS) dispatching rule is used to select one train among the candidate trains, which are the trains waiting at neighbour stations of the track that want to use the same track. If the all conditions to move are suitable for a candidate train, which arrived first to one of neighbour station of the track it will begin to trip, else the same check is made for another train arrived second. Checking goes on until a suitable train is found.
- Trains that have reverse directions can cross each other only at the real stations.
- The number of currently running trains on the corridor can not exceed total available capacity of the stations minus one (e.g. 20 in our hypothetic case)

4.2.4 Blockage Preventive Algorithm

A common potential deadlock is exhibited in Figure 4.11 where there are four trains; two trains are the WB trains and located at S(*i*) and the other trains are the EB trains and located at S(*i*+1). As can be seen in this figure, the system has a deadlock. Deadlock situation goes on until one of those trains reverses its direction.

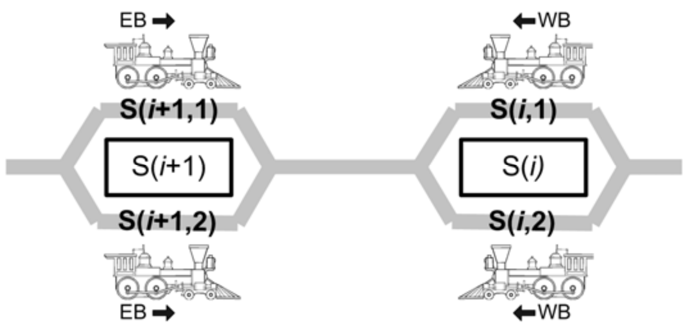


Figure 4.11 An example of a deadlock

Another example of a deadlock is shown in Figure 4.12. There are two WB trains at S(*i*), two EB trains at S(*i*+2) and two trains (not important to be a WB or an EB train) at S(*i*+1). Since we avoid reversing the direction of trains the deadlock problem can not be solved. In order to obtain a feasible train timetable we must take course of action to prevent a deadlock.

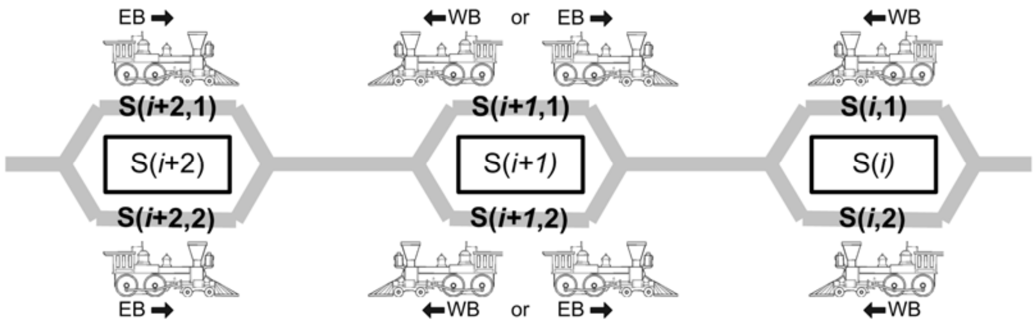


Figure 4.12 Deadlock between six trains

In order to avoid a deadlock, which prevents the movement of trains, the whole system is checked in the direction of the train before permitting the train to departure from its current station. The Blockage Preventive Algorithm, given in Table 4.9, is developed to obtain a deadlock free system and consequently a feasible train timetable.

4.2.5 Verification of the Simulation Model

The simulation model is verified by developing the model in a modular manner, using interactive debuggers, substituting constants for random variables, manually checking the results and animating the system.

In order to develop the simulation model in a modular manner a step by step approach was used. This approach gives an ability to systematically model a complex system. First the railway network that includes the links, the intersections and the stations was modelled. Then, the track failures and repairs were added, and the train movement logic on the railway corridor was modelled. Next, we added a rule for track allocation to the candidate trains. The fixed train speeds were relaxed, and then additional unplanned delays at the stations were modelled. The number of trains in the system was increased and randomness was added to the planned initial train timetable. As a last step, animation of the system was developed. The animation part of the simulation model was built by using *Animate* tool of the ARENA, to see if model is working as intended, and to understand the system clearer.

Table 4.9 Blockage preventive algorithm

Assume that a WB train is at the $S(i, j)$ station and aimed to travel to the $S(i+1, 1)$ station where $i = 1, \ldots, 9$ denotes the station number and $j = 1, 2$ denotes the platform number.

Step = 1
- **$S(i+1, 1)$ empty ?**
  - *Yes*: Go next.
  - *No*: Prohibit the train to travel, STOP the algorithm for that train.
- **$i+1 = 10$?**
  - *Yes*: Permit the train to travel, STOP checking blockage.
  - *No*: Go next.

Step = 2
- **$S(i+1, 2)$ empty *or* allocated to an EB train ?**
- ***or* $S(i+2, j)$ empty *or* allocated to a WB train ?**
  - *Yes*: Go next.
  - *No*: Prohibit the train to travel, STOP the algorithm for that train.
- **$i+2 = 10$?**
  - *Yes*: Permit the train to travel, STOP checking blockage.
  - *No*: Go next.

Step = 3
- **$S(i+1, 2)$ empty *or* allocated to an EB train ?**
- ***or* { $S(i+2, j)$ empty? }**
- ***or* $S(i+3, j)$ empty *or* allocated to a WB train ?**
  - *Yes*: Go next.
  - *No*: Prohibit the train to travel, STOP the algorithm for that train.
- **$i+3 = 10$?**
  - *Yes*: Permit the train to travel, STOP checking blockage.
  - *No*: Go next.

From Step = 4 to Step = 8
- **$S(i+1, 2)$ empty *or* allocated to an EB train ?**
- ***or* { $S(i+k, j)$ empty?; $k = 2, \ldots,$ (Step-1)}**
- ***or* $S(i+\text{Step}, j)$ empty *or* allocated to a WB train ?**
  - *Yes*: Go next.
  - *No*: Prohibit the train to travel, STOP the algorithm for that train.
- **$i+\text{Step} = 10$?**
  - *Yes*: Permit the train to travel, STOP checking blockage.
  - *No*: Go next.

Step = 9
- **$S(i+1, 2)$ empty *or* allocated to an EB train ?**
- ***or* { $S(i+k, j)$ empty?; $k = 2, \ldots,$ (Step-1)}**
- ***or* $S(i+\text{Step}, j)$ empty *or* allocated to a WB train ?**
  - *Yes*: Permit the train to travel, STOP the algorithm for that train.
  - *No*: Prohibit the train to travel, STOP the algorithm for that train.

4.3 Discussion

In this section we discuss on some results of the simulation model. As a first step, we begin with an initial train timetable we obtained. This timetable is infeasible since it includes conflicts. Secondly, we focus on a feasible, conflict free initial train timetable we obtained by the deterministic simulation model. Lastly, a feasible train timetable we obtained by the stochastic simulation model is given by detail.

4.3.1 Infeasible Planned Initial Train Timetable

The trains are scheduled on a single track corridor regarding the initial train timetable, where all the inputs are deterministic, given in Table 4.2. The corridor has 10 real and 20 dummy, totally 30 stations. There is no randomness in the planned train arrival and departure times, failures and repairs are excluded, train speeds are fixed at 100 kilometres / an hour, and no additional delay is added to the dwell times.

We begin our discussion on an empty corridor, on which only the WB1 train is running. The simulation model is run for this scenario and the train-station diagram exhibited in Figure 4.13 is obtained. The train travel time of the WB1 is calculated as 16350 seconds.

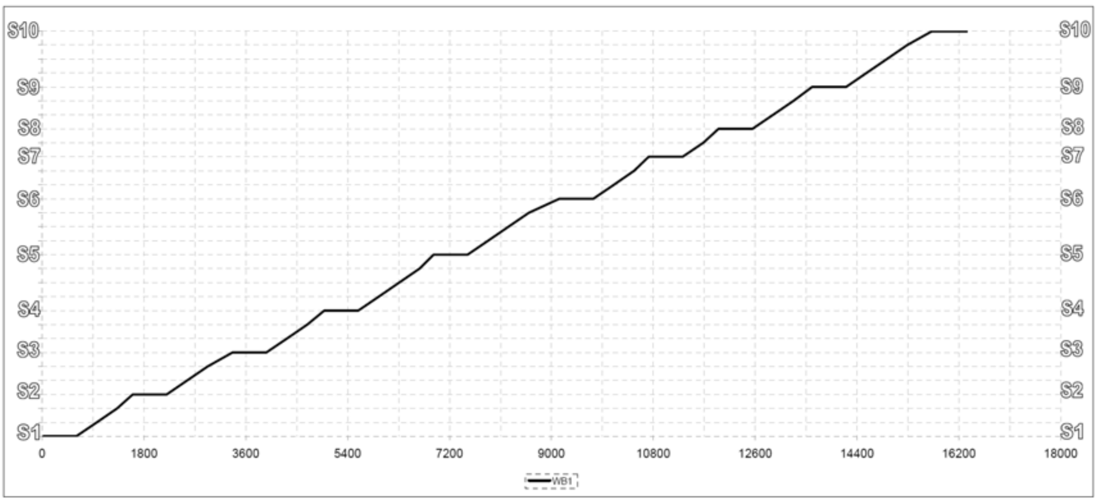


Figure 4.13 Train-station diagram for the WB1

The next scenario is related to the EB1 train that is running on the empty corridor. The train-station diagram exhibited in Figure 4.14 is obtained by the deterministic simulation model. The calculated train travel time is the same as the previous one, 16350 seconds.

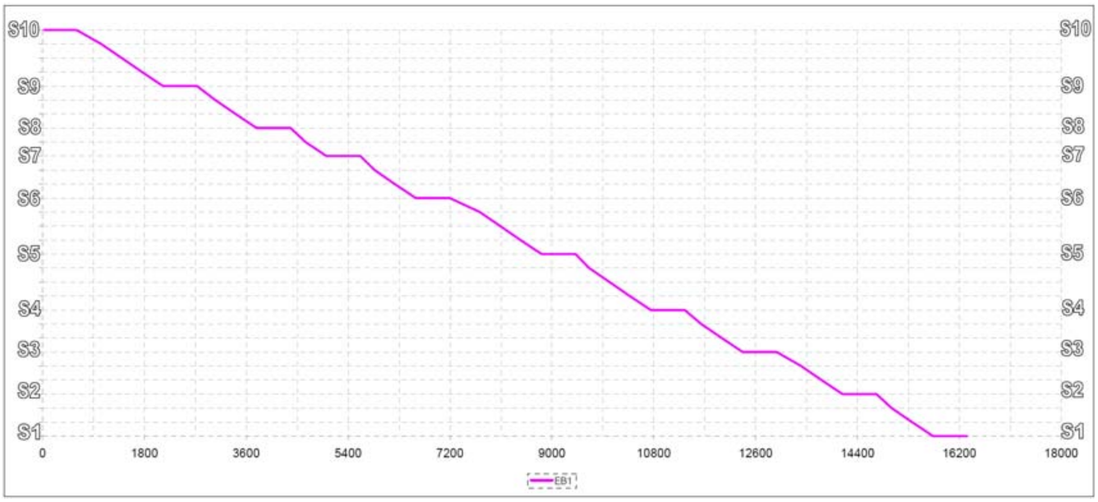


Figure 4.14 Train-station diagram for the EB1

After that, the train-station diagram for the planned initial train timetable given in Table 4.2 is manually obtained. This diagram is based on the WB1 and the EB1 train timetables, and depicted in Figure 4.15.

The timetable in Figure 4.15 is not conflict free, it is infeasible. The conflict locations are indicated by dotted line circles, and it is calculated that there are 44 conflicts to be solved in such a deterministic system.

To introduce the problem clearer the conflicts between the WB3 and some EB trains are displayed in Figure 4.16. As it is seen in this figure, the WB3 will have the first conflict with the EB1 train between the S1 and the S2. The other conflicts will be with the EB2, the EB3, the EB4 and the EB5 trains between the S3 - S4, between the S5 - S6, between the S7 - S8, and between the S9 - S10, respectively. The infeasible initial train timetable is given in Table 4.10.

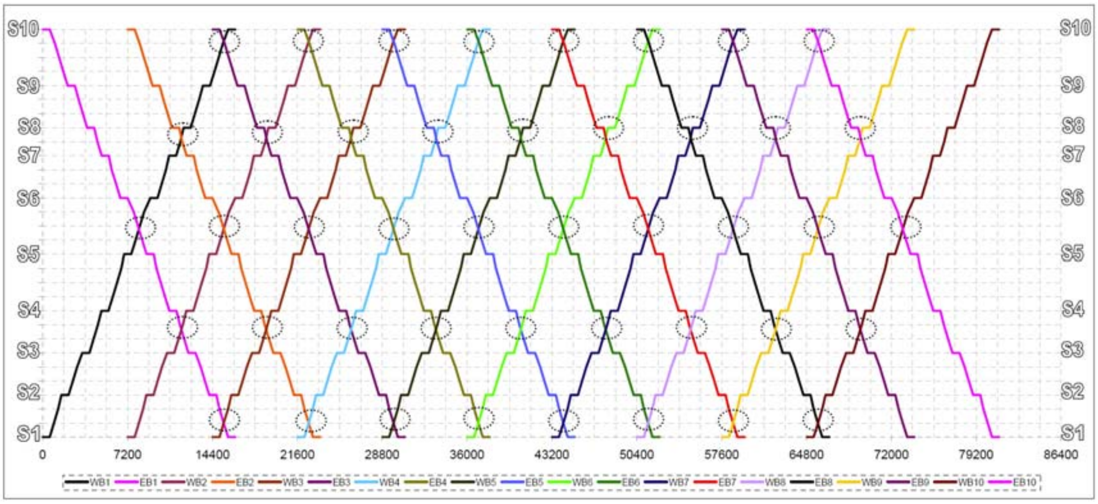


Figure 4.15 Infeasible train-station diagram for the planned initial train timetable

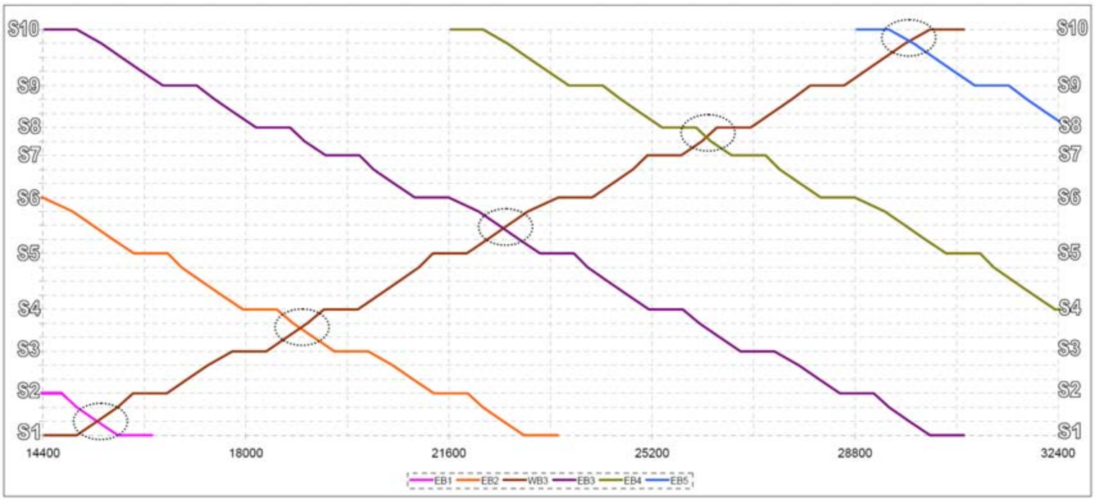


Figure 4.16 Infeasible train-station diagram for the WB3

Table 4.10 Infeasible initial train timetable

Train#	WB/ EB	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		Travel time	
		ArT	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*		
1	WB1	32	600	1600	2200	3363	3963	4986	5586	6919	7519	9140	9740	10719	11319	11952	12552	13607	14207	15734	16334	16350	04:32:30
2	EB1	15734	16334	14134	14734	12371	12971	10748	11348	8816	9416	6595	7195	5016	5616	3782	4382	2127	2727	32	600	16350	04:32:30
3	WB2	7232	7801	8800	9400	10563	11163	12186	12786	14119	14719	16340	16940	17919	18519	19152	19752	20807	21407	22934	23534	16350	04:32:30
4	EB2	22934	23534	21334	21934	19571	20171	17948	18548	16016	16616	13795	14395	12216	12816	10982	11582	9327	9927	7232	7801	16350	04:32:30
5	WB3	14432	15001	16000	16600	17763	18363	19386	19986	21319	21919	23540	24140	25119	25719	26352	26952	28007	28607	30134	30734	16350	04:32:30
6	EB3	30134	30734	28534	29134	26771	27371	25148	25748	23216	23816	20995	21595	19416	20016	18182	18782	16527	17127	14432	15001	16350	04:32:30
7	WB4	21632	22201	23200	23800	24963	25563	26586	27186	28519	29119	30740	31340	32319	32919	33552	34152	35207	35807	37334	37934	16350	04:32:30
8	EB4	37334	37934	35734	36334	33971	34571	32348	32948	30416	31016	28195	28795	26616	27216	25382	25982	23727	24327	21632	22201	16350	04:32:30
9	WB5	28832	29401	30400	31000	32163	32763	33786	34386	35719	36319	37940	38540	39519	40119	40752	41352	42407	43007	44534	45134	16350	04:32:30
10	EB5	44534	45134	42934	43534	41171	41771	39548	40148	37616	38216	35395	35995	33816	34416	32582	33182	30927	31527	28832	29401	16350	04:32:30
11	WB6	36032	36601	37600	38200	39363	39963	40986	41586	42919	43519	45140	45740	46719	47319	47952	48552	49607	50207	51734	52334	16350	04:32:30
12	EB6	51734	52334	50134	50734	48371	48971	46748	47348	44816	45416	42595	43195	41016	41616	39782	40382	38127	38727	36032	36601	16350	04:32:30
13	WB7	43232	43801	44800	45400	46563	47163	48186	48786	50119	50719	52340	52940	53919	54519	55152	55752	56807	57407	58934	59534	16350	04:32:30
14	EB7	58934	59534	57334	57934	55571	56171	53948	54548	52016	52616	49795	50395	48216	48816	46982	47582	45327	45927	43232	43801	16350	04:32:30
15	WB8	50432	51001	52000	52600	53763	54363	55386	55986	57319	57919	59540	60140	61119	61719	62352	62952	64007	64607	66134	66734	16350	04:32:30
16	EB8	66134	66734	64534	65134	62771	63371	61148	61748	59216	59816	56995	57595	55416	56016	54182	54782	52527	53127	50432	51001	16350	04:32:30
17	WB9	57632	58201	59200	59800	60963	61563	62586	63186	64519	65119	66740	67340	68319	68919	69552	70152	71207	71807	73334	73934	16350	04:32:30
18	EB9	73334	73934	71734	72334	69971	70571	68348	68948	66416	67016	64195	64795	62616	63216	61382	61982	59727	60327	57632	58201	16350	04:32:30
19	WB10	64832	65401	66400	67000	68163	68763	69786	70386	71719	72319	73940	74540	75519	76119	76752	77352	78407	79007	80534	81134	16350	04:32:30
20	EB10	80534	81134	78934	79534	77171	77771	75548	76148	73616	74216	71395	71995	69816	70416	68582	69182	66927	67527	64832	65401	16350	04:32:30
																		Average train travel time				16350	04:32:30

4.3.2 Feasible Planned Initial Train Timetable

The developed simulation model has the ability to solve the conflicts and create a conflict free feasible train timetable. The feasible train-station diagram obtained by our deterministic simulation model is given in Figure 4.17.

Figure 4.18 shows the conflict free feasible train station diagram for the WB3. We see that the EB1 train waits at the S2 and permits the WB3 to travel from the S1 to the S2, and the EB2 waits the WB3 at the S4 for empty the track part between the S4 and the S3. On the other hand the WB3 waits at S5 to permit EB3 to travel from the S6 to the S5. Since the WB3 spends additional time while waiting the EB3, the WB3 meets the EB4 at the S7 and passes without spending additional time. The WB3 waits the EB5 at the S9 and then finishes its trip. The feasible train timetable calculated by the deterministic simulation model is given in Table 4.11. The calculated average train travel time for this scenario is 17956 seconds.

In order to observe the change in the computer running time versus the number of trains in the deterministic simulation model, the model is run for different number of trains, the result are shown in Figure 4.19.

As it is seen in Figure 4.19(a) the computer running time is increasing nonlinearly when we increase the number of trains in the system. On the other hand, Figure 4.19(b) shows the increments in computer running time versus the added two trains. It is seen that the amount of increment caused by addition of two trains into the system depends on the number of current trains in the system. To make it clearer, we assume that there are 4 trains in the system. In this case, the computer running time is 9.0 seconds. When two new trains are added in the system (i.e. there will be 6 trains in the system), the computer running time will be 15.0 seconds (i.e. the increase in computer running time will be 6.0 seconds). On the other hand the increment will be 17.4 seconds if two trains are also added into the system that has 18 trains.

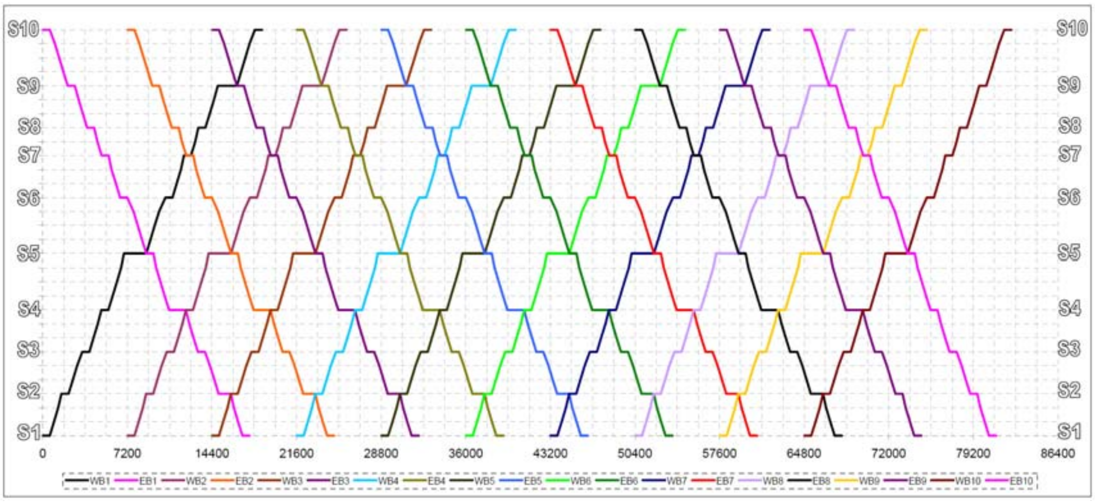


Figure 4.17 Feasible train-station diagram for the planned initial train timetable

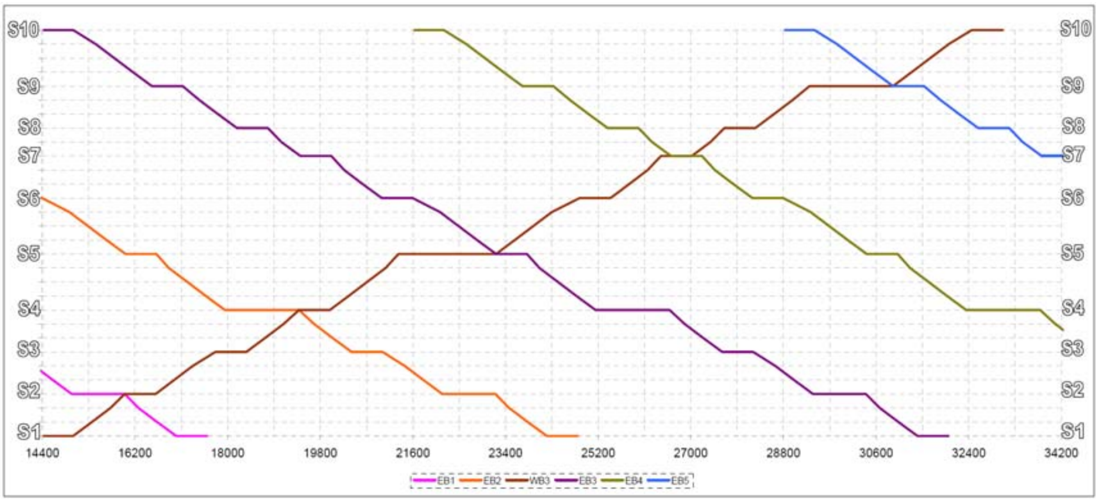


Figure 4.18 Feasible train-station diagram for the WB3

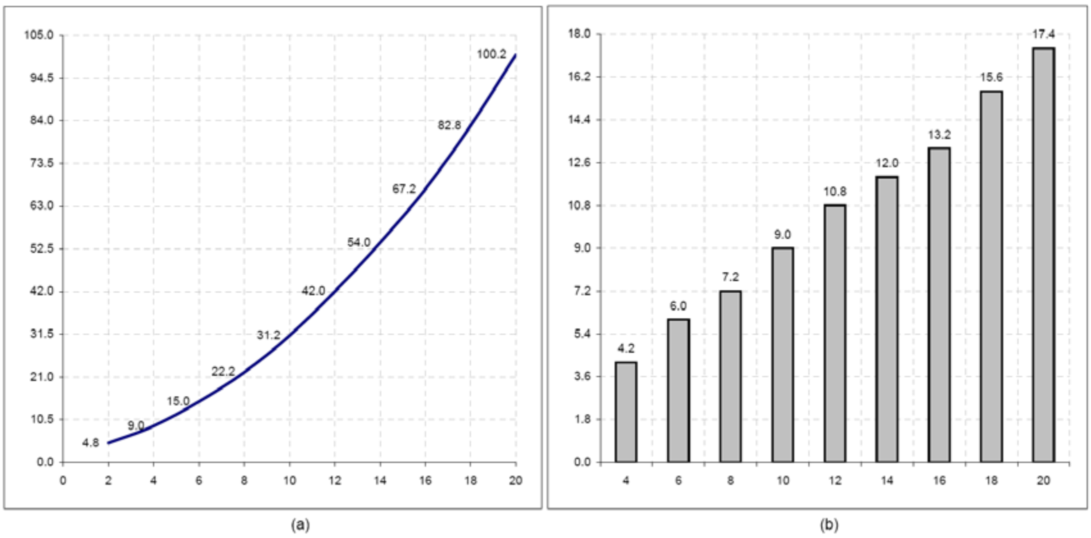


Figure 4.19 Computer running time for 20 replications versus the number of trains in the system

Table 4.11 Feasible planned initial train timetable

Train#	WB/ EB	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		Travel time	
		ArT	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*		
1	WB1	32	600	1600	2200	3363	3963	4986	5586	6919	8816	10438	11038	12017	12617	13250	13850	14905	16527	18053	18653	18670	05:11:10
2	EB1	17000	17600	14972	16000	13209	13809	10748	12186	8816	9416	6595	7195	5016	5616	3782	4382	2127	2727	32	600	17616	04:53:36
3	WB2	7232	7801	8800	9400	10563	11163	12186	12786	14118	16016	17638	18238	19217	19817	20450	21050	22105	23727	25254	25854	18670	05:11:10
4	EB2	24200	24800	22172	23200	20409	21009	17948	19386	16016	16616	13794	14394	12215	12815	10982	11582	9327	9927	7232	7801	17616	04:53:36
5	WB3	14431	15000	16000	16600	17763	18363	19386	19986	21318	23216	24838	25438	26417	27017	27650	28250	29305	30927	32454	33054	18670	05:11:10
6	EB3	31400	32000	29372	30400	27609	28209	25148	26586	23216	23816	20994	21594	19415	20015	18182	18782	16527	17127	14431	15000	17616	04:53:36
7	WB4	21631	22200	23200	23800	24963	25563	26586	27186	28518	30416	32038	32638	33617	34217	34850	35450	36505	38127	39654	40254	18670	05:11:10
8	EB4	38600	39200	36572	37600	34809	35409	32348	33786	30416	31016	28194	28794	26615	27215	25382	25982	23727	24327	21631	22200	17616	04:53:36
9	WB5	28831	29400	30400	31000	32163	32763	33786	34386	35718	37616	39238	39838	40817	41417	42050	42650	43705	45327	46854	47454	18670	05:11:10
10	EB5	45800	46400	43772	44800	42009	42609	39548	40986	37616	38216	35394	35994	33815	34415	32582	33182	30927	31527	28831	29400	17616	04:53:36
11	WB6	36031	36600	37600	38200	39363	39963	40986	41586	42918	44816	46438	47038	48017	48617	49250	49850	50905	52527	54054	54654	18670	05:11:10
12	EB6	53000	53600	50972	52000	49210	49810	46748	48186	44816	45416	42594	43194	41015	41615	39782	40382	38127	38727	36031	36600	17616	04:53:36
13	WB7	43231	43800	44800	45400	46563	47163	48186	48786	50118	52016	53638	54238	55217	55817	56450	57050	58105	59727	61254	61854	18670	05:11:10
14	EB7	60200	60800	58172	59200	56410	57010	53948	55386	52016	52616	49794	50394	48215	48815	46982	47582	45327	45927	43231	43800	17616	04:53:36
15	WB8	50431	51000	52000	52600	53763	54363	55386	55986	57318	59216	60838	61438	62417	63017	63650	64250	65305	66927	68454	69054	18670	05:11:10
16	EB8	67400	68000	65372	66400	63610	64210	61148	62586	59216	59816	56994	57594	55415	56015	54182	54782	52527	53127	50431	51000	17616	04:53:36
17	WB9	57631	58200	59200	59800	60963	61563	62586	63186	64518	66416	68038	68638	69617	70217	70850	71450	72505	73105	74632	75232	17648	04:54:08
18	EB9	74172	74772	72572	73172	70810	71410	68348	69786	66416	67016	64194	64794	62615	63215	61382	61982	59727	60327	57631	58200	17188	04:46:28
19	WB10	64832	65401	66400	67000	68163	68763	69786	70386	71718	73616	75238	75838	76817	77417	78051	78651	79705	80305	81832	82432	17648	04:54:08
20	EB10	80534	81134	78935	79535	77172	77772	75548	76148	73616	74216	71395	71995	69815	70415	68582	69182	66927	67527	64832	65401	16350	04:32:30
																		Average train travel time				17956	04:59:17

4.3.3 Feasible Train Timetable

In this section we discuss on a feasible train timetable obtained by our stochastic simulation model. The train-station diagram of the feasible solution of which the calculated average train timetable is 24218 seconds is given in Figure 4.20, and the related train timetable is depicted in Table 4.12.

In order to see what happens in the system after a failure was occurred, Figure 4.21 should be examined. In this figure, while the simulation model is running through 39600-79200 seconds a part of the system between the stations dS31 and the dS81 is displayed.

As can be seen in the dotted line circle denoted by 1, a failure occurred after the EB8 train begins its trip from the S5 to the S4. Therefore, the EB8 waits at the dS43 during the repair, and then the EB8 and the EB9 trains traverse on the track part between the S5 - S4. More than one train that have the same direction can use the same track with time headway between them.

While the simulation model is running through 39600-70200 seconds, a part of the system between dS43 and the dS71 is displayed in Figure 4.22, which is the dotted line rectangle denoted by 2 in Figure 4.21. In this figure, if we look at the dotted line rectangle, there is a track failure between the S6 and S7. After the track is repaired the trains can travel. But at that time there are more than one candidate trains waiting for using the repaired track part. To make it clearer we exhibited the train positions pi (i = 1, …, 39) on the track part between the S5 - S7 step by step in Figures 4.23(a) - 4.23(f).

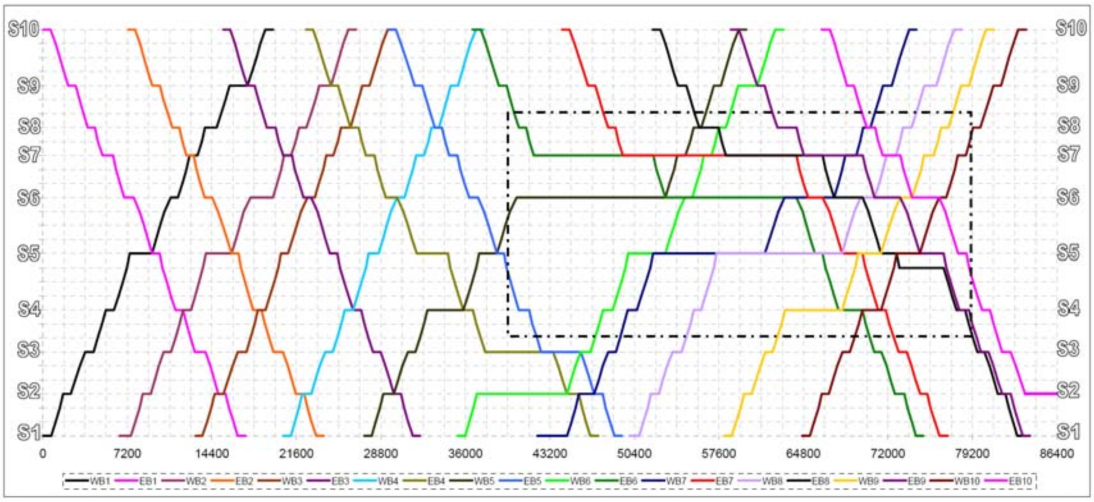


Figure 4.20 Feasible train-station diagram

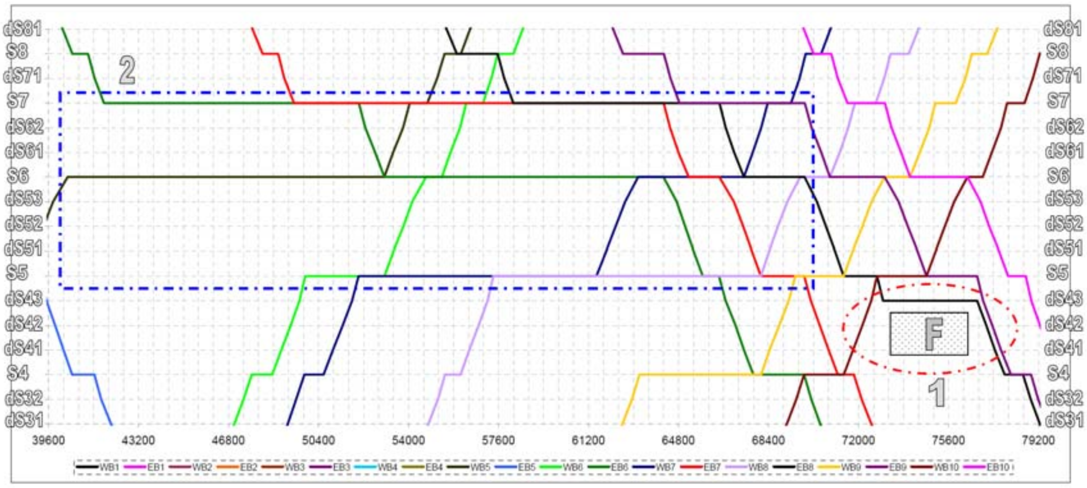


Figure 4.21 Feasible train-station diagram for the dS31-dS81 part from 39600 to 79200 seconds

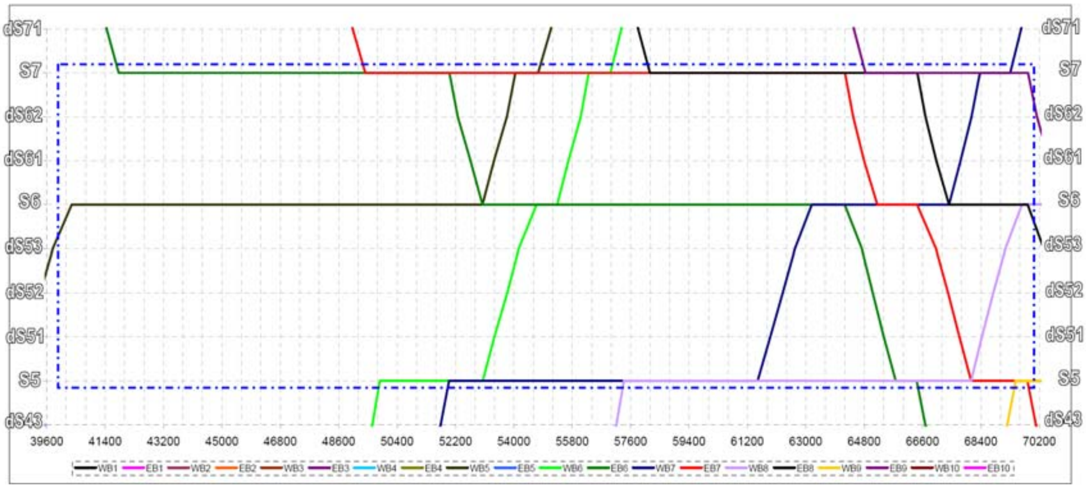


Figure 4.22 Feasible train-station diagram for the dS43-dS71 part from 39600 to 70200 seconds

Table 4.12 Feasible train timetable

Train#	WB/ EB	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		Travel time	
		ArT	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*	*ArT*	*DpT*		
1	WB1	32	710	1718	2370	3610	4334	5369	6058	7405	9305	10892	11500	12493	13179	13809	14801	15907	17423	18945	19620	19637	05:27:17
2	EB1	16591	17269	14957	15573	12943	13853	11303	11937	9305	9951	6932	7740	5097	5980	3829	4447	2159	2797	32	650	17285	04:48:05
3	WB2	6548	7504	8509	9188	10290	10922	11937	12622	13893	16067	17704	19598	20541	21152	21829	22458	23558	24545	26036	26679	20180	05:36:20
4	EB2	23290	23924	21554	22270	19653	20448	17916	18623	16067	16670	13801	14437	12227	12854	11029	11639	9325	9968	7283	7857	16690	04:38:10
5	WB3	13020	13591	14638	15388	16611	17340	18326	18976	20294	20895	22540	23171	24136	24749	25406	26202	27217	27838	29383	30010	17038	04:43:58
6	EB3	31527	32135	29896	30517	28123	28777	26433	27125	24447	25093	22209	22849	20483	21226	19079	19838	17423	18049	15379	15958	16803	04:40:03
7	WB4	20482	21132	22091	22868	24065	24696	25727	26433	27761	28604	30172	30819	31800	32469	33108	33738	34753	35360	36890	37550	17115	04:45:15
8	EB4	46640	47324	44649	45661	37693	43468	35829	36631	31849	34501	29219	30172	27521	28253	26202	26909	24545	25155	22421	23018	24950	06:55:50
9	WB5	27385	28019	29001	29896	31104	31747	32767	35829	37166	38678	40369	53030	54044	54747	55439	56092	57165	57770	59293	59933	32596	09:03:16
10	EB5	48714	49327	46972	47699	42436	45825	40544	41435	38678	39280	36323	37067	34621	35323	33398	34008	31670	32329	29492	30072	19883	05:31:23
11	WB6	35341	35989	36985	44649	45825	46653	47731	48549	49867	53040	54700	55333	56306	56976	57567	58193	59230	60861	62387	63080	27786	07:43:06
12	EB6	74382	75006	72588	73402	70830	71441	67792	69818	65769	66416	53030	64195	41820	52014	40556	41181	38850	39521	36611	37316	38443	10:40:43
13	WB7	42111	44659	45661	46972	48174	48822	49834	50604	52001	61513	63183	67413	68363	69297	69884	70503	71577	72320	73816	74426	32362	08:59:22
14	EB7	76396	77102	74750	75359	72906	73599	71171	71812	68094	69828	65202	66426	49420	64205	48164	48787	46450	47068	44232	44874	32917	09:08:37
15	WB8	49975	50809	51791	52485	53650	54414	55464	56081	57392	68094	69661	70854	71854	72687	73278	74006	75089	76121	77664	78281	28354	07:52:34
16	EB8	83023	83697	81360	82003	79611	80222	77844	78578	71400	72740	67413	69838	58193	66436	55932	57567	54105	54822	51945	52603	31800	08:50:00
17	WB9	58059	58681	59752	60392	61551	62183	63237	68104	69470	71400	73043	74060	75066	75882	76526	77146	78229	78912	80349	81026	23015	06:23:35
18	EB9	83488	84113	81735	82464	79913	80555	78091	78897	74716	76741	70854	73043	64837	69848	62580	64215	60861	61498	58604	59293	25557	07:05:57
19	WB10	64678	65342	66358	66960	68129	68751	69818	71410	72740	74716	76355	76967	77940	78636	79243	79855	80986	81638	83094	83775	19144	05:19:04
20	EB10	88473	89102	83680	87468	81722	82446	80019	80686	77981	78685	74060	76355	71558	73053	70199	70907	68497	69114	66341	67052	22808	06:20:08
																					Average train travel time	24218	06:43:39

In the first position (p1) the time is just after 39600 seconds, all the three stations are empty, the WB5 is travelling between the S5 - S6, and there is a failure event between the S6 - S7. In p2, the WB5 is at the S6 and the failure is still going on. In p3, the EB6 is at the S7, and in p4 the other part of the S7 is occupied by the EB7 since the trains are waiting because of the failure is still going on. In p5, the WB6 occupies the S5, and in p6 the other part of the S5 is occupied by the WB7. Although there is an empty capacity at the S6, both the WB6 and the WB7 stop at the S5 since a movement from the S5 to the S6 will cause a blockage. The failure is still going on, and the five trains are waiting for the track repair. In p7, it is seen that the failure track has been repaired and has opened for the candidate trains. Although the first train in the queue is the WB5, its move will cause a blockage. Thus, the EB6 moves and uses the repaired track part. All the other positions related to Figure 4.22 are depicted in Figures 4.23(a) - 4.23(f), and in the followings we explain what happens in these positions.

- p13; the WB5 has just left and there are four trains at this part (the track part between the S5 - S7) of the system.
- p16; the WB6 has just left and there are three trains at this part.
- p17; a new train, the WB8, has just entered from S5 and there are four trains at this part.
- p18; a new train, the EB8, has just entered from S7 and there are five trains at this part.
- p23; a new train, the EB9, has just entered from S7 and there are six trains at this part.
- p26; the EB6 has just left from S5 and there are five trains at this part.
- p34; the WB7 has just left from S7 and there are four trains at this part.
- p35; a new train, the WB9, has just entered from S5 and there are five trains at this part.
- p37; the EB7 that is the last one entered this track part before its repair has just left from S5 and there are four trains at this part.
- p39; the simulation time is 70200 seconds and there are four trains that have entered this part after the repair.

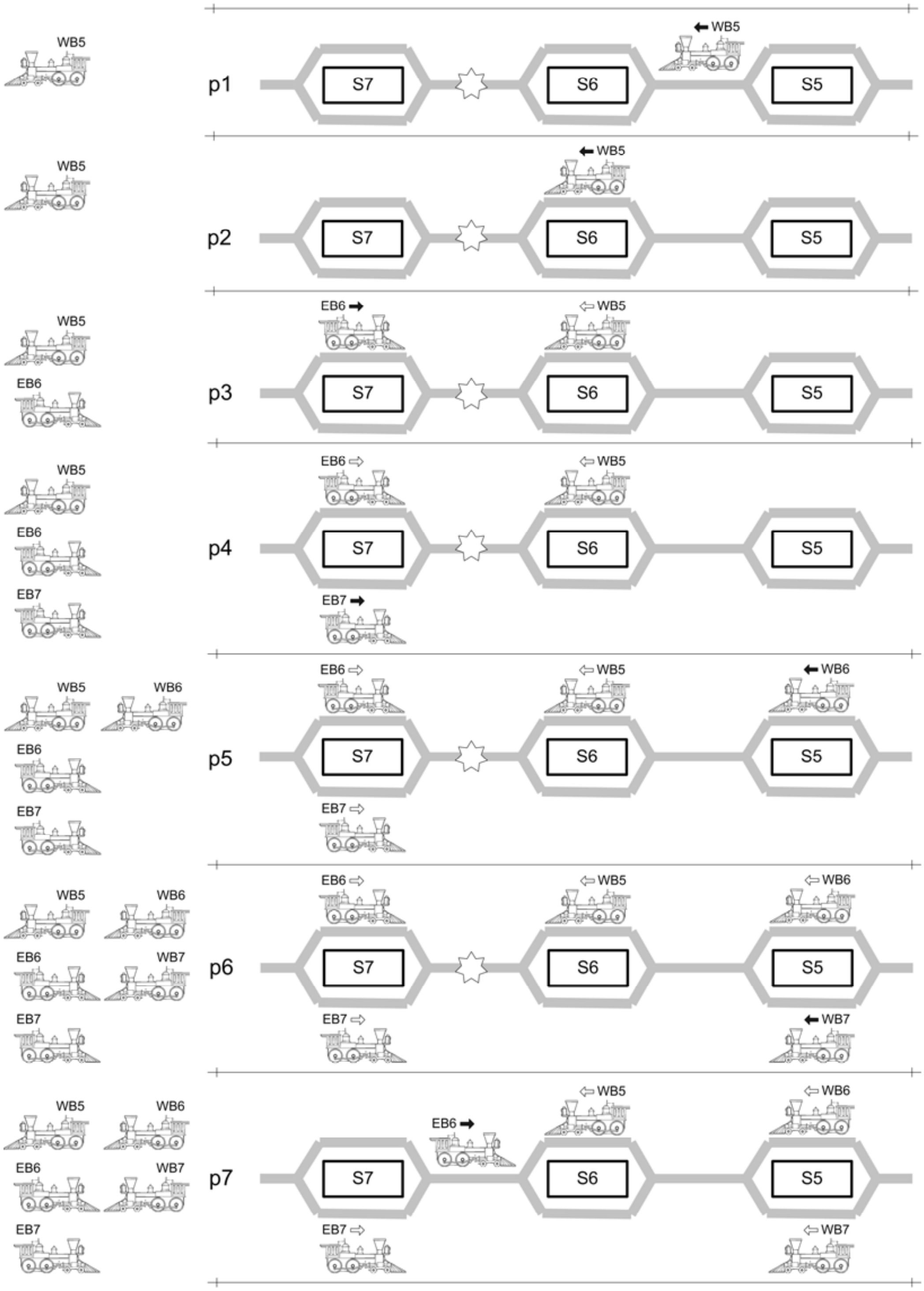


Figure 4.23(a) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

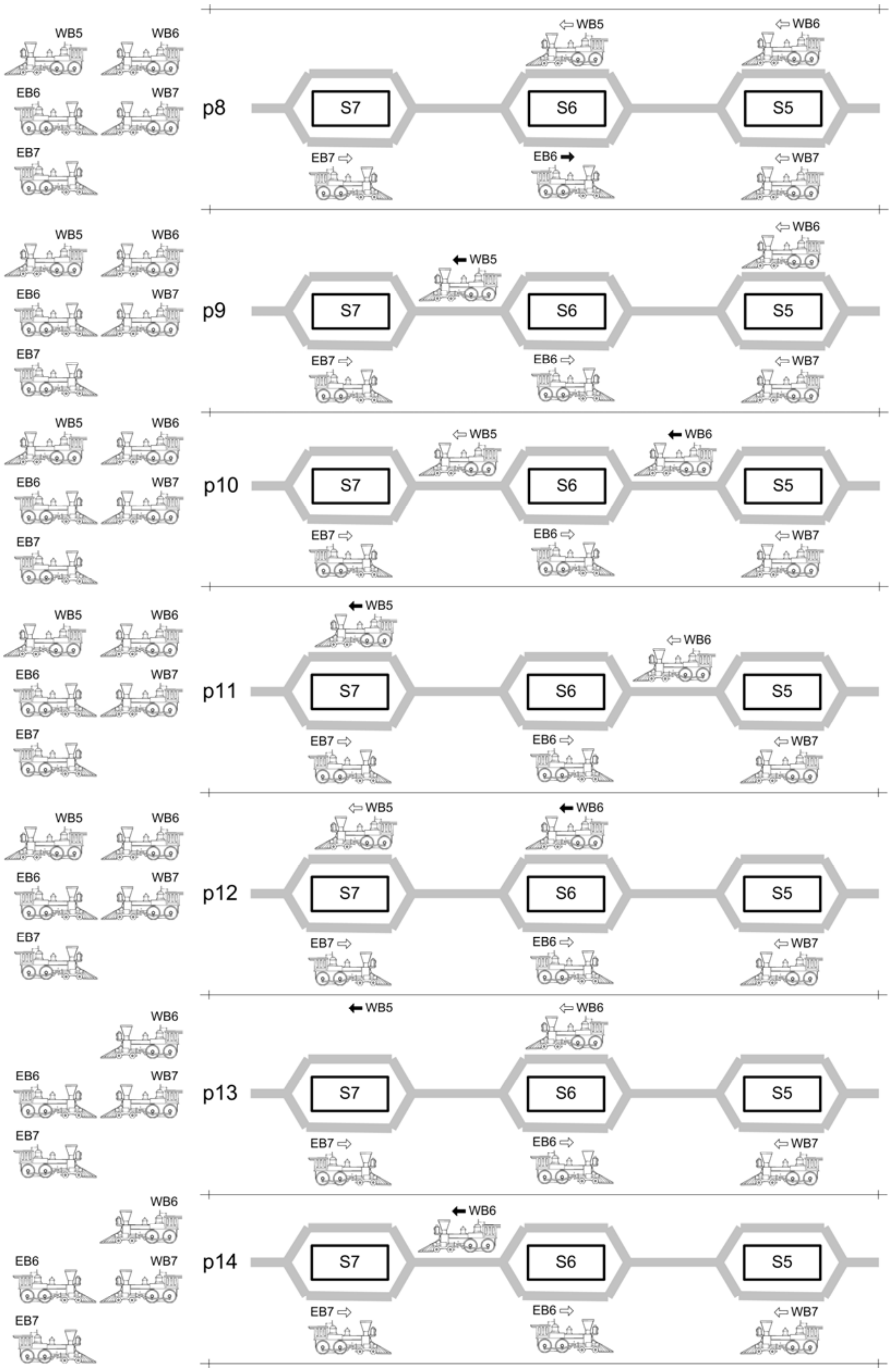


Figure 4.23(b) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

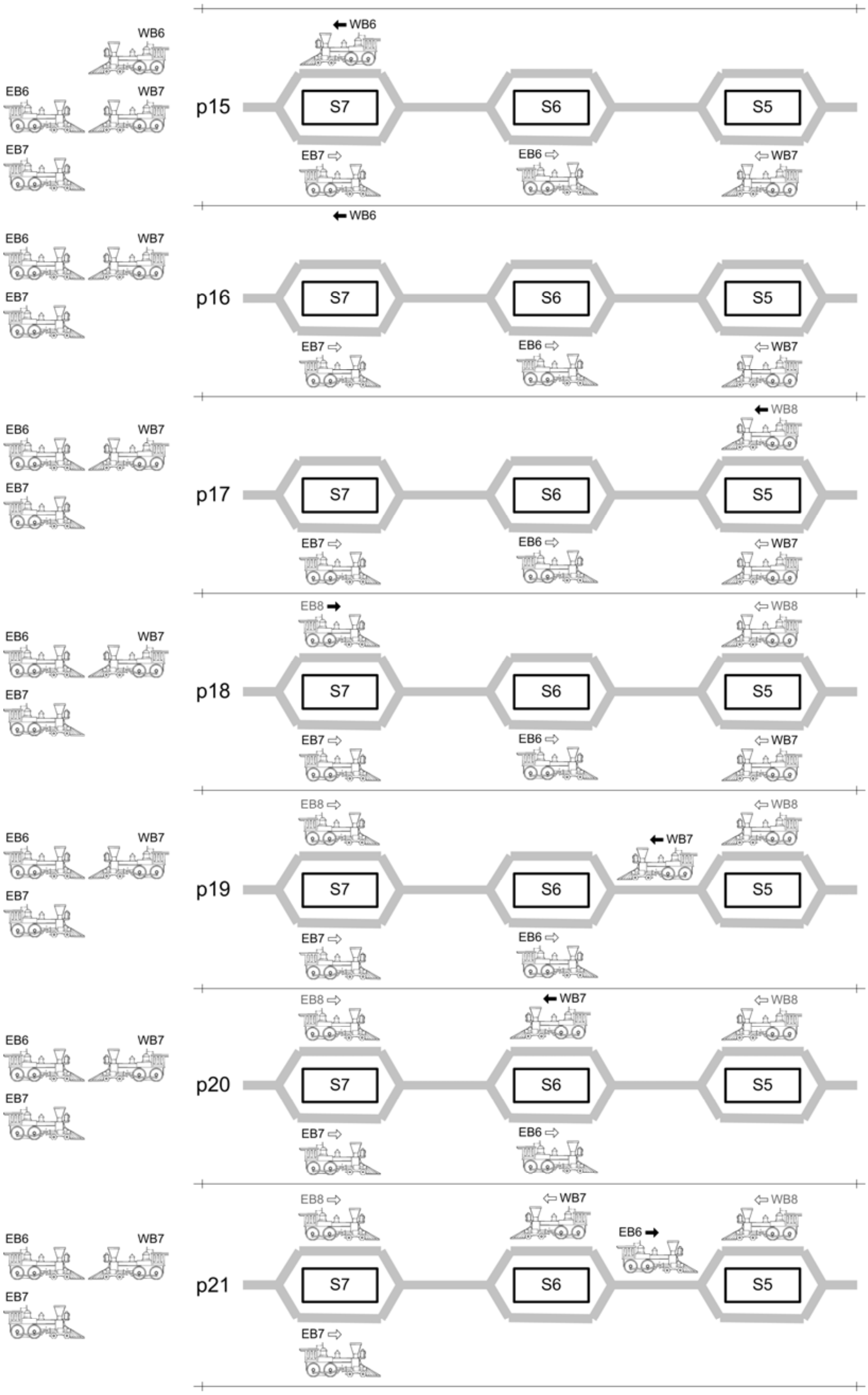


Figure 4.23(c) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

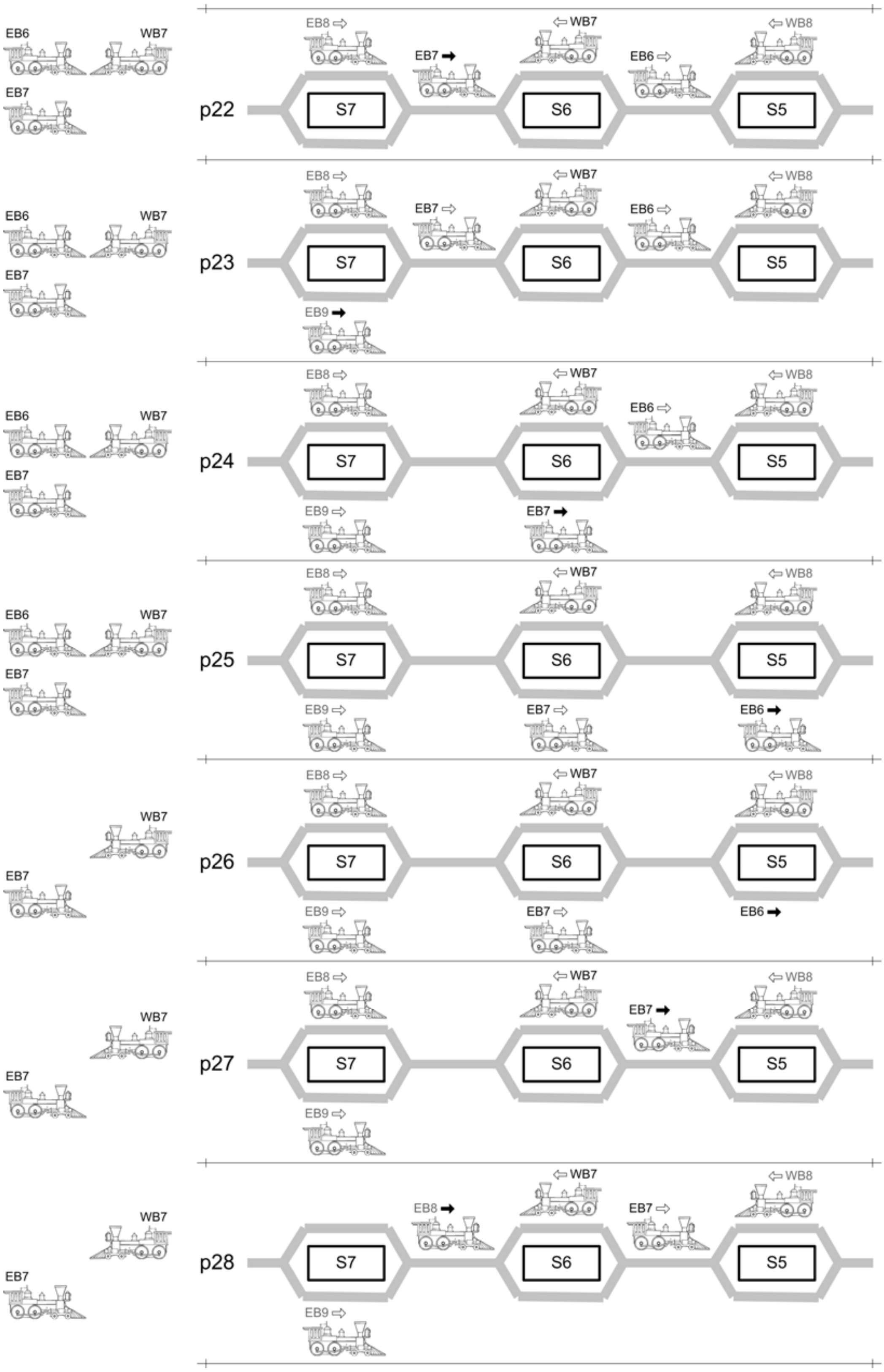


Figure 4.23(d) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

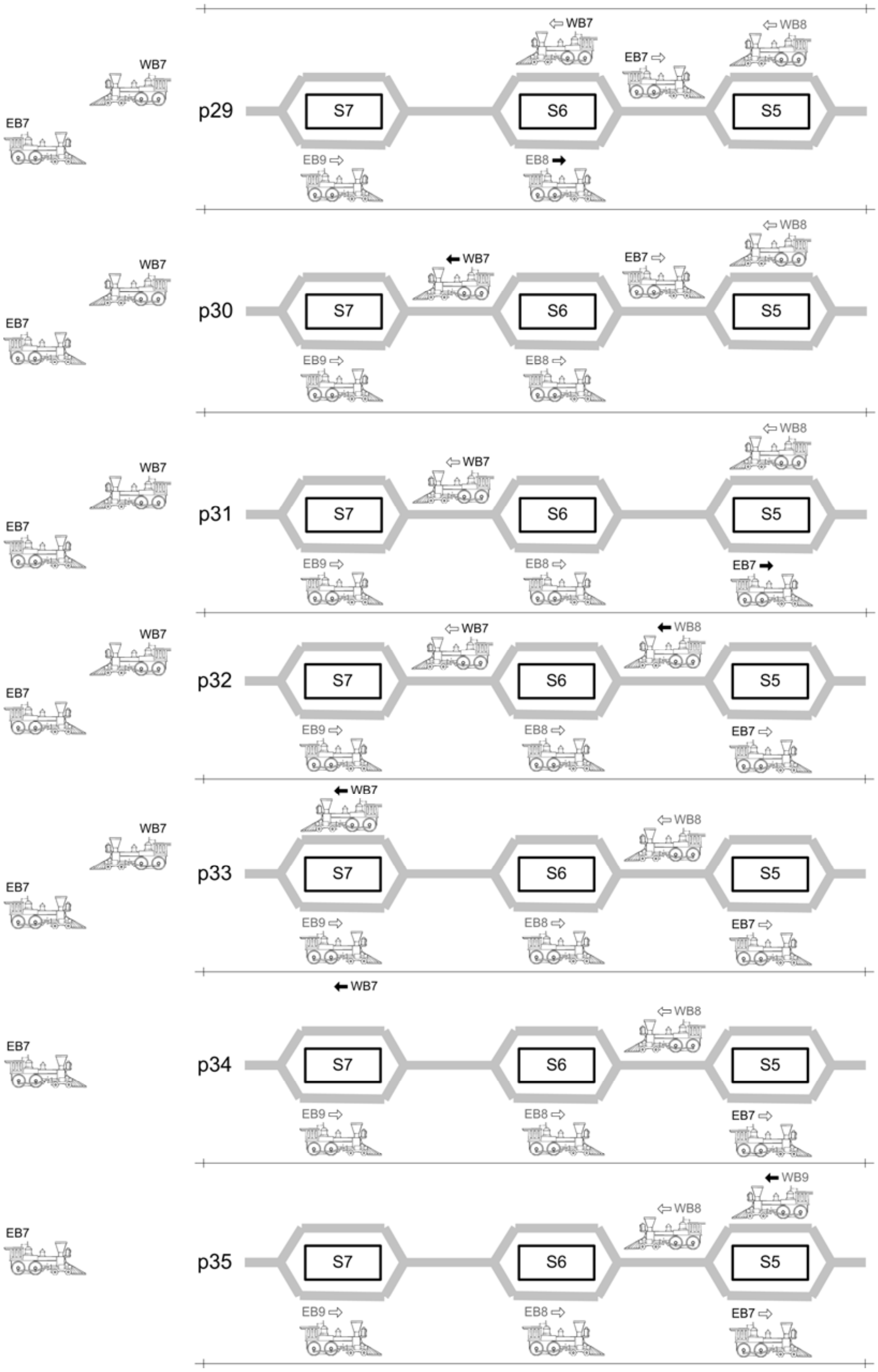


Figure 4.23(e) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

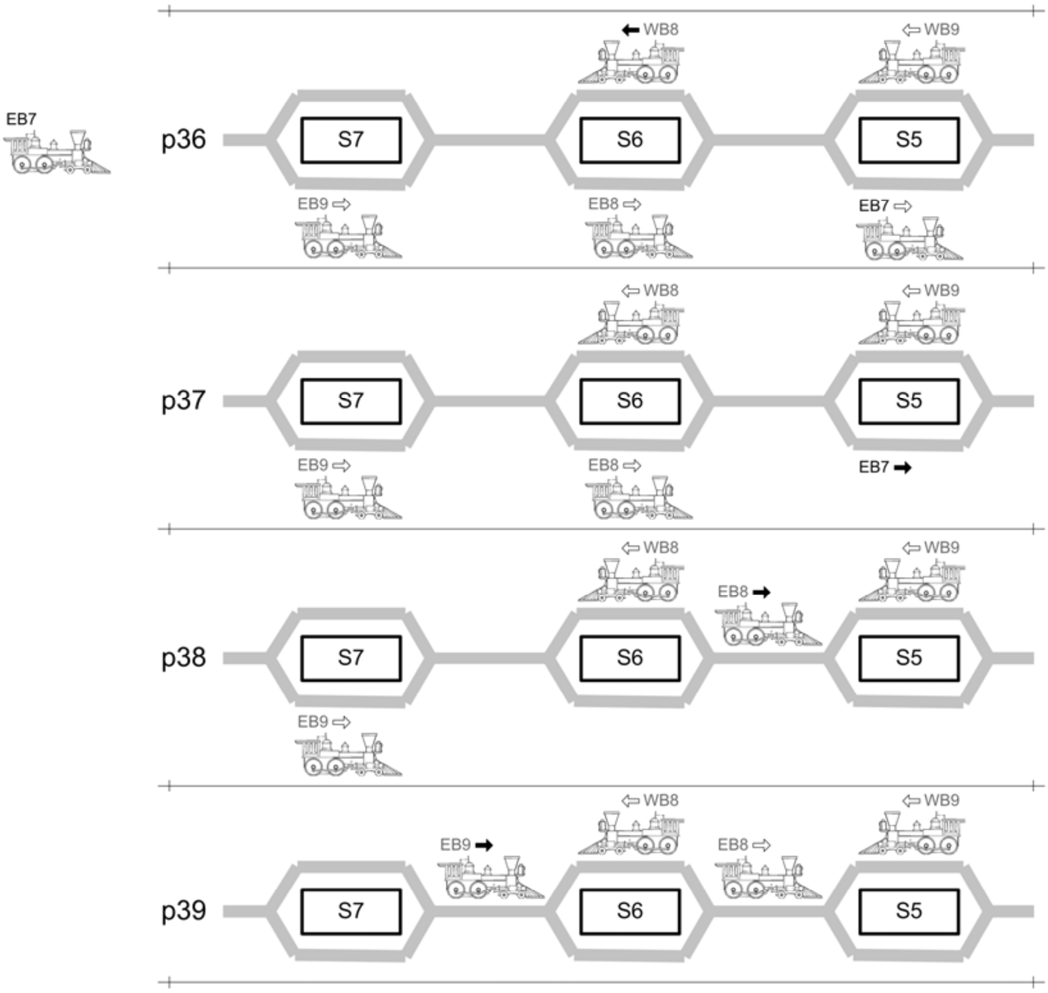


Figure 4.23(f) Train positions on the part dS43-dS71 from 39600 to 70200 seconds

CHAPTER FIVE
SIMULATION INTEGRATED GENETIC AND HYBRID GENETIC ALGORITHMS FOR TRAIN SCHEDULING PROBLEM

In this chapter, a simulation integrated GA and three local search (LS) embedded hybrid GAs for the *TrnSchPrb* are given. The objective is to obtain a feasible train timetable with optimized average train travel time. This chapter includes two subsections. In the first subsection, a simulation integrated GA (*SimGA*) and three hybrid GAs integrated with simulation are developed. In the second subsection, these algorithms are employed to solve the hypothetic *TrnSchPrb* presented in chapter four and the results are given in detail.

5.1 The SimGA for the Hypothetic TrnSchPrb

Here, encoding of a chromosome and parameters of the GA are denoted.

5.1.1 Representation

In our GA, a chromosome (solution) in the initial population is composed of nine genes each of which is related with a decision point and indicates a dispatching rule. As it is shown in Figures 4.2(a)-4.2(c), since there are nine main track parts between the real stations, a chromosome in the proposed GA has nine genes. The first gene in the chromosome is the dispatching rule that is used for the candidate trains waiting in the queue of the track between the S1 - S2, the second gene is the dispatching rule that is used for the candidate trains waiting in the queue of the track between the S2 - S3 and etc. A gene can take values in a range of (1, 6), and each value in this range 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.

A chromosome structure is exhibited in Figure 5.1, in which the first gene value 1 means the dispatching rule used for the candidate trains waiting in the queue of the track between the S1 - S2 is FCFS rule, the fourth gene value 2 means the dispatching rule used for the candidate trains waiting in the queue of the track between the S4 - S5 is LCFS rule.

decision point	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}
track between	S1-S2	S2-S3	S3-S4	S4-S5	S5-S6	S6-S7	S7-S8	S8-S9	S9-S10
a chromosome	1	3	4	2	5	1	6	2	4

Figure 5.1 Representation of a chromosome

For this chromosome representation, there are 6^9 = 10,077,696 possible feasible solutions for the hypothetic *TrnSchPrb*. Since the GAs 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 replications for each possible feasible solution, totally 201,553,920 replications are needed to obtain fitness values of all the possible feasible solutions. One replication lasts 0.26 seconds, thus we need 52,404,019 seconds (more than 606 days) for testing all the possible feasible solutions. That gives us an evidence to say the problem is NP Hard with this representation. Fortunately the developed *SimGA* has an ability to solve the problem in an acceptable time.

The chromosome representation used in the *SimGA* is flexible for both the changes in the railway infrastructure and in the total number of dispatching rules. While the changes in the railway infrastructure can influence on the length of the chromosome, and the changes in the total number of dispatching rules can influence on the value which a gene takes. Therefore, this chromosome representation can be used for the several types of *TrnSchPrb*s. Because it is not matter which railway infrastructures they 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 the system. With the encoding presented here any changes in the number of trains or number of meetings that increase the complexity have no damaging influence on the chromosome those strengths the developed chromosome structure to cope with more complex systems. Also this chromosome structure generates feasible chromosomes all time and provides the simulation model to go on generating feasible train timetables. So the chromosome structure tightens the search space by preventing us to deal with infeasible solutions that will, no doubt, take a huge calculation time.

5.1.2 Initial Population and Evaluation

The chromosomes in 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 tracks are rearranged according to the gene values of the chromosome. Then, the simulation model is run for 20 replications and the average train travel time value is noted to be the fitness value of the chromosome. The procedure for generating the initial population and evaluation is given in Table 5.1.

5.1.3 Parent Selection and Crossover

The procedure for parent selection and crossover is exhibited in Table 5.2. 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 20 replications, and the average train travel time value is hold to be the fitness value of the child.

Table 5.1 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
    Chr(k) = [- , - , - , - , - , - , - , - , -]
      From m =1 to m = 9
        $\boldsymbol{i_m}$; the gene related with m^{th} decision point
        i_m = takes a random value from the set S = {1, 2, 3, 4, 5, 6}
      Next m
    Chr(k) = [i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9]
    Rearrange the simulation model
      From m =1 to m = 9
        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 = 20
        Catch average train travel time of 20 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

Table 5.2 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 n = 1 to n = 2
          cChr(*k,k+1,n*); n^{th} child from the parents rChr(k) and rChr(k+1)
          cChr(k,k+1,n) = [i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9]
          Rearrange the simulation model (Sub steps are given in Table 5.1)
          Run the rearranged simulation model (Sub steps are given in Table 5.1)
          Calculate the fitness value
        Next n
    k = k +1
    Next k

5.1.4 Mutation

The procedure for mutation and replacement strategy is denoted in Table 5.3. 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 taken from the set S = {1, 2, 3, 4, 5, 6} except the current value of the gene. Then, the simulation model is rearranged, the model is run for 20 replications, and the average train travel time value is noted to be the fitness of the mutated child. Mutation goes on according to the predetermined mutation rate.

Table 5.3 Procedure for mutation and replacement strategy

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, i_3, i_4, i_5, i_6, i_7, i_8, i_9]
            Select a gene of the child randomly
            $\boldsymbol{i_j}$; the selected gene related with j^{th} decision point, j = 1, …, 9
                If i_j = 1 change the i_j with a random value from the set S = {2, 3, 4, 5, 6}
                If i_j = 2 change the i_j with a random value from the set S = {1, 3, 4, 5, 6}
                If i_j = 3 change the i_j with a random value from the set S = {1, 2, 4, 5, 6}
                If i_j = 4 change the i_j with a random value from the set S = {1, 2, 3, 5, 6}
                If i_j = 5 change the i_j with a random value from the set S = {1, 2, 3, 4, 6}
                If i_j = 6 change the i_j with a random value from the set S = {1, 2, 3, 4, 5}
            mcChr(*k*); the mutated child
            mcChr(k) = [i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9]
            Rearrange the simulation model (Sub steps are given in Table 5.1)
            Run the rearranged simulation model (Sub steps are given in Table 5.1)
            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

5.1.5 Termination Criteria and Replacement Strategy

The procedure for termination and replacement strategy is denoted in Table 5.4. By using the termination criteria procedure, if it is desired, the *SimGA* can be stopped at the generation that has successively result the same best fitness value in a predetermined number of previous generations. The replacement is made according to 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 5.4 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 hypothetic *TrnSchPrb* is exhibited in Table 5.5.

Table 5.5 Framework of the *SimGA*

```
Begin SimGA

Generate INITIAL POPULATION (Sub steps are given in Table 5.1)
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 (Sub steps are given in Table 5.2)
        MUTATION (Sub steps are given in Table 5.3)
        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 (Sub steps are given in Table 5.4)
        Form the population of the next generation
            Form by using the ranked population adhered to the PS
    Next GN

Stop SimGA
```

5.2 Hybridization of the SimGA with Local Searches

Although the *SimGA* itself is a hybrid of the simulation model with GA, we also hybridized the GA with problem specific local searches (LS) in order to improve its searching capability and then we integrated each hybrid GA with simulation. Three LS algorithm are used for hybridization,

- LS on the best,
- LS on the first and the second best, and
- LS on the best and the worst.

In the rest of the study, three hybrid GA are denoted by *GAb*, *GAfs,* and *GAbw*.

5.2.1 Simulation integrated GAb (SimGAb)

While implementing the GA, after the best chromosome in the new generation is found, some problem specific neighbours of the best chromosome are searched via the *SimGAb* given in Table A.6 in Appendix.

Verbally to obtain a problem specific neighbour, every time only one gene of the best chromosome is changed using the *opposite pair* rule. For instance, if the dispatching rule on use is FCFS, it is altered to LCFS, or if it is ShrCTT it is altered to LngCTT, or if it is ShrRTP it is altered to LngRTP. Each time only one gene is changed, and thus for each generation nine neighbours are obtained by the LS on the best algorithm. The simulation model is rearranged and run for 20 replications for the each neighbour, and the average train travel time value is noted to be the fitness value of the neighbour.

5.2.2 Simulation integrated GAfs (SimGAfs)

While implementing the GA, after the first and the second best chromosomes in the new generation are found, some problem specific neighbours of the chromosomes are searched via the LS on the first and the second best algorithm depicted in Table A.7 in Appendix.

5.2.3 Simulation integrated GAbw (SimGAbw)

While implementing the GA, after the best and the worst chromosomes in the new generation are found, some problem specific neighbours of the chromosomes are searched via the LS on the best and the worst algorithm exhibited in Table A.8 in Appendix.

5.3 Application of Simulation Integrated GA and Hybrid GAs on the Hypothetic TrnSchPrb and Discussion on the Results

At first, we run the stochastic simulation model for the different dispatching rules, but the same rules for all the decision points. For instance a chromosome that is composed of the nine same gene values, for instance 1, means that all dispatching rules used for the candidate trains waiting in the queues of the tracks are FCFS. On the other hand, if all the gene values are 5, all dispatching rules used are ShrRTP. Since there are six different dispatching rules, there are six chromosomes that are composed of the nine same gene values, exhibited in Table 5.6.

After the simulation model was rearranged with FCFS rule for all the decision points, the model is run for 20 replications. At the end of the simulation run average train travel time value is found as 21408 seconds. The other average train travel time values for other chromosomes that have the same dispatching rules for all decision points are given in Table 5.6. As can be seen, the minimum average travel time value (i.e. 21230 seconds) is obtained by the LCFS rule.

Table 5.6 Average train travel time (in seconds)

Dispatching rule	FCFS	LCFS	ShrCTT	LngCTT	ShrRTP	LngRTP
Chromosome	111111111	222222222	333333333	444444444	555555555	666666666
Average train travel time	21408	21230	21465	21346	21346	21465

For different values of the GA parameters, the population size (PS), the crossover rate (CR) and the mutation rate (MR), the presented algorithms are run, and the observed best average travel time values are recorded. The utilized values are; 10, 20 and 30 for PS, 40%, 60% and 80% for CR, and 10%, 20% and 30% for MR. So there are 3.3.3 = 27 different combinations of the parameters. These combinations will be represented by PS/CR/MR for short. The presented algorithms are run for each PS/CR/MR combination, and all the results are displayed in Table A.9 in Appendix.

As displayed in Table A.9, the first conducted algorithm is the *SimGA*. The all PS/CR/MR combinations in the GA were applied. The best average travel time value is 20742.93 seconds. The earliest generation number, in which the best 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 10/0.6/0.1 combination. On the other hand the earliest time to reach the best is 6.59 minutes, and the latest time is 46.28 minutes.

The second algorithm is the *SimGAb*. In all PS/CR/MR combinations the best value is reached. The earliest generation number is three, the related combinations are 30/0.4/0.3 and 30/0.6/0.2 combinations, and the latest generation number is 56 and the combination is 10/0.4/0.1. On the other hand the earliest time to reach the best is 10.40 minutes and the latest time is 68.81 minutes.

The third algorithm is the *SimGAfs*. The best value is reached in all PS/CR/MR combinations. The earliest generation number is two and the combination is 10/0.4/0.1. We note that the *SimGAfs* has the *minimum generation number* (i.e. two) among the presented algorithms. The latest generation number is 34 and the combination is 10/0.6/0.1. On the other hand the earliest time to reach the best is 4.85 minutes (i.e. the *best time* among the presented algorithms). The latest time is 88.40 minutes.

The fourth algorithm is the *SimGAbw*. Except the combination of 10/0.8/0.1, all the other PS/CR/MR combinations have the best value. The earliest generation number is three and appeared in combinations 10/0.4/0.1, 20/0.8/0.2 and 30/0.4/0.3. The earliest time is to reach the best is 6.85 minutes. Although the *SimGAbw* is run for 100 generations for the combination of 10/0.8/0.1, that lasts 234.87 minutes, the best fitness value can not be approached, but it gets too close to the best found so far and only in this combination the best value can not be arrived. The reached average train travel time is 20753.02 seconds that is only 10.09 seconds (0.049%) more than the best (20742.93 seconds). In this combination the *SimGAbw* trapped in a local optimum in the second generation, and can not get out.

We see that only one combination in 108 combinations get trapped in a local optimum. Although some of the other 107 combinations get trapped in a local optimum, they were able to get out.

In the parts given below, in order to see the variation in fitness values, one of the three parameters examined above is allowed to take different values while the other two are held constant. In Figures A3-A11, given in Appendix, *A*1 represents the *SimGA*, *A*2 represents the *SimGAb*, *A*3 represents the *SimGAfs* and *A*4 represents the *SimGAbw*.

At first we look the results of 10/0.4/MR combinations, while the parameter MR takes the values 10%, 20% or 30%, respectively, the PS and CR are set to 10 and 40%, respectively. The earliest generation number belongs to the *SimGAfs* and is equal to two, in the combination of 10/0.4/0.1. It is the *best generation number*, and to reach the generation, only lasts 4.85 minutes (the best time). The latest generation number, 56, appears in *SimGAb* in the combination of 10/0.4/0.1. It takes 68.81 minutes to reach. It is the worst generation number and time among the combinations of 10/0.4/MR.

The best fitness values of the generations for combination of 10/0.4/0.1 are exhibited in Figure A.3 in Appendix. It is seen that;

- The *SimGA* reached the best in 25th generation in 6 steps in; 1st, 5th, 6th, 8th, 20th and 25th generations.

- The *SimGAb* reached the best in 56^{th} generation in 3 steps in; 2^{nd}, 3^{rd} and 56^{th} generations.
- The *SimGAfs* reached the best in 2^{nd} generation, which is the *best generation number*, in 2 steps in; 1^{st} and 2^{nd} generations.
- The *SimGAbw* reached the best in 3^{rd} generation in 3 steps in; 1^{st}, 2^{nd} and 3^{rd} generations.
- Although all algorithms reached the best in an acceptable time the *SimGAfs* and the *SimGAbw* arrived too fast.

The best fitness values of the generations for 10/0.4/0.2 combination are denoted in Figure A.4 in Appendix. It is seen that;

- The *SimGA* reached the best in 16^{th} generation in 5 steps in; 1^{st}, 6^{th}, 7^{th}, 10^{th} and 16^{th} generations.
- The *SimGAb* and the *SimGAfs* reached the best in the same generation; 29^{th} generation in the same 4 steps in; 1^{st}, 2^{nd}, 3^{rd} and 29^{th} generations.
- The *SimGAbw* reached the best in 20^{th} generation in 5 steps in; 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 20^{th} generations.
- All algorithms reached the best in an acceptable time, the fastest is The *SimGA*.

The best fitness values of the generations for 10/0.4/0.3 combination are depicted in Figure A.5 in Appendix. It is seen that;

- The *SimGA* reached the best at 13^{th} generation in 6 steps in; 1^{st}, 4^{th}, 6^{th}, 7^{th}, 10^{th} and 13^{th} generations.
- The *SimGAb*, the *SimGAfs* and the *SimGAbw* reached the best in the same generation, 28^{th} generation, in the same 4 steps that are in; 1^{st}, 2^{nd}, 3^{rd} and 28^{th} generations.
- The fastest is The *SimGA*, the others show same pattern.

Secondly when we look the results of 20/CR/0.2 combinations, while the parameter MR takes the values 40%, 60% or 80%, respectively, the PS and MR are set to 20 and 20%, respectively. The earliest generation number belongs to the *SimGAbw* and is equal to three, in the combination of 20/0.8/0.2. It takes 11.61

minutes to reach the best is also the earliest time in this set of combinations. The latest generation number, 19, appears in *SimGAb* in the combination of 20/0.4/0.2. It takes 36.31 minutes to reach the best. It is the worst generation number and time among the combinations of 20/CR/0.2.

The best fitness values of the generations for 20/0.4/0.2 combination are exhibited in Figure A.6 in Appendix. It is seen that;

- The *SimGA* reached the best in 12th generation in 5 steps in; 1st, 2nd, 4th, 5th and 12th generations. The fitness value reached in 5th generation, 20743.05 seconds, is a local optimum and it is too close to the reached optimum, so from now on we will call this local as the *nearest local*. It is only 0.12 seconds more that the best that is 20742.93 seconds.
- The *SimGAb* reached the best in 19th generation in 3 steps in; 1st, 2nd and 19th generations.
- The *SimGAfs* reached the best in 6th generation in 3 steps in; 1st, 2nd and 6th generations, and it is seen that it reached to the nearest local in an early (2nd) generation.
- The *SimGAbw* reached the best in 6th generation in 4 steps in; 1st, 2nd, 3rd and 6th generations, and it reached to the nearest local in 3rd generation.
- The *SimGAfs* and the *SimGAbw* reached the best fast. Except the *SimGAb,* all algorithms visited the nearest local in early generations but they had the ability to escape.

The best fitness values of the generations for 20/0.6/0.2 combination are denoted in Figure A.7 in Appendix. It is seen that;

- The *SimGA* reached the best in 15th generation in 5 steps in; 1st, 2nd, 5th, 8th and 15th generations.
- The *SimGAb* and the *SimGAbw* reached the best in the same 7th generation in the same 4 steps that are in; 1st, 2nd, 3rd and 7th generations, and both reached to the nearest local in 3rd generation.

- The *SimGAfs* has nearly the same pattern with the *SimGAb* and the *SimGAbw*, reached the best in 6^{th} generation in 4 steps in; 1^{st}, 2^{nd}, 3^{rd} and 6^{th} generations, and with the nearest local reached in 3^{rd} generation.
- Except the *SimGA*, the others reached the best in early generations and also visited the nearest local, and achieved to escape.

The best fitness values of the generations for 20/0.8/0.2 combination are depicted in Figure A.8 in Appendix. It is seen that;

- The *SimGA* reached the best in 17^{th} generation in 8 steps which are in; 1^{st}, 2^{nd}, 5^{th}, 6^{th}, 8^{th}, 9^{th}, 12^{th} and 17^{th} generations.
- Both the *SimGAb* and the *SimGAfs* reached the best in the same 8^{th} generation in the same 4 steps in; 1^{st}, 2^{nd}, 3^{rd} and 8^{th} generations, in addition they both reached to the nearest local in 3^{rd} generation.
- The *SimGAbw* reached the best too fast in 3^{rd} generation in 3 steps in; 1^{st}, 2^{nd} and 3^{rd} generations.
- Except the *SimGA*, the others reached the best in early generations and also visited the nearest local, and achieved to escape.

Lastly when we look the results of PS/0.8/0.1 combinations, while the parameter PS takes the values 10, 20 or 30, respectively, the CR and MR are set to 80% and 10%, respectively. The earliest generation number belongs to the *SimGA* and is equal to six, in the combination of 30/0.8/0.1. It takes 16.64 minutes to reach the best is also the earliest time in this set of combinations. Although the *SimGAbw* is run for 100 generations, which lasts 234.87 minutes, the best fitness value can not be reached in 10/0.8/0.1 combination, but it closes to the best and is the only combination that can not reach the best. In this combination, in an early (2^{nd}) generation the *SimGAbw* trapped in one of a local (20753.02 seconds) and can not get away from the local although the algorithm is run an additional 98 generations.

The best fitness values of the generations for 10/0.8/0.1 combination are exhibited in Figure A.9 in Appendix. It is seen that;

- The *SimGA* reached the best in 31st generation in 5 steps in; 1st, 2nd, 3rd, 26th and 31st generations.
- The *SimGAb* and the SimGA*fs* both reached the best in the same 14th generation in the same 4 steps that are in; 1st, 2nd, 3rd and 14th generations.
- The *SimGAbw* trapped in a local in 2nd generation and can not get away in additional 98 generations.
- Except the *SimGAbw*, the other algorithms reached the best in acceptable times.

The best fitness values of the generations for 20/0.8/0.1 combination are denoted in Figure A.10 in Appendix. It is seen that;

- The *SimGA* reached the best in 10th generation in 5 steps in; 1st, 2nd, 3rd, 5th and 10th generations.
- The *SimGAb* reached the best in 16th generation in 4 steps in; 1st, 2nd, 3rd and 16th generations, with the nearest local reached in an early (3rd) generation.
- Both the *SimGAfs* and the *SimGAbw* reached the best in the same 7th generation in the same 4 steps that are in; 1st, 2nd, 3rd and 7th generations, and it is seen that the fitness value reached at the 3rd generation is the nearest local.
- All algorithms reached the best in acceptable times. Except the *SimGA*, the others visited the nearest local, and achieved to escape.

The best fitness values of the generations for the 30/0.8/0.1 combination are depicted in Figure A.11 in Appendix. It is seen that;

- The *SimGA* reached the best in 6th generation in 6 steps in; 1st, 2nd, 3rd, 4th, 5th and 6th generations.
- The *SimGAb* reached the best in 10th generation in 4 steps in; 1st, 2nd, 3rd and 10th generations, with the nearest local reached in 3rd generation.
- The *SimGAfs* reached the best in 10th generation in 5 steps in; 1st, 2nd, 3rd, 5th and 10th generations and also reached the nearest local in 5th generation.

- The *SimGAbw* reached the best in a late generation (38th) in 4 steps in; 1st, 2nd, 3rd and 38th generations, but it reached the nearest local that is too close to the best in an early (just 3rd) generation.
- All algorithms reached the best in acceptable times. Except the *SimGA*, the others visited the nearest local, and achieved to escape.

It is seen that all algorithms have an ability to escape from a very near local point and reach the best, but generally the *SimGAb*, the *SimGAfs* and the *SimGAbw* visited the nearest local in early generations.

CHAPTER SIX
CONCLUSIONS

The *TrnSchPrb* is the problem of determining a timetable for a set of trains which satisfies some operational constraints without violating track capacities. In this thesis the aim is to present an approach to solve this problem. To realize this aim, first the relevant studies published in the range of 1966-2009 were reviewed. We divided the studies in the *TrnSchPrb* literature into two main parts; *scheduling/timetabling* and *rescheduling/dispatching*. The objective in the scheduling is to prepare a train timetable that includes arrival and departure times of all trains at the visited stations. The objective in the rescheduling is to reschedule the trains after disturbances. This thesis focused on the train scheduling/timetabling problem.

Simulation gives a chance to researchers to model complex problems that have stochastic nature. Although simulation modelling has been used in a few articles those focused on the scheduling/timetabling, none of them includes a comprehensive framework. In this thesis, we developed a comprehensive feasible timetable generator simulation modelling framework for the train scheduling/timetabling problem. The simulation model was developed to cope with the disturbances, therefore stochastic events were allowed in the simulation model. To cope with disturbances is also the interest of rescheduling/dispatching. Therefore, the simulation framework can also be used for the train rescheduling/dispatching problem if it can be feed by the real time data. The study is located in the train scheduling/timetabling problems that used simulation modelling class, which includes too few articles. The feasible timetable generator simulation modelling framework was developed with the objective of obtaining a feasible train timetable that includes train arrival and departure times at all visited stations and calculated average train travel time for all trains in a railway system.

By using the presented approaches, all the railway transportation systems can be modelled with only problem/infrastructure specific modifications and feasible solutions can be easily attained. In order to avoid a deadlock, a general Blockage

Preventive Algorithm is developed. This algorithm can be embedded in the simulation model and can be easily adapted to problem/infrastructure specific modifications.

Although a few article studied integration of simulation with GAs to solve the *TrnSchPrb*, they did not handle the problem comprehensively. In addition, since GA provides flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem we developed simulation integrated hybrid GAs. To get an optimum average train travel time, first a GA was integrated with the feasible timetable generator simulation model, and second three local search embedded hybrid GAs were also developed and integrated with the feasible timetable generator simulation model. To the best of our knowledge our study is the first one that integrates stochastic simulation model with GA and hybrid GA to deal with the train scheduling/timetabling problem. On the other hand, this is the first time that local search embedded GAs were exhibited and integrated to simulation model for the train scheduling/timetabling problem.

Using the encoding structure presented in the current study, any changes in the number of trains or number of train meetings can be overcame easily and consequently the simulation model generates feasible train timetables. So the chromosome structure tightens the search space by preventing us to deal with infeasible solutions that will, no doubt, take a huge calculation time.

Finally, the feasible timetable generator simulation model, the simulation integrated GA, and the three hybrid GAs were applied on a hypothetic *TrnSchPrb* to compare their performance at average train travel time. The problem is based on an artificial infrastructure inspired by a real railway line system. The line structure in the problem is a single track corridor as analogous to many lines in the literature and also real railway systems. By the application of the presented approaches on the hypothetic *TrnSchPrb* good results were obtained. We conducted several experiments with different population size, cross over rate and mutation rate parameters of the GA, and found that all algorithms have an ability to reach the best solution in a reasonable time depicted in Table 6.1.

Table 6.1 The best generation numbers and the best times

Algorithm	**Generation**		**Time**	
	Best	Combination(s)	Best	Combination(s)
SimGA	6	10/0.8/0.3 30/0.8/0.1	6.59	10/0.8/0.3
SimGAb	3	30/0.4/0.3 30/0.6/0.2	10.40	30/0.4/0.3
SimGAfs	2	10/0.4/0.1	4.85	10/0.4/0.1
SimGAbw	3	10/0.4/0.1 20/0.8/0.2 30/0.4/0.3	6.85	10/0.4/0.1

In Table 6.1, the best generation number and the best time belong to the *SimGAfs*, (i.e. simulation integrated local search on the first and the second best embedded hybrid GA), with parameter combination 10/0.4/0.1. For this hypothetical *TrnSchPrb* the *SimGAfs* with these parameter values is preferred.

Future work directions can be as follows;

- So far we dealt with the problem from the service provider (train operating authority) point of view, but there are also the service users (passengers or freight transporting companies) in the system. The simulation modelling framework can be extended by including the service users.
- The performance criterion of the study is the average train travel time. Different criteria of service users and energy consumptions can be considered. Energy consumption is getting more and more important since the energy resources are limited and minimizing energy consumption is essential to have an inhabitable world in future.
- The single objective problem can be extended to a multi objective problem with additional criteria, and the multi objective evolutionary algorithms can be used in order to solve the problem.
- The infrastructure considered in the hypothetic problem is a single track corridor. It can be extended to have double track parts that can also be

modelled as one way, and the system can contain not only a corridor but also a network.

- If very long multi platforms where more than one train can accommodate are regarded, the problem of sequencing of trains on the multi platforms arises and new decision variables are needed to solve the problem.
- In the thesis, the speed of trains is assumed to be uniformly distributed random variable. The speed which influences the energy consumption can be also modelled as a decision variable and the optimum speeds at different parts of the track can be calculated.

REFERENCES

Abril, M., Salido, M.A., Barber, F., Ingolotti, L., Tormos, P., & Lova, A. (2006). Distributed constraint satisfaction problems to model railway scheduling problems. *Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems*, 54-61.

Adenso-Diaz, B., Gonzalez, M.O., & Gonzalez-Torre, P. (1999). On-line timetable re-scheduling in regional train services. *Transportation Research Part B, 33*, 387-398.

Amit, I., & Goldfarb, D. (1971). The time-table problem for railways. *Developments in Operations Research, 2,* 379-387.

Araya, S., Abe, K., & Fukumori, K. (1983). An optimal rescheduling for online train traffic control in disturbed situations. *22nd IEEE Conference on Decision and Control*, 489-494.

Assad, A.A. (1980). Models for rail transportation. *Transportation Research A, 14A*, 205-220.

Assad, A.A. (1982). A class of train-scheduling problems. *Transportation Science, 16,* 281-310.

Assis, W.O., & Milani, B.E.A. (2004). Generation of optimal schedules for metro lines using model predictive control. *Automatica, 40*, 1397-1404.

Bai-gen, C., & Ju-zhen, W. (1993). A study of the expert system for train dispatching. *IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, 2,* 730-733.

Barber, F., Salido, M.A., Ingolotti, L., Abril, M., Lova, A., & Tormos, P. (2004). An interactive train scheduling tool for solving and plotting running maps. *Lecture Notes in Artificial Intelligence, 3040,* 646-655.

Brännlund, U., Lindberg, P.O., Nou, A., & Nilsson, J.-E. (1998). Railway timetabling using lagrangian relaxation. *Transportation Science, 32* (4), 358-369.

Brucker, P., Heitmann, S., & Knust, S. (2002). Scheduling railway traffic at a construction site. *OR Spectrum, 24*, 19-30.

Bussieck, M.R., Winter, T., & Zimmermann, U.T. (1997). Discrete optimization in public rail transport. *Mathematical Programming, 79*, 415-444.

Cacchiani, V., Caprara, A., & Toth, P. (2008). A column generation approach to train timetabling on a corridor. *4OR: A Quarterly Journal of Operations Research, 6* (2), 125-142.

Cai, X., & Goh, C.J. (1994). A fast heuristic for the train scheduling problem. *Computers and Operations Research, 21*, 499-510.

Cai, X., Goh, C.J., & Mees, A.I. (1998). Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions, 30*, 481-493.

Caimi, G., Burkolter, D., Herrmann, T., Chudak, F., & Laumanns, M. (2007). Design of a new railway scheduling model for dense services. *2nd International Seminar on Railway Operations Modelling and Analysis*.

Caimi, G., Fuchsberger, M., Laumanns, M., & Schüpbach, K. (2007). Periodic railway timetabling with event flexibility. *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*.

Caprara, A., Fischetti, M., Guida, P.L., Monaci, M., Sacco, G., & Toth, P. (2001). Solution of real-word train timetabling problems. *34th Hawaii International Conference on System Sciences*, 1-10.

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research, 50*, 851-861.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Passenger railway optimization. *Handbooks in Operations Research and Management Science, 14,* 129-187.

Caprara, A., Monaci, M., Toth, P., & Guida, P.L. (2006). A lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics, 154,* 738-753.

Carey, M. (1994a). A model and strategy for train pathing with choice of lines, platforms, and routes. *Transportation Research B, 28B*, 333-353.

Carey, M. (1994b). Extending a train pathing model from one-way to two-way track. *Transportation Research B, 28B*, 395-400.

Carey, M., & Carville, S. (2003). Scheduling and platforming trains at busy complex stations. *Transportation Research Part A, 37*, 195-224.

Carey, M., & Crawford, I. (2007). Scheduling trains on a network of busy complex stations. *Transportation Research Part B, 41*, 159-178.

Carey, M., & Lockwood, D. (1995). A model, algorithms and strategy for train pathing. *Journal of Operational Research Society, 46*, 988-1005.

Castillo, E., Gallego, I., Ureña, J.M., & Coronado, J.M. (2009). Timetabling optimization of a single railway track line with sensitivity analysis. *Top, 17* (2), 256-287.

Chang, S.-C., & Chung, Y.-C. (2005). From timetabling to train regulation-a new train operation model. *Information and Software Technology, 47*, 575-585.

Chang, Y.-H., Yeh, C.-H., & Shen, C.-C. (2000). A multiobjective model for passenger train services planning: Application to Taiwan's high-speed rail line. *Transportation Research B, 34*, 91-106.

Chiang, T.-W., & Hau, H.-Y. (1993). Knowledge-based railway scheduling system. *3rd International Conference on Industrial Fuzzy Control and Intelligent Systems*, 42-46.

Chiang, T.-W., & Hau, H.-Y. (1995). Railway scheduling system using repair-based approach. *7th International Conference on Tools with Artificial Intelligence*, 71-78.

Chiang, T.-W., & Hau, H.-Y. (1996). Cycle detection in repair-based railway scheduling system. *IEEE International Conference on Robotics and Automation, 3*, 2517-2522.

Chiang, T.-W., Hau, H.-Y., Chiang, H.-M., Ko, S.-Y., & Hsieh, C.-H. (1998). Knowledge-based system for railway scheduling. *Data and Knowledge Engineering, 27*, 289-312.

Cheng, Y. (1998a). Hybrid simulation for resolving resource conflicts in train traffic rescheduling. *Computers in Industry, 35* (3), 233-246.

Cheng, Y. (1998b). Rule-based train traffic reactive simulation model. *Applied Artificial Intelligence, 12*, 5-27.

Cheng,Y.-H., & Yang, L.-A. (2009). A fuzzy petri nets approach for railway traffic control in case of abnormality: Evidence from Taiwan railway system. *Expert Systems with Applications, 36*, 8040-8048.

Chiu, C.K., Chou, C.M., Lee, J.H.M., Leung, H.F., & Leung, Y.W. (1996). A constraint-based interactive train rescheduling tool. *Lecture Notes in Computer Science, 1118*, 104-118.

Coley, D.A. (2003). *An introduction to genetic algorithms for scientists and Engineers* (3rd ed.). Singapore: World Scientific Press.

Cordeau, J.-F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science, 32* (4), 380-404.

Crainic, T.G. (2003). Long haul freight transportation. *Handbook of Transportation Science*, 451-516.

Cury, J.E., Gomide, F.A.C., & Mendes, M.J. (1979). A methodology for generation of optimal schedules for an underground railway system. *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, 18*, 897-902.

Cury, J.E., Gomide, F.A.C., & Mendes, M.J. (1980). A methodology for generation of optimal schedules for an underground railway system. *IEEE Transactions on Automatic Control, AC-25* (2), 217-222.

D’Ariano, A., Pranzo, M., & Hansen, I.A. (2007). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems, 8*, (2), 208-222.

Dorfman, M.J., & Medanic, J. (2004). Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B, 38*, 81-98.

Eiben, A.E., & Smith, J.E. (2003). *Introduction to evolutionary computing*. Berlin: Springer.

Fay, A. (2000). A fuzzy knowledge-based system for railway traffic control. *Engineering Applications of Artificial Intelligence, 13*, 719-729.

Ferreira, L. (1997). Planning Australian freight rail operations: An overview. *Transportation Research Part A, 31* (4), 335-348.

Ferreira, L., & Higgins, A. (1996). Modeling reliability of train arrival times. *Journal of Transportation Engineering, 122* (6), 414-420.

Fischer, F., Helmberg, C., Janßen, J., & Krostitz, B. (2008). Towards solving very large scale train timetabling problems by lagrangian relaxation. *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*.

Flamini, M., & Pacciarelli, D. (2008). Real time management of a metro rail terminus. *European Journal of Operational Research, 189*, 746-761.

Frank, O. (1966). Two-way traffic on a single line of railway. *Operations Research, 14* (5), 801-811.

Fukumori, K., Sano, H., Hasegawa, T., & Sakai, T. (1987). Fundemental algorithm for train scheduling based on artificial intelligence. *Systems and Computers in Japan, 18,* 52-63.

Geske, U. (2006). Railway scheduling with declarative constraint programming. *Lecture Notes in Artificial Intelligence, 4369*, 117-134.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering desig*. New York: John Wiley & Sons, Inc.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: John Wiley & Sons, Inc.

Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: Multiobjective genetic algorithm approach*. London: Springer.

Ghoseiri, K., & Morshedsolouk, F. (2006). ACS-TS: Train scheduling using ant colony system. *Journal of Applied Mathematics and Decision Sciences*, 1-28.

Ghoseiri, K., Szidarovszky, F., & Asgharpour, M.J. (2004). A multi-objective train scheduling model and solution. *Transportation Research Part B, 38*, 927-952.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Massachusetts: Addison Wesley Longman, Inc.

Goh, C.J., & Mees, A.I. (1991). Optimal control on a graph with application to train scheduling problems. *Mathematical Computer Modelling, 15*, 49-58.

Haupt, R.L., & Haupt, S.E. (2004). *Practical genetic algorithms* (2nd ed.). New Jersey: John Wiley & Sons, Inc.

Hellström, P., Frej, I., Gideon, A., & Sandblad, B. (1997). Algorithms and control systems for computer-aided train dispatching. *World Congress on Railway Research, C*, 613-619.

Higgins, A., Kozan, E., & Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research B, 30*, 147-161.

Higgins, A., Kozan, E., & Ferreira, L. (1997). Heuristic techniques for single line train scheduling. *Journal of Heuristics, 3*, 43-62.

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbour: The University of Michigan Press.

Huisman, D., Kroon, L.G., Lentink, R.M., & Vromans, M.J.C.M. (2005). Operations research in passenger railway transportation. *Statistica Neerlandica, 59* (4), 467-497.

Ingolotti, L., Barber, F., Tormos, P., Lova, A., Salido, M.A., & Abril, M. (2004). An efficient method to schedule new trains on a heavily loaded railway network. *Lecture Notes in Artificial Intelligence, 3315*, 164-173.

Ingolotti, L., Barber, F., Tormos, P., Lova, A., Salido, M.A., & Abril, M. (2006). A scheduling order-based method to solve timetabling problems. *Lecture Notes in Artificial Intelligence*, 4177, 52-61.

Ingolotti, L., Lova, A., Barber, F., Tormos, P., Salido, M.A., & Abril, M. (2006). New heuristics to solve the "CSOP" railway timetabling problem. *Lecture Notes in Artificial Intelligence, 4031*, 400-409.

Ingolotti, L., Tormos, P., Lova, A., Barber, F., Salido, M.A., & Abril, M. (2004). A decision support system (DSS) for the railway scheduling problem. *1st IFIP Conference on Artificial Intelligence Applications and Innovations*, 465-474.

Isaai, M.T. (2005). An intelligent search technique for solving train scheduling problems: simulated annealing & constraint satisfaction. *35th International Conference on Computers and Industrial Engineering*, 2269-2276.

Isaai, M.T., & Cassaigne, N.P. (2001). Predictive and reactive approaches to the train-scheduling problem: A knowledge management perspective. *IEEE Transactions on System, Man, and Cybernetics Part C: Applications and Reviews, 31*, 476-484.

Isaai, M.T., & Singh, M.G. (2000). An object-oriented, constraint-based heuristic for a class of passenger-train scheduling problems. *IEEE Transactions on System, Man, and Cybernetics Part C: Applications and Reviews, 30*, 12-21.

Isaai, M.T., & Singh, M.G. (2001). Hybrid applications of constraint satisfaction and meta-heuristics to railway timetabling: *A comparative study. IEEE Transactions on System, Man, and Cybernetics Part C: Applications and Reviews, 31*, 87-95.

Iyer, R.V., & Ghosh, S. (1991). DARYN, A distributed decision-making algorithm for railway networks: Modeling and simulation. *IEEE International Conference on Systems, Man, and Cybernetics, 1*, 269-274.

Iyer, R.J., & Ghosh, S. (1995). DARYN - A distributed decision-making algorithm for railway networks: Modeling and simulation. *IEEE Transactions on Vehicle Technology, 44*, 180-191.

Jovanovic, D., & Harker, P.T. (1989). SCAN: A decision support system for railroad scheduling. *Railroad Conference,* 97-105.

Jovanovic, D., & Harker, P.T. (1990). A decision support system for train dispatching: An optimization-based methodology. *Journal of the Transportation Research Forum 31*, 25-37.

Jovanovic, D., & Harker, P.T. (1991a). Decision support system for train dispatching: An optimization-based methodology. *Transportation Research Records, 1314*, 31-40.

Jovanovic, D., & Harker, P.T. (1991b). Tactical scheduling of rail operations: The SCAN I system. Transportation Science, 25, 46-64.

Kauppi, A., Wikström, J., Sandblad, B., & Andersson, A.W. (2006). Future train traffic control: control by re-planning. *Cognition, Technology and Work, 8* (1), 50-56.

Khan, M.B., Zhang, D., Jun, M.S., & Li, Z.J. (2006). An intelligent search technique to train scheduling problem based on genetic algorithm. *2nd International Conference on Emerging Technologies,* 593-598.

Komaya, K., & Fukuda, T. (1991). A knowledge-based approach for railway scheduling. *7th IEEE Conference on Artificial Intelligence*, 404-411.

Kraay, D.R., & Harker, P.T. (1995). Real-time scheduling of freight railroads. *Transportation Research B, 29B*, 213-229.

Kraay, D., Harker, P.T., & Chen, B. (1991). Optimal pacing of trains in freight railroads: Model formulation and solution. *Operations Research, 39*, 82-99.

Kraft, E.R. (1987). A branch-and-bound procedure for optimal train dispatching. *Journal of the Transportation Research Forum, 28*, 263-267.

Kreuger, P., Carlsson, M., Olsson, J., Sjöland, T., & Aström, E. (1997a). Trip scheduling on single truck networks-The TUFF train scheduler. *In Workshop on Industrial Constraint-Directed Scheduling*, 1-12.

Kreuger, P., Carlsson, M., Olsson, J., Sjöland, T., & Aström, E. (1997b). The tuff train scheduler - Two duration trip scheduling on multi directional track networks. *In Workshop on Tools and Environments for Constraint Logic Programming*, 1-13.

Kroon, L.G., & Peeters, L.W. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science, 37* (2), 198-212.

Kwan, R.S.K., & Mistry, P. (2003). A co-evolutionary algorithm for train timetabling. *IEEE Congress on Evolutionary Computation*, 2142-2148.

Lee, Y., Chen, C.-Y. (2009). A heuristic for the train pathing and timetabling problem. *Transportation Research Part B, 43*, 837-851.

Liebchen, C. (2008). The first optimized railway timetable in practice. *Transportation Science, 42* (4), 420-435.

Liebchen, C., & Möhring, R.H. (2007). The modeling power of the periodic event scheduling problem: Railway timetables - and beyond. *Lecture Notes in Computer Science, 4359*, 3-40.

Lin, H.-C., & Hsu, C.-C. (1994). An interactive train scheduling workbench based on artificial intelligence. *6th International Conference on Tools with Artificial Intelligence*, 42-48.

Lindner, T., & Zimmermann, U.T. (2005). Cost optimal periodic train scheduling. Mathematical Methods of Operations Research, 62, 2281-295.

Liu, S.Q., & Kozan, E. (2009). Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Computers and Operations Research, 36*, 2840-2852.

Lova, A., Tormos, P., Barber, F., Ingolotti, L., Salido, M.A., & Abril, M. (2007). Intelligent train scheduling on a high-loaded railway network. *Lecture Notes in Computer Science, 4359*, 219-232.

Luethi, M., Medeossi, G., & Nash, A. (2009). Structure and simulation evaluation of an integrated real-time rescheduling system for railway networks. *Networks and Spatial Economics, 9*, 103-121.

Luethi, M., Weidmann, U., Laube, F., & Medeossi, G. (2007). Rescheduling and train control: A new framework for railroad traffic control in heavily used networks. *86th Transportation Research Board Annual Meeting*.

Mazzarello, M., & Ottaviani, E. (2007). A traffic management system for real-time traffic optimisation in railways. *Transportation Research Part B, 41*, 246-274.

Medanic, J., & Dorfman, M.J. (2002a). Time efficient greedy strategy for scheduling trains on a line. *15th International Federation of Automatic Control World Congress*.

Medanic, J., & Dorfman, M.J. (2002b). Energy efficient strategies for scheduling trains on a line. *15th International Federation of Automatic Control World Congress*.

Medanic, J., & Dorfman, M. J. (2002c). Efficient scheduling of traffic on a railway line. *Journal of Optimization Theory and Applications, 115*, 587-602.

Mees, A.I. (1991). Railway scheduling by network optimization. *Mathematical Computer Modelling, 15*, 33-42.

Mills, RG.J., Perkins, S.E., & Pudney, P.J. (1991). Dynamic rescheduling of long-haul trains for improved timekeeping and energy conservation. *Asia-Pacific Journal of Operational Research, 8*, 146-165.

Mladenović, S., & Čangalović, M. (2007). Heuristic approach to train rescheduling. *Yugoslav Journal of Operations Research, 17* (1), 9-29.

Nachtigall, K., & Voget, S. (1996). A genetic algorithm approach to periodic railway synchronization. *Computers and Operations Research, 23* (5), 453-463.

Nagarajan, V., & Ranade, A.G. (2008). Exact train pathing. *Journal of Scheduling, 11* (4), 279-297.

Nemhauser, G. L. (1969). Scheduling local and express service. *Transportation Science, 3* (2), 164-175.

Newman, A.M., Nozick, L.K., & Yano, C.A. (2002). Optimization in the rail industry. *Handbook of Applied Optimization*, 704-719.

Odijk, M.A. (1996). A constraint generation algorithm for construction of periodic railway timetables. *Transportation Research B, 30* (6), 455-464.

Oliveira, E., & Smith, B.M. (2000). A job-shop scheduling model for the single-track railway scheduling problem. *Research Report, 2000.21*, University Of Leeds.

Pacciarelli, D., & Pranzo, M. (2001). A tabu search algortihm for the railway scheduling problem. *4th Metaheuristics International Conference*, 159-163.

Parkes, D.C., & Ungar, L.H. (2001). An auction-based method for decentralized train scheduling. *5th International Conference on Autonomous Agents*, 43-50.

Peeters, L., & Kroon, L. (2001). A cycle based optimization model for the cyclic railway timetabling problem. *Computer-aided Scheduling of Public Transport Series: Lecture Notes in Economics and Mathematical Systems, 505*, 275-296.

Petersen, E.R. (1974). Over-the-road transit time for a single track railway. *Transportation Science, 8*, 65-74,.

Petersen, E.R., & Merchant, R. (1981). Scheduling of trains in a linear railway system. *INFOR Journal, 19*, 246-258.

Petersen, E.R., & Taylor, A.J. (1982). A structured model for rail line simulation and optimization. *Transportation Science, 16*, 192-206.

Petersen, E.R., Taylor, A.J. & Martland, C.D. (1986). An introduction to computer-assisted train dispatching. *Journal of Advanced Transportation, 20*, 63-72.

Ping, L., Axin, N., Limin, J., & Fuzhang, W. (2001). Study on intelligent train dispatching. *IEEE Intelligent Transportation Systems Conference*, 949-953.

Rebreyend, P. (2005). DisTrain: A simulation tool for train dispatching. *8th International IEEE Conference on Intelligent Transportation Systems*, 801-806.

Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B, 41* (2), 231-245.

Şahin, İ. (1999), Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research Part B, 33*, 511-534.

Salido, M.A., Abril, M., Barber, F., Ingolotti, L., Tormos, P., & Lova, A. (2004). Topological constraint in periodic train scheduling. *Frontiers in Artificial Intelligence and Applications 117*, 11-20.

Salido, M.A., Abril, M., Barber, F., Ingolotti, L., Tormos, P., & Lova, A. (2007). Domain-dependent distributed models for railway scheduling. *Knowledge-Based Systems, 20*, 186-194.

Salido, M.A., Barber, F., Abril, M., Tormos, P., Lova, A., & Ingolotti, L. (2004). A topological model based on railway capacity to manage periodic train scheduling. *Applications and Innovations in Intelligent Systems XII*, 107-120.

Salim, V., & Cai, X. (1997). A genetic algorithm for railway scheduling with environmental considerations. *Environmental Modelling and Software, 12* (4), 301-309.

Salzborn, F. J. M. (1969). Timetables for a suburban rail transit system. *Transportation Science, 3* (4), 297-316.

Sauder, R.L., & Westerman, W.M. (1983). Computer aided train dispatching: Decision support through optimization. *Interfaces, 13*, 24-37.

Schaefer, H. (1995). Computer-aided train dispatching with expert systems. *International Conference on Electric Railways in a United Europe*, 28-32.

Semet, Y., & Schoenauer, M. (2005). An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. *IEEE Congress on Evolutionary Computation, 3*, 2752-2759.

Semet, Y., & Schoenauer, M. (2006). On the benefits of inoculation, an example in train scheduling. *Genetic and Evolutionary Computation Conference*.

Smith, M., Patel, P.K., Resor, R.R., & Kondapalli, S. (1990). Benefits of the meet/pass planning and energy management subsystems of the advanced railroad electronics system (ARES). *Journal of the Transportation Research Forum, 30*, 301-309.

Su, J.-M., & Huang, J.-Y. (2006). Using ant colony optimization to solve train timetabling problem of mass rapid transit. *Joint Conference on Information Science*.

Szpigel, B. (1973). Optimal train scheduling on a single track railway. *Operational Research'72*, 343-352.

Törnquist, J. (2005). Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. *5th Workshop on Algorithmic Methods and Models for Optimization of Railways*.

Törnquist, J., & Persson, J.A. (2005). Train traffic deviation handling using tabu search and simulated annealing. *38th Hawaii International Conference on System Sciences*.

Törnquist, J., & Persson, J.A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B, 41*, 342-362.

Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., & Salido, M.A. (2008). A genetic algorithm for railway scheduling problems. *Studies in Computational Intelligence, 128*, 255-276.

Tsuruta, S., & Matsumoto, K. (1988). A knowledge-based interactive train scheduling system - aiming at large-scale complex planning expert systems. *International Workshop on Artificial Intelligence for Industrial Applications*, 490-495.

Wikström, J., Kauppi, A., Hellström, P., Andersson, A.W., & Sandblad, B. (2004). Train traffic control by re-planning in real-time. *Computers in Railways IX*.

Wong, O., & Rosser, M. (1978). Improving system performance for a single line railway with passing loops. *New Zealand Operational Research, 6*, 137-155.

Yalçınkaya, Ö., & Bayhan, G.M. (2005). An optimization study for an urban public transport system. *35th International Conference on Computers and Industrial Engineering*, 2101-2106.

Yalçınkaya, Ö., & Bayhan, G.M. (2008). Average travel time optimization for a metro line by simulation and response surface methodology. *2nd International Railway Symposium and Raiway Trade Exhibition*, 1361-1371.

Yalçınkaya, Ö., & Bayhan, G.M. (2009). Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. *European Journal of Operational Research, 196*, 225-233.

Yang, L., Li, K., & Gao, Z. (2009). Train timetable problem on a single-line railway with fuzzy passenger demand. *IEEE Transactions on Fuzzy Systems, 17* (3), 617-629.

Yu, X., & Gen, M. (2010). *Introduction to evolutionary algorithms*. London: Springer.

Zhou, X., & Zhong, M. (2005). Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research, 167* (3), 752-771.

Zhou, X., & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B, 43*, 320-341.

APPENDICES

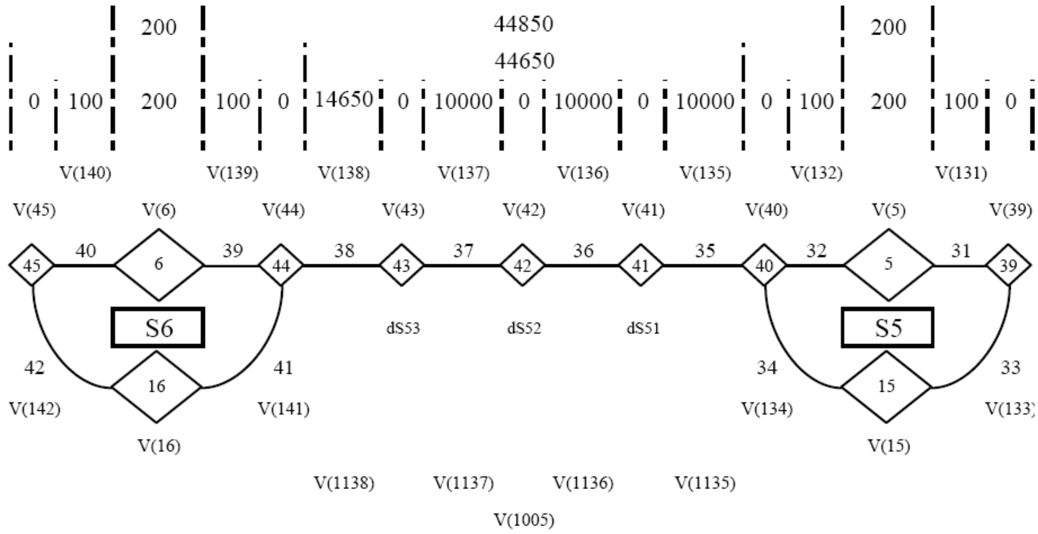


Figure A.1 The line-station diagram of the track between the S5 and the S6

Table A.1 SIMAN View of failure model logic related to track between the S5 and the S6

```
1$            DELAY:         Expo(33283):2$;
2$            ASSIGN:        V1005_fS5=1:3$;
3$            BRANCH,        1:
                         With,0.25:4$;
                         With,0.25:5$;
                         With,0.25:6$;
                         With,0.25:7$;
4$            ASSIGN:        V1135_fLnk35=1:8$;
5$            ASSIGN:        V1136_fLnk36=1:8$;
6$            ASSIGN:        V1137_fLnk37=1:8$;
7$            ASSIGN:        V1138_fLnk38=1:8$;
8$            DELAY:         Expo(4673):9$;
9$            ASSIGN:        V1005_fS5=0:
                             V1135_fLnk35=0:
                             V1136_fLnk36=0:
                             V1137_fLnk37=0:
                             V1138_fLnk38=0:10$;
10$           BRANCH,        1:
                  If,(#ofWestboundTrains+#ofEastboundTrains)==0:11$;
                  Else:1$;
11$           DISPOSE:
```

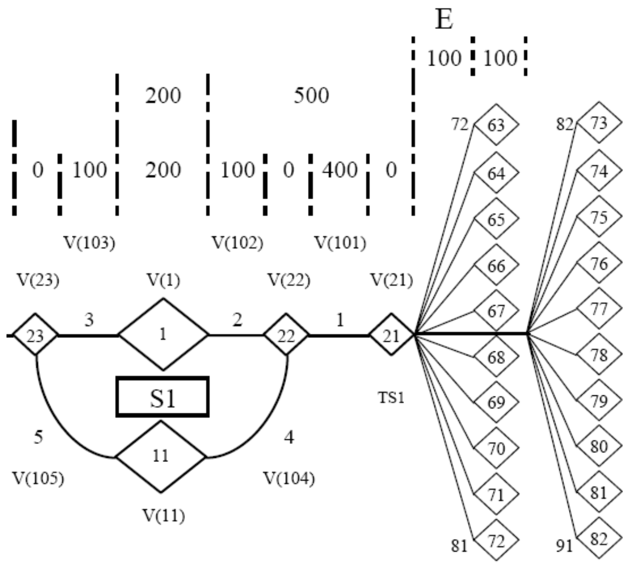


Figure A.2 Line-station diagram of the TS1 and its neighbourhood

Table A.2 SIMAN View of the train movement logic from the park area to the TS1

```
1$            CREATE,        1:2$;
2$            BRANCH,        1:
                             If,(ed(211)==1):3$;
                             If,(ed(212)==1):4$;
                             Else:5$;
3$            ASSIGN:        m=21:
                             FromStation=63:
                             ToStation=21:
                             Destination=1:
                             v(21)=v(21)+1:
                             v(101)=v(101)+1:
                             v(22)=v(22)+1:
                             v(102)=v(102)+1:
                             v(1)=v(1)+1:
                             v(103)=v(103)+1:
                         #ofWestboundTrains=#ofWestboundTrains-1:6$;
4$            ASSIGN:        m=21:
                             FromStation=63:
                             ToStation=21:
                             Destination=11:
                             v(21)=v(21)+1:
                             v(101)=v(101)+1:
                             v(22)=v(22)+1:
                             v(104)=v(104)+1:
                             v(11)=v(11)+1:
                             v(105)=v(105)+1:
                         #ofWestboundTrains=#ofWestboundTrains-1:6$;
5$            SCAN:          (ed(211)==1).or.(ed(212)==1):2$;
6$            DUPLICATE:     7$;
                             1:9$;
7$            REQUEST,       1:TrainFleet(sds,Train#):8$;
8$            TRANSPORT:     TrainFleet,TS1;
```

```
9$            BRANCH,        1:
                             If,#ofWestboundTrains==0:11$;
                             Else:10$;
10$           DELAY:         7200+unif(-900,900):2$;
11$           DISPOSE:

EXPRESSIONS:

211,FromParkViaTS1ToS1p1,
((ed(21101)==1).and.(ed(21102)==1)):

21101,FromParkViaTS1ToS1p1_1,
(v(21)==0).and.(v(101)==0).and.(v(22)==0).and.(v(102)==0).and.(v(1)=
=0).and.(v(103)==0).and.((v(11)<=0).or.(v(2)>=0).or.(v(12)>=0)).and.
((v(11)<=0).or.((v(2)==0).or.(v(12)==0)).or.(v(3)>=0).or.(v(13)>=0))
.and.((v(11)<=0).or.((v(2)==0).or.(v(12)==0).or.(v(3)==0).or.(v(13)=
=0)).or.(v(4)>=0).or.(v(14)>=0)).and.((v(11)<=0).or.((v(2)==0).or.(v
(12)==0).or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0)).or.(
v(5)>=0).or.(v(15)>=0)).and.((v(11)<=0).or.((v(2)==0).or.(v(12)==0).
or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0).o
r.(v(15)==0)).or.(v(6)>=0).or.(v(16)>=0)).and.((v(11)<=0).or.((v(2)=
=0).or.(v(12)==0).or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)=
=0).or.(v(5)==0).or.(v(15)==0).or.(v(6)==0).or.(v(16)==0)).or.(v(7)>
=0).or.(v(17)>=0)).and.((v(11)<=0).or.((v(2)==0).or.(v(12)==0).or.(v
(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0).or.(v(
15)==0).or.(v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0)).or.(v
(8)>=0).or.(v(18)>=0)):

21102,FromParkViaTS1ToS1p1_2,
((v(11)<=0).or.((v(2)==0).or.(v(12)==0).or.(v(3)==0).or.(v(13)==0).o
r.(v(4)==0).or.(v(14)==0).or.(v(5)==0).or.(v(15)==0).or.(v(6)==0).or
.(v(16)==0).or.(v(7)==0).or.(v(17)==0).or.(v(8)==0).or.(v(18)==0)).o
r.(v(9)>=0).or.(v(19)>=0)).and.((v(11)<=0).or.((v(2)==0).or.(v(12)==
0).or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0
).or.(v(15)==0).or.(v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0
).or.(v(8)==0).or.(v(18)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(10)>=
0).or.(v(20)>=0)):

212,FromParkViaTS1ToS1p2,
((ed(21201)==1).and.(ed(21202)==1)):

21201,FromParkViaTS1ToS1p2_1,
(v(21)==0).and.(v(101)==0).and.(v(22)==0).and.(v(104)==0).and.(v(11)
==0).and.(v(105)==0).and.((v(1)<=0).or.(v(2)>=0).or.(v(12)>=0)).and.
((v(1)<=0).or.((v(2)==0).or.(v(12)==0)).or.(v(3)>=0).or.(v(13)>=0)).
and.((v(1)<=0).or.((v(2)==0).or.(v(12)==0).or.(v(3)==0).or.(v(13)==0
)).or.(v(4)>=0).or.(v(14)>=0)).and.((v(1)<=0).or.((v(2)==0).or.(v(12
)==0).or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0)).or.(v(5
)>=0).or.(v(15)>=0)).and.((v(1)<=0).or.((v(2)==0).or.(v(12)==0).or.(
v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0).or.(v
(15)==0)).or.(v(6)>=0).or.(v(16)>=0)).and.((v(1)<=0).or.((v(2)==0).o
r.(v(12)==0).or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).o
r.(v(5)==0).or.(v(15)==0).or.(v(6)==0).or.(v(16)==0)).or.(v(7)>=0).o
r.(v(17)>=0)).and.((v(1)<=0).or.((v(2)==0).or.(v(12)==0).or.(v(3)==0
).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0).or.(v(15)==0
).or.(v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0)).or.(v(8)>=0
).or.(v(18)>=0)):
```

```
21202,FromParkViaTS1ToS1p2_2,
((v(1)<=0).or.((v(2)==0).or.(v(12)==0).or.(v(3)==0).or.(v(13)==0).or
.(v(4)==0).or.(v(14)==0).or.(v(5)==0).or.(v(15)==0).or.(v(6)==0).or.
(v(16)==0).or.(v(7)==0).or.(v(17)==0).or.(v(8)==0).or.(v(18)==0)).or
.(v(9)>=0).or.(v(19)>=0)).and.((v(1)<=0).or.((v(2)==0).or.(v(12)==0)
.or.(v(3)==0).or.(v(13)==0).or.(v(4)==0).or.(v(14)==0).or.(v(5)==0).
or.(v(15)==0).or.(v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0).
or.(v(8)==0).or.(v(18)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(10)>=0)
.or.(v(20)>=0)):
```

Table A.3 SIMAN View of the train movement logic at the TS1

```
1$             STATION,       TS1;2$;
2$             BRANCH,        1:
                              If,FromStation==1:3$;
                              If,FromStation==11:3$;
                              Else:7$;
3$             DUPLICATE:     4$;
                              1:6$;
4$             ASSIGN:   TravelTimeOfTrain(Train#,1)=tnow-TimeIn:12$;
5$             DELAY:         5:6$;
6$             ASSIGN:        v(22)=v(22)+1:
                              v(101)=v(101)+1:
                              v(21)=v(21)+1:23$;
7$             BRANCH,        1:
                              If,Destination==1:8$;
                              If,Destination==11:10$;
8$             ASSIGN:        FromStation=21:
                              ToStation=1:
                              TimeIn=tnow:9$;
9$             TRANSPORT:     TrainFleet,S1_1;
10$            ASSIGN:        FromStation=21:
                              ToStation=11:
                              TimeIn=tnow:$11;
11$            TRANSPORT:     TrainFleet,S1_2;
12$            BRANCH,        1:
                              If,(nx(73)==0).and.(ndx(82)==0):13$;
                              If,(nx(74)==0).and.(ndx(83)==0):14$;
                              If,(nx(75)==0).and.(ndx(84)==0):15$;
                              If,(nx(76)==0).and.(ndx(85)==0):16$;
                              If,(nx(77)==0).and.(ndx(86)==0):17$;
                              If,(nx(78)==0).and.(ndx(87)==0):18$;
                              If,(nx(79)==0).and.(ndx(88)==0):19$;
                              If,(nx(80)==0).and.(ndx(89)==0):20$;
                              If,(nx(81)==0).and.(ndx(90)==0):21$;
                              If,(nx(82)==0).and.(ndx(91)==0):22$;
13$            TRANSPORT:     TrainFleet,Sta73_park;
14$            TRANSPORT:     TrainFleet,Sta74_park;
15$            TRANSPORT:     TrainFleet,Sta75_park;
16$            TRANSPORT:     TrainFleet,Sta76_park;
17$            TRANSPORT:     TrainFleet,Sta77_park;
18$            TRANSPORT:     TrainFleet,Sta78_park;
19$            TRANSPORT:     TrainFleet,Sta79_park;
20$            TRANSPORT:     TrainFleet,Sta80_park;
21$            TRANSPORT:     TrainFleet,Sta81_park;
22$            TRANSPORT:     TrainFleet,Sta82_park;
23$            DISPOSE:
```

Table A.4 SIMAN View of the train movement logic at the first part of the S5

```
1$            STATION,       S5_1:2$;
2$            BRANCH,        1:
                             If,FromStation==4:3$;
                             If,FromStation==14:3$;
                             If,FromStation==6:5$;
                             If,FromStation==16:5$;
3$            ASSIGN:        TrainArrivalTime(Train#,14)=tnow:
                             v(35)=v(35)-1:
                             v(127)=v(127)-1:
                             v(36)=v(36)-1:
                             v(128)=v(128)-1:
                             v(37)=v(37)-1:
                             v(129)=v(129)-1:
                             v(38)=v(38)-1:
                             v(130)=v(130)-1:
                             v(39)=v(39)-1:
                             FromStation=5:4$;
4$            DELAY:         600+expo(90):7$;
5$            ASSIGN:        TrainArrivalTime(Train#,14)=tnow:
                             v(44)=v(44)+1:
                             v(138)=v(138)+1:
                             v(43)=v(43)+1:
                             v(137)=v(137)+1:
                             v(42)=v(42)+1:
                             v(136)=v(136)+1:
                             v(41)=v(41)+1:
                             v(135)=v(135)+1:
                             v(40)=v(40)+1:
                             FromStation=5:6$;
6$            DELAY:         600+expo(90):29$;
7$            ASSIGN:        v(2000)=v(2000)+1:
                             TrainOrderInQ=v(2000):8$;
8$            QUEUE,         Q5:9$;
9$            SCAN:          (v(1005)==0):10$;
10$           SEIZE,         R5,1:11$;
11$           DELAY:         0.000001:12$;
12$           BRANCH,        1:
                             If,FromStation==5:13$;
                             If,FromStation==15:16$:
                             If,FromStation==6:21$;
                             If,FromStation==16:24$;

13$           BRANCH,        1:
                             If,(ed(51)==1):14$;
                             If,(ed(52)==1):15$;
                             Else:19$;
14$           RELEASE:       R5,1:51$;
15$           RELEASE:       R5,1:52$;
16$           BRANCH,        1:
                             If,(ed(51)==1):17$;
                             If,(ed(52)==1):18$;
                             Else:19$;
17$           RELEASE:       R5,1:related with 2.part of S5;
18$           RELEASE:       R5,1:related with 2.part of S5;
19$           RELEASE:       R5,1:20$;
```

```
20$            SCAN:          (ed(51)==1).or.(ed(52)==1):8$;
21$            BRANCH,        1:
                              If,(ed(63)==1):22$;
                              If,(ed(64)==1):23$;
                              Else:27$;
22$            RELEASE:       R5,1:related with 1.part of S6;
23$            RELEASE:       R5,1:related with 1.part of S6;
24$            BRANCH,        1:
                              If,(ed(63)==1):25$;
                              If,(ed(64)==1):26$;
                              Else:27$;
25$            RELEASE:       R5,1:related with 2.part of S6;
26$            RELEASE:       R5,1:related with 2.part of S6;
27$            RELEASE:       R5,1:28$;
28$            SCAN:          (ed(63)==1).or.(ed(64)==1):8$;
29$            ASSIGN:        v(2000)=v(2000)+1:
                              TrainOrderInQ=v(2000):30$;
30$            QUEUE,         Q4:31$;
31$            SCAN:          (v(1004)==0):32$;
32$            SEIZE,         R4,1:33$;
33$            DELAY:         0.000001:34$;
34$            BRANCH,        1:
                              If,FromStation==4:35$;
                              If,FromStation==14:38$;
                              If,FromStation==5:43$;
                              If,FromStation==15:46$;
35$            BRANCH,        1:
                              If,(ed(41)==1):36$;
                              If,(ed(42)==1):37$;
                              Else:41$;
36$            RELEASE:       R4,1:related with 1.part of S4;
37$            RELEASE:       R4,1:related with 1.part of S4;
38$            BRANCH,        1:
                              If,(ed(41)==1):39$;
                              If,(ed(42)==1):40$;
                              Else:41$;
39$            RELEASE:       R4,1:related with 2.part of S4;
40$            RELEASE:       R4,1:related with 2.part of S4;
41$            RELEASE:       R4,1:42$;
42$            SCAN:          (ed(41)==1).or.(ed(42)==1):30$;
43$            BRANCH,        1:
                              If,(ed(53)==1):44$;
                              If,(ed(54)==1):45$;
                              Else:49$;
44$            RELEASE:       R4,1:57$;
45$            RELEASE:       R4,1:58$;
46$            BRANCH,        1:
                              If,(ed(53)==1):47$;
                              If,(ed(54)==1):48$;
                              Else:49$;
47$            RELEASE:       R4,1:related with 2.part of S5;
48$            RELEASE:       R4,1:related with 2.part of S5;
49$            RELEASE:       R4,1:50$;
50$            SCAN:          (ed(53)==1).or.(ed(54)==1):30$;
51$            ASSIGN:        ToStation=6:
                              v(40)=v(40)+1:
                              v(135)=v(135)+1:
                              v(41)=v(41)+1:
                              v(136)=v(136)+1:
```

```
                              v(42)=v(42)+1:
                              v(137)=v(137)+1:
                              v(43)=v(43)+1:
                              v(138)=v(138)+1:
                              v(44)=v(44)+1:
                              v(139)=v(139)+1:
                              v(6)=v(6)+1:
                              v(140)=v(140)+1:
                              TrainDepartureTime(Train#,14)=tnow:53$;
52$           ASSIGN:         ToStation=16:
                              v(40)=v(40)+1:
                              v(135)=v(135)+1:
                              v(41)=v(41)+1:
                              v(136)=v(136)+1:
                              v(42)=v(42)+1:
                              v(137)=v(137)+1:
                              v(43)=v(43)+1:
                              v(138)=v(138)+1:
                              v(44)=v(44)+1:
                              v(141)=v(141)+1:
                              v(16)=v(16)+1:
                              v(142)=v(142)+1:
                              TrainDepartureTime(Train#,14)=tnow:53$;
53$           DUPLICATE:      54$;
                              1:55$;
54$           TRANSPORT:      TrainFleet,dS51,unif(25.00,30.56);
55$           DELAY:          10:56$;
56$           ASSIGN:         v(131)=v(131)-1:
                              v(5)=v(5)-1:
                              v(132)=v(132)-1:63$;
57$           ASSIGN:         ToStation=4:
                              v(39)=v(39)-1:
                              v(130)=v(130)-1:
                              v(38)=v(38)-1:
                              v(129)=v(129)-1:
                              v(37)=v(37)-1:
                              v(128)=v(128)-1:
                              v(36)=v(36)-1:
                              v(127)=v(127)-1:
                              v(35)=v(35)-1:
                              v(124)=v(124)-1:
                              v(4)=v(4)-1:
                              v(123)=v(123)-1:
                              TrainDepartureTime(Train#,14)=tnow:59$;
58$           ASSIGN:         ToStation=14:
                              v(39)=v(39)-1:
                              v(130)=v(130)-1:
                              v(38)=v(38)-1:
                              v(129)=v(129)-1:
                              v(37)=v(37)-1:
                              v(128)=v(128)-1:
                              v(36)=v(36)-1:
                              v(127)=v(127)-1:
                              v(35)=v(35)-1:
                              v(126)=v(126)-1:
                              v(14)=v(14)-1:
                              v(125)=v(125)-1:
                              TrainDepartureTime(Train#,14)=tnow:59$;
```

```
59$           DUPLICATE:     60$;
                             1:61$;
60$           TRANSPORT:     TrainFleet,dS43,unif(25.00,30.56);
61$           DELAY:         10:62$;
62$           ASSIGN:        v(132)=v(132)+1:
                             v(5)=v(5)+1:
                             v(131)=v(131)+1:63$;
63$           DISPOSE:

EXPRESSIONS:

41,FromS4p1OrS4p2ToS5p1,
(v(35)>=0).and.(v(127)>=0).and.(v(36)>=0).and.(v(128)>=0).and.(v(37)
>=0).and.(v(129)>=0).and.(v(38)>=0).and.(v(130)>=0).and.(v(39)>=0).a
nd.(v(131)==0).and.(v(5)==0).and.(v(132)==0).and.(v(1004)==0).and.((
v(15)<=0).or.(v(6)>=0).or.(v(16)>=0)).and.((v(15)<=0).or.((v(6)==0).
or.(v(16)==0)).or.(v(7)>=0).or.(v(17)>=0)).and.((v(15)<=0).or.((v(6)
==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0)).or.(v(8)>=0).or.(v(18
)>=0)).and.((v(15)<=0).or.((v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(
v(17)==0).or.(v(8)==0).or.(v(18)==0)).or.(v(9)>=0).or.(v(19)>=0)).an
d.((v(15)<=0).or.((v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0)
.or.(v(8)==0).or.(v(18)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(109)>=
0).or.(v(20)>=0)):

42,FromS4p1OrS4p2ToS5p2,
(v(35)>=0).and.(v(127)>=0).and.(v(36)>=0).and.(v(128)>=0).and.(v(37)
>=0).and.(v(129)>=0).and.(v(38)>=0).and.(v(130)>=0).and.(v(39)>=0).a
nd.(v(133)==0).and.(v(15)==0).and.(v(134)==0).and.(v(1004)==0).and.(
(v(5)<=0).or.(v(6)>=0).or.(v(16)>=0)).and.((v(5)<=0).or.((v(6)==0).o
r.(v(16)==0)).or.(v(7)>=0).or.(v(17)>=0)).and.((v(5)<=0).or.((v(6)==
0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0)).or.(v(8)>=0).or.(v(18)>
=0)).and.((v(5)<=0).or.((v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(1
7)==0).or.(v(8)==0).or.(v(18)==0)).or.(v(9)>=0).or.(v(19)>=0)).and.(
(v(5)<=0).or.((v(6)==0).or.(v(16)==0).or.(v(7)==0).or.(v(17)==0).or.
(v(8)==0).or.(v(18)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(109)>=0).o
r.(v(20)>=0)):

51,FromS5p1OrS5p2ToS6p1,
(v(40)>=0).and.(v(135)>=0).and.(v(41)>=0).and.(v(136)>=0).and.(v(42)
>=0).and.(v(137)>=0).and.(v(43)>=0).and.(v(138)>=0).and.(v(44)>=0).a
nd.(v(139)==0).and.(v(6)==0).and.(v(140)==0).and.(v(1005)==0).and.((
v(16)<=0).or.(v(7)>=0).or.(v(17)>=0)).and.((v(16)<=0).or.((v(7)==0).
or.(v(17)==0)).or.(v(8)>=0).or.(v(18)>=0)).and.((v(16)<=0).or.((v(7)
==0).or.(v(17)==0).or.(v(8)==0).or.(v(18)==0)).or.(v(9)>=0).or.(v(19
)>=0)).and.((v(16)<=0).or.((v(7)==0).or.(v(17)==0).or.(v(8)==0).or.(
v(18)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(10)>=0).or.(v(20)>=0)):

52,FromS5p1OrS5p2ToS6p2,
(v(40)>=0).and.(v(135)>=0).and.(v(41)>=0).and.(v(136)>=0).and.(v(42)
>=0).and.(v(137)>=0).and.(v(43)>=0).and.(v(138)>=0).and.(v(44)>=0).a
nd.(v(141)==0).and.(v(16)==0).and.(v(142)==0).and.(v(1005)==0).and.(
(v(6)<=0).or.(v(7)>=0).or.(v(17)>=0)).and.((v(6)<=0).or.((v(7)==0).o
r.(v(17)==0)).or.(v(8)>=0).or.(v(18)>=0)).and.((v(6)<=0).or.((v(7)==
0).or.(v(17)==0).or.(v(8)==0).or.(v(18)==0)).or.(v(9)>=0).or.(v(19)>
=0)).and.((v(6)<=0).or.((v(7)==0).or.(v(17)==0).or.(v(8)==0).or.(v(1
8)==0).or.(v(9)==0).or.(v(19)==0)).or.(v(10)>=0).or.(v(20)>=0)):
```

```
53,FromS5p1OrS5p2ToS4p1,
(v(39)<=0).and.(v(130)<=0).and.(v(38)<=0).and.(v(129)<=0).and.(v(37)
<=0).and.(v(128)<=0).and.(v(36)<=0).and.(v(127)<=0).and.(v(35)<=0).a
nd.(v(124)==0).and.(v(4)==0).and.(v(123)==0).and.(v(1004)==0).and.((
v(14)>=0).or.(v(3)<=0).or.(v(13)<=0)).and.((v(14)>=0).or.((v(3)==0).
or.(v(13)==0)).or.(v(2)<=0).or.(v(12)<=0)).and.((v(14)>=0).or.((v(3)
==0).or.(v(13)==0).or.(v(2)==0).or.(v(12)==0)).or.(v(1)<=0).or.(v(11
)<=0)):

54,FromS5p1OrS5p2ToS4p2,
(v(39)<=0).and.(v(130)<=0).and.(v(38)<=0).and.(v(129)<=0).and.(v(37)
<=0).and.(v(128)<=0).and.(v(36)<=0).and.(v(127)<=0).and.(v(35)<=0).a
nd.(v(126)==0).and.(v(14)==0).and.(v(125)==0).and.(v(1004)==0).and.(
(v(4)>=0).or.(v(3)<=0).or.(v(13)<=0)).and.((v(4)>=0).or.((v(3)==0).o
r.(v(13)==0)).or.(v(2)<=0).or.(v(12)<=0)).and.((v(4)>=0).or.((v(3)==
0).or.(v(13)==0).or.(v(2)==0).or.(v(12)==0)).or.(v(1)<=0).or.(v(11)<
=0)):

63,FromS6p1OrS6p2ToS5p1,
(v(44)<=0).and.(v(138)<=0).and.(v(43)<=0).and.(v(137)<=0).and.(v(42)
<=0).and.(v(136)<=0).and.(v(41)<=0).and.(v(135)<=0).and.(v(40)<=0).a
nd.(v(132)==0).and.(v(5)==0).and.(v(131)==0).and.(v(1005)==0).and.((
v(15)>=0).or.(v(4)<=0).or.(v(14)<=0)).and.((v(15)>=0).or.((v(4)==0).
or.(v(14)==0)).or.(v(3)<=0).or.(v(13)<=0)).and.((v(15)>=0).or.((v(4)
==0).or.(v(14)==0).or.(v(3)==0).or.(v(13)==0)).or.(v(2)<=0).or.(v(12
)<=0)).and.((v(15)>=0).or.((v(4)==0).or.(v(14)==0).or.(v(3)==0).or.(
v(13)==0).or.(v(2)==0).or.(v(12)==0)).or.(v(1)<=0).or.(v(11)<=0)):

64,FromS6p1OrS6p2ToS5p2,
(v(44)<=0).and.(v(138)<=0).and.(v(43)<=0).and.(v(137)<=0).and.(v(42)
<=0).and.(v(136)<=0).and.(v(41)<=0).and.(v(135)<=0).and.(v(40)<=0).a
nd.(v(134)==0).and.(v(15)==0).and.(v(133)==0).and.(v(1005)==0).and.(
(v(5)>=0).or.(v(4)<=0).or.(v(14)<=0)).and.((v(5)>=0).or.((v(4)==0).o
r.(v(14)==0)).or.(v(3)<=0).or.(v(13)<=0)).and.((v(5)>=0).or.((v(4)==
0).or.(v(14)==0).or.(v(3)==0).or.(v(13)==0)).or.(v(2)<=0).or.(v(12)<
=0)).and.((v(5)>=0).or.((v(4)==0).or.(v(14)==0).or.(v(3)==0).or.(v(1
3)==0).or.(v(2)==0).or.(v(12)==0)).or.(v(1)<=0).or.(v(11)<=0)):
```

Table A.5 SIMAN View of train movement logic at the dummy station dS51

```
0$            STATION,       dS51:1$;
1$            ASSIGN:        TrainArrivalTime(Train#,15)=tnow:2$;
2$            BRANCH,        1:
                             If,ToStation==5:3$;
                             If,ToStation==15:3$;
                             If,ToStation==6:4$;
                             If,ToStation==16:4$;
3$            SCAN:          v(1135)==0:5$;
4$            SCAN:          v(1136)==0:5$;
5$            ASSIGN:        TrainDepartureTime(Train#,15)=tnow:6$;
6$            BRANCH,        1:
                             If,ToStation==5:7$;
                             If,ToStation==15:8$;
                             If,ToStation==6:9$;
                             If,ToStation==16:9$;
7$            TRANSPORT:     TrainFleet,S5_1,unif(23.6,26.4);
8$            TRANSPORT:     TrainFleet,S5_2,unif(23.6,26.4);
9$            TRANSPORT:     TrainFleet,dS52,unif(23.6,26.4);
```

Table A.6 Framework of the *SimGAb*

Begin *SimGAb*

Generate INITIAL POPULATION (Sub steps are given in Table 5.1)
Rank the initial population
Select best of the initial population

GN; generation number
maxGN; maximum generation number
From GN = 1 to GN = maxGN
  CROSSOVER (Sub steps are given in Table 5.2)
  MUTATION (Sub steps are given in Table 5.3)
  Rank the population
    The current population includes; individuals from the previous generation population, children and mutated child(ren)
  Select best of the population
  LOCAL SEARCH on the best
  Search neighbours of the best chromosome
  bChr(GN); the best chromosome in the current generation
  bChr(GN) = [$i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9$]
    From p =1 to p = 9
      $\boldsymbol{i_p}$; the gene related with p^{th} decision point
        If i_p = 1 change the value to 2
        If i_p = 2 change the value to 1
        If i_p = 3 change the value to 4
        If i_p = 4 change the value to 3
        If i_p = 5 change the value to 6
        If i_p = 6 change the value to 5
          nbChr(GN, *p*); p^{th} neighbour of the best chromosome
          Rearrange the simulation model (Sub steps are given in Table 5.1)
          Run the rearranged simulation model (Sub steps are given in Table 5.1)
          Calculate the fitness value
      Next *p*
  Rank the population
    The current population includes; individuals from the previous generation population, children, mutated child(ren) and nine neighbours of the best chromosome
  Select best of the population
  Check TERMINATION CRITERIA (Sub steps are given in Table 5.4)
  Form the population of the next generation
Next GN

Stop *SimGAb*

Table A.7 Framework of the *SimGAfs*

Begin *SimGAfs*

Generate INITIAL POPULATION (Sub steps are given in Table 5.1)
Rank the initial population
Select best of the initial population

GN; generation number
maxGN; maximum generation number
From GN = 1 to GN = maxGN
    CROSSOVER (Sub steps are given in Table 5.2)
    MUTATION (Sub steps are given in Table 5.3)
    Rank the population
        The current population includes; individuals from the previous generation population, children and mutated child(ren)
    Select best of the population
    LOCAL SEARCH on the best (Sub steps are given in Table A.6)
    Select the second best of the population
        Record the second best fitness value and related chromosome
    LOCAL SEARCH on the second best
    Search neighbours of the second best chromosome
    sbChr(GN); the second best chromosome in the current generation
    sbChr(GN) = [i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9]
        From p =1 to p = 9
            $\boldsymbol{i_p}$; the gene related with p^{th} decision point
                If i_p = 1 change the value to 2
                If i_p = 2 change the value to 1
                If i_p = 3 change the value to 4
                If i_p = 4 change the value to 3
                If i_p = 5 change the value to 6
                If i_p = 6 change the value to 5
            nsbChr(GN, *p*); p^{th} neighbour of the second best chromosome
            Rearrange the simulation model (Sub steps are given in Table 5.1)
            Run the rearranged simulation model (Sub steps are given in Table 5.1)
            Calculate the fitness value
        Next *p*
    Rank the population
        The current population includes; individuals from the previous generation population, children, mutated child(ren), nine neighbours of the best chromosome, and nine neighbours of the second best chromosome
    Select best of the population
    Check TERMINATION CRITERIA (Sub steps are given in Table 5.4)
    Form the population of the next generation
Next GN

Stop *SimGAfs*

Table A.8 Framework of the *SimGAbw*

Begin *SimGAbw*

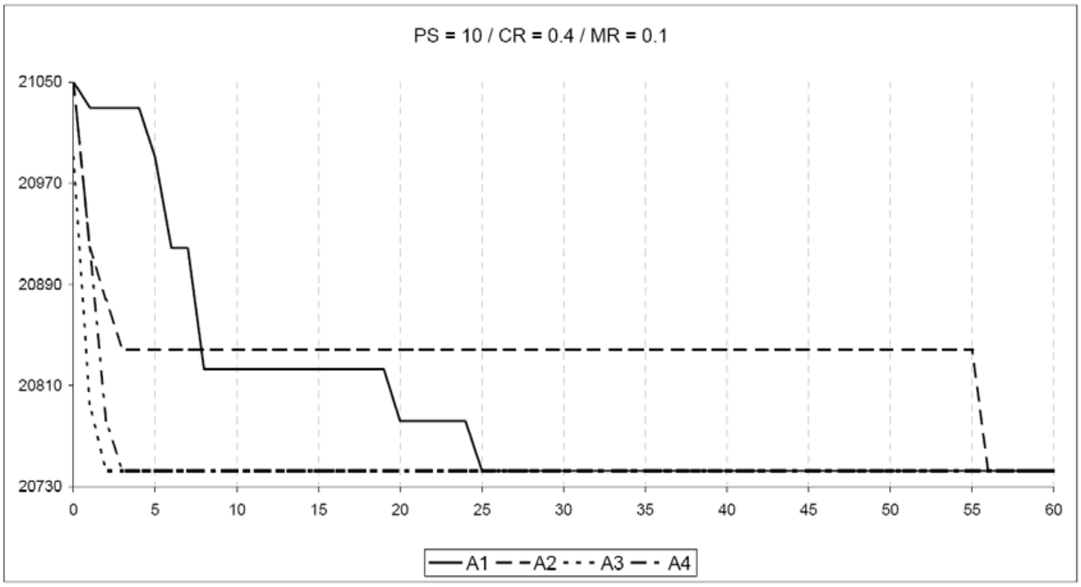
Generate INITIAL POPULATION (Sub steps are given in Table 5.1)
Rank the initial population
Select best of the initial population

GN; generation number
maxGN; maximum generation number
From GN = 1 to GN = maxGN
    CROSSOVER (Sub steps are given in Table 5.2)
    MUTATION (Sub steps are given in Table 5.3)
    Rank the population
        The current population includes; individuals from the previous generation population, children and mutated child(ren)
    Select best of the population
    LOCAL SEARCH on the best (Sub steps are given in Table A.6)
    Select the worst of the population
        Record the worst fitness value and related chromosome
    LOCAL SEARCH on the worst
    Search neighbours of the worst chromosome
    wChr(GN); the worst chromosome in the current generation
    wChr(GN) = [$i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9$]
        From p =1 to p = 9
            $\boldsymbol{i_p}$; the gene related with p^{th} decision point
                If i_p = 1 change the value to 2
                If i_p = 2 change the value to 1
                If i_p = 3 change the value to 4
                If i_p = 4 change the value to 3
                If i_p = 5 change the value to 6
                If i_p = 6 change the value to 5
            nwChr(GN, *p*); p^{th} neighbour of the worst chromosome
            Rearrange the simulation model (Sub steps are given in Table 5.1)
            Run the rearranged simulation model (Sub steps are given in Table 5.1)
            Calculate the fitness value
        Next *p*
    Rank the population
        The current population includes; individuals from the previous generation population, children, mutated child(ren), nine neighbours of the best chromosome, and nine neighbours of the worst chromosome
    Select best of the population
    Check TERMINATION CRITERIA (Sub steps are given in Table 5.4)
    Form the population of the next generation
Next GN

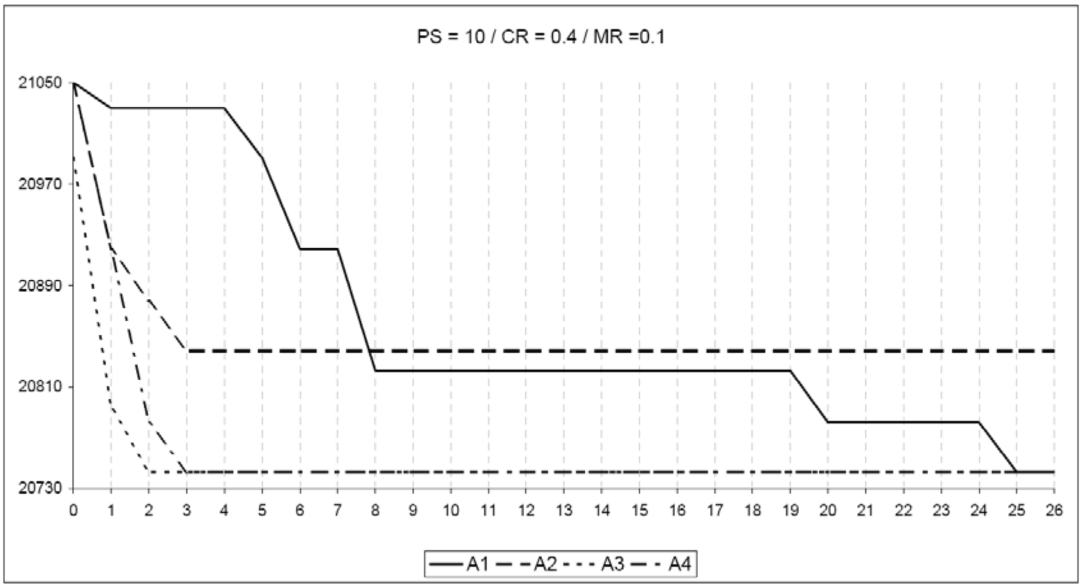
Stop *SimGAbw*

Table A.9 Results of the algorithms

				SimGA			*SimGAb*			*SimGAfs*			*SimGAbw*		
	PS	**CR**	**MR**	**Chromosome**	**Fitness**	**GN**	**Chromosome**	**Fitness**	**GN**	**Chromosome**	**Fitness**	**GN**	**Chromosome**	**Fitness**	**GN**
1	10	0.4	0.1	6,3,4,2,4,4,4,1,4	20742.93	25	1,3,1,2,2,4,4,1,4	20742.93	56	1,6,4,2,4,1,5,3,5	20742.93	2	6,3,4,2,2,4,4,3,5	20742.93	3
2	10	0.4	0.2	1,3,1,2,4,4,5,6,4	20742.93	16	1,3,1,2,4,4,4,1,4	20742.93	29	1,3,1,2,4,4,4,1,4	20742.93	29	1,3,1,2,4,1,4,6,5	20742.93	20
3	10	0.4	0.3	1,3,1,2,4,4,2,6,4	20742.93	13	1,3,1,2,4,4,4,1,4	20742.93	28	1,3,1,2,4,4,4,1,4	20742.93	28	1,3,1,2,4,4,4,1,4	20742.93	28
4	10	0.6	0.1	6,3,4,2,2,4,4,1,4	20742.93	37	1,3,4,2,2,4,4,1,4	20742.93	38	1,3,1,2,2,4,4,1,4	20742.93	34	1,3,1,2,2,4,4,1,4	20742.93	34
5	10	0.6	0.2	1,3,5,2,4,4,5,1,4	20742.93	27	1,3,1,2,2,4,4,1,4	20742.93	27	1,3,1,2,2,4,4,1,4	20742.93	27	1,3,1,2,2,4,4,1,4	20742.93	27
6	10	0.6	0.3	6,3,4,2,2,5,5,1,4	20742.93	13	1,3,1,2,2,4,4,1,4	20742.93	18	1,3,4,2,2,4,4,1,4	20742.93	23	6,3,1,2,4,1,4,6,5	20742.93	5
7	10	0.8	0.1	6,3,5,2,4,4,5,3,4	20742.93	31	6,3,4,2,2,4,4,1,4	20742.93	14	6,3,4,2,2,4,4,1,4	20742.93	14	1,3,1,2,4,1,4,6,2	20753.02	100
8	10	0.8	0.2	1,6,5,2,4,4,5,1,4	20742.93	20	6,3,4,2,2,4,4,1,4	20742.93	9	6,3,1,2,2,4,4,1,4	20742.93	16	6,3,4,2,4,1,4,6,5	20742.93	8
9	10	0.8	0.3	6,3,4,2,2,4,4,1,4	20742.93	6	6,3,1,2,2,4,4,1,4	20742.93	7	6,3,1,2,2,4,4,1,4	20742.93	7	1,3,1,2,4,1,5,6,4	20742.93	14
10	20	0.4	0.1	6,3,1,2,2,4,2,6,4	20742.93	19	6,3,5,2,2,1,2,6,4	20742.93	9	6,6,5,2,2,1,2,6,4	20742.93	9	6,3,5,2,2,1,2,6,4	20742.93	9
11	20	0.4	0.2	3,6,5,2,5,4,4,1,4	20742.93	12	6,3,1,2,5,4,4,1,4	20742.93	19	6,3,5,2,2,1,2,1,4	20742.93	6	6,3,5,2,2,4,4,1,4	20742.93	6
12	20	0.4	0.3	6,3,1,2,5,1,4,6,4	20742.93	13	6,3,4,2,2,1,2,6,4	20742.93	6	6,3,5,2,2,1,4,1,4	20742.93	5	6,3,4,2,2,1,2,6,4	20742.93	6
13	20	0.6	0.1	6,3,1,2,4,4,5,1,4	20742.93	17	1,3,1,2,2,1,5,6,4	20742.93	27	1,3,1,2,2,1,5,6,4	20742.93	27	1,3,1,2,2,1,5,6,4	20742.93	27
14	20	0.6	0.2	6,3,5,2,5,1,2,1,4	20742.93	15	6,6,5,2,2,1,2,6,4	20742.93	7	6,3,5,2,2,1,2,6,4	20742.93	6	6,6,5,2,2,1,2,1,4	20742.93	7
15	20	0.6	0.3	1,3,1,2,4,4,5,1,5	20742.93	24	6,3,5,2,2,1,2,6,4	20742.93	9	6,3,5,2,2,5,2,6,4	20742.93	8	6,3,5,2,2,1,2,6,4	20742.93	9
16	20	0.8	0.1	6,3,5,2,2,1,2,1,4	20742.93	10	6,6,5,2,2,1,2,6,4	20742.93	16	6,3,5,2,2,1,2,6,4	20742.93	7	6,3,5,2,2,1,2,6,4	20742.93	7
17	20	0.8	0.2	3,3,5,2,4,1,5,6,4	20742.93	17	6,6,5,2,2,1,2,6,4	20742.93	8	6,6,5,2,2,1,5,6,4	20742.93	8	6,3,4,2,2,1,2,6,4	20742.93	3
18	20	0.8	0.3	1,3,1,2,4,5,4,6,4	20742.93	9	6,3,5,2,2,1,2,6,4	20742.93	5	6,6,5,2,2,1,2,6,4	20742.93	6	6,3,5,2,2,1,2,6,4	20742.93	6
19	30	0.4	0.1	6,3,4,2,5,1,2,1,4	20742.93	12	6,3,1,2,4,4,4,1,4	20742.93	30	6,3,1,2,4,4,4,1,4	20742.93	30	6,3,1,2,4,4,4,1,4	20742.93	30
20	30	0.4	0.2	6,3,4,2,4,5,2,6,4	20742.93	7	1,6,5,2,4,4,4,1,4	20742.93	14	3,6,5,2,2,1,2,6,5	20742.93	9	6,3,4,2,4,4,4,1,4	20742.93	7
21	30	0.4	0.3	6,3,1,2,4,1,2,1,4	20742.93	7	6,3,4,2,4,4,4,1,4	20742.93	3	6,3,4,2,4,4,2,1,4	20742.93	3	6,3,4,2,4,4,4,1,4	20742.93	3
22	30	0.6	0.1	6,3,4,2,4,1,2,1,4	20742.93	24	6,3,4,2,4,4,4,1,4	20742.93	24	6,3,4,2,4,4,4,1,4	20742.93	7	6,3,4,2,4,4,4,1,4	20742.93	24
23	30	0.6	0.2	6,3,5,2,2,4,2,1,4	20742.93	7	6,3,5,2,2,1,2,1,4	20742.93	3	6,3,4,2,4,4,4,1,4	20742.93	3	6,3,4,2,4,4,4,6,4	20742.93	4
24	30	0.6	0.3	6,3,5,2,4,1,2,3,4	20742.93	8	6,3,4,2,4,4,4,1,4	20742.93	4	6,3,4,2,4,4,4,1,4	20742.93	3	6,3,4,2,4,4,4,1,4	20742.93	4
25	30	0.8	0.1	6,3,5,2,4,4,2,3,4	20742.93	6	1,6,4,2,2,5,2,3,5	20742.93	10	6,6,5,2,2,1,4,1,4	20742.93	10	1,6,1,2,2,5,2,6,5	20742.93	38
26	30	0.8	0.2	6,3,1,2,5,4,2,3,4	20742.93	11	6,3,4,2,2,1,2,6,5	20742.93	6	6,3,4,2,4,4,2,6,5	20742.93	9	6,3,5,2,2,5,5,6,5	20742.93	4
27	30	0.8	0.3	6,3,4,2,4,1,2,3,4	20742.93	12	6,3,4,2,4,4,5,1,4	20742.93	7	6,3,4,2,5,4,4,1,4	20742.93	4	6,3,4,2,4,4,4,1,4	20742.93	7



(a)



(b)

Figure A.3 Best fitness values of the generations for 10/0.4/0.1

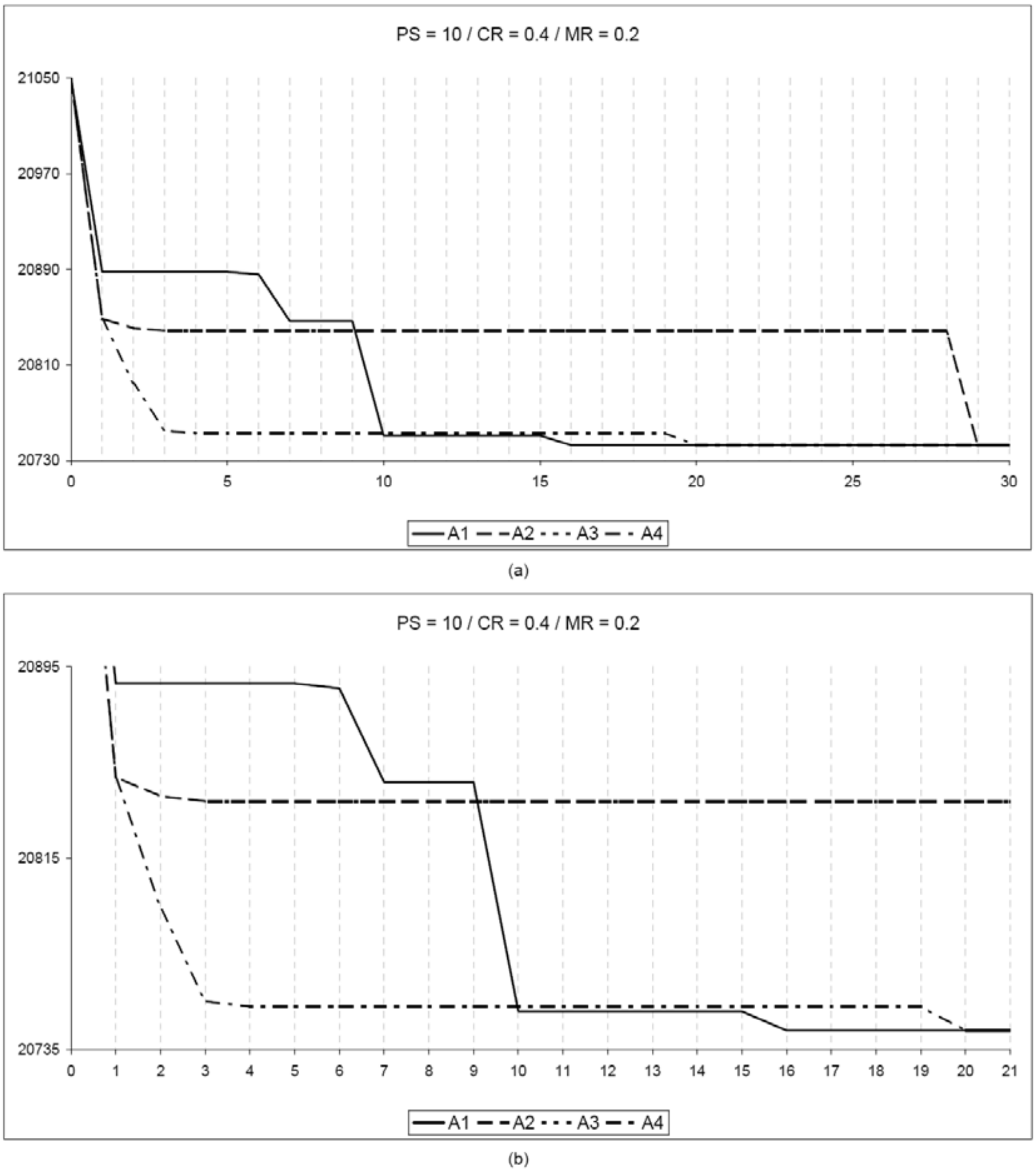
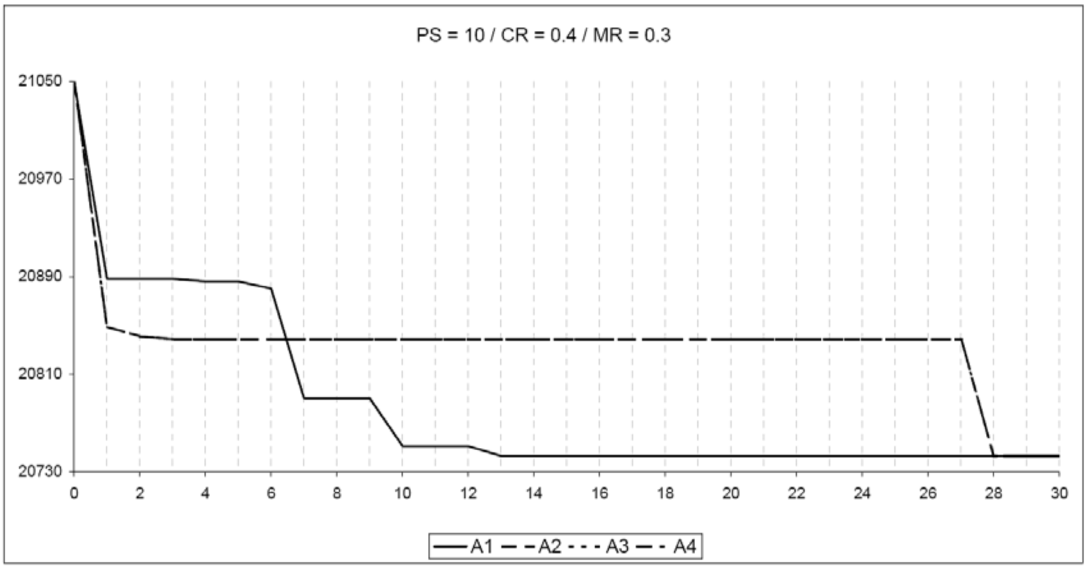
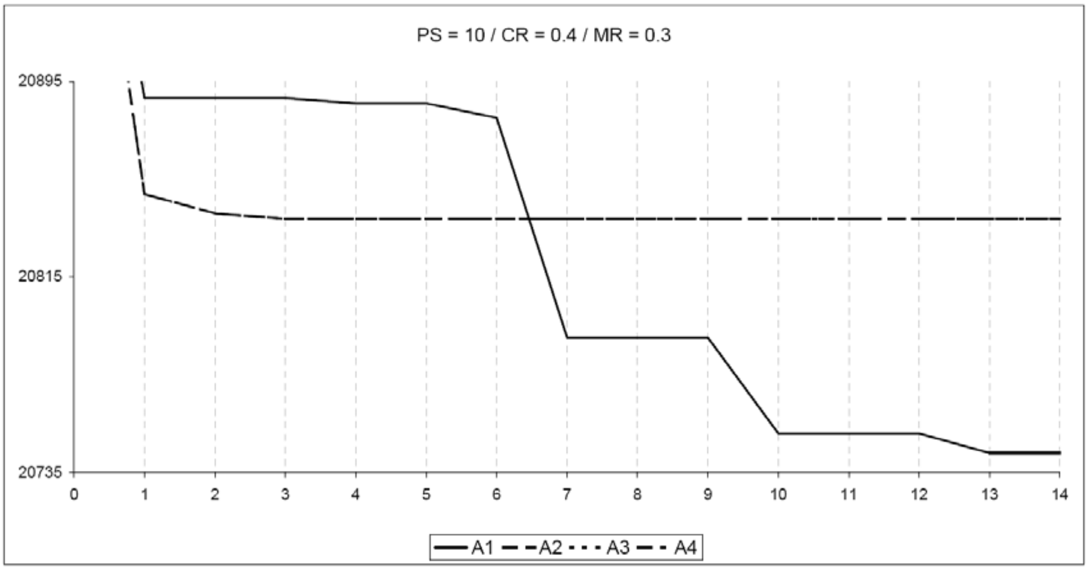


Figure A.4 Best fitness values of the generations for 10/0.4/0.2

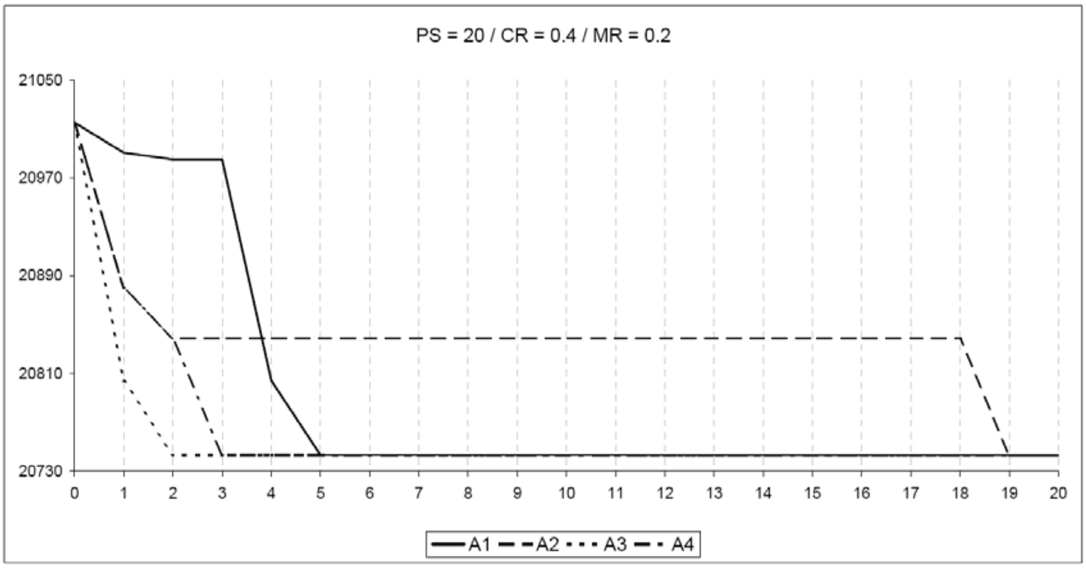


(a)

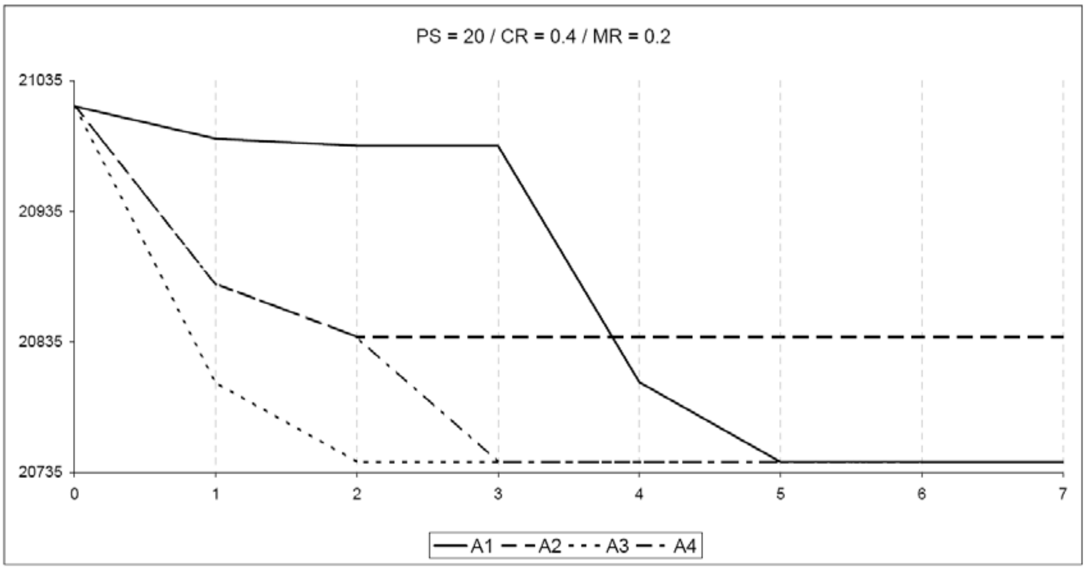


(b)

Figure A.5 Best fitness values of the generations for 10/0.4/0.3

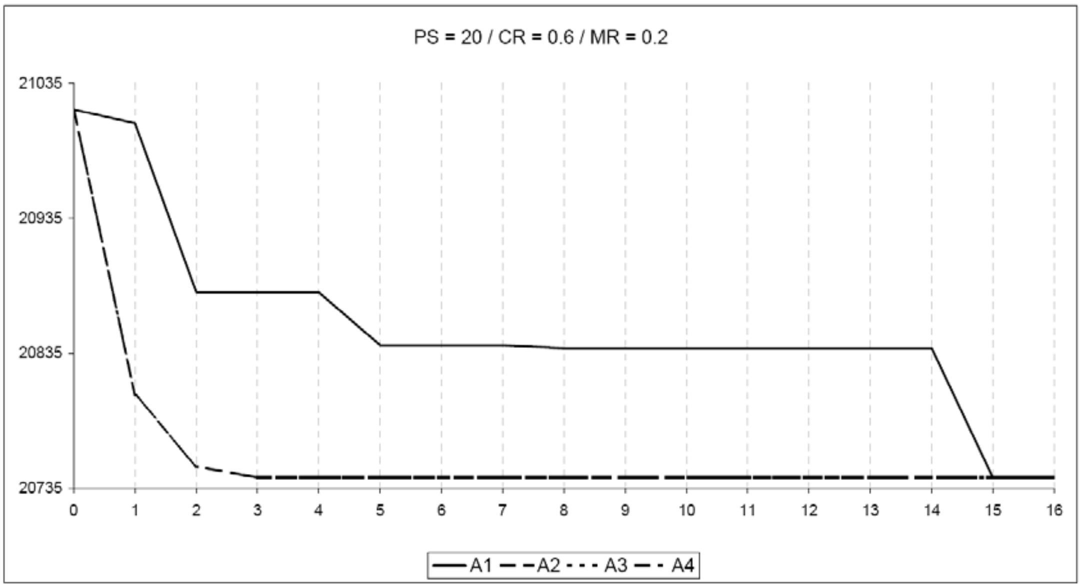


(a)

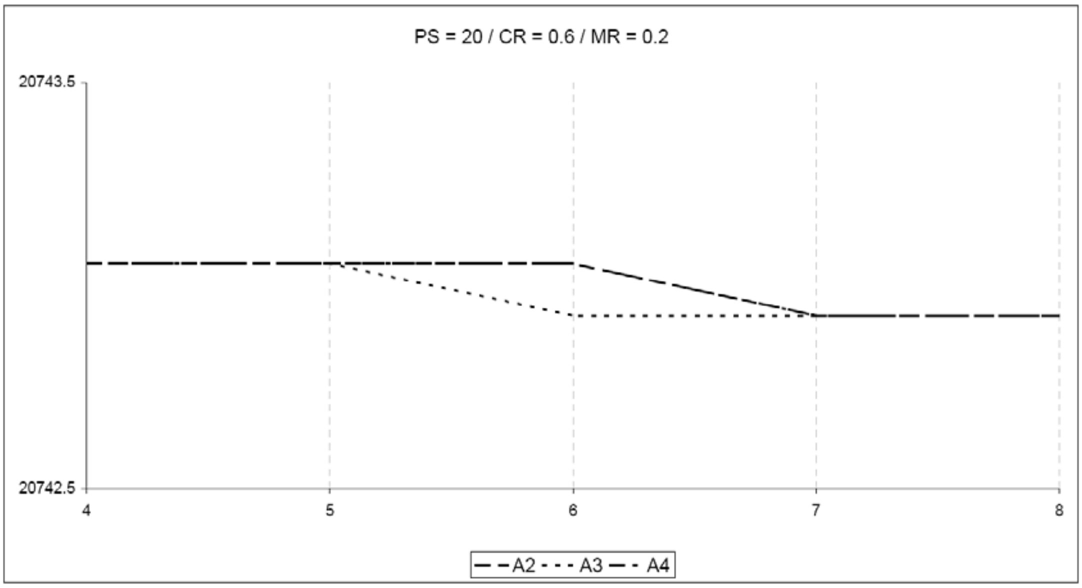


(b)

Figure A.6 Best fitness values of the generations for 20/0.4/0.2



(a)



(b)

Figure A.7 Best fitness values of the generations for 20/0.6/0.2

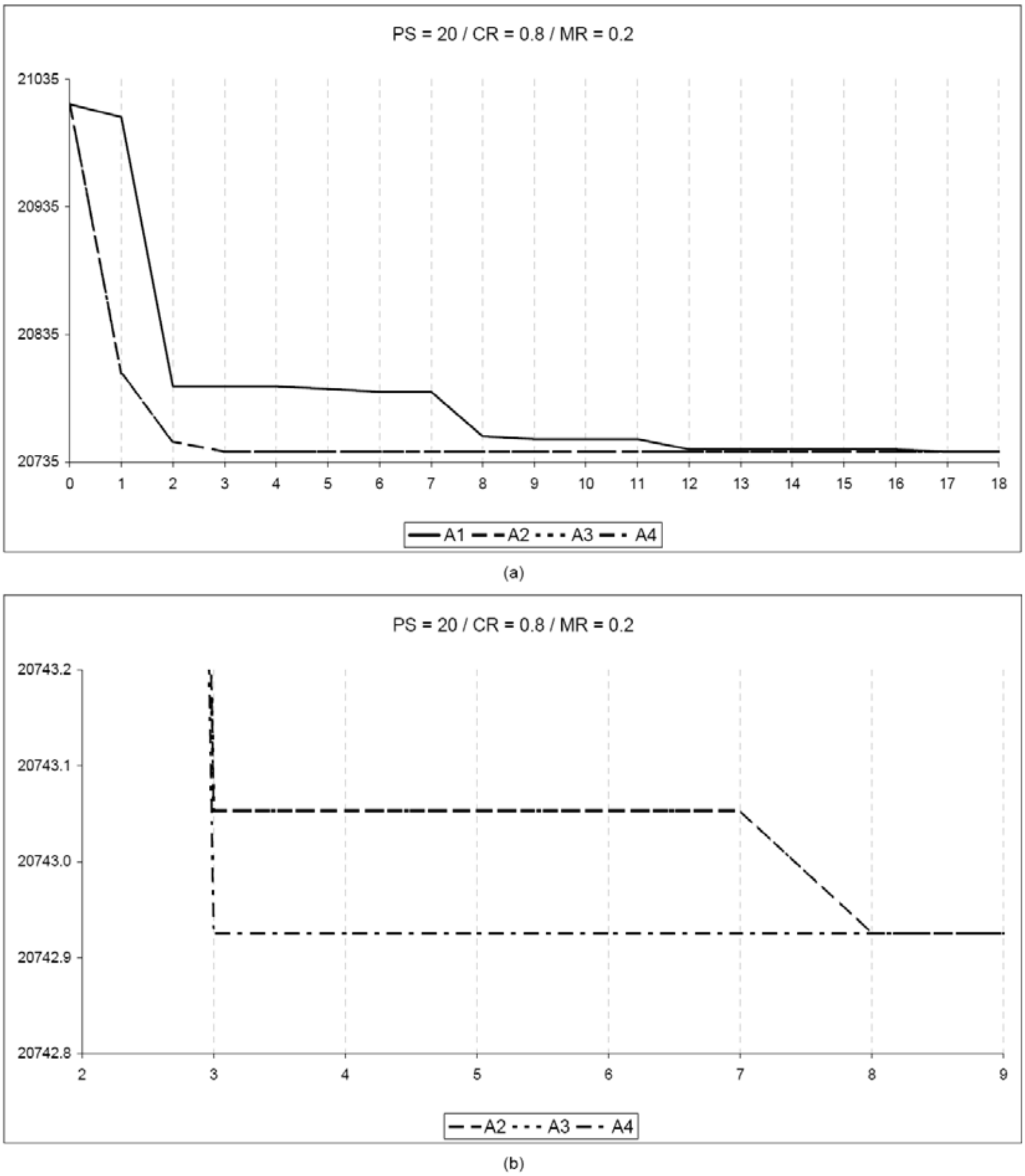


Figure A.8 Best fitness values of the generations for 20/0.8/0.2

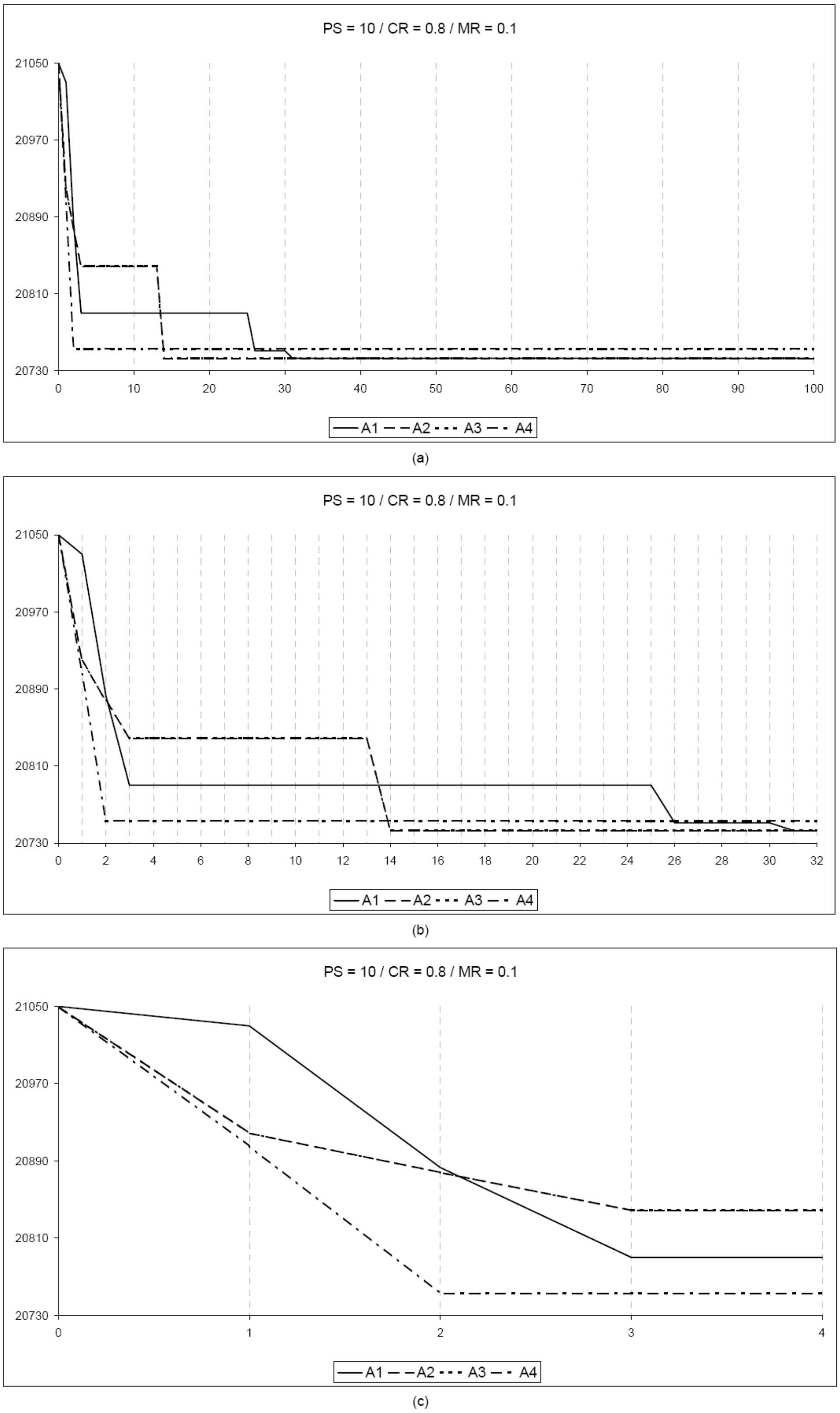
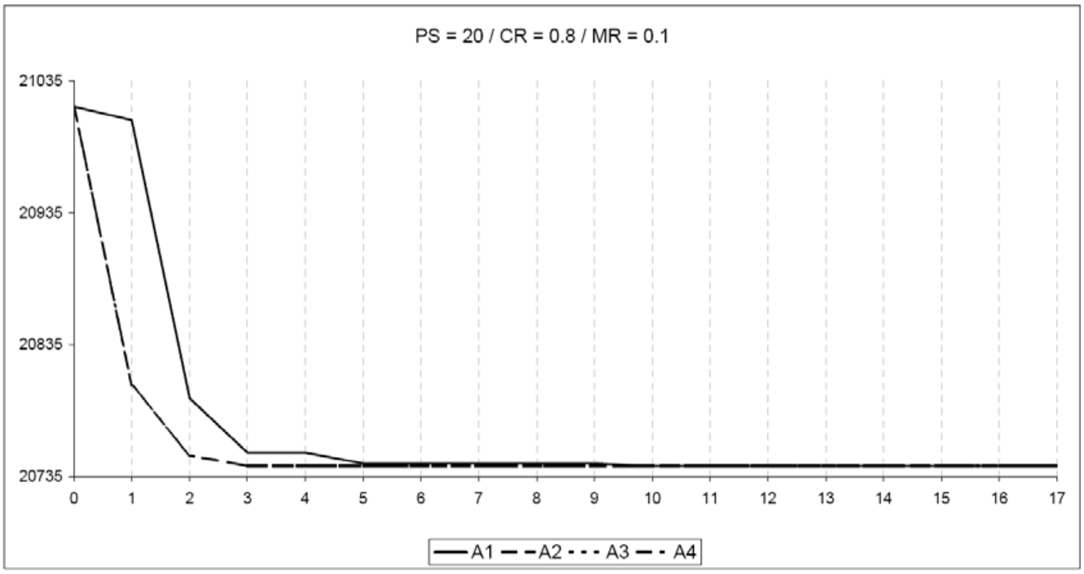
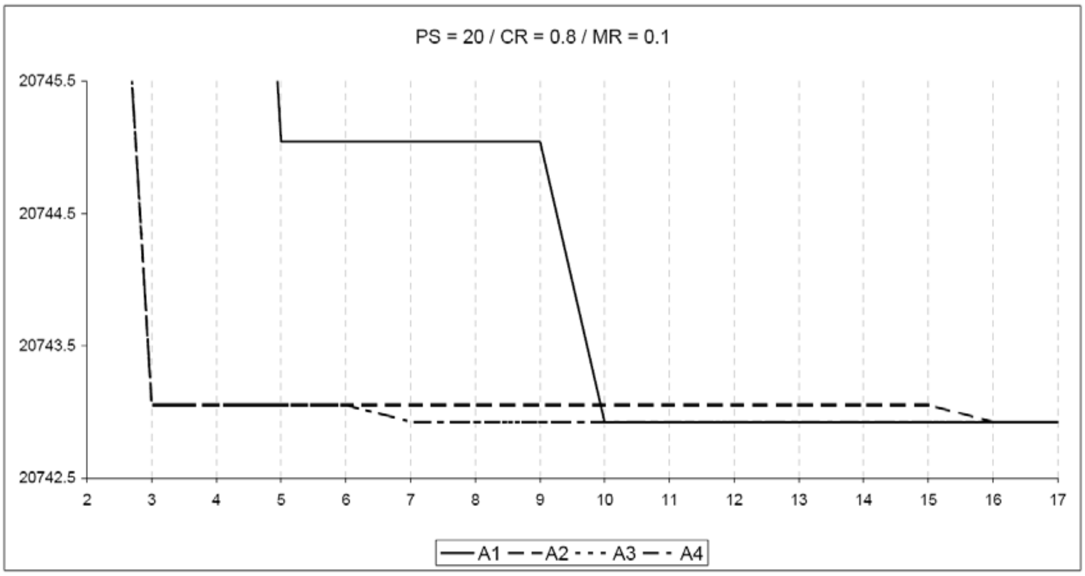


Figure A.9 Best fitness values of the generations for 10/0.8/0.1

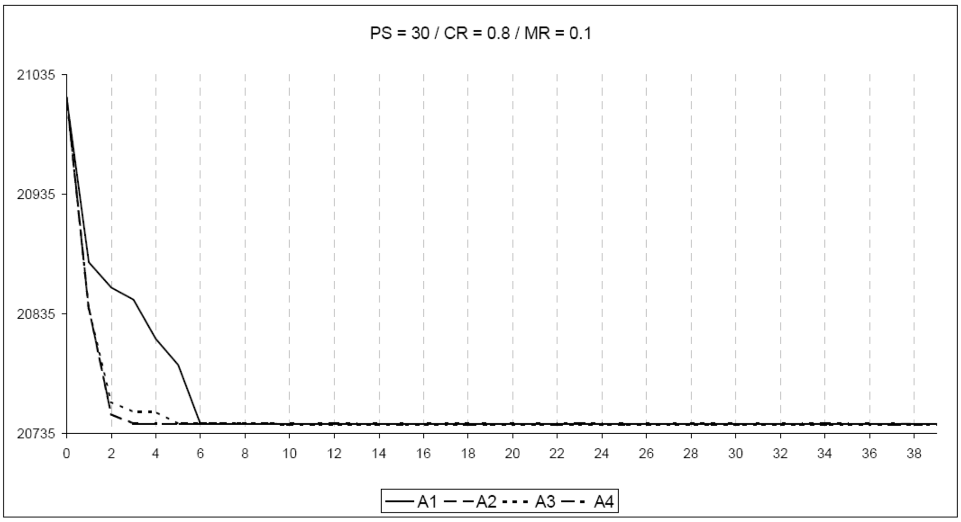


(a)

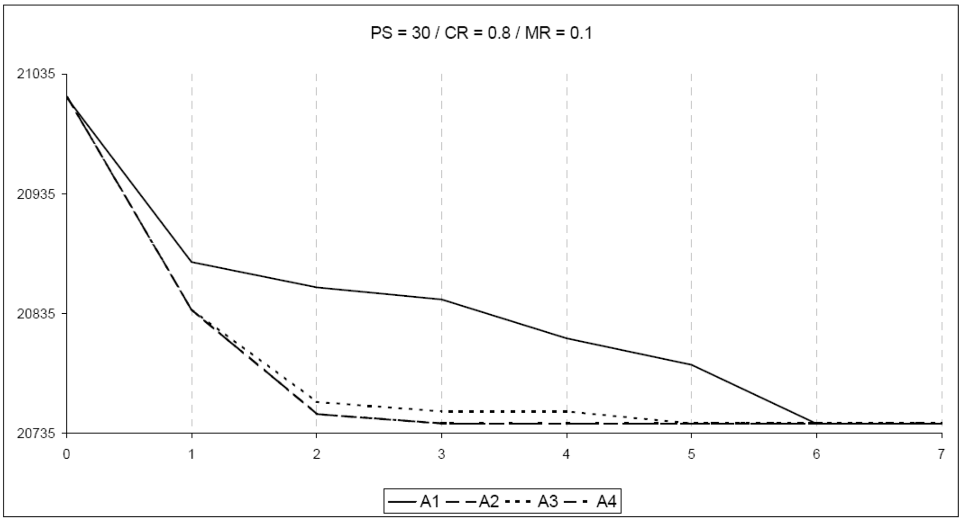


(b)

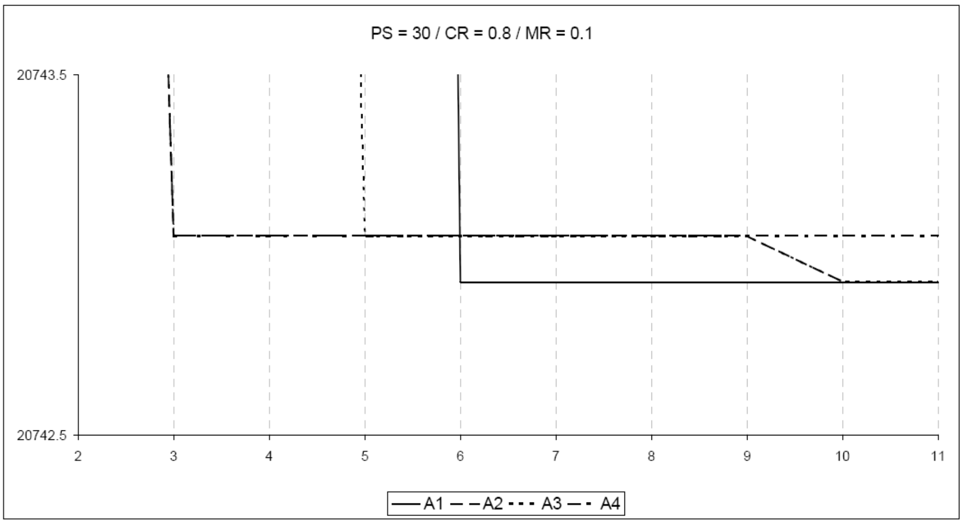
Figure A.10 Best fitness values of the generations for the 20/0.8/0.1



(a)



(b)



(c)

Figure A.11 Best fitness values of the generations for 30/0.8/0.1

Table A.10 Articles on *TrnSchPrb* in chronological order

No	Author(s) (year)	Reference(s)	Citation(s)	Problem type	Infrastructure railway	station	Objective(s)	Model structure(s)	Solution approach(es)
1	Frank(1) (1966)		(6, 7, 10, 13, 28, 55, 61, 87, 106, 135, 138, 139, 140)	railway planning problem *scheduling (timetabling)*	single tracked *line*	double tracked	to find the optimum traffic system that maximizes the traffic capacity	analytic model	solution on a set of theorems
2	Salzborn(1) (1969)		(10, 12, 55, 64)	suburban railway timetabling problem *scheduling (timetabling)*	double tracked *line*	*no clear information*	to find stop-schedules with minimum number of carriage miles or to find stop-schedules with minimum number of passenger stops	mathematical model	dynamic programming algorithm
3	Nemhauser(1) (1969)		(10, 11, 12, 55, 64)	train scheduling problem *scheduling (timetabling)*	unrestricted tracked *line*	*no clear information*	to find a schedule that yields maximum total profit	mathematical model	dynamic programming algorithm
4	Amit(1), Goldfarb(1) (1971)		(7, 10, 61, 87, 106, 138, 139)	railway timetabling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize the overall passage time of trains	nonlinear programming model	heuristic algorithm based on one train at a time
5	Szpigel(1) (1973)		(7, 10, 22, 23, 28, 34, 37, 42, 43, 53, 55, 57, 61, 62, 70, 72, 74, 79, 87, 94, 101, 103, 105, 106, 114, 115, 127, 131, 139)	train scheduling problem *scheduling (timetabling)*	single tracked *line*	double tracked	to minimize the weighted average of train travel times	linear programming model programmed in FORTRAN	branch and bound algorithm, dual simplex algorithm

6	Petersen(1) (1974)	1	(7, 10, 13, 28, 55, 57, 83, 106)	railway planning problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to estimate the congestion delays and the interaction between different types of trains	analytic model	solution on a set of linear equations
7	Wong(1), Rosser(1) (1978)	1, 4, 5, 6		train scheduling problem *scheduling (timetabling)*	single tracked *line*	double or quadruple tracked	to minimize the sum of weighted costs of delaying trains at passing loops	simulation model programmed in FORTRAN	iterative heuristic procedure based on one train at a time
8	Cury(1), Gomide(1), Mendes(1) (1979)	9	(9)	metro line scheduling problem *scheduling (timetabling)*	double tracked *line*	double tracked	to minimize the total cost	analytic model	hierarchical multilevel method
9	Cury(2), Gomide(2), Mendes(2) (1980)	8	(8, 15, 32, 41, 47, 56, 86, 106)	metro line scheduling problem *scheduling (timetabling)*	double tracked *line*	double tracked	to minimize the total cost	analytic model	hierarchical multilevel method
10	Assad(1) (1980)	1, 2, 3, 4, 5, 6	(20, 27, 28,37, 40, 55, 57, 61, 62, 64, 75, 81, 83, 87, 94, 99, 101, 102, 106, 115, 126, 135, 139)	reported on the literature models for rail transportation					
11	Petersen(2), Merchant(1) (1981)	3	(70, 74)	train scheduling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize the train operating cost	linear programming model	dynamic programming algorithm, branch and bound algorithm, heuristic algorithm

12	Assad(2) (1982)	2, 3	(55, 64)	train scheduling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize total car-hours of delay	integer programming model programmed in PL/1	heuristic algorithm, branch and bound algorithm
13	Petersen(3), Taylor(1) (1982)	1, 6	(16, 27, 28, 53, 55, 57, 61, 62, 63, 79, 87, 97, 101, 115, 135, 138, 139)	moving trains over a line problem *scheduling (timetabling)*	unrestricted tracked *line*	unrestricted tracked	to minimize the terminating times of trains	discrete event simulation model programmed in FORTRAN	heuristic algorithm based on one train at a time
14	Sauder(1), Westerman(1) (1983)		(16, 20, 22, 23, 24, 27, 28, 34, 37, 52, 55, 57, 60, 61, 62, 72, 75, 78, 87, 101, 105, 135)	train dispatching problem *rescheduling (dispatching)*	single tracked *network*	double tracked	to minimize total train delay	simulation model, linear programming model	shortest path algorithm, branch and bound algorithm
15	Araya(1), Abe(1), Fukumori(1) (1983)	9	(17, 19, 30, 31, 56, 106)	train traffic control problem *rescheduling (dispatching)*	double tracked *line*	double tracked	to minimize the length of the total delay	zero-one mixed integer programming model, linear programming model, knowledge-based model programmed in FORTRAN	rule-based heuristic algorithm, branch and bound algorithm
16	Petersen(4), Taylor(2), Martland(1) (1986)	13, 14	(22, 23, 24, 28, 34, 37, 40, 42, 53, 55, 57, 61, 79, 81, 87, 105, 115, 116, 135, 139)	train dispatching problem *rescheduling (dispatching)*	single tracked *line*	double tracked	to minimize overall train delay	mathematical model	*no clear information*

17	Fukumori(2), Sano(1), Hasegawa(1), Sakai(1) (1987)	15	(29, 36, 38, 41, 44, 47, 56)	train scheduling problem *scheduling (timetabling)*	double tracked *line*	double or triple tracked	to determine the arrival and departure times of each train at each station	discrete event model programmed in FORTRAN	diagram rough-sketching algorithm
18	Kraft(1) (1987)		(28, 42, 43, 55, 76, 77, 78, 79, 84, 94, 105)	train dispatching problem *rescheduling (dispatching)*	single or double tracked *line*	*no clear information*	to minimize weighted average of train delays	simulation model programmed in FORTRAN	branch and bound based dispatching algorithm
19	Tsuruta(1), Matsumoto(1) (1988)	15	(36, 56, 58, 59)	subway train scheduling problem *scheduling (timetabling)*	*no clear information*	*no clear information*	to generate a balanced train schedule	knowledge-based model programmed in C	rule-based approach
20	Jovanovic(1), Harker(1) (1989)	10, 14, 27	(27, 28)	train scheduling problem *scheduling (timetabling)*	single or double tracked *network*	double tracked	to find all or the prespecified number of feasible solution	mixed integer programming model programmed in PASCAL	branch and bound algorithm
21	Smith(1), Patel(1), Resor(1), Kondapalli(1) (1990)		(40)	meet/pass planning problem *rescheduling (dispatching)*	*no clear information line*	*no clear information*	to meet required running times while minimizing fuel consumption	simulation model, mixed integer nonlinear programming model	a meet/pass planning algorithm
22	Jovanovic(2), Harker(2), (1990)	5, 14, 16, 23, 28	(23, 40, 55, 61, 105, 106)	train dispatching problem *rescheduling (dispatching)*	presented a methodological framework for the role of computer-aided train-dispatching systems				

23	Jovanovic(3), Harker(3), (1991a)	5, 14, 16, 22, 28	(22, 40, 67)	train dispatching problem *rescheduling (dispatching)*	presented a methodological framework for the role of computer-aided train-dispatching systems				
24	Mills(1), Perkins(1), Pudney(1) (1991)	14, 16, 25, 28	(42, 43, 50, 52, 53, 79, 97)	train scheduling problem *rescheduling (dispatching)*	single or double tracked *line*	double tracked	to minimize the overall cost of train lateness and energy consumption	nonlinear programming model programmed in PASCAL and GAMS	discrete dynamic rescheduling algorithm based on one train at a time
25	Mees(1) (1991)		(24, 26, 33, 42, 53, 54, 57, 63, 67, 79, 97, 110, 133)	railway scheduling problem *scheduling (timetabling)*	single tracked *network*	double tracked	to minimize the total cost	integer linear programming model	modified shortest path algorithm
26	Goh(1), Mees(2) (1991)	25	(33, 53, 57, 79, 97)	train scheduling problem *scheduling (timetabling)*	single tracked *network*	double tracked	to minimize total fuel consumption	mixed integer nonlinear programming model	shortest path algorithm
27	Jovanovic(4), Harker(4) (1991b)	10, 13, 14, 20	(20, 28, 34, 37, 40, 42, 53, 55, 58, 59, 60, 62, 67, 70, 74, 75, 78, 79, 81, 83, 87, 103, 114, 115, 116, 122, 127, 131, 138, 139, 140)	train scheduling problem *scheduling (timetabling)*	single or double tracked *network*	double tracked	to find all or the prespecified number of feasible solution	mixed integer programming model programmed in PASCAL	branch and bound algorithm

28	Kraay(1), Harker(5), Chen(1) (1991)	1, 5, 6, 10, 13, 14, 16, 18, 20, 27	(22, 23, 24, 34, 37, 40, 42, 43, 50, 53, 55, 57, 61, 62, 69, 75, 78, 79, 81, 83, 87, 116, 120, 133, 135, 138, 139, 140)	train pacing problem *rescheduling (dispatching)*	single tracked *line*	double tracked	to minimize total fuel consumption	mixed integer nonlinear programming model	implicit enumeration algorithm, set generation algorithm, heuristic rounding procedure based on one train at a time
29	Iyer(1), Ghosh(1) (1991)	17, 38	(38)	railway scheduling problem *rescheduling (dispatching)*	single tracked *network*	*no clear information*	to minimize cost	simulation model programmed in C	distributed decision-making algorithm
30	Komaya(1), Fukuda(1) (1991)	15	(32, 36, 41, 44 47, 56, 58, 59, 101, 102, 106, 126, 135, 139)	railway scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	*no clear information line*	*no clear information*	to minimize total delay time	simulation model, knowledge-based model programmed in C	rule-based approach
31	Bai-gen(1), Ju-zhen(1) (1993)	15		train dispatching problem *rescheduling (dispatching)*	*no clear information*	*no clear information*	to find a feasible solution	simulation model, knowledge-based model programmed in C	rule-based approach
32	Chiang(1), Hau(1) (1993)	9, 30	(41, 47, 56)	railway scheduling problem *scheduling (timetabling)*	double tracked *line*	quadruple tracked	to generate a feasible local schedule	simulation model, knowledge-based model	rule-based approach, constraint propagation technique

33	Cai(1), Goh(2) (1994)	25, 26	(43, 50, 54, 57, 62, 63, 66, 67, 70, 74, 79, 97, 100, 103, 106, 110, 114, 123, 127, 131, 133)	train scheduling problem *scheduling (timetabling)*	single tracked *line*	double tracked	to minimize the total cost	integer programming model	heuristic algorithm based on one train at a time
34	Carey(1) (1994a)	5, 14, 16, 27, 28, 35, 37	(35, 37, 51, 55, 58, 59, 63, 81, 87, 99, 100, 101, 115, 116, 124, 125, 128, 139, 140)	train pathing problem *scheduling (timetabling)*	double or more tracked *network*	double or more tracked	to minimize the cost	zero-one mixed integer programming model programmed in GAMS	heuristic algorithm based on one train at a time, branch and bound algorithm
35	Carey(2) (1994b)	34, 37	(34, 37, 55, 58, 59, 81, 101, 115, 116, 139, 140)	train pathing problem *scheduling (timetabling)*	single tracked *network*	double or more tracked	to minimize the cost	zero-one mixed integer programming model programmed in GAMS	heuristic algorithm based on one train at a time, branch and bound algorithm
36	Lin(1), Hsu(1), (1994)	17, 19, 30	(44, 59, 97, 101)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single or double tracked *line*	*no clear information*	to minimize the delay time	knowledge-based model programmed in C	rule-based approach
37	Carey(3), Lockwood(1) (1995)	5, 10, 14, 16, 27, 28, 34, 35	(34, 35, 53, 55, 58, 59, 62, 66, 67, 68, 70, 74, 75, 80, 81, 87, 97, 101, 103, 106, 114, 115, 116, 127, 128, 131, 133, 138, 139, 140)	train pathing problem *scheduling (timetabling)*	double tracked *line*	*no clear information*	to minimize the cost	zero-one mixed integer programming model programmed in FORTRAN and GAMS/ZOOM	heuristic algorithm based on one train at a time, depth-first branching based branch and bound algorithm

38	Iyer(2), Ghosh(2) (1995)	17, 29	(29, 76, 84, 101, 126)	railway scheduling problem *rescheduling (dispatching)*	single tracked *network*	*no clear information*	to minimize cost	simulation model programmed in C	distributed decision-making algorithm
39	Schaefer(1) (1995)		(58, 59)	train dispatching problem *rescheduling (dispatching)*	single or double tracked *line*	double tracked	to minimize cost	simulation model	greedy algorithm
40	Kraay(2), Harker(6) (1995)	10, 16, 21, 22, 23, 27, 28	(55, 69, 87, 94, 101, 106, 115, 118, 138, 139, 140)	train scheduling problem *rescheduling (dispatching)*	single or double tracked *network*	triple or fivefold tracked	to minimize the time based objective function	mixed integer nonlinear programming model	simplicial decomposition algorithm, network flow solution algorithm, heuristic rounding procedure based on one train at a time, local search method
41	Chiang(2), Hau(2) (1995)	9, 17, 30, 32	(44, 47, 56, 102)	railway scheduling problem *scheduling (timetabling)*	double tracked *line*	quadruple tracked	to minimize the total running time and the start time deviation of each train	mathematical model	local search based route preprocessing algorithm, tabu search / simulated annealing and earliest-conflict -first based iterative repair algorithm

42	Higgins(1), Kozan(1), Ferreira(1) (1996)	5, 16, 18, 24, 25, 27, 28	(43, 50, 55, 62, 64, 66, 79, 83, 87, 94, 97, 110, 115, 133, 138, 139, 140)	train scheduling problem *rescheduling (dispatching)*	single or double tracked *line*	double tracked	to minimize the cost function includes fuel consumption and train delays	mixed integer nonlinear programming model programmed in FORTRAN	branch and bound algorithm, tabu search
43	Ferreira(2), Higgins(2), (1996)	5, 18, 24, 28, 33, 42		train scheduling problem *scheduling (timetabling)*	single tracked *line*	double tracked	to minimize train trip times, while maximising reliability of arrival times	mathematical model	branch and bound algorithm
44	Chiu(1), Chou(1), Lee(1), Leung(1), Leung(1) (1996)	17, 30, 36, 41	(63, 79, 97, 123)	train scheduling problem *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to minimize passenger delay and the number of station visit modifications	constraint satisfaction model programmed in C++	propagation based constraint solver, heuristic algorithm
45	Odijk(1) (1996)		(51, 55, 71, 81, 82, 89, 95, 99, 114, 116, 119, 124, 125, 127, 129, 131)	periodic railway timetabling problem *scheduling (timetabling)*	*no clear information*	multiple tracked	to find a feasible timetable structure	mathematical model programmed in PASCAL	cut generation algorithm
46	Nachtigall(1), Voget(1) (1996)		(55, 82, 99, 106, 114, 119, 125, 129, 131)	periodic railway scheduling problem *scheduling (timetabling)*	double tracked *network*	*no clear information*	to minimize waiting time for passengers changing trains	mathematical model programmed in C++	greedy heuristic , local improvement procedure, genetic algorithms
47	Chiang(3), Hau(3) (1996)	9, 17, 30, 32, 41	(56)	railway scheduling problem *scheduling (timetabling)*	double tracked *line*	quadruple tracked	to minimize the total running time and the start time deviation of each train	mathematical model	earliest-conflict -first based iterative repair algorithm

48	Kreuger(1), Carlsson(1), Olsson(1), Sjöland(1), Aström(1) (1997a)		(66, 69, 98, 123)	train scheduling problem *scheduling (timetabling)*	single tracked *network*	*no clear information*	to schedule a set of train trips over a fixed network of predetermined paths	constraint model programmed in OZ	constraint solver
49	Kreuger(2), Carlsson(2), Olsson(2), Sjöland(2), Aström(2) (1997b)		(85, 90, 91, 92, 121)	train scheduling problem *scheduling (timetabling)*	single tracked *network*	*no clear information*	to schedule a set of train trips over a fixed network of predetermined paths	constraint model programmed in OZ	constraint solver
50	Higgins(3), Kozan(2), Ferreira(3) (1997)	24, 28, 33, 42	(62, 67, 70, 74, 83, 97, 101, 102, 103, 106, 114, 127, 131, 136, 138)	train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	double tracked	to minimize the total weighted travel time	mathematical model	local search heuristic, genetic algorithms, branch and bound algorithm, tabu search, hybrid algorithms
51	Bussieck(1), Winter(1), Zimmermann(1) (1997)	34, 45	(55, 64, 79, 85, 87, 89, 90, 91, 92, 95, 96, 101, 114, 119, 121, 129)	focused on some new aspects of the planning process, and on some new results which lead to more comprehensive planning and optimization of railroad network systems					
52	Hellström(1), Frej(1), Gideon(1), Sandblad(1) (1997)	14, 24		performed a survey on a set of interviews of train dispatchers and other experts on train traffic control					
53	Ferreira(4) (1997)	5, 13, 16, 24, 25, 26, 27, 28, 37		reviewed some of the research effort designed to provide rail planners with optimization and simulation based tools to undertake train planning, train and locomotive scheduling, as well as track maintenance planning					

54	Salim(1), Cai(2) (1997)	25, 33	(101)	railway/train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	unrestricted tracked	to minimize the cost of stopping and waiting for trains	genetic model	genetic algorithms
55	Cordeau(1), Toth(1), Vigo(1) (1998)	1, 2, 3, 5, 6, 10, 12, 13, 14, 16, 18, 22, 27, 28, 34, 35, 37, 40, 42, 45, 46, 51, 60	(68, 75, 76, 77, 78, 79, 81, 83, 84, 85, 87, 89, 94, 95, 99, 101, 102, 105, 106, 107, 108, 109, 113, 114, 115, 116, 120, 121, 126, 128, 139)	presented a survey of optimization models for the most commonly studied rail transportation problems					
56	Chiang(4), Hau(4), Chiang(1), Ko(1), Hsieh(1) (1998)	9, 15, 17, 19, 30, 32, 41, 47	(98, 101, 106)	railway/train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single or double tracked *network*	multiple tracked	to minimizing the number of constraint violations and the average stopover time of trains	knowledge-based system model programmed in C	earliest-conflict -first based iterative repair algorithm
57	Cai(3), Goh(3), Mees(3) (1998)	5, 6, 10, 13, 14, 16, 25, 26, 28, 33	(101, 102, 115, 128, 139)	train scheduling problem *scheduling (timetabling)*	single tracked *network*	double tracked	to minimize the total cost due to train stopping and waiting	integer programming model	greedy heuristic algorithm based on one train at a time
58	Cheng(1) (1998a)	19, 27, 30, 34, 35, 37, 39	(101, 106, 126, 135)	train traffic rescheduling problem *rescheduling (dispatching)*	double tracked *line*	*no clear information*	to minimize the total delay of trains	simulation model	hybrid algorithm

59	Cheng(2) (1998b)	19, 27, 30, 34, 35, 36, 37, 39		train traffic control problem *rescheduling (dispatching)*	double tracked *line*	*no clear information*	to confirm the applicability of a train traffic schedule	simulation model programmed in NEXPERT OBJECT	rule-based heuristic algorithm
60	Brännlund(1), Lindberg(1), Nou(1), Nilsson(1) (1998)	14, 27	(55, 70, 74, 75, 94, 97, 101, 103, 114, 115, 127, 130, 138, 139, 140)	railway timetabling problem *scheduling (timetabling)*	single tracked *line*	double or more tracked	to maximize the total profit	integer programming model programmed in MATLAB, C and FORTRAN	Lagrangian relaxation heuristic algorithm based on one train at a time
61	Şahin(1) (1999)	1, 4, 5, 10, 13, 14, 16, 22, 28	(66, 67, 79, 87, 98, 101, 102, 105, 115, 120, 126, 128, 135, 138, 139)	railway traffic control and train scheduling problem *rescheduling (dispatching)*	single tracked *line*	double or more tracked	to minimize the sum of deviation of the expected arrival times of trains from their scheduled times within a pre-specified time horizon	zero-one mixed integer programming model, linear programming model programmed in BASIC, simulation model programmed in BASIC	heuristic algorithm based on one conflict at a time
62	Adenso-Diaz(1), Gonzalez(1), Gonzalez-Torre(1) (1999)	5, 10, 13, 14, 27, 28, 33, 37, 42, 50	(94, 102, 115, 120, 128, 138, 139)	timetable rescheduling problem *rescheduling (dispatching)*	single or double tracked *network*	*no clear information*	to maximise the number of passenger transported	mixed integer programming model	heuristic algorithm based on backtracking, branch and bound algorithm
63	Isaai(1), Singh(1) (2000)	13, 25, 33, 34, 44	(66, 67, 79, 97)	train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize total waiting time of the trains	object-oriented model programmed in BORLAND C++	constraint-based lookahead heuristic algorithm

64	Chang(1), Yeh(1), Shen(1) (2000)	2, 3, 10, 12, 42, 51	(87, 94)	train service planning problem *scheduling (timetabling)*	*no clear information line*	*no clear information*	to minimize the total operating cost and to minimize the total travel time loss	multi objective linear programming model	fuzzy approach
65	Fay(1) (2000)		(135)	train traffic control problem *rescheduling (dispatching)*	*no clear information*	*no clear information*	to ensure optimal train traffic performance and to minimize the impacts of schedule deviations	object-oriented simulation model, fuzzy Petri Net model programmed in VISUAL BASIC	fuzzy Petri Net approach integrated knowledge-based decision support system
66	Oliveira(1), Smith(1) (2000)	33, 37, 42, 48, 61, 63	(90, 91, 92, 103, 114, 133, 136)	railway scheduling problem *scheduling (timetabling)*	single or double tracked *line*	infinitive tracked	to minimize the total delay	constraint programming model	shortest processing time heuristic
67	Isaai(2), Cassaigne(1) (2001)	23, 25, 27, 33, 37, 50, 61, 63	(102)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single or double tracked *line*	double tracked	to minimize total local waiting time summed over all the trains	database model programmed in MS ACCESS	knowledge-based heuristic algorithm
68	Isaai(3), Singh(2) (2001)	37, 55	(100, 101)	train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	double tracked	to minimize total waiting time of the trains	object-oriented model programmed in C	constraint-based heuristic algorithm, hybrid of the constraint-based with tabu search, hybrid of the constraint-based with simulated annealing

69	Parkes(1), Ungar(1) (2001)	28, 40, 48	(93, 104)	decentralized train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	infinitive tracked	to maximize the total net value	mixed integer programming model programmed in CPLEX	LP-based branch and bound algorithm
70	Caprara(1), Fischetti(1), Guida(1), Monaci(1), Sacco(1), Toth(2) (2001)	5, 11, 27, 33, 37, 50, 60, 74	(80, 90, 91, 92, 93, 100, 104, 130, 140)	train timetabling problem *scheduling (timetabling)*	double tracked *line*	infinitive tracked	to maximize sum of the profits of the scheduled trains	graph theoretic model, integer linear programming model programmed in C	Lagrangian heuristic algorithm based on one train at a time
71	Peeters(1), Kroon(1) (2001)	45, 82	(82, 99, 103, 125, 127)	cyclic railway timetabling problem *scheduling (timetabling)*	multiple tracked *network*	*no clear information*	to minimize halting and transfer times	mixed integer programming model programmed in CPLEX	branch and bound algorithm
72	Ping(1), Axin(1), Limin(1), Fuzhang(1) (2001)	5, 14	(101, 102, 126)	train dispatching problem *rescheduling (dispatching)*	double tracked *line*	*no clear information*	to minimize total delay time	genetic model, simulation model	genetic algorithms
73	Pacciarelli(1), Pranzo(1) (2001)		(101, 102, 106, 136)	railway scheduling problem *scheduling (timetabling)*	single tracked *network*	double tracked	to find a feasible solution	alternative graph model	tabu search
74	Caprara(2), Fischetti(2), Toth(3) (2002)	5, 11, 27, 33, 37, 50, 60	(70, 89, 93, 101, 103, 110, 114, 115, 125, 127, 129, 130, 131, 139, 140)	train timetabling problem *scheduling (timetabling)*	double tracked *line*	infinitive tracked	to maximize sum of the profits of the scheduled trains	graph theoretic model, integer linear programming model programmed in C	Lagrangian heuristic algorithm based on one train at a time

75	Newman(1), Nozick(1), Yano(1) (2002)	10, 14, 27, 28, 37, 55, 60	(115)	described optimization problems that arise in the rail industry					
76	Medanic(1), Dorfman(1) (2002a)	18, 38, 55	(77, 78, 84)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single or double tracked *line*	*no clear information*	to obtain suboptimal time-efficient schedules	discrete event model	greedy travel advance strategy
77	Medanic(2), Dorfman(2) (2002b)	18, 55, 76	(78, 84)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to obtain suboptimal energy-efficient schedules	discrete event model	greedy travel advance strategy
78	Medanic(3), Dorfman(3) (2002c)	14, 18, 27, 28, 55, 76, 77	(84)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to obtain suboptimal time-efficient and energy-efficient schedules	discrete event model	greedy travel advance strategy
79	Brucker(1), Heitmann(1), Knust(1) (2002)	5, 13, 16, 18, 24, 25, 26, 27, 28, 33, 42, 44, 51, 55, 61, 63		train rescheduling problem *rescheduling (dispatching)*	single tracked *line*	double tracked	to minimize the maximum lateness or to minimize the weighted maximum lateness	graph model programmed in C	reaching algorithm, iterative improvement, tabu seach

80	Kwan(1), Mistry(1) (2003)	37, 70	(93, 104, 131)	train timetabling problem *scheduling (timetabling)*	*no clear information network*	double tracked	to minimize the weighted sum of violations expressed in time units	evolutionary model	co-operating co-evolutionary algorithm, simulated annealing
81	Carey(4), Carville(1) (2003)	10, 16, 27, 28, 34, 35, 37, 45, 55	(98, 99, 101, 114, 116, 124, 128, 140)	train platforming problem *scheduling (timetabling)*	multiple tracked *no clear information*	multiple tracked	to minimize cost of deviations	set of modules programmed in C	heuristic algorithm based on one train at a time
82	Kroon(2), Peeters(2) (2003)	45, 46, 71	(71, 99, 114, 115, 119, 125, 127, 131, 140)	cyclic railway timetabling problem *scheduling (timetabling)*	multiple tracked *network*	*no clear information*	to obtain a feasible timetable	periodic event scheduling model	decision support system
83	Crainic(1) (2003)	6, 10, 27, 28, 42, 50, 55	(101)	presented the main freight transportation planning and management issues, briefly reviewed the associated literature, described a number of major developments, and identified trends and challenges					
84	Dorfman(4), Medanic(4) (2004)	18, 38, 55, 76, 77, 78	(98, 101, 115, 120, 128, 139)	train scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	single or double tracked *network*	double tracked	to obtain suboptimal time-efficient and energy-efficient schedules	discrete event model	greedy travel advance strategy
85	Ingolotti(1), Barber(1), Tormos(1), Lova(1), Salido(1), Abril(1) (2004)	49, 51, 55	(101)	train scheduling problem *rescheduling (dispatching)*	single tracked *network*	*no clear information*	to minimize the traversal time of each new train	model programmed in C ++	iterative based sequential algorithm

86	Assis(1), Milani(1) (2004)	9	(132, 134)	metro line train time scheduling problem *scheduling (timetabling)*	double tracked *line*	double tracked	to minimize the performance index	linear programming model, predictive control formulation	solution on a software package
87	Ghoseiri(1), Szidarovszky(1), Asgharpour(1) (2004)	1, 4, 5, 10, 13, 14, 16, 27, 28, 34, 37, 40, 42, 51, 55, 61, 64	(101, 106, 138, 139)	multi objective train scheduling problem *scheduling (timetabling)*	single or multiple tracked *network*	multiple tracked	to lower the fuel consumption cost and to shorten the total passenger time	mathematical programming model	pareto frontier and distance based algorithm
88	Wikström(1), Kauppi(1), Hellström(2), Andersson(1), Sandblad(2) (2004)			train traffic control problem *rescheduling (dispatching)*	*no clear information*	*no clear information*	to minimize the overall delay	simulation model, questionnaire	control by re-planning strategy
89	Ingolotti(2), Tormos(2), Lova(2), Barber(2), Salido(2), Abril(2) (2004)	45, 51, 55, 74	(136)	railway scheduling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize average traversal time	mixed integer programming model	constraint satisfaction integrated solver process module of decision support system
90	Barber(3), Salido(3), Ingolotti(3), Abril(3), Lova(3), Tormos(3) (2004)	49, 51, 66, 70	(91, 92, 100)	train scheduling problem *scheduling (timetabling)*	single or double tracked *network*	double or more tracked	to obtain a correct and optimized running map	mixed integer programming model, linear programming model	heuristics, constraint solver

91	Salido(4), Abril(4), Barber(4), Ingolotti(4), Tormos(4), Lova(4) (2004)	49, 51, 66, 70, 90		periodic train scheduling problem *scheduling (timetabling)*	single or double tracked *network*	double or more tracked	to minimize the journey time of all trains	mixed integer programming model, linear programming model	topological constraint optimization technique
92	Salido(5), Barber(5), Abril(5), Tormos(5), Lova(5), Ingolotti(5) (2004)	49, 51, 66, 70, 90		periodic train scheduling problem *scheduling (timetabling)*	single or double tracked *network*	double or more tracked	to minimize the journey time of all trains	mixed integer programming model, linear programming model	topological constraint optimization technique
93	Semet(1), Schoenauer(1) (2005)	69, 70, 74, 80	(104)	train timetabling problem *rescheduling (dispatching)*	multiple tracked *network*	multiple tracked	to minimize the total accumulated delay	mixed integer programming model programmed in CPLEX, evolutionary model	branch and bound algorithm, evolutionary algorithm, hybrid greedy algorithm based on one train at a time
94	Zhou(1), Zhong(1) (2005)	5, 10, 18, 40, 42, 55, 60, 62, 64	(101, 115, 136, 139)	bicriteria train scheduling problem *scheduling (timetabling)*	double tracked *line*	*no clear information*	to minimize the variation of interdeparture times for high-speed trains and to minimize the total travel time	integer programming model programmed in VISUAL C ++ 6.0 and GAMS/ZOOM	train-based breadth-first search included branch and bound algorithm with effective dominance rules, beam search algorithm with utility evaluation rule and with random selection rule

95	Lindner(1), Zimmermann(2) (2005)	45, 51, 55	(127, 136)	periodic train scheduling problem *scheduling (timetabling)*	*no clear information network*	*no clear information*	to minimize cost assignment of different train types	graph model, mixed integer linear programming model programmed in CPLEX	decomposition based branch and bound algorithm
96	Rebreyend(1) (2005)	51	(135)	train dispatching problem *rescheduling (dispatching)*	single or double tracked *network*	*no clear information*	to minimize the number of delayed trains	event based simulation model programmed in C	genetic algorithms, branch and bound algorithm
97	Isaai(4) (2005)	13, 24, 25, 26, 33, 36, 37, 42, 44, 50, 60, 63		train scheduling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize total local waiting time summed over all the trains	object-oriented model	hybrid algorithm combination of the lookahead constraint-based heuristic with simulated annealing
98	Yalçınkaya(1), Bayhan(1) (2005)	48, 56, 61, 81, 84		metro line scheduling problem *scheduling (timetabling)*	double tracked *line*	double or triple tracked	to minimize the average passenger travel time and to reach fifty percent fullness rate of the carriages	discrete event based simulation model programmed in ARENA, simulation metamodel	response surface methodology, desirability functions based multi response optimization procedure, genetic algorithms
99	Huisman(1), Kroon(3), Lentink(1), Vromans(1) (2005)	10, 34, 45, 46, 55, 71, 81, 82	(114, 124)	gave an overview of operations research models and techniques used in passenger railway transportation					

100	Chang(1), Chung(1) (2005)	33, 34, 68, 70, 90	(101)	train timetabling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	double tracked *line*	*no clear information*	to minimize the average travel time of passengers and to maximize the utilization of trains	genetic model	genetic algorithms
101	Törnquist(1) (2005)	5, 10, 13, 14, 30, 34, 35, 36, 37, 38, 40, 50, 51, 54, 55, 56, 57, 58, 60, 61, 68, 72, 73, 74, 81, 83, 84, 85, 87, 94, 100	(126)	railway traffic scheduling problem *scheduling (timetabling)* and *rescheduling (dispatching)*	reviewed the researches carried out within the area of railway scheduling and dispatching				
102	Törnquist(2), Persson(1) (2005)	10, 30, 41, 50, 55, 57, 61, 62, 67, 72, 73	(126)	train rescheduling problem *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to minimize the total delay for the trains and to minimize the costs due to the different delays	mixed integer programming model and linear programming model programmed in AMPL and CPLEX, simulation model programmed in JAVA	simulated annealing, tabu search, branch and bound algorithm

103	Caprara(3), Monaci(2), Toth(4), Guida(2) (2006)	5, 27, 33, 37, 50, 60, 66, 71, 74	(114, 127, 130, 131, 140)	train timetabling problem *scheduling (timetabling)*	double tracked *line*	double or more tracked	to maximize sum of the profits of the scheduled trains	graph theoretic model, integer linear programming model programmed in C	Lagrangian heuristic included traffic capacity management algorithm based on one train at a time
104	Semet(2), Schoenauer(2) (2006)	69, 70, 80, 93		train rescheduling problem *rescheduling (dispatching)*	multiple tracked *network*	multiple tracked	to minimize the total accumulated delay	mixed integer programming model programmed in CPLEX, evolutionary model	branch and bound algorithm, evolutionary algorithm
105	Khan(1), Zhang(1), Jun(1), Li(1) (2006)	5, 14, 16, 18, 22, 55, 61		train rescheduling problem *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to minimize the total delay of trains	genetic model	genetic algorithms
106	Ghoseiri(2), Morshedsolouk(1) (2006)	1, 4, 5, 6, 9, 10, 15, 22, 30, 33, 37, 40, 46, 50, 55, 56, 58, 73, 87		train scheduling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize the total train delays in the stations	mathematical model	ant colony system
107	Ingolotti(6), Lova(6), Barber(6), Tormos(6), Salido(6), Abril(6) (2006)	55, 121	(131)	railway timetabling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize the delay and time based deviation	constraint satisfaction model programmed in C ++	train priorities based decision support system

108	Ingolotti(7), Barber(7), Tormos(7), Lova(7), Salido(7), Abril(7) (2006)	55, 121	train timetabling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize the average deviation	model programmed in C ++	track priorities based irrevocable heuristic-driven search included branch and bound algorithm
109	Abril(8), Salido(8), Barber(8), Ingolotti(8), Tormos(8), Lova(8) (2006)	55, 121	railway scheduling problem *scheduling (timetabling)*	single tracked *network*	double or more tracked	to minimize the journey time of all trains	distributed constraint satisfaction model	constraint solver, graph partitioning
110	Su(1), Huang(1) (2006)	25, 33, 42, 74	train timetabling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to minimize sum of headway difference	time-space diagram model programmed in MICROSOFT VISUAL C ++.NET	ant colony
111	Kauppi(2), Wikström(2), Sandblad(3), Andersson(1) (2006)		train traffic control problem *rescheduling (dispatching)*	single tracked *line*	*no clear information*	to minimize train delays	simulation model	control by re-planning strategy
112	Geske(1) (2006)		railway scheduling problem *scheduling (timetabling)*	single tracked *network*	double or triple tracked	to reduce the lateness of trains	constraint based deterministic simulation model	one train at a time based heuristic, genetic algorithms

113	Salido(9), Abril(9), Barber(9), Ingolotti(9), Tormos(9), Lova(9) (2007)	55		railway scheduling problem *scheduling (timetabling)*	single tracked *network*	double or more tracked	to minimize the journey time of all trains	distributed constraint satisfaction model	graph partitioning, constraint solver
114	Caprara(4), Kroon(4), Monaci(3), Peeters(1), Toth(5) (2007)	5, 27, 33, 37, 45, 46, 50, 51, 55, 60, 66, 74, 81, 82, 99, 103	(127, 139)	focused on passenger transportation in European, and surveyed operational planning problems such as; line planning, timetabling, platforming, rolling stock circulation, shunting, and crew planning problems					
115	Zhou(2), Zhong(2) (2007)	5, 10, 13, 16, 27, 34, 35, 37, 40, 42, 55, 57, 60, 61, 62, 74, 75, 82, 84, 94	(138, 139, 140)	train timetabling problem *scheduling (timetabling)*	single tracked *line*	multiple tracked	to minimize the total train travel time	integer programming model programmed in VISUAL C ++	branch and bound algorithm, longest path algorithm, Lagrangian relaxation approach, beam search algorithm
116	Carey(5), Crawford(1) (2007)	16, 27, 28, 34, 35, 37, 45, 55, 81	(128, 140)	train scheduling problem *scheduling (timetabling)*	single or multiple tracked *line*	multiple tracked	to minimize cost of deviations	set of modules programmed in C	heuristic algorithms based on one train at a time
117	Rodriguez(1) (2007)		(140)	train scheduling problem *rescheduling (dispatching)*	*no clear information*	*no clear information*	to minimize the sum of delays	simulation model, constraint programming model	branch and bound algorithm, decision support system

118	Mazzarello(1), Ottaviani(1) (2007)	40	(137)	real-time traffic regulation problem *rescheduling (dispatching)*	quadruple tracked *network*	triple tracked	to minimize delays	alternative graph model	heuristic algorithms based on one train at a time
119	Liebchen(1), Möhring(1) (2007)	45, 46, 51, 82	(125, 129)	periodic railway timetabling problem *scheduling (timetabling)*	single or more tracked *network*	double or more tracked	to obtain a feasible periodic timetable	graph theoretic model, mixed integer linear programming model	*no clear information*
120	D'Ariano(1), Pranzo(2), Hansen(1) (2007)	28, 55, 61, 62, 84, 126	(137)	train dispatching problem *rescheduling (dispatching)*	quadruple tracked *network*	double or more tracked	to minimize the maximum delay caused by conflicts	alternative graph model	iterative rescheduling algorithm, dispatching rules, greedy heuristic, branch and bound algorithm
121	Lova(10), Tormos(10), Barber(10), Ingolotti(10), Salido(10), Abril(10) (2007)	49, 51, 55	(107, 108, 109)	train scheduling problem *rescheduling (dispatching)*	single or double tracked *network*	*no clear information*	to minimize the traversal time of each new train	model programmed in C ++	iterative based sequential algorithm
122	Luethi(1), Weidmann(1), Laube(1), Medeossi(1) (2007)	27		train rescheduling and control problem *rescheduling (dispatching)*	single, double or triple tracked *network*	multiple tracked	to minimize the total delay of all trains	simulation model	iterative based process comparing algorithm

123	Mladenović(1), Čangalović(1) (2007)	33, 44, 48		train rescheduling problem *rescheduling (dispatching)*	single tracked *line*	double or more tracked	to minimize maximum tardiness or to minimize maximum weighted tardiness or to minimize total tardiness or to minimize total weighted tardiness or to minimize maximum slack of trains in stations or to minimize makespan or to minimize the number of late trains	constraint satisfaction model programmed in OPL	bound heuristic, separation heuristics, search heuristics, constraint solver
124	Caimi(1), Burkolter(1), Herrmann(1), Chudak(1), Laumanns(1) (2007)	34, 45, 81, 99	(125, 137)	train scheduling problem *scheduling (timetabling)*	single, double, triple or quadruple tracked *network*	multiple tracked	to minimize time loose	independent set in a conflict graph model, integer linear programming model	fixed-point iteration based heuristic
125	Caimi(2), Fuchsberger(1), Laumanns(2), Schüpbach(1) (2007)	34, 45, 46, 71, 74, 82, 119, 124		periodic train timetabling problem *scheduling (timetabling)*	single or double tracked *network*	*no clear information*	to minimize a weighted sum of the passenger times or to optimize both the quality of the timetable and the time slots	mixed integer linear programming model programmed in MATLAB and MOSEK	solver

126	Törnquist(3), Persson(2) (2007)	10, 30, 38, 55, 58, 61, 72, 101, 102	(120)	railway traffic rescheduling problem *rescheduling (dispatching)*	single, double or n-tracked *network*	quadruple tracked	to minimize the total final delay of the traffic or to minimize the total cost associated with delays	mixed integer linear programming model programmed in AMPL and CPLEX	branch and bound algorithm, solver
127	Cacchiani(1), Caprara(5), Toth(6) (2008)	5, 27, 33, 37, 45, 50, 60, 71, 74, 82, 95, 103, 114	(130)	train timetabling problem *scheduling (timetabling)*	single tracked *line*	*no clear information*	to maximize sum of the profits of the scheduled trains	graph theoretic model, integer linear programming model programmed in C and CPLEX	column generation and one train at a time based heuristic algorithm, branch and cut and price algorithm
128	Flamini(1), Pacciarelli(2) (2008)	34, 37, 55, 57, 61, 62, 81, 84, 116	(132, 134)	train routing/ sequencing problem, metro terminus scheduling problem *rescheduling (dispatching)*	*no clear information*	multiple tracked	to minimize of the sum of total tardiness plus total earliness and to minimize of the difference between the off-line and the actual headway	alternative graph model, simulation model programmed in C++	one train at a time based heuristic algorithm, polynomial time algorithm
129	Liebchen(2) (2008)	45, 46, 51, 74, 119		periodic railway timetabling problem *scheduling (timetabling)*	double tracked *network*	double or more tracked	to minimize the weighted sum of passenger waiting times and to minimize the number of trains that is required to operate the timetable	graph model, integer programming model programmed in CPLEX	simple approximation algorithm, cut heuristic, genetic algorithms

130	Fischer(1), Helmberg(1), Janssen(1), Krostitz(1), (2008)	60, 70, 74, 103, 127	train timetabling problem *scheduling (timetabling)*	*no clear information network*	*no clear information*	to obtain a feasible timetable	graph model, integer linear programming model programmed in C ++ and CPLEX and ConicBundle, simulation model	Lagrangian relaxation algorithm, bundle cutting plane approach, rounding heuristic
131	Tormos(11), Lova(11), Barber(11), Ingolotti(11), Abril(11), Salido(11) (2008)	5, 27, 33, 37, 45, 46, 50, 74, 80, 82, 103, 107	train timetabling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize the average delay of the new trains	genetic model	iterative heuristic based on random sampling methods, genetic algorithms
132	Yalçınkaya(2), Bayhan(2) (2008)	86, 128, 134	metro line scheduling problem *scheduling (timetabling)*	double tracked *line*	double or triple tracked	to minimize the average passenger travel time and to reach a satisfactory carriage fullness rate	discrete event based simulation model programmed in ARENA, simulation metamodel	response surface methodology, desirability functions based multi response optimization procedure
133	Nagarajan(1), Ranade(1) (2008)	25, 28, 33, 37, 42, 66	single train pathing problem *scheduling (timetabling)*	*no clear information network*	*no clear information*	to minimize the new train's journey time without affecting the schedules for the old trains	graph model	shortest path based train path algorithm

134	Yalçınkaya(3), Bayhan(3) (2009)	86, 128	(132)	metro line scheduling problem *scheduling (timetabling)*	double tracked *line*	double or triple tracked	to minimize the average passenger travel time and to reach a satisfactory carriage fullness rate	discrete event based simulation model programmed in ARENA, simulation metamodel	response surface methodology, desirability functions based multi response optimization procedure
135	Cheng(1), Yang(1) (2009)	1, 10, 13, 14, 16, 28, 30, 58, 61, 65, 96		railway traffic control problem *rescheduling (dispatching)*	*no clear information network*	*no clear information*	to minimize the total passenger delay	fuzzy Petri Net model	dispatching decision support system
136	Liu(1), Kozan(3) (2009)	50, 66, 73, 89, 94, 95		train scheduling problem *scheduling (timetabling)*	single tracked *network*	double or triple tracked	to minimize the makespan	alternative graph model programmed in VISUAL C++	improved shifting bottleneck procedure includes; the topological-sequence algorithm, a modified Carlier algorithm, an extended algorithm based on Jackson rules, tabu search and simulated annealing, feasibility satisfaction procedure based on one train at a time, insertion algorithm

137	Luethi(2), Medeossi(2), Nash(1) (2009)	118, 120, 124	real-time train rescheduling problem *rescheduling (dispatching)*	single, double or triple tracked *network*	multiple tracked	to minimize the total delay of all trains	simulation model	two-loop real-time rescheduling approach
138	Yang(1), Li(1), Gao(1) (2009)	1, 4, 13, 27, 28, 37, 40, 42, 50, 60, 61, 62, 87, 115	train timetabling problem *scheduling (timetabling)*	single tracked *line*	double or more tracked	to minimize the total passengers' time and to minimize the the total delay time	fuzzy mixed integer goal programming model	fuzzy simulation based branch and bound algorithm
139	Castillo(1), Gallego(1), Ureña(1), Coronado(1) (2009)	1, 4, 5, 10, 13, 16, 27, 28, 30, 34, 35, 37, 40, 42, 55, 57, 60, 61, 62, 74, 84, 87, 94, 114, 115	train timetabling problem *scheduling (timetabling)*	single or double tracked *line*	*no clear information*	to minimize the relative travel times of all trains and determine the train priorities and to minimize the sum of the train entry times to all segments and to minimize total dwell times of all trains	mixed integer linear programming model programmed in CPLEX and GAMS	iteration based bisection method included three step heuristic algorithm
140	Lee(1), Chen(1) (2009)	1, 27, 28, 34, 35, 37, 40, 42, 60, 70, 74, 81, 82, 103, 115, 116, 117	train pathing and timetabling problem *scheduling (timetabling)*	single or double tracked *line*	multiple tracked	to minimize the sum of weights of tracks assigned to all services at all stations and to minimize the sum of the difference between the services' scheduled departure time and the target departure time	binary integer programming model programmed in CPLEX and C++, linear programming model programmed in CPLEX and C++	iteration based four step heuristic algorithm, threshold accepting heuristic