ONLINE RESCHEDULING FOR MASS RAPID TRANSIT SYSTEM USING EVOLUTIONARY TECHNIQUES WITH FUZZY AGGREGATION OF MULTIPLE OBJECTIVES

WANG ZHENYU

NATIONAL UNIVERSITY OF SINGAPORE

2005

ONLINE RESCHEDULING FOR MASS RAPID TRANSIT SYSTEM USING EVOLUTIONARY TECHNIQUES WITH FUZZY AGGREGATION OF MULTIPLE OBJECTIVES

WANG ZHENYU

(B.ENG)

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

ACKNOWLEDGEMENTS

The author would like to express his most sincere gratitude to his supervisor, Associate Professor C.S Chang for his invaluable guidance, advice and encouragement throughout the course of the project.

The author would also like to thank all research scholars/students in the power system lab for their assistance in the duration of the project.

Thanks are also given to the Power System Laboratory Technician Mr. S.H. Seow, for his help and assistance throughout this research project.

Last but not least, I would like to thank my friends and all those, who have helped me in one way or another.

PAPER WRITTEN ARISING FROM WORK IN THIS THESIS

C.S. Chang, ZY. Wang, "Multi-objective Optimization for Conflict Resolution of Mass Rapid Transit System", Proceeding of IEE International Conference on Railway Engineering Development into 21 st Century, Hong Kong, March 15-20, 2005

TABLE OF CONTENTS

SUMMARY

Rescheduling strategies integrated in railway traffic control system play a crucial role in improving the performance of a mass rapid transit (MRT) system in terms of passenger service and operation cost. Without any form of rescheduling activities, the performance of a MRT system, which operates according to pre-determined time-tables, is prone to degradation caused by operational disturbances. In contrast, if an intelligent rescheduling system is implemented, the effect of these disturbances can be avoided or reduced to an acceptable level.

The purpose of this thesis is to present a new on-line rescheduling strategy, in which Differential Evolution (DE) with fuzzy logic is combined for multi-objective optimisation for conflict resolution as well as maintaining optimal overall performance. A simulation software package integrated with decision support systems, called Automatic Train Regulator (ATR), is developed to evaluate performance of the proposed strategy for the study of MRT systems under various scenarios. The proposed optimization is based on the passenger-flow profiles of a typical medium-sized mass transit system.

ATR is a complex real-time system, which is programmed in multithreaded mode to simulate the train movement and traffic control systems that run in parallel at the two levels, namely: Local Processing Units (LPU) and Operation Center Control (OCC). Object-oriented techniques are used to simulate the operations of the MRT system. The proposed strategy divides the study/control period into time windows of equal length, which slides from one time window to another according to a real-time clock. Using real-time information of the MRT system, ATR predicts the timing of each

inter-station run and other performance indices occurring in the present time window. According to the latest simulation results, ATR executes a DE-optimized train rescheduler to adjust all the train dispatch and dwell times, and train coasting levels of all inter-station runs within each time window. The DE-optimized train rescheduler also identifies potential conflict scenarios for fast, robust and near-optimal solutions. The traffic condition of each potential conflict is represented by four fuzzy performance indices, namely [2]: (i) regularity, (ii) overall energy consumption, (iii) platform congestion and (iv) train congestion, which also form the basis of train rescheduling. Besides DE, its counterpart, Genetic Algorithm (GA) is applied to ATR as a comparative study to prove DE's high efficiency on solving the online optimization problems.

The simulation tests reveal that the proposed strategy implemented in ATR has a great potential for online detecting and resolving conflicts within strict time constraints as well as maintaining a satisfactory overall operational performance under both normal and disturbed running conditions.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The first section of this chapter briefs the background of this research. The second section reviews the previous works on the train control and scheduling problems. In the third section, the objectives of the research and new techniques adopted are listed. Lastly, the organization of the thesis is given.

1.1 Background and Aims of Study

Mass rapid transit (MRT) system services have increasingly become more competitive. Hence, they demand more new features in the planning process and online real-time rescheduling operations. The increasing traffic density on railways desires sophisticated operation control systems, which provide intelligent tools to help operators to meet changes in passenger demand.

With the advancement in communication network and computer technology, efficient and easily represented evolutionary computing techniques, such as Differential Evolution (DE), can be applied to the realm of on-line rescheduling of MRT systems to solve some complicated multi-objective optimisation problems. This research project aims to devise an on-line rescheduling strategy to optimize passenger service with low electrical energy cost, which is one of the major components of the overall MRT operating costs.

1.2 Review of Literature

In recent years, the vast processing power available in the distributed computing environment has been used to assist railway engineers in the design of new services and the operation of existing systems. Noteworthy contributions have been made in this area.

Taskin and Allan [1] from the London Underground Limited and University of Birmingham explore the ongoing Service Regulation subsystem development for the

Central Line. Their study discusses the feasibility of designing a Service Regulation subsystem that reflects the passenger flow dynamics of the line, as well as the signaling system characteristics. It identifies the requirements for a Service Regulation subsystem as represented by a set of cost functions. These cost functions penalize the delays to the nominal passenger journey times of a maximum number of passengers.

Yasunobu and his colleagues [2] developed a predictive fuzzy controller (PFC) for automatic train operation (ATO) that controls the train's departure, speed regulation and stopover at target points at each station. A microcomputer with built-in program composed of the control rules established based on skilled operators' experience and their evaluation method was incorporated in the on-board equipment. Shinji and Sone [3] proposed an online traffic control system for modifying predetermined schedules. The control scheme is accomplished by adjusting station stopover times and/or inter-station running times to ensure the regularity of trains and quality of passenger service. Chua [4] devised a traffic control algorithm, which operates within the constraints of minimum run times, minimum stopover times and minimum headways to recover the train service from disturbance. In his design, the source of the train delays is identified and the regularity adjustments are carried out in each of the three regions. The on-line control instructions are to be conveyed to the trains at the stations based on the sources of train delay. Chang and Quek [5] developed a simulation package that employs a dwell time and dispatch frequency controller which makes use of fuzzy decision-making to determine the optimal schedules for train dwell times and dispatch intervals based on the criteria of regularity of service, energy consumption, train congestion as well as platform congestion levels. These railway control schemes in terms of individual train control and station-based decision-making were most frequently applied when train service providers update existing systems during the period of transition from manual to automatic operation.

Apart from train automatic operations, train scheduling is one of the most challenging problems in railway traffic control. In the traditional approach, the train traffic scheduling was often analysed by mathematical modelling. Higgins and E.Kozan [6] have studied the problem of dispatching freight trains in a single-line track aiming to minimize the total weighted travel times. They formulated a mixed integer linear problem in which the arrival and departure times are modelled as continuous decision variables and the conflicts resolution as binary variables. Local search heuristics, genetic algorithms, tabu search, and related hybrid algorithms were developed to optimise a train schedule on a single line.

Cai and Goh [7] proposed a greedy construction heuristic for timetabling and dispatching trains on a single-track railroad. The heuristic approach is capable of resolving the conflicts of planned schedule by selecting the train(s) to traverse the conflict segment according to a greedy local criterion.

Raymond and Mistry [8] applied a co-evolutionary algorithm for an automatic generation of planning train timetables. The departure time, scheduled runtimes and resource option at a station are identified as key solution variables. Based on co-operative co-evolution, the train timetabling problem is decomposed into sub-components that are evolved by a separate evolutionary algorithm. A collaborative schemes is operated between the co-evolved species so that solution from each component can be pooled together to derive a better evaluation than if members of species were evaluated separately. Kraay and Harker [9] proposed a model for optimising freight train schedules over the entire rail network with the intention of using them as part of a real-time control system. Its goal was to help coordinate train dispatchers by determining the target time for each train at major points of its itinerary. Ho and Norton [10] applied dynamic programming to an event-based traffic flow model of fixed-block signalling system to realize a delay-optimized traffic controller for a converging railway junction to resolve conflicts. Brannlund [11] proposed an integer programming model to obtain a profit maximizing timetable in which profit was measured by estimating the value of running different services at specified times, and solve it using Lagrangian relaxation.

Simulation analysis approach can be well applied to solve problems that do not have analytical and mathematical solutions, and makes it possible to study the complex internal interactions of the system being simulated. It is now also used as a decision support tool in some facets of transportation engineering. Chang [5, 12] applied different computational intelligence methods, such as fuzzy logic and tabu search to a computer-based simulation for testing and evaluating their performance in optimizing the traffic schedule for MRTS.

From the review of the past works, it can be summarized that most of the control schemes in the area of automatic train control were applied to individual train or station only. Although these conventional automatic train controllers are able to add some intelligence to train operations to reduce human interference, their lack of a system view cannot ensure the optimality of the overall system performance. As discussed, the studies of train scheduling problem often adopted the mathematical modelling approach to optimise train timetable using various optimisation techniques in an offline mode to get a predetermined timetable. This approach can only achieve an optimal system performance in a theoretical environment without external disturbance. To date, no work has been done to integrate the automatic train controller and train scheduling into a live system to do an online optimisation, which takes into account the various disturbances that may arise. This thesis proposes an online rescheduling algorithm which optimises the dynamic working timetable using Evolutionary algorithms with the aid of the existing train control schemes to achieve an optimised operational performance for the whole MRT line.

1.3 Objective of the Study and New Technique Adopted

This project aims to devise and test an on-line rescheduling strategy that optimizes passenger service with low energy cost on a simulated real-time environment. In the previous work done by Quek, a dwell time and dispatch frequency controller using fuzzy decision-making was employed in each station to determine the optimal train dwell time and dispatch interval for individual train dwelling at the station. This decomposition approach that deals with one train at a time can lead to good or near-optimal solutions for each train. However, this may also result in a poor global solution since the schedule of trains is made independently and does not consider its impact of conflicting schedules. In this project, a new approach with a system level view is introduced. The active timetable of the whole track line under study is observed and optimised as a background task to provide online operational parameters for the foreground train movement simulation task. A new simulation method, multithreaded programming, is adopted in a simulated ATR system to accommodate the proposed approach. In Quek's work, four fuzzy performance indices were calculated simply according to their corresponding membership functions to evaluate the operational performance of the MRT system. The final decision-making consists of evaluating 30 potential solutions. From each potential solution, the minimum (worst case) value among the four performance indices is used for comparison with other solutions. The dwell time obtained from the solution with the highest minimum value is considered the optimal dwell time. The simplified optimisation process may was not able to fully achieve a satisfactory overall performance. In this thesis, all four performance indices and fuzzy inference rule base form a fuzzy expert evaluation system. A complete fuzzy logic control process including fuzzification, rule inference, and defuzzification is performed to aggregate all four performance indices into a fitness value that reflects the expert's evaluation of the corresponding solution. With the fitness value from fuzzy evaluation system, EA-based optimization can identify the near optimal solution from thousands of other potential solutions in the system level solution space. Finally, a set of optimized control variables is generated as inputs of active timetable for real time operations.

In summary, the objectives of this research are listed below:

1. To build a comprehensive real-time simulation, which performs all the important functions of the real MRT system and provides the platform for the research's analytical work under different scenarios. In order to make the simulation truly reflect the nature of the real system, the multithreading technique [12] should be applied to simulate the train movement and traffic

control systems that run in parallel at the two levels, namely: Local Processing Units (LPU) and Operation Center Control (OCC).

2. To propose online rescheduling method that is able to perform the functions of scenario analysis and optimization through maintaining an Active Timetable, which includes planning information of all trains along the track line and is updated online by the rescheduler.
3. To identify possible conflicts in advance and to solve these conflicts in an optimal way so that passenger service can be optimized while at the same time, a low operational cost can be maintained.
4. To apply a feasible optimization method to solve the multi-objective optimization problem based on the analysis of Active Timetable. Evolutionary algorithms (EA) such as Genetic Algorithm (GA) and Differential Evolution (DE) are applied to do the optimization search. Inferred by a series of adjustable fuzzy rules, the performance indices of regularity, energy, and train and platform congestion are mapped into an overall performance index which functions as the fitness value of EAs.

1.4 Scope of the Thesis

This thesis comprises of seven chapters, which are briefly described as follows:

Chapter 1 introduces the previous works on the train control and scheduling problems, as well as the objectives of the research and new techniques adopted.

Chapter 2 gives an overview of the MRT system and a brief description of automatic signaling and train control systems as they play a key role in the satisfactory performance of MRT operations.

Chapter 3 describes two train-rescheduling algorithms at the Operational Control Centre and Local Process Units in ATR.

Chapter 4 describes the development of Automatic Train Regulator (ATR) simulation software, which is used to simulate the train movement and train control system. The software is programmed in Visual C++ 6.0 environment.

Chapter 5 develops Evolutionary Algorithms such as Genetic Algorithms and Differential Evolution with a fuzzy fitness in the ATR controller.

Chapter 6 discusses the simulation results obtained after optimization for the normal and disturbed running conditions of train operation.

Chapter 7 summarizes the main achievements of the research. Recommendations for future work are given.

CHAPTER 2

BASICS OF MASS RAPID TRANSIT SYSTEM

This Chapter provides a general study of the basics of automatic train control in existing MRT system. The overall structure and functional components of automatic signaling and train control system are illustrated. The signaling scheme and mechanism of Automatic Train Control are discussed.

2.1 General Introduction to Mass Rapid Transit System

The overall development of modern metropolitan cities around the world in terms of population, trade and commerce has increased the traffic tremendously over the years causing congestion, delays and pollution, which in turn will adversely affect the future economic growth of the city. This situation has made policy makers of many developing countries envisage the need of an efficient, economical, equitable and environment friendly MRT system for achieving sustainable city development.

The MRT system, or metro system, is reputed for its capability to upgrade the quality of life of a highly populated city's inhabitants by providing significant improvements as such as: a) to provide a convenient means of public transportation; b) to relax the tense contradiction of transportation; c) to reduce overall fuel and energy consumption; and d) to reduce environmental pollution and noise.

Compared with long-distance railway systems, MRT systems experience higher speed and dispatch frequency, which demand high levels of safety and reliability and great system flexibility to deal with variations in traffic density. The purpose of this chapter is to give a brief description of automatic signaling and train control system that play a key role in MRT operations for the satisfactory performance.

2.2 Automatic Signaling and Train Control

2.2.1 Overall Structure and Functional Components

MRT signaling system is designed to achieve four basic objectives [13]:

1. Ensure safety of trains that the route ahead is clear and remains so until after the train is clear.
2. Maintain the regularity of transport service. A good regularity means a small deviation from pre-determined timetable.
3. Ensure swift response to passenger demands.
4. Enable the railway to be operated at minimum cost with maximum passenger satisfactions.

Signaling and train control facilities used by MRT can be classified into 4 distinct categories. They are:

1. Signaling system
2. Automatic Train Supervision (ATS) system
3. Automatic Train Control (ATC) system
4. Automatic Train Regulator (ATR) system

The signaling system in MRT includes the conventional equipment such as point machines, colored light signals, signaling panels and relay interlocking. All the above signaling and train control systems are under the overall surveillance of the Automatic Train Supervision (ATS) system. The ATS system performs functions of train location monitoring, train routing and timetable running. Fig 2.1 shows the schematic on signaling and train control system.

The computer based ATS system monitors and controls the whole signaling system in accordance with timetable working. ATS consists of the Operating Control Center (OCC), the Interlocking Stations, and the Intermediate Stations. The OCC, which is equipped with the Main Computer and the Scheduler Compiler, amongst other signaling and SCADA systems, is the heart of the ATS system. The Main Computer controls the railway operation whilst the schedule Compiler compiles schedules and works as a redundancy or ‘standby’ for the Main Computer, in case it fails.

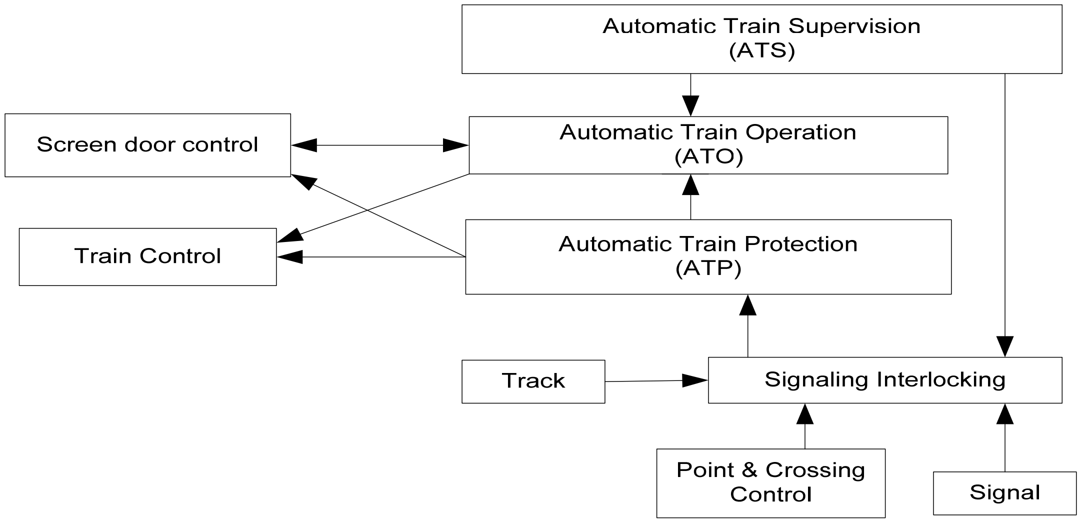


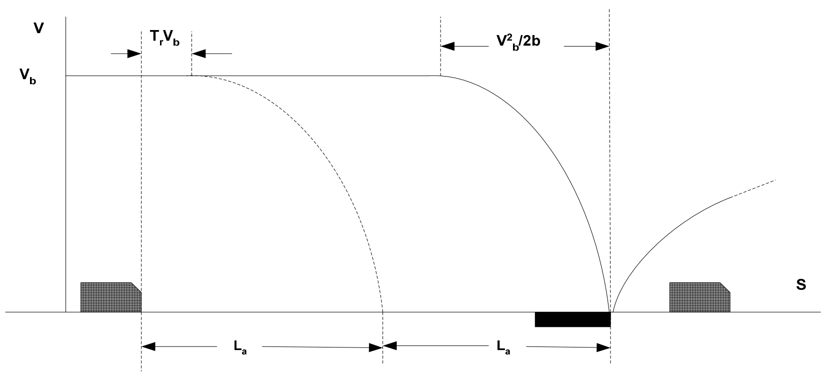
Fig 2.1 Schematic on signaling and train control system

The MRT system employs a network of distributed processing computing units and does not rely solely on the Main Computer at the OCC to perform all automated task. Although the Main Computer retains overall control of the system, the individual Local Processing Units (LPU) situated at the Interlocking and Intermediate Stations actually are the ones controlling the trains. This feature enables the MRT network to continue operating with minimal disruption even when part of the system fails to function.

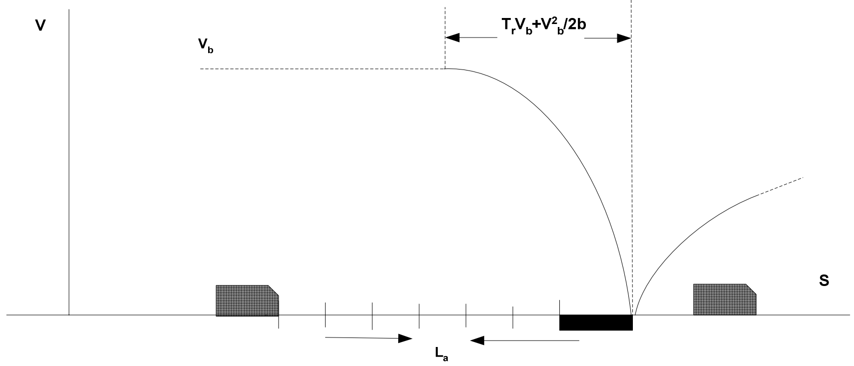
2.2.2 Signaling Scheme in Mass Rapid Transit System

Automatic block, track circuit based operation continues to be the most widely used form of signaling for MRT systems. Two forms of blocking signaling schemes are most often employed for automatic train protection, namely the Multiple Aspect Equi-Block Sigalling Scheme (EBSS) and the Moving Block Signaling (MBSS) [14].

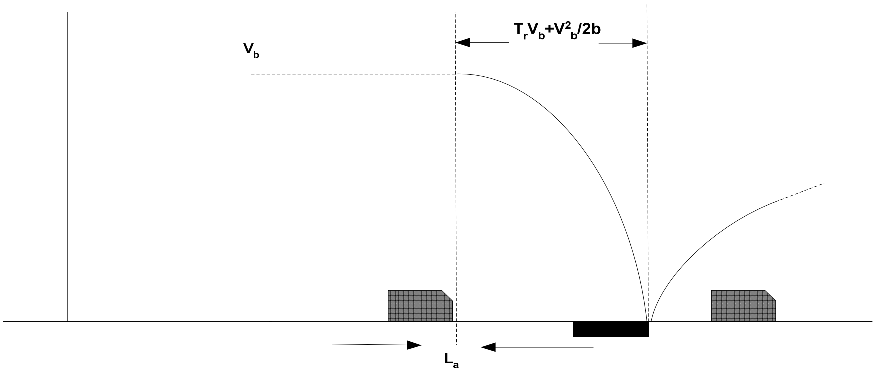
Under ATP block signaling of EBSS, the railway track is divided into ATP blocks whose length can be made shorter than the braking distance of the trains. Advance warning of restrictive speed conditions is given up to n blocks ahead, which reduces the separation of successive trains, thus improving the headway performance. Reducing the headway brings about an increase in the overall line capacity of the system which in turn can ease congestion levels during the peak periods of train operation.



a.



b.



c.

Fig 2.2 Trajectories for block signaling

a. Block length > braking distance

b. Block length < braking distance

c. Block length = 0

As shown in Fig 2.2, under ATP block signaling of EBSS, the railway track is divided into ATP blocks whose length can be made shorter than the braking distance of the trains. Advance warning of restrictive speed conditions is given up to n blocks ahead, which reduces the separation of successive trains, thus improving the headway performance. Reducing the headway brings about an increase in the overall line capacity of the system which in turn can ease congestion levels during the peak periods of train operation.

If the block length L_a is reduced to zero, it can represent a complete accuracy of the train's position. Theoretically, the headway between successive trains will become minimum (refer to Fig 2.2 c). To implement the MBSS, each train needs to know its own position, its speed, and its distance to the next station stopover point and to the rear of the leading train. This distance must be dynamically fed to each train, thus making continuous and reliable bi-directional data transmission crucial between trackside circuit and each train.

EBSS represents a crude form of positional resolution used in modern mass transit systems. Our proposed ATR models the Moving Block Signaling System which theoretically allows a much greater number of trains to be run at peak times since the trains would be separated by the minimum safe distance. Under this signaling scheme, the train would continually monitor its position using trackside beacons. It would send this data to the central computer, along with data about what speed it was doing. The computer then sends back information about what speed it should be doing. Hence, continuous and reliable bi-directional data transmission is crucial between trackside circuit and each train. However, in the software simulation this signaling scheme can be implemented simply through adding communication functions between the train objects and the object of Operational Control Center to update the trains' position and other related status information.

2.3 Basics of Automatic Train Control System

Playing an indispensable role for making high-speed operation into reality, Automatic Train Control (ATC) is comprised of Automatic Train Protection (ATP) and Automatic Train Operation (ATO). ATP is the 'vital' system that protects against collision, switch malfunction, overspeeds, door malfunctions, and other related operational problems. The ATO system drives the train automatically to achieve the prescribed operational performance.

The ATP that consists of the ATP Trackside system and the ATP Train-carried system provide ‘over-speed protection’ for the railway system. When the ATP Trackside system detects a train, it transmits the Maximum Safe Speed (MSS) and Target Speed (TS) to the ATP Train-carried equipment. The speed restriction is displayed in the drives cab and the speed restriction is conveyed to the braking control system. If the actual speed exceeds the MSS, the emergency brakes will be applied, bringing the train to a halt.

The ATO system uses an onboard database containing lineside data to control train braking, coasting and acceleration to suit the relevant conditions at each station. During operation, information from the ATP system, instructions from the ATS system and other data such as gradient, distance to station stop, braking rates, etc, are collected. ATO also gives a precision and repeatability of performance impossible of attainment with manual driving thus enabling inter-station run times, headways and energy consumption to be optimized.

The ATO system uses an onboard database containing lineside data to drive the train in accordance with the desired operational performance by carrying out the following functions: Speed regulation, accurate station stopping, execution of signaling stops and automatic restart, coasting control, indications to train operator to open/close doors, enabling of door open/close controls. ATO gives a precision and repeatability of performance impossible of attainment with manual driving thus enabling inter-station run times, headways and energy consumption to be optimized. The train running in this mode always has the fail-safe backup of the ATP system, which makes an emergency brake application if incorrect operation of the ATO equipment causes the Maximum Safe Speed to be exceeded. Fig 2.3 shows the ATO system. The equipment comprises a train borne ATO controller, ATO antennae and trackside marker and data loops.

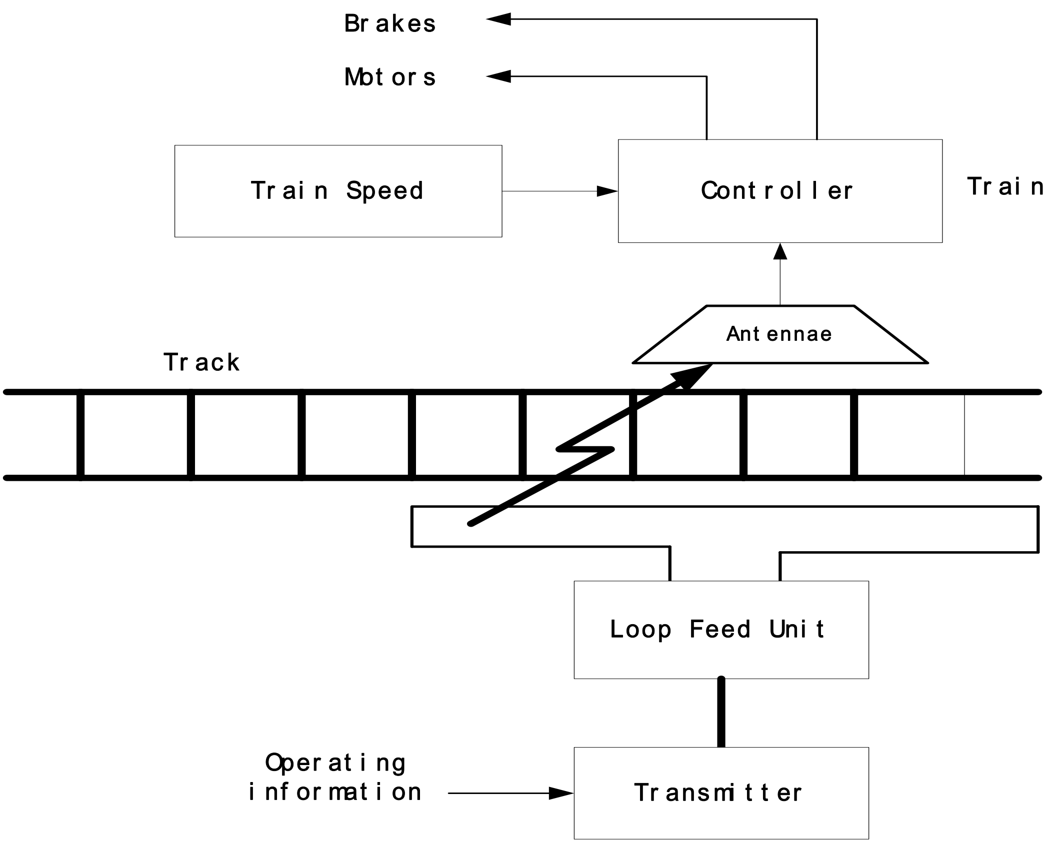


Fig 2.3 ATO block diagram

It has been seen that the ATO controller receives inputs of target speed, train and operating mode from the non-vital ATP unit. From the trackside loop via the train's ATO antennae it receives information for the next inter-station run and from the train operator's panel start button commands, driving mode and direction of travel. The ATO controller provides outputs to the train's traction control system, calling for motors or brakes and to the train operator's panel indicating open or close doors and alarms

The ATO system uses an onboard database containing lineside data to control train braking, coasting and acceleration to suit the relevant conditions at each station. During operation, information from the ATP system, instructions from the ATS system and other data such as gradient, distance to station stop, braking rates, etc, are collected. ATO also gives a precision and repeatability of performance impossible of attainment with manual driving thus enabling inter-station run times, headways and energy consumption to be optimized.

2.4 Cost Functions of Mass Rapid Transit

As described, the main objective of MRT operation is to enhance the efficiency of passenger service, that is, improving or maintaining passenger satisfaction while utilizing the resources in system economically. To evaluate the status of resource utilization, the Singapore Mass Rapid Transport (SMRT) looks into a series of cost components that are compiled into one spreadsheet. The spreadsheet is used to calculate the daily operational cost of one MRT track in an offline mode. It consists of 6 main input sections: time periods, service information, coast level, total of spare and double ending hours, train operations and transport allowances, and miscellaneous data. The 3 output sections are costs, train usage, and passenger load and headway. The derivation of the cost equations for the total operational cost and its components, such as fixed cost (rolling stock, infrastructure, track and rolling stock maintenance, etc) and variable cost (power consumption and train operational manpower), can also be found in the APPENDIX. The rest of this section is mainly focused on identifying the variables that would affect the cost functions.

2.5 Relation between Variables in Cost Equations

It was observed that the terms, dwell time, run time, layover time and headway exist in most of the cost equations. Dwell time denotes the amount of time a train remains at a particular station. The run time is the amount of time a train spends on each inter-station journey. Layover time is the amount of time taken for the change of shift and other mandatory checks at the terminal/interchange stations. Headway is defined as the dispatching time/interval between two consecutive trains from the terminal station.

It was seen from the various costs as shown in Appendix D that an increase in any of the three variables (layover, dwell time and run time) would increase the cost, while an increase in the headway would decrease the total operating costs. In this thesis, the fixed costs are not considered since train operation has no effect on the fixed cost (whether or not trains are run fixed costs are incurred). For the variable cost aroused by power consumption and train operational manpower, only the cost of electrical energy for traction power and air-conditioning are identified as factors affecting operating costs which can be effectively optimized by the proposed online rescheduling algorithm in the thesis. For the purpose of simplicity but without losing generality, layover time is not being included into the factors affecting the operational costs. Therefore, the three factors, headway, dwell time and coast level (run time is related to the coast level of the train) are adopted as primary variables for the online rescheduling optimization in this research.

CHAPTER 3

AUTOMATIC TRAIN REGULATOR FOR ONLINE TRAIN RESCHEDULING AND CONFLICT RESOLUTION OF MASS RAPID TRANSIT SYSTEM

Based on the structure of Automatic Train Control (ATC) and signaling system introduced in Chapter 2, an innovative Automatic Train Regulator is proposed in this Chapter in order to fulfill the operational objectives commonly set by railway industry. Correspondingly, fundamental performance indices are presented to evaluate the performance of each individual train or platform. The two train-rescheduling algorithms at the Operational Control Centre and Local Process Units in ATR are presented.

3.1 Overview

Advancement in communication networks and high speed computers have provided control systems with real time information on the position and speed of the trains, which allow automation of many real time functions in train operations such as controlling and rescheduling of train services. This project proposes an innovative train rescheduling strategy known as Automatic Train Regulator (ATR) at the OCC and LPU levels to optimize the overall performance of mass transit systems. ATR employs soft computing techniques, such as fuzzy logic system and evolutionary computing and the fuzzy performance indices of regularity, energy, train as well as platform congestion for online train rescheduling.

3.1.1 Operational Objectives

System safety, operational efficiency, and passenger satisfaction are most often adopted indices to evaluate the performance of a MRT system. To remain competitive in the transport industry, railway operators aim to achieve certain important operational objectives as follows:

1. To keep train operate in high safety (identification and resolution of possible conflicts)
2. To maintain operational efficiency (service regularity and energy efficiency)
3. To ensure passenger satisfaction (punctuality, train and platform congestion).

Passenger satisfaction relates to factors that have an effect on the quality of transit service, which includes service regularity, train congestion level and platform congestion level. Regularity of service is denoted by the degree to which the trains actually operate according to their pre-defined timetables. The performance index of regularity penalizes on the early or late arrival or dispatch of trains, since irregular waiting times for trains can bring about frustration as well as other inconveniences for commuters.

Commuter dissatisfaction can also arise from the experiences in an overcrowded train, whereas under-utilization of train cabins can be considered being highly inefficient in terms of operating costs to dispatch an empty train at a close time interval from the previous one. Hence, the performance index of train congestion penalizes on a deviation of passenger number in the train away from a nominal number.

Platform congestion is a critical indication of the demand of service, since a huge number of passengers waiting at the next platform prompts for an early train dispatch to satisfy the increasing demand for service and return the platform congestion to normal. In the worst, deteriorating platform congestion poses a potential safety hazard to the system. However, a low platform congestion level is an indication of low train utility, which should warrant a delay in train dispatch either from its terminal or dwelling station. As such, the performance index of platform congestion rewards an adherence to a nominal commuter number on the platform and penalizes on a deviation from it.

Operational efficiency refers to the efficiency in the utilization of resources such as electricity, trains and other inventories. The economy of operation of the railway depends very much on these factors as well. Total train energy consumption is sought to be minimized as a criteria that brings with it higher operational efficiency which in turn means lower operating costs for the mass transit operator. The window period for consideration is the minimum service headway and encompasses trains operating on the up and down tracks of the railway line simulated.

One of most important objectives for train schedulers is to avoid a headway conflict or any train encroaching upon another train's headway leading to an unplanned stop. To this end, the railway traffic controller continuously monitors train traffic to identify any abnormality that can lead to a headway conflict, under which. ATC/ATP devices will take over and automatically stop a train. In this project, two typical conflict scenarios are represented by a set of operational constraints to detect potential conflicts.

3.1.2 Performance Indices

For situations that the most important information comes from human experts, fuzzy control would be the best choice. Fuzzy controllers provide a systematic and efficient framework to incorporate linguistic information from human experts. It can act as a human operator by translating experts' linguistic description into an algorithm suitable for computer programming.

The objectives discussed in section 3.1.1 can be evaluated by incorporating into four fuzzy performance indices all the information gained from experience of field experts or comprehensive survey data collected by MRT Company. To gain the reference data regarding passenger satisfaction such as regularity, train and platform congestion, the best way for service provider is to conduct various questionnaires since they are customer-oriented. Therefore, the shapes of those three membership functions derived from the survey data need to be adjusted accordingly only when the trains and platforms in the whole MRT system are upgraded in order to accommodate a growing number of passengers. In contrast to other membership functions, the parameters that determine the shape of membership function for evaluating energy consumption can be varied according to the predicted maximum value and minimum value of energy consumption in each time window. It is designed to estimate the condition of energy consumption for potential solutions and function as a reference to counter the side effect introduced by solely considering the factors in passenger satisfaction.

The following linguistics levels are used: Regularity {Early, Okay, Late}, Energy {Low, Normal, High}, Train/platform congestion {Empty, Satisfactory, Congested}. Fuzzy memberships of these performance indices are shown in Figs 3.1-3.4.

Performance Index 1-Regularity

The performance index of regularity reflects how closely the train operation follows the pre-defined timetable. The train is considered absolutely early if it arrives 60 seconds before or earlier, and it is considered absolutely late if it arrives 90 seconds after or later. The in-between arrival times are described by a mix of the fuzzy membership functions known as “Early”, “Okay”, and “Late” as in Fig 3.1.

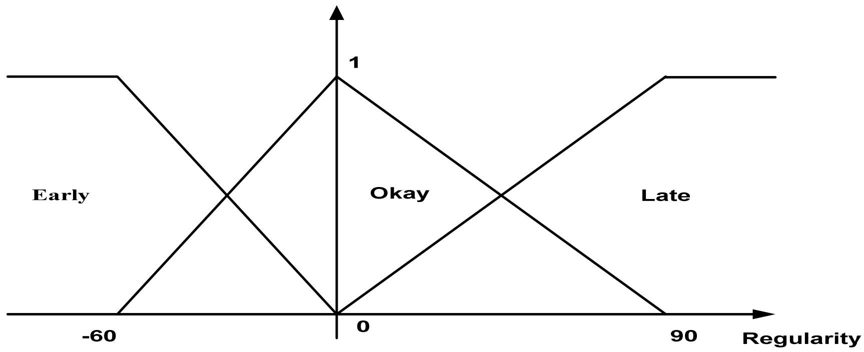


Fig 3.1 Fuzzy memberships for regularity performance index

Performance Index 2-Energy Consumption

The total energy consumption is calculated as the aggregate sum of the traction energy of all trains running in each time window. The maximum and minimum values of the energy consumption in the time-window are predicted by adjusting the dispatch interval, dwell time and coast level. Based on this, a convincing membership function can be derived in the following ways:

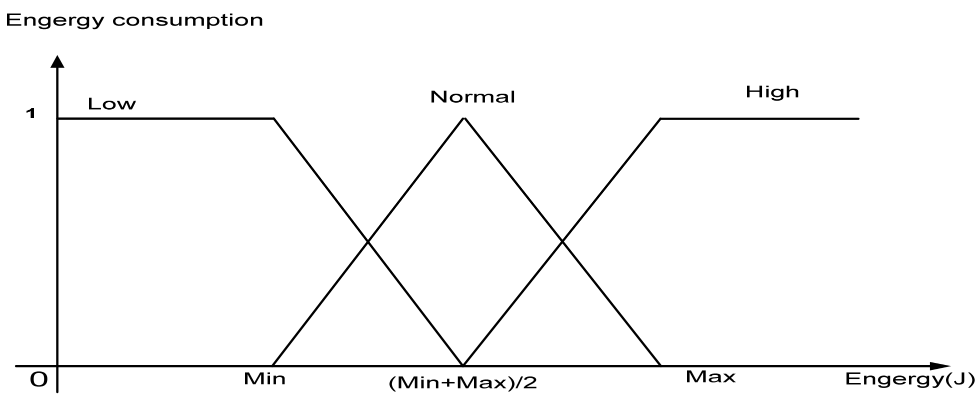


Fig 3.2 Fuzzy memberships for energy consumption performances index

Performance Index 3-Train Congestion

The performance index of train congestion determines the total number of passengers residing on a train. The nominal passenger capacity of a train is set at 650, and the maximum passenger congestion level allowable is set at 1500. The train congestion performance index is expressed as follows:

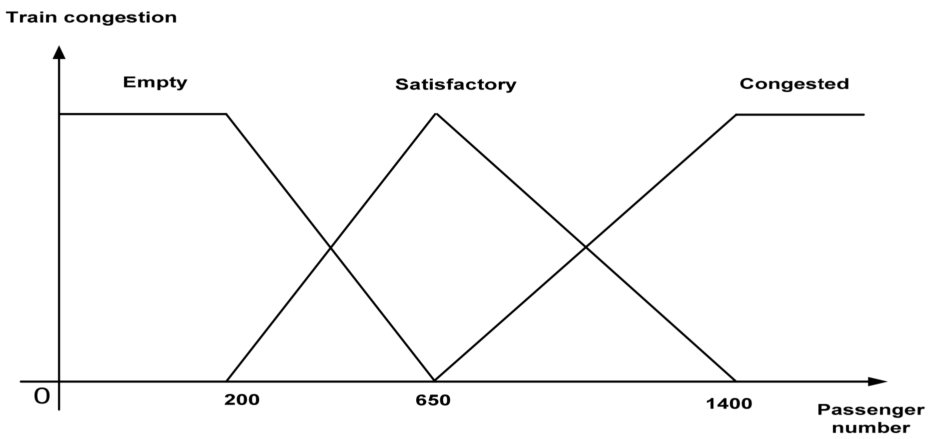


Fig 3.3 Fuzzy memberships for train congestion performance index

Performance Index 4-Platform Congestion

The performance index of platform congestion expresses the congestion level on a platform in any time window. The nominal commuter packing capacity of the platform is set at 350 and the maximum platform congestion level allowable is set at 800. The platform congestion performance index is measured according to the following fuzzy membership functions:

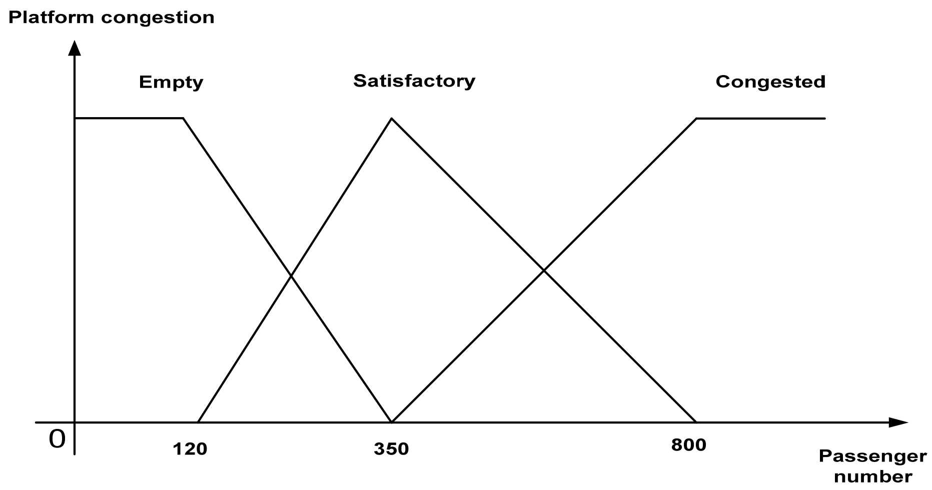


Fig 3.4 Fuzzy memberships for platform congestion performance index

3.2 Functional Description of Proposed Automatic Train Regulator

The proposed ATR optimizes the passenger service while maintaining low operational cost. Due to complexity of the railway operations, a MRT system commonly operates with a master schedule strategy. According to this strategy, a feasible timetable is produced in advance for each scheduled train. The predefined timetable must provide sufficient operational flexibility for (1) a consistent operating plan that can achieve train schedule with high probability, and (2) a platform for making operational variations as a result of unexpected disturbances in passenger flows as well as interferences and delays encountered by each train.

While detailed tactical schedules are developed for all trains through the predefined timetable, unforeseen events can arise and require timetable modifications without short notice. These events can take the form of train headway conflicts, distributed or sharp rises in passenger load at some stations. Accordingly, ATR produces an active timetable, and updates it regularly for online train rescheduling and conflict resolution.

In the modeling of MRT system, a clear separation is made between the train movement and traffic control system. The former models train movements on tracks. The latter is regulated by a proposed intelligent Decision Support System (DSS) to resolve conflicts, maintain train regularity, and reduce train congestion, platform congestion and energy consumption. The proposed DSS performs two main functions, namely Scenario Analysis and Intelligent Decision-making.

Accordingly, ATR performs scenario analysis as a key function to keep track of train traffic evolution. ATR also looks one time window ahead to predict the optimal dwell times and coast levels with the help of scenario analysis.

With the assistance of scenario analysis, ATR implements intelligent decision-making to work out a near-optimal solution. For instance, when a present or potential headway conflict has been detected, ATR assesses its impact by examining the active timetable along with the local state of the MRT system (including its regularity, congestion, and energy consumption) to generate a solution with satisfactory overall performance and steer the system clear of conflict. Finally, all operational constraints and the four performance indices associated with each possible solution are updated. Some local controllers employed in traditional methods detect and solve the conflict without consideration of the aggravated delays, which can induce new conflicts in other parts of line. As conflict resolution is an optimization problem, a conflict-solving module equipped with evolution algorithm is applied to optimize the conflict resolution process, taking into account the state of the whole line within a predefined time window.

In current MRT practice, conflicts are resolved mainly manually. The proposed computerized DSS has been designed and implemented to support the dispatchers to quickly and effectively reschedule the train movements en route.

As shown in Fig 3.5, from a systematic view, the proposed DSS will be divided into a hierarchy of three sub-algorithms:

a. **Train Rescheduling Algorithm at Operational Control Centre** (OCC): This scheme controls the whole railway line under study, helping monitoring operators by making decisions on dispatch interval, dwell times and coast levels.

b. **Train Rescheduling Algorithm at Local Process Units** (LPU): This distributed scheme detects severe abnormalities or disturbances arising locally within a short time and modifies decisions on dwell times and coast levels at station controllers.

c. **Coordinating Train Rescheduling Algorithms**: This is a master schedule strategy for providing global control to coordinate the two schemes at the OCC and LPU for online rescheduling under various disturbance scenarios.

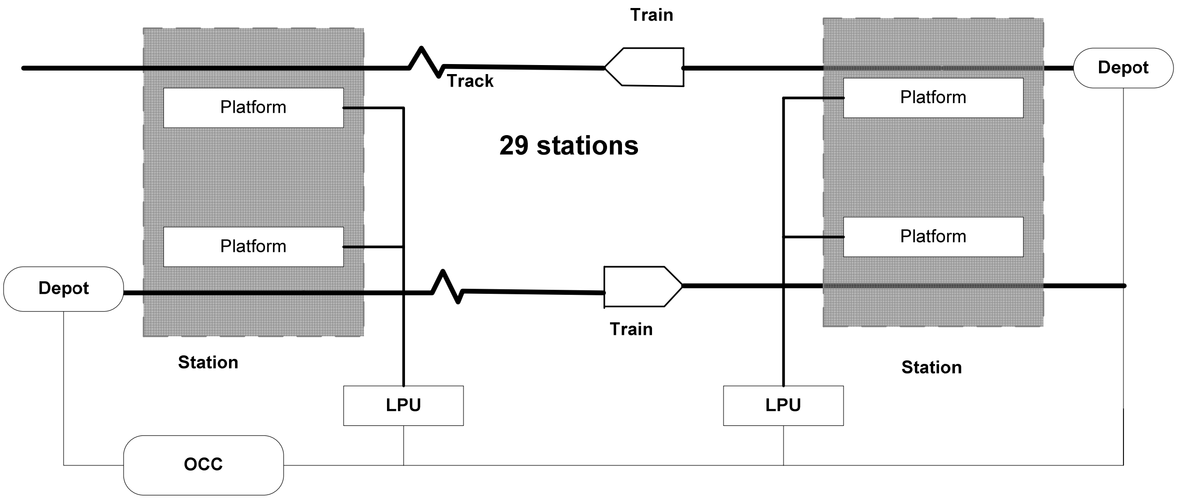


Fig 3.5 Layout of the proposed train rescheduling at Operational Control Centre and Local Process Units

3.3 Proposed Train Rescheduling Algorithm at Operational Control Centre

This algorithm is implemented at the OCC to employ primary rescheduling that takes into account all trains and stations in each track as objects of an optimization problem.

3.3.1 Control Variables

In each time window, the controller in ATS is triggered to predict the controls and movements of all trains over the time window through scenario analysis. Whenever a conflict is detected, a new dynamic control vector is derived $\underline{\Delta P}_{i,G} = (\Delta t_1, \Delta t_2 .. \Delta t_n, l_1, l_2 ... l_m, tm)$ to synthesize a new active timetable, where Δt, l, and tm are the dwell time, coast level, and dispatch interval respectively. n is the number of passenger stations included in each time window, and m is the number of trains included in each time window. The dynamic control vector is represented by chromosomes in evolutionary computing. The three control variables are defined as follows:

1. Dispatch Interval tm **:** This is defined as the time difference between successive trains (known as time margin) to be dispatched from a terminal station. The train rescheduling algorithm adjusts tm to maintain system performance at the OCC level.

2. Dwell Time Δt **:** This is the time taken by a train to stop over at a particular station. Adjustment to the dwell time of a train will alter the time margin with the train ahead, the time margin with the train behind, the passenger loading time. This variable is affected by passenger flow (rate of entering and leaving) and the time taken by passengers to board/alight each train.

3. Coast Level l**:** This describes the train movement profile in the journey between each pair of successive stations (or inter-station journey). The run time of each inter-station journey a train depends on the amount of coasting and powering in the journey. As illustrated in an inter-station velocity-time profile (Fig 3.6), the train is powered on and accelerated after departure. When it reaches a desired velocity, the train can either (**a**) remain powered and maintain the desired velocity (Coast Level 0) to cover a given distance within the shortest runtime but requiring the highest energy consumption. (**b**) turn the motive power on and off alternately at certain preset velocities (Coast Level 1), or (**c**) turn off the motive power completely (Coast Level 2) for the longest run time but consuming the lowest energy. Adjustment to coast level will alter the energy usage during the inter-station run.

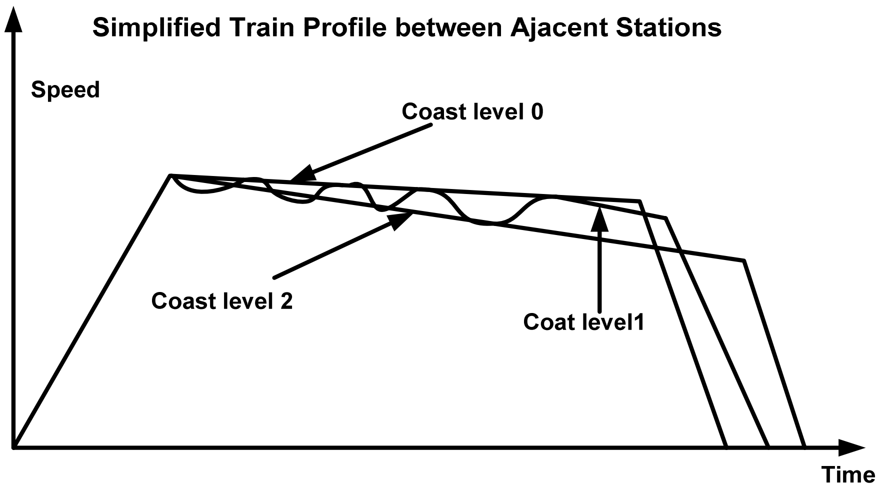


Fig 3.6 Train run-time profiles at different coasting levels

3.3.2 Functions of Proposed Train Rescheduling Algorithm at Operational Control Centre

The rescheduler at OCC executes an evolutionary-algorithm for optimizing train rescheduling to adjust all the train dispatch and dwell times, and train coasting levels of all inter-station runs within each time window. Based on the analysis of the real-time information on the MRT system such as the passenger incoming or outgoing rate at the stations, the online rescheduling algorithm at OCC predicts the runtime and other performance indices of trains entering each time window, adjusts the train dispatch times and dwell times, and hence updates an active timetable to satisfy all operating requirements for the whole line. All the adjustable parameters are sampled as the input parameter vector of the evolutionary algorithms to make a fast and robust numerical optimization in order to generate a near optimal solution. A group of operational constraints are set in order to detect potential conflicts, which may occur under different conflict scenarios as a result of the various possible solutions. A fuzzy fitness aggregated with the four sets of fuzzy performance indices has been used to measure the effects of rescheduling.

3.3.3 Active Timetable

Starting from the predefined timetable, MRT operators maintain several active timetables using rescheduling computer software. The proposed ATR reads one of these timetables for performing scenario analysis and dynamic decision-making according to the latest operational circumstances for conflict identification and resolution. In an object-oriented environment, ATR creates an active timetable as a live object with a structure as in Fig 3.7.

Object Name: Active timetable
Data attribute: Arrive time; Departure time; Coast level;
Associate methods: Search Monitored stations; Retrieve or update arrive time; Retrieve or update departure time; Retrieve or update coast level;

Fig 3.7 Structure of proposed active timetable

As discussed, at the beginning of a control transaction, active timetable is originally a copy of predefined timetable. During the control transaction, the active timetable is retrieved by DSS for scenario analysis and updated by new schedules after going through the evolutionary-algorithm-optimised rescheduler. Fig 3.8 illustrates the processes of scenario analysis on the active timetable through sliding time windows.

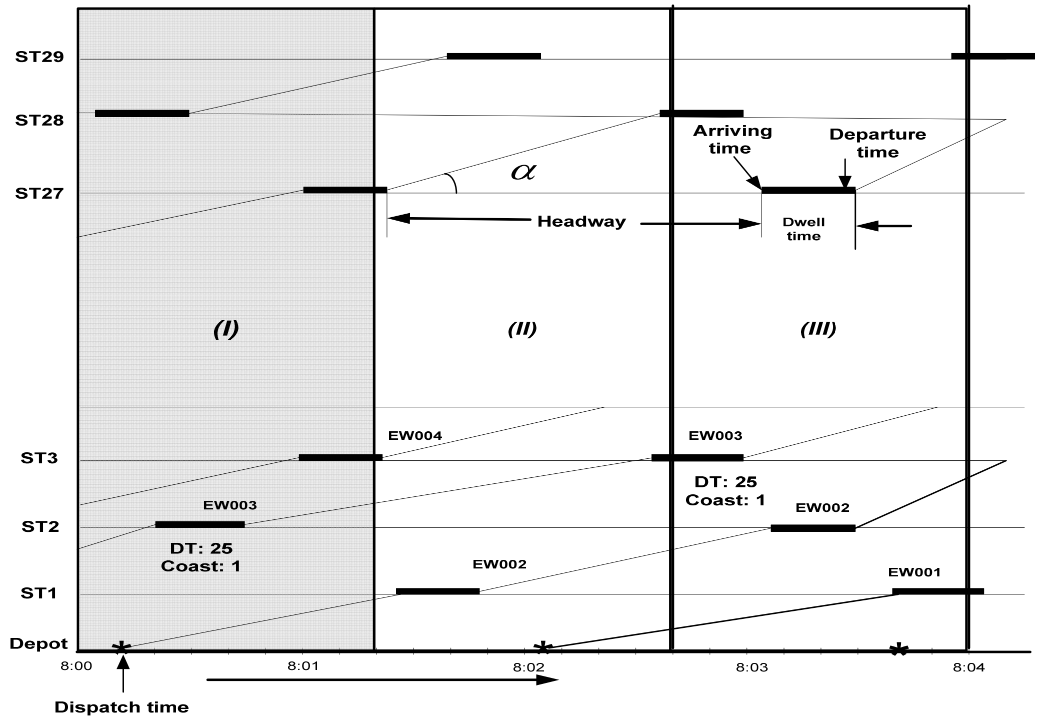
As shown in Fig 3.8, the active timetable stores schedule information of every train including its dispatch time, arrival time and departure time. The slope angle α of lines connecting two stations reflects the coast-level information. From the information above, dwell times for every station and headway between two consecutive trains can be deduced:

$$dwelltime_{t,s} = Arrivaltime_{t,s} - Departuretime_{t,s} \quad (3.1)$$

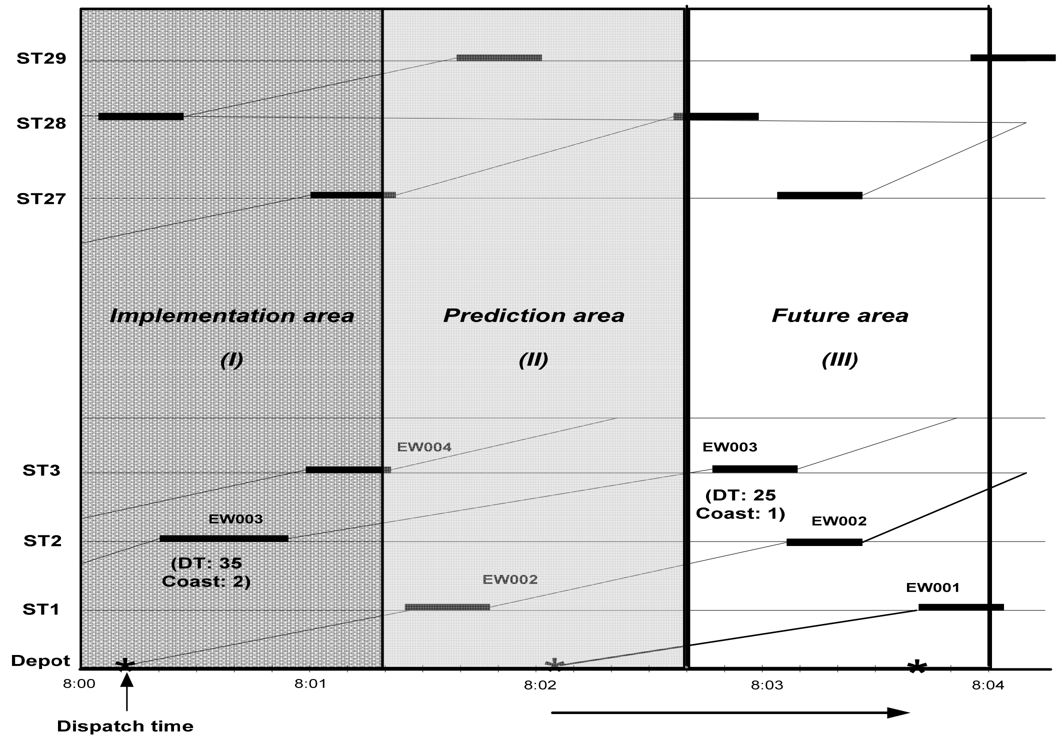
$$headway_{t1,t2} = Arrivaltime_{t2,s} - Arrivaltime_{t1,s} \quad (3.2)$$

where $dwelltime_{t,s}$ is the dwell time for train t at station s, $headway_{t1,t2}$ is the headway between train t1 and t2.

ATR divides the study/control period into time windows of equal length, sliding from one time window to another according to a real-time clock. As illustrated in Fig 3.8(b), time windows distributed along the active timetable are labeled with implementation, prediction, and future areas for reference purposes. For instance, in Fig 3.8(a), arrival time of train EW003 for station 2 and similarly that of train EW004 for station 3, are within the scope of time window I (shadow area). Hence using real-time information of the MRT system, ATR predicts the timing of each inter-station run and other performance indices occurring in the present time window. Adjustment to the dwell time and coast level of a train will alter the headway with the train ahead, the headway with the train behind, the passenger loading time and the energy usage during an inter-station runtime.



(a)



(b)

Fig 3.8 Illustration of scenario analysis on Active Timetable

Accordingly the adjustment to the value of t_{dwell} and the selection of the coast level will affect the following:

1. The regularity of train service, Reg [($t_{dwell,}$ cl)].
2. The energy consumption, Eng [($t_{dwell,}$ cl)].
3. The congestion level on the train, TrnCg [($t_{dwell,}$ cl)].
4. The congestion level on the next platform, PFCg [($t_{dwell,}$ cl)].

Reg, Eng, PFCg, and TrnCg are performance indices that reflect regularity, energy, and congestion. t_{dwell} and cl are the dwell time and the coast level respectively.

The performance index for the regularity of train service Reg(t_{dwell}) is a measure of the deviation of the train departure time from the predetermined timetable.

$$\text{Reg}[(t_{dwell,}\ \text{cl})] = \text{Current Time} + (t_{dwell} - \text{planned departure time}) + (\text{runtime}[\text{cl}] - \text{runtime}[\text{CL}]) \quad (3.3)$$

where runtime[CL] is the runtime at predefined coast level CL and runtime[cl] is the runtime of the current coast level.

Eng [($t_{dwell,}$ cl)] is the performance index that measures energy consumption, which depends on the control variable. Runtime-profiles, which correspond to different coast levels, indicate unique inter-station energy consumption curves.

The performance index for the congestion level on the current platform TrnCg (t_{dwell}) is measured by the number of passengers on the dwelling train when the train doors are closed. Before the calculation of PFCg(t_{dwell}), the number of passengers to be loaded onto the train PsgToBeLoad(t_{dwell}), the maximum number of passengers able to aboard in t_{dwell} seconds MaxPsgLoad(t_{dwell}), and the maximum available space on the train MaxAvailSpc(t_{dwell}) have to be calculated.

$$\text{PsgToBeLoad}(t_{dwell}) = \text{PFCurrPsg} + \text{PFRate}_i * t_{dwell} \qquad (3.4)$$

$$\text{MaxPsgLoad}(t_{dwell}) = (t_{dwell} - \text{APsg/LRate}) * \text{LRate} \qquad (3.5)$$

$$\text{MaxAvailSpc} = \text{TrnSpc} - \text{TrnPsg} - \text{APsg} \qquad (3.6)$$

where,

PFCurrPsg - refers to the current number of passengers on platform

PFRate_i - refers to the rate of passengers entering the platform i

APsg - refers to the number of alighting passengers

LRate - refers to the rate of passenger flow

The number of passengers actually loaded is:

$$\text{LPsg } [(t_{dwell,}\ \text{cl})] = \min(\text{PsgToBeLoad}(t_{dwell}),\ \text{MaxPsgLoad}(t_{dwell}),\ \text{MaxAvailSpc}) \qquad (3.7)$$

$$\text{Hence, TrnCg } [(t_{dwell,}\ \text{cl})] = \text{TrnCurrPsg} + \text{LPsg} \qquad (3.8)$$

The performance index for the congestion level on the next platform PFCg(t_{dwell}) is predicted from the rate of passenger entering the next platform PFRate_{i+1}.

$$PFCg\,[(t_{dwell,}\,cl)] = [t_m^{curr} + Reg[(t_{dwell,}\,cl)] * PFRate_{i+1} \quad (3.9)$$

where tm^{curr} refers to the current time interval.

After the prediction, ATR slides forward from time window I to time window II whilst the ATR executes an evolutionary-algorithm-optimized train re-scheduler to adjust all the train dispatch and dwell times, and train coasting levels of all inter-station runs within each time window. Details of EA optimization process are presented in the Section 5.3.

3.3.4 Runtime Constraints for Two Types of Conflict Scenarios

The decision support system in the ATR is subject to various runtime constraints to ensure safe operations. These constraints are also be used to reduce the space of search and computing time required for optimal solution.

For each train t traveling between two adjacent stations s and s+1:

$$Arrivaltime_{t,s+1} - Departuretime_{t,s} = runtime_{s,s+1}. \quad (3.10)$$

$$Departuretime_{t,s} - Arrivaltime_{t,s} \geq \min\ dwelltime_{t,s}.$$

For any two consecutive trains t1 and t2:

$$Arrivaltime_{t2,s} \geq Arrivaltime_{t1,s} + \min headway.$$

$$Departuretime_{t2,s} \geq Departuretime_{t1,s} + \min headway. \tag{3.11}$$

The above safety constraints are used to detect and avoid head-to-tail collisions. Some specific constraints can be introduced to detect and solve conflicts arising locally in a given area. For instance, one typical conflict scenario arising in the terminals of MRT system was simulated in the ATR and relevant constraints were attached for conflict detection and resolution.

Taking E1, E2, W3, and W4 as the four conflicting points along the tracks at the terminal station as in Fig 3.9:

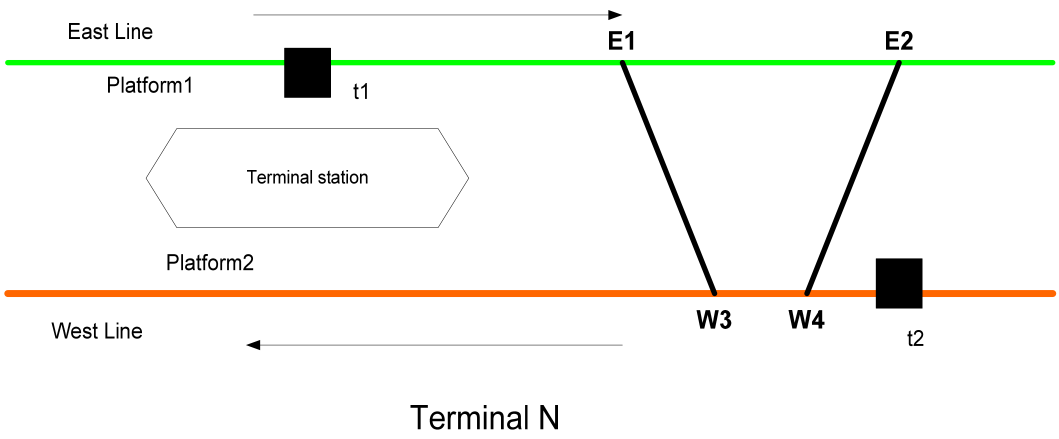


Fig 3.9 Typical conflict scenario arising at terminal stations

For train t1 departing from Terminal to east, there were two possible pathings

a) $Platform1 -> E1 -> E2 -> Eastline$ (3.12)

b) $Platform2 -> W3 -> W4 -> E2 -> Eastline$. (3.13)

For train t2 arriving at Terminal from east to west, there were two optional pathings:

c) $Westline -> W4 -> W3 -> Platform2$ (3.14)

d) $Westline -> W4 -> W3 -> E1 -> Platform1$ (3.15)

Accordingly, the following safety constraints were added to the three conflict scenarios when two of the above mentioned pathings occurred simultaneously resulting in a conflict:

a) and d) $Departuretime_{t1,E1} - Arrivaltime_{t2,E1} \geq Safety\ Interval$ (3.16)

b) and c) $Departuretime_{t1,w4} - Arrivaltime_{t2,w4} \geq Safety\ Interval$ (3.17)

b) and c) $Departuretime_{t1,w3} - Arrivaltime_{t2,w3} \geq Safety\ Interval$ (3.18)

3.3.5 Evolutionary-algorithm-optimised Rescheduler

In principle, finding the optimal rescheduling strategy requires the analysis of a very large solution space. An exhaustive search is not compatible with the real-time constraint imposed by the application. A routine transaction, consisting of the active timetable update, conflict detection and construction of a rescheduling strategy such as ATR, is fast enough for real-time, if it is completed within the span of each time windows.

To aim for such computing-time requirement, DE is considered as an exceptionally simple strategy that has the promise of providing robust and real-time optimizations. The design and the implementation of the proposed DE-optimized rescheduler are discussed in detail in Chapter 5. Genetic algorithm (GA), another candidate among all evolution strategies, is applied to ATR to compare performance with DE in Chapter 6 for solving the train rescheduling problem.

3.4 Proposed Train Rescheduling Algorithm at Local Process Units

LPUs receive rescheduled operational information from the OCC and regulate the departure time and routing of all trains within its own area. The train rescheduling algorithm at LPU serve as the auxiliary strategy to the one at OCC in case severe disturbances take place after the real-time information of system states is sampled by OCC at the start of a time window for rescheduling within predefined time window. For instance, if the passenger flow increases in some stations or a certain accident delays the train over a set value, some adjustment must be applied to command station-based LPU to subdue the ensuing unpredictable influences and hence avoid losing critical real-time information. LPU controllers complement and adjust if necessary the commands issued by the OCC controller at all stations from the beginning of each time window. The OCC controller examines the status of the MRT system states at the end of each time window for fine-tuning the commands in the next time window.

LPU controllers make use of fuzzy decision-making to adjust schedules based on the same fuzzy performance indices of regularity of service, energy consumption, train congestion as well as platform congestion levels. LPUs equipped with the on-line rescheduling algorithm that controls the departure of a train from the passenger station effectively controls the dwell time of that train at the station.

When a train arrives at a passenger station and the LPU controllers are set to “ON”, the algorithm will be activated to retrieve the dwell time from the Active timetable and reschedule the train departure time if necessary. The schematic of the events from the time the train arrives at the station is given in Fig 3.10.

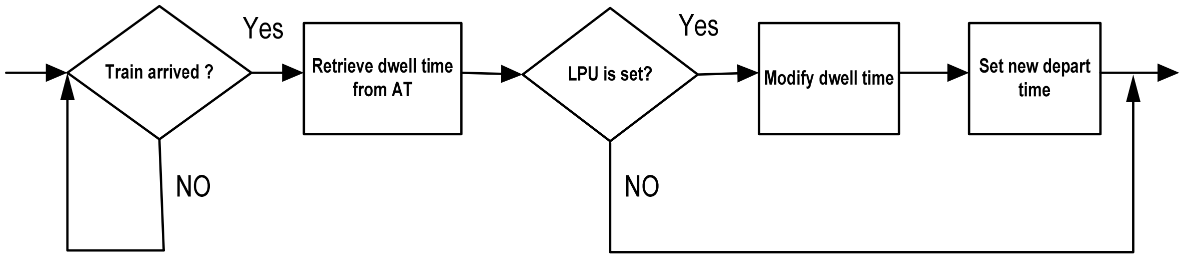


Fig 3.10 Schematic of train dwell time rescheduling in LPU

Adjustment to dwell time of the train will vary headway of the train with respect to the leading and lagging trains as well as the passenger loading time. Although this algorithm makes adjustment to headway, the variation is relatively minor compared to the on-line rescheduling of train dispatch interval from the terminal stations. However, it should be noted that once the train is dispatched from the terminal station, the on-line rescheduling of station dwell time becomes an important available means to regulate the dynamic headway of the train, besides other alternatives like train coasting, etc.

The Master Schedule Scheme is designed as “a global control” to coordinate the two schemes. It set the scenarios of triggering the corresponding schemes. Fig 3.11 shows the overall DSS block diagram, which applies the DE-based method for train rescheduling.

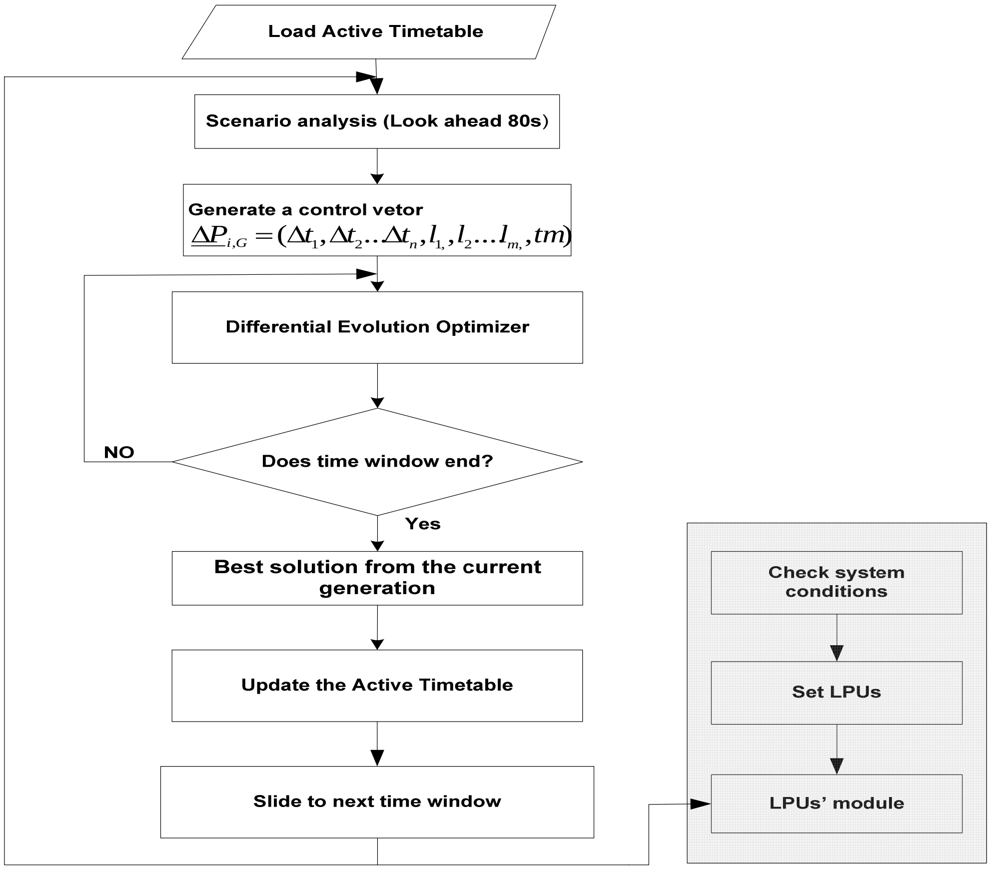


Fig 3.11 Block diagram of Decision Support System for ATR

CHAPTER 4

OBJECT-ORIENTED SIMULATION OF MASS RAPID TRANSIT SYSTEM WITH AUTOMATIC TRAIN REGULATOR

This Chapter explores the simulation of MRT that provides a platform to accommodate ATR rescheduling algorithms discussed in Chapter 3. The architecture of proposed ATR in simulation is illustrated. The application of object-oriented, multi-threading programming, and online database management techniques on ATR simulation is described. Lastly, typical train and platform operational profiles are presented as simulation outputs.

4.1 Overview

4.1.1 Layout of Object-oriented Mass Rapid Transit Simulation

Simulation attempts to develop models to synthesize and study the dynamic behavior of actual real-life systems. Precise and comprehensive knowledge representations of the MRT system are often beyond the scope of a simplified mathematical model for its complexity. These representations must be based on well-defined objects, which have data attributes, associate methods, and structures of their counterparts in the real world.

ATR has a simulation program integrating all train operation and train control functions. It is a time-driven simulation of train movement being incremented by a real-time clock in fixed steps to monitor and record the current states. It also models a decision support system for train dispatchers to schedule trains in real-time in an optimal way and as a planning tool to evaluate the impacts of possible timetable changes on the overall operational performance for the selected line track under study.

To model all ATR functions, the simulation model has to accommodate both discrete events and continuous train movements on the same simulation platform for modeling dynamic behavior of the MRT system. The main process of simulation is described as below:

1. Initialize the system state and simulation time
2. In each time increment until simulation is finished:

(a) Collect and save the current state described by the travel time, train chainage, speed, energy consumption, and the number of passenger on platforms or trains.

(b) Simulate all events that occurred between last step and the present time, which include system, platform and ATR controls and other discrete events on the trains such as ‘open the door’, ‘close the door’, “dispatch the train”, and “ run scenario analysis”.

(c) Increment simulation time

The development of simulation software for the operation of a MRT system demands enormous effort on understanding the operation, preparing the study data, and programming language. Hence, reusability is a highly desirable feature. Because of the enormous amount of data to be processed online, efficient data organization and management are indispensable for robustness. The coordination and interaction between different functional components require one mechanism that is able to reflect the nature of real-life system. ATR applies object-oriented programming, relational database management and multithreaded programming techniques to satisfy the above requirements.

4.1.2 Review on Object-oriented Representations for MRT System Simulation

Object Oriented Programming (OOP) [15] has long been acclaimed as an efficient methodology for minimizing development time and maximizing code reusability. OOP creates a modular design that is easily modified without having to restructure the entire system. OOP organizes objects into classes, which inherit properties from their ancestors and be reused in future projects. In our proposed ATR, three basic classes were defined: TRAIN, STATION, and TRACK.

The class TRAIN is created to represent the drop off and pick up of passengers at every station. It will stop the train at every station on its route. It keeps track of the total trip time for each passenger if the passenger has arrived at his or her final destination. TRAIN records each train's its current location, destination, energy consumption, speed, travel time, and the total number of passengers. At the end of the simulation, TRAIN returns the current state for display and further studies.

The class STATION is created to accommodate passengers and implement functions of LPU controller. It records the total number of passengers and the arrival and departure times for all trains. STATION also operates the train for its arrival and departure. At the end of simulation, STATION returns the current state for display and further studies.

The class TRACK is more of an arbitrator class. TRACK imposes global controls to keep track of the objects, coordinate the arrival and departure of TRAINs, define the schedules of TRAINs and resources, and print results. It also reschedules trains in operations and provides safety protection.

4.2 Simulation Detail of Automatic Train Regulator

4.2.1 Development of Automatic Train Regulator in Visual C++

The simulation is developed in Visual C++ including four main parts: Main User Interface, Train Movement Simulation, Decision Support System, and Reports Systems. Fig 4.1 illustrates the architecture of the ATR simulation.

Main User Interface is programmed through the Microsoft Functions Class in VC++ to enable human-machine interactions. It provides user-friendly interfaces to configure the layout of MRT systems or design parameters of reschedule controller, thus greatly benefiting the study of performance under different system settings.

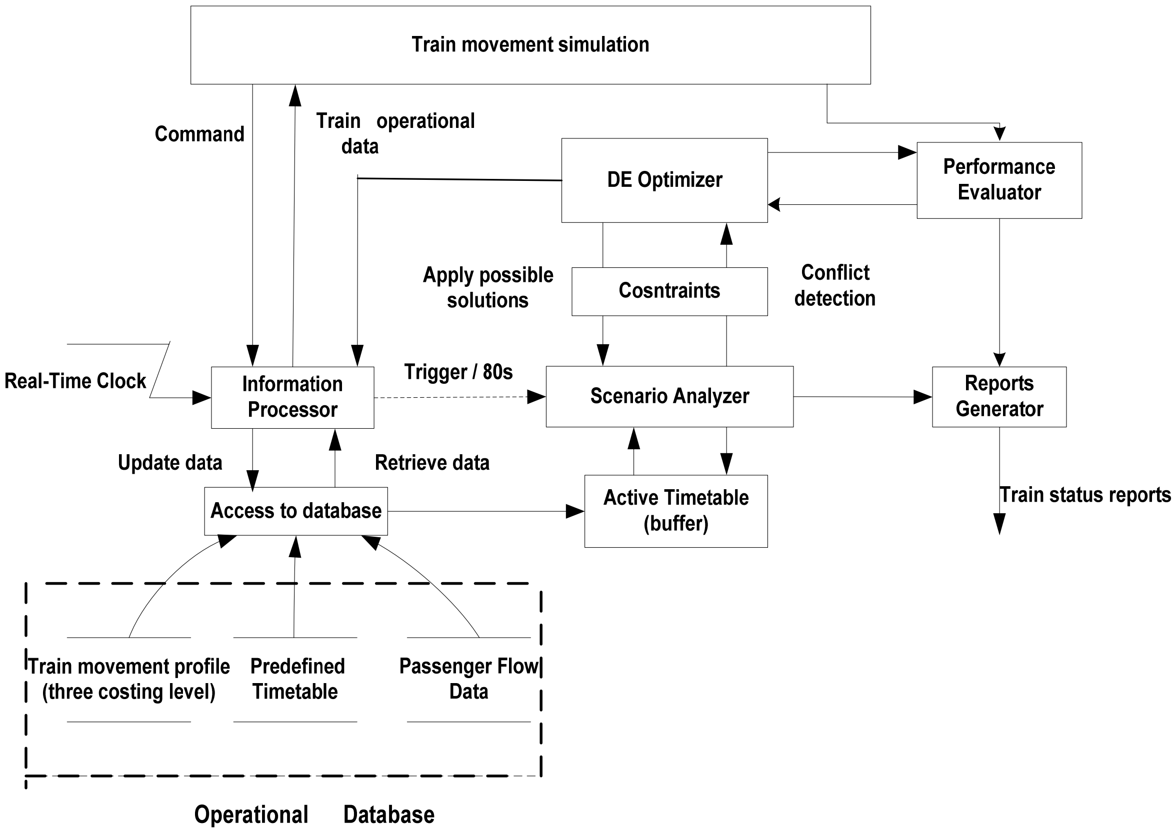


Fig 4.1 Architecture of proposed ATR and data flows

Train movement simulation is built based on train movement profiles organized in a relational database and on operational controller that fulfills basic operation of train in MRT system. ATR sends a command message to activate information manager that periodically updates and retrieves operational data stored in the information database.

Then Controller applies these data to update the state of all the station and train objects in operation.

Report systems serve to present high quality output graphics (time-distance, platform occupation, and speed profiles) and various performance indices of regularity, energy, train and platform congestion in tabulated form. They include two kinds of reports: real time graphics and static graphics.

4.2.2 Implementation of Multithreading Technique in Automatic Train Regulator

ATR simulation is conducted using multithreading techniques in Visual C++6.0. The simulation of each inter-station train movement is programmed as a thread because of its common features and close interaction to other train movements. A worker thread is created for Decision Support System running in the background to do scenario analysis, conflict resolution and optimisation. These two threads of execution are run concurrently and cooperate with each other through thread synchronization objects and thread communication mechanism (as provided by Windows), which simulate the real-time communications between trains, LPU and OCC.

4.3 Design for Real Time MRT Systems

Since the ATR simulation is designed for an online rescheduling task, the real time requirements or constraints are always crucial in the design process for its actual implementation in a real life MRT control system. A real time system must respond to events such as disturbances or accidents within a finite and specifiable delay in order to recover its service to normal and to prevent the system from any disaster.

The unpredictable disturbances could be caused by traction equipments faults, communication link failure or excessive boarding times that are beyond the control of railway engineers. While system safety is assured through software and mechanical fail-safe interlocks, the disturbances can affect the smooth functioning of the system. In this research the focus is on the disturbances caused by the excessive boarding time because of the sudden increase of passenger demand. Without any adjustment on the scheduling factors such as dispatch time, dwell time or coast level, the delay to one train may eventually slow down the trains behind and hence magnify a minor disturbance into a total system failure. Hence, Controllers in MRT monitor and control system are typically organized hierarchically to ensure its real time reliability. One or more digital controllers at the lowest level directly control the physical devices. A second level controller typically performs scheduling functions to achieve a higher-level goal. In particular, it tries to find one of the most desirable timings for individual trains among all possible timings that meet the constraints of the system and optimises energy consumption. A timely rescheduling to react and minimize any negative effect of the disturbance keeps the system within its acceptable operating range.

The second level control structure for the proposed real time system is designed as shown in Figure 4.2.

Do the following in each 0.125 second cycle:

1. Validate sensor data and select data source, such as trains' distance, speed, and current, and store the real time data into buffers.

2. In the presence of failures (data transmission failure or exceptional data, etc), reset the system to its last safe schedule configuration.
3. Do the following computations, once every 4 cycles (0.5 second):
   - Perform data normalization and coordinate transformation
   - Do speed regulation and control with ATP according to the two speed parameters: Maximum Safe Speed and The Target Speed.
   - Encode and package the control commands and parameters
4. Do the following computations, every 8 cycles (1 second) using the data produced by task 3.
   - Update the database with real-time train operational data and display the information to train controllers on monitor
   - Retrieve passenger flow data, detect disturbance, and update the database
5. Do the online rescheduling algorithms every 240 ~ 640 (30~80 seconds) cycles using data provided by task 4 to optimise terminal dispatch time, inter-station dwell time and coasting level in terms of regularity, energy consumption, train and platform congestion satisfying various constraints.
6. Output command
7. Wait until the beginning of the next cycle

Figure 4.2 Software control structure of real time MRT control system

The ATO system indirectly controls the embedded components in lower levels of the control hierarchy. The system needs to compute some control laws to monitor and control the transitions between different schedules as well as algorithms for estimating and predicting times between stations. These values correlate directly with the smooth operation of the trains in dispersing the passengers from an overloaded station. These time-critical computations tend to be simpler and more deterministic and have periods in the order of seconds.

ATO system detects the disturbances arising in the system and then updates the system parameters according to the online sampled real-time data. The rescheduling processes are triggered again within a specified time from the detection of the disturbances. With updated system information, the Main computer in OCC has minutes to do rescheduling to achieve the optimal solution or otherwise accept sub-optimal solutions when there is insufficient time. By restricting the allowable computational time of the rescheduling analysis, the best available solution is always posted in real time to the controller for an action to be taken.

4.4 Online Database Management

ATR is an information-based system in which one relational database is created to organize various operational data that supply ATR with prompt and accurate information about the monitored railway scenario. Using a database greatly enhances the ease of such complex data management since the program can handle single-database files more easily than it does hundreds of different data files. Moreover, because the data is arranged logically in a relational database in a tabular form, it makes the job of programming and data retrieval and updating more efficient and systematic. Three types of operational data are provided by railway industry to simulate the train movement based on real-life scenarios.

4.4.1 Train Movement Profiles

These profiles can be generated by an 'Interstation Train Movement' program developed by C.S.Chang [13], which generates inter-station single train run-time profiles for energy and signaling studies.

4.4.2 Passenger Flow during Busy Time or Less Busy Time

Accurate passenger forecasts provide the essential information for evaluating online control under all conditions. With the advancements in communication networks and the introduction of the Automatic Fare Collection (AFC) system, predicting the passenger flow becomes available. In this project, the demand of passengers during different operating hours, such as normal, peak, and sudden sharp rise is simulated, and the approximate passenger flow data is used.

During simulation, passengers enter the platform i at a rate of r_i. Given that the number of passengers on the platform i at time k is $PF_{i,k}$, the number of passengers on the platform i at time t_k is given by [16]:

$$PF_{i,k+1} = PF_{i,k} + r_i \times \Delta t - u_{m,k} \times \Delta t \times \alpha_k \qquad (4.1)$$

$$TP_{i,k+1} = TP_{i,k} + u_{m,k} \times \Delta t \times \alpha_k - l_{m,k} \times \beta_k$$

where $TP_{i,k+1} \leq$ passenger carrying capacity of train m, $u_{m,k}$ is the rate of passenger boarding the train from the platform, $l_{m,k}$ is the rate of passengers leaving the train. α_k and β_k are scaling coefficients to the passenger flow of passengers boarding the train and the passengers alighting the train respectively. α_k and β_k can be estimated by onsite passenger survey.

4.4.3 Predefined Timetable

This is predetermined using experience and knowledge of prevailing conditions or off-line optimization. This timetable is accessed by ATR in read-only mode during traffic control operations. Predetermined timetable is the products of an off-line optimization on the train scheduling on different operational hours. Originating from the initial optimized timetable, the simulation software provides a platform for the investigation and fine-tuning of the schedules to further the optimization into an online mode.

4.5 Results from Simulation

4.5.1 Study System

The simulation covers a medium-sized MRT line 29 passenger stations, and 30 trains in a simulation time of 2.5 hours. Layout of the simulation is shown in Fig 4.2

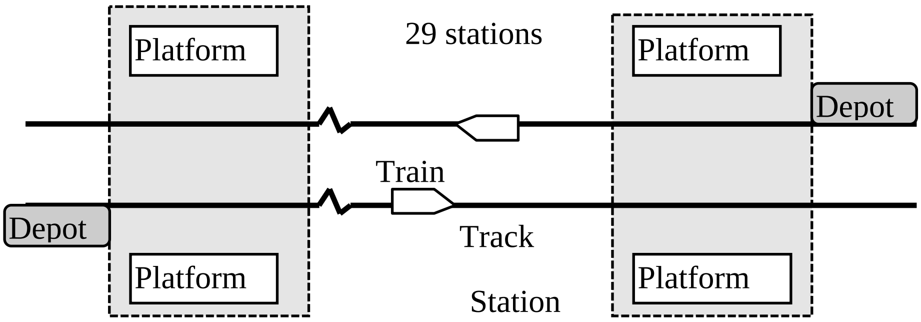


Fig 4.3 Layout of study system

After initialization, the ATS module is simulated to dispatch the trains according to the pre-defined or active timetable. The ATP provides over-speed protection, and the ATO regulates the movement of train.

4.5.2 Simulation Output

Typical simulated speed-time and distance-time profiles are shown in Fig 4.3 and Fig 4.4. Other information such as coast level in each inter-section run, dwell time at each passenger station and other displays is described in Section 4.2.1.

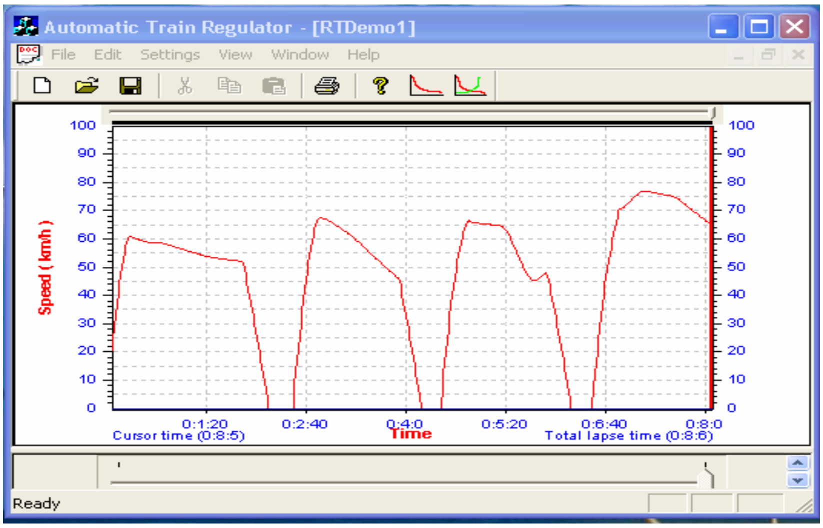


Fig 4.4 Speed-time profile under normal condition

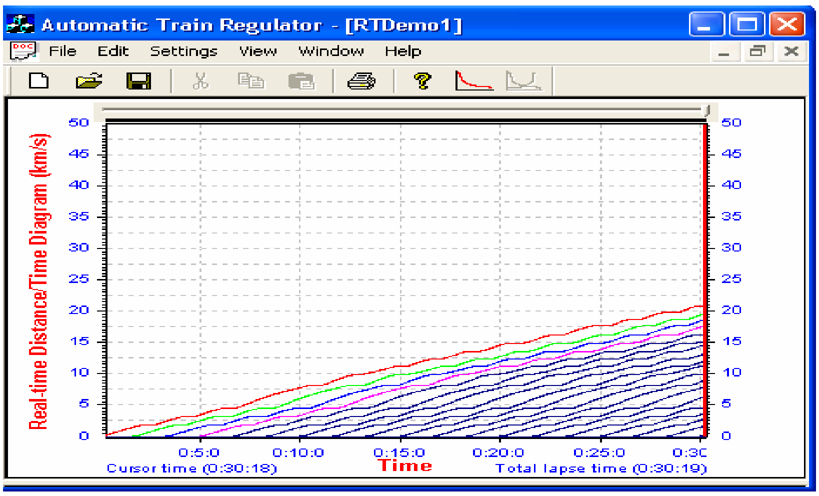


Fig 4.5 Distance-time graph of trains operating with a 2 minutes time margin

Fig 4.5 illustrates the passenger congestion on a particular platform under normal conditions, where passengers enter the platform at rate 1.2 person/sec.

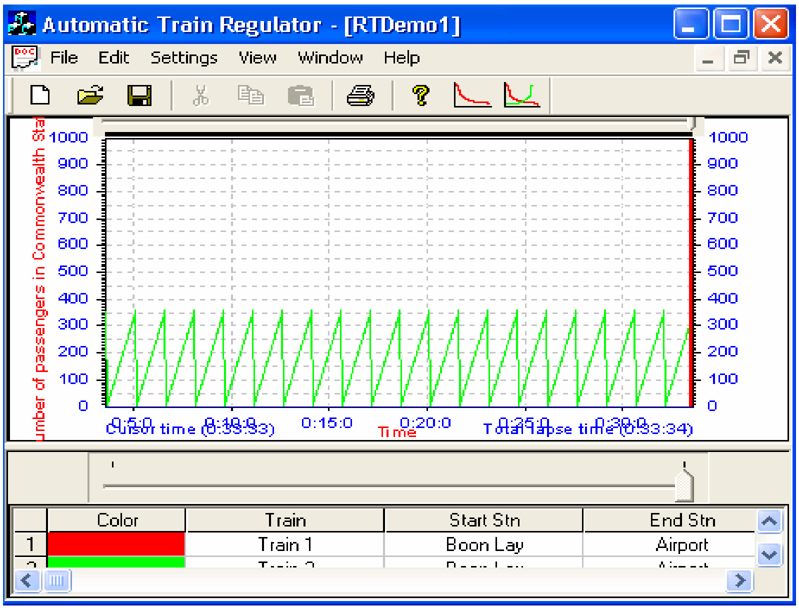


Fig 4.6 Platform congestion under normal condition

Fig 4.6 illustrates the passenger congestion on three trains, in which the congestion on trains grows to the top on stations at the central district of the city and then is appeased until the destination depot.

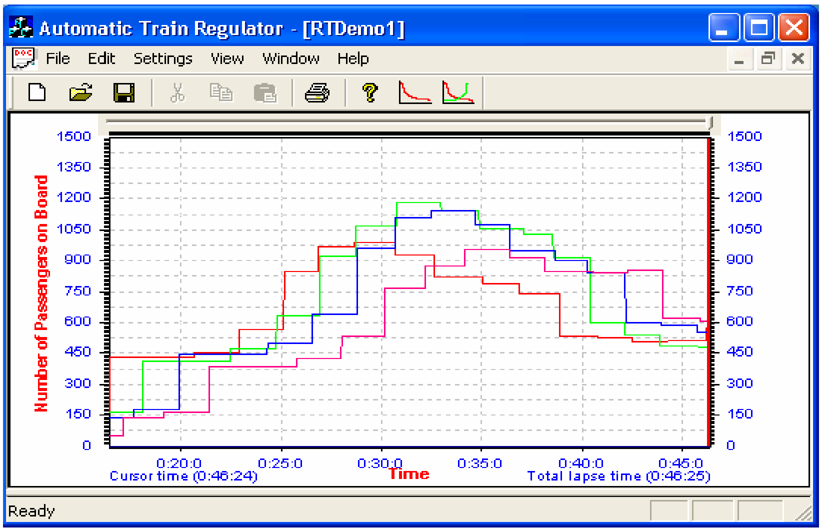


Fig 4.7 Train congestion under normal condition

4.6 Summary of the Simulation Studies

The simulation studies are conducted to study the performance of MRT systems under different operating conditions, which is evaluated by regularity, congestion on both platforms and trains, and energy consumption.

The regularity in service is affected by the intervention of the ATO and ATP systems. By maintaining an appropriate headway between successive trains, intervention from the ATO and ATP systems can be prevented. The number of passengers at the platform can be lowered if trains arrive more frequently. Hence, the congestion problem can be reduced by decreasing dispatch interval. In the operation of the MRT system, the number of trains in service has to meet the passenger demand. As the high number of trains in service will lead to high costs, and the low number of trains in service will lead to passenger dissatisfaction, the dispatch frequency has to be adjusted so that optimal utilization of trains is achieved. It is apparent that all these performance indices are affected by the time margin between successive trains.

In this research project, a Decision Support System equipped with evolutionary algorithms is devised to improve the performance of the MRT system. The on-line rescheduling strategy improves the performance of the MRT system by making adjustment to headway. By rescheduling the dispatch of trains from the terminal station, and by rescheduling the departure of trains from passenger stations, the headway between successive trains can be adjusted. Details on these rescheduling algorithms will be presented in Chapter 5.

CHAPTER 5

APPLICATION OF EVOLUTIONARY ALGORITHMS OPTIMIZATION WITH FUZZY FITNESS

In Chapters 3 and 4, the two ATR train-rescheduling algorithms at OCC and LPU and the simulation mechanism are outlined. In this Chapter, the design and implementation of evolutionary algorithms for decision-making in ATR are discussed in detail.

5.1 Introduction of Evolutionary Algorithms

The pioneer work on evolutionary strategy (ES) [17] was developed in 1964 at the Technical University of Berlin by Rechenberg and Schwefel as an experimental optimization technique. Evolutionary algorithms (EAs) are global stochastic search algorithm whose search methods model some natural processes: genetic inheritance and Darwinian strife for survival. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution without early convergence to local optima. Each individual is evaluated to find its fitness by substitution into the fitness function, which measures the quality of the individual. A new set of individuals is created by the process of selecting individuals based on their fitness and breeding them together using operators such as selection, recombination, mutation and immigration.

In evolutionary algorithm (EA) and some other parameter optimization techniques, a set of variables is optimized to maximize some target such a profit or to minimize cost or error. They are treated as a black box with a series of control parameters, and the only output of the black box is a value returned by an evaluation function indicating how well a particular combination of parameter settings solves the optimization problem.

Evolutionary algorithms have become established as the method for solving optimization problems that are too complex to be solved by exact methods, such as linear programming and gradient search. Typically, evolutionary algorithms are applied on nonlinear problems where it is not possible to treat each parameter as an independent variable that can be solved in isolation from the other variables. There are interactions such that the combined effects of the parameters must be considered in order to maximize or minimize the output of the black box. Dynamic programming is an option for solving high-dimensional and nonlinear discrete optimization problem. It breaks down a multistage decision process containing many interdependent decision variables into a series of sub-problems, each containing only a few. Hence the common feature of the problems that could be solved by dynamic programming is that they can be divided into stages with a decision required at each stage. If the train rescheduling problem discussed in this thesis is decomposed into subproblems and all combinations of possible solutions are examined in the solving process, the combination numbers are so huge that the optimum combination cannot be solved in practical process time which constrains this online optimization problem. Therefore, EA as a proven technique is more efficient in solving such kind of optimization problems with large number of possible solutions.

In real-life problems, functions of many variables have a large number of local minima and maxima. EAs differ from more traditional optimization techniques in that they involve a search from populations of individuals instead of a single point. This way the search is performed in a parallel manner, covering a wider solution space. Moreover, many search techniques require some not readily available information in order to work properly. For example, the Hill climbing method requires derivatives of the fitness function, which is fairly complex for a multi-parameter function, in order to guide the search direction. To perform an effective search, EAs only require objective or fitness function values associated with individual strings. Once the EAs know the current measure of fitness about a point, the values can be used to continue searching for the optimum.

In the industry, control systems are usually designed by using one of two general approaches. The first approach is the mathematical model-based approach. However, in a large number of instances, the system to be controlled is too complex to be understood and modeled, or is time varying. The lack of system information makes the mathematical modeling difficult or completely impossible. To deal with this problem, the control strategy and control system elements are selected based on the knowledge, experience and insight of experts of the processes for evaluation. In the event that only expert knowledge or experience rather than a mathematical input-output relationship is available, the second approach, knowledge or rule based controller such as fuzzy logic controller, is a possible alternative to build up control systems.

For the online train rescheduling problem presented, the number of control parameters and their relationship with the objectives are decided by real time information collected from the simulated system. Hence, with a flexible length solution set, the EA provides a natural representation for this problem. Fuzzy logic algorithm interpolates the action of any individual rule from the available finite rule base using a choice of fuzzy logic operation and defuzzification method.

In this research, the effectiveness of two EA-based optimization methods (DE and GA) in solving the online rescheduling problem is compared. In each test, one of the EA-based methods is integrated into the online rescheduling strategy to do multi-objectives optimization based on the analysis of an active timetable.

This chapter is organized into 6 sections. Section 5.2 specifies the system variables for optimization and coding of control variables into chromosome. Section 5.3 defines optimization objectives and presents a fuzzy expert evaluation system for calculating fitness of individuals in evolutionary algorithms. Performance indices and rules base are created according to human expert experiences. Section 5.4 and 5.5 present the design and implementation of the GA and DE accordingly for run-time train scheduling. Section 6 gives the choice of the look-ahead time window and results out of optimization.

5.2 Application of Fuzzy Evolutionary Algorithms in Online Rescheduling Strategy

5.2.1 Objectives of Optimization

As stated in the Chapter 3, ATR assists railway operator with train rescheduling by aiming for the operational goals of passenger satisfaction and operational efficiency under operational safety constraints. The performance indices relating to the operational goals of passenger satisfaction and operational efficiency include regularity of service, energy consumption, train congestion and platform congestion.

In the EA-based ATR, a set of control variables is fed through analysis of active timetable stored in memory. The number of trains being included in the current time window governs the number of control variables. A population of candidate solutions are generated and applied to the active timetable to predict the fuzzy performance indices (Section 3.1.1) of all trains included in the time window. . The EA-based optimization then finds the near-optimal solution aiming for no conflicts with the following:

1. Average performance index of regularity
2. Performance index of overall energy consumption
3. Average performance index of platform congestion
4. Average performance index of train congestion

5.2.2 Fuzzy Fitness [18]

EAs associate to each individual a value corresponding to the fitness function. Based on this fitness, some of better candidates are chosen to seed the next generation by applying recombination and mutation.

In case of multi-objective optimization problems, there is often a dilemma as how to determine if one solution is better than another because of multiple conflicting objectives.

The weighted sum approach is commonly used to aggregate the objectives together using different weighting coefficients for each one of them. The strength of this method is its efficiency. However its main disadvantage is the difficulty in determining appropriate weights while there is no enough information available about the problem.

A fuzzy expert evaluation system is introduced in the ATR to take advantage of the experts' perceptions and reasoning in evaluation of solutions. The four performance indices defined above are aggregated by the fuzzy expert evaluation system to generate a fitness value that reflects the expert's evaluation of the corresponding solution. The structure of Fuzzy Expert Evaluation system created for ATR is illustrated in the Fig 5.1.

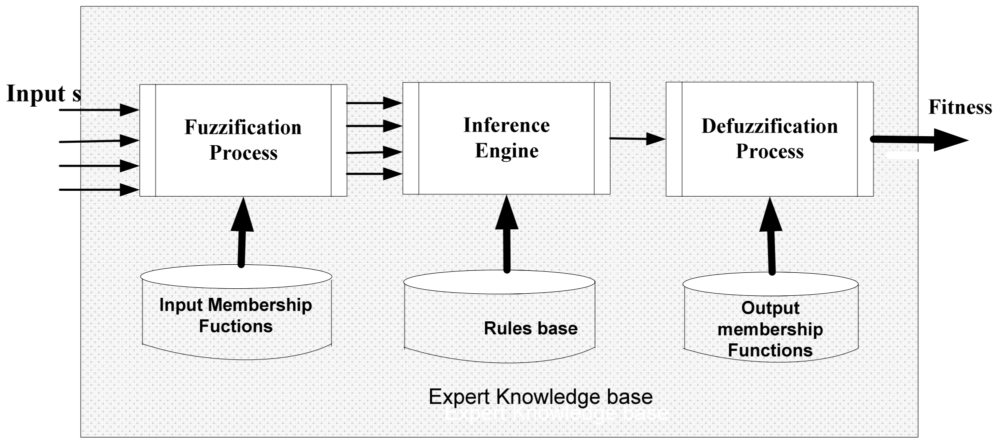


Fig 5.1 Structure of fuzzy expert evaluation system

5.2.3 Fuzzy Inference Rule Base

Fuzzy inference rule base contains fuzzy inference rules, which are usually characterized by a set of linguistic statements generated by human experts. The fuzzy inference rule base is initially constructed by the 'rule of thumb' human intuition and heuristics, and then trained and tested with measured training data through trial and error. In this project, all the possible combinations of fuzzy rules have been tried out in the simulation to come out with the fuzzy inference rule base that gives the best optimization performance.

Twelve fuzzy inference rules are used for evaluating the performance for all trains in the time window. The membership function for each input estimates each objective's values. The antecedent and the consequence of the fuzzy rules indicate combinations of objectives and human reasoning for the associated combination:

R1: **IF** regularity is OK **AND** energy is Low **AND** train congestion is Satisfactory **AND** platform congestion is Satisfactory **THEN** overall performance is Good

R2: **IF** regularity is OK **AND** energy is Normal **AND** train congestion is Satisfactory **AND** platform congestion is Satisfactory **THEN** overall performance is Good

R3: **IF** regularity is Early **AND** energy is Normal **AND** train congestion is Satisfactory **AND** platform congestion is Satisfactory **THEN** overall performance is Normal

R4: **IF** regularity is Late **AND** energy is Normal **AND** train congestion is Satisfactory **AND** platform congestion is Satisfactory **THEN** overall performance is Normal

R5: **IF** regularity is OK **AND** energy is High **AND** train congestion is Satisfactory **AND** platform congestion is Satisfactory **THEN** overall performance is Normal

R6: **IF** regularity is OK **AND** energy is Normal **AND** train congestion is Empty **AND** platform congestion is Satisfactory **THEN** overall performance is Normal

R7: **IF** regularity is OK **AND** energy is Normal **AND** train congestion is Satisfactory **AND** platform congestion is Empty **THEN** overall performance is Normal

R8: **IF** regularity is Early **AND** train congestion is Crowded **AND** platform congestion is Satisfactory **THEN** overall performance is Bad

R9: **IF** regularity is **AND** energy is **AND** train congestion **AND** platform congestion is **THEN** overall performance is Bad

R10: **IF** regularity is Early **AND** train congestion is Satisfactory **AND** platform congestion is Crowded **THEN** overall performance is Bad

R11: **IF** regularity is Late **AND** train congestion is Crowded **AND** platform congestion is Satisfactory **THEN** overall performance is Bad

R12: **IF** regularity is Late **AND** train congestion is Satisfactory **AND** platform congestion is Crowded **THEN** overall performance is Bad

More fuzzy rules generated could be added to the inference engine from expert knowledge or human experience, when necessary.

The equivalent rules for fuzzy logic operation using the root-sum-square method are:

R1: μ_{Good1} =min (μ_{RO} , μ_{EL} , μ_{TS} , μ_{PS}) (5.1)

R2: μ_{Good2} =min (μ_{RO} , μ_{EN} , μ_{TS} , μ_{PS}) (5.2)

Then $$\mu_{Good} = (\mu_{Good1}\text{^}2 + \mu_{Good2}\text{^}2)^{0.5} \qquad (5.3)$$

Similarly $\mu_{Normal,}$ μ_{Bad} can be calculated:

$$\mu_{Normal} = (\mu_{Normal1}\text{^}2 + \mu_{Normal2}\text{^}2 + \mu_{Normal3}\text{^}2 + \mu_{Normal4}\text{^}2 + \mu_{Normal5}\text{^}2)^{0.5} \qquad (5.4)$$

$$\mu_{Bad} = (\mu_{Bad1}\text{^}2 + \mu_{Bad2}\text{^}2 + \mu_{Bad3}\text{^}2 + \mu_{Bad4}\text{^}2 + \mu_{Bad5}\text{^}2)^{0.5} \qquad (5.5)$$

μ_{GD} , μ_{NM} and μ_{BD} are membership values for overall performance corresponding to linguistic levels Good, Normal, and Bad. The output membership function is shown in Fig 5.2.

5.2.4 Fuzzy Fitness after Defuzzification

Defuzzification is used to obtain a crisp fitness output t by combining the results of the above inference process and then computing the "fuzzy centroid" of the area. The weighted strengths of all output member functions are multiplied by their respective centers of each output membership function as shown in Fig 5.2, and then summed. Finally, the resulted sum (area) is divided by the sum of the weighted member function strengths and the result is taken as the crisp output:

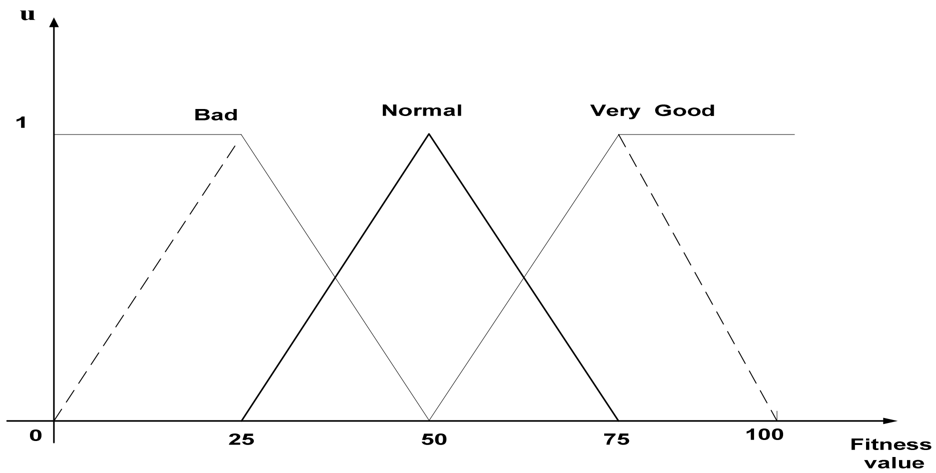


Fig 5.2 Output fuzzy performance index membership function

$$Fitness = \frac{Bad\_Centre \times \mu_{Bad} + Normal\_Centre \times \mu_{Normal} + Good\_Centre \times \mu_{Good}}{\mu_{Bad} + \mu_{Normal} + \mu_{Good}} \tag{5.6}$$

5.3 Coding Scheme and Control Variables for Evolutionary Algorithms

ATR carries out scenario analysis (Section 3.2) in every time window. Whenever a conflict is detected, ATR synthesizes a dynamic control vector for a new active timetable. The vector contains the dwell time, coast level, and dispatch interval to be adjusted within the time window.

In the proposed DE and GA optimization, a population is a collection of solutions each of which is represented by a chromosome made up of a set of genes. The coding of variables has a great impact on search performance. Whilst the most commonly used representation in GAs is the binary alphabet {0, 1}. An integer representation [19, 20] is used in both DE and GA in this research because of following reasons:

1). Both the dispatch time and dwell time are varied in steps of 1sec whereas the coast level is an integer of 0, 1, or 2.

2). The use of integer-valued genes can increase the efficiency of the GA as there is no need to convert chromosomes to phenotypes before each function evaluation; hence, less memory is required as efficient integer internal computer representation can be used directly.

In chromosomes, each gene has two attributes, namely: dwell time and coast level, except the last gene that is left to the dispatch time included in the working time window. The length of the chromosome representation for the train schedule solution may vary when the time window slides from one to another. An example is illustrated in Fig 5.3, which is the chromosome representation of the train schedule solution in Fig 3.8.

1	Dwell time	Coast level	2	Dwell time	Coast level	. . . . . .	n-1	Dwell time	Coast level	n	Disptach inerval

Fig 5.3 Structure of a chromosome for the train schedule solutions

5.4 Implementation of Genetic Algorithm in Online Rescheduling

GA initially begins search with random points in the solution space then incrementally generate new points by applying operators. These points are pursued is controlled by a probabilistic decision procedure that guides the method into optimal regions of the solution space. Fig 5.4 shows the block diagram that applies the GA to the on line train rescheduling. A review of the basic GA is given in [Appendix B].

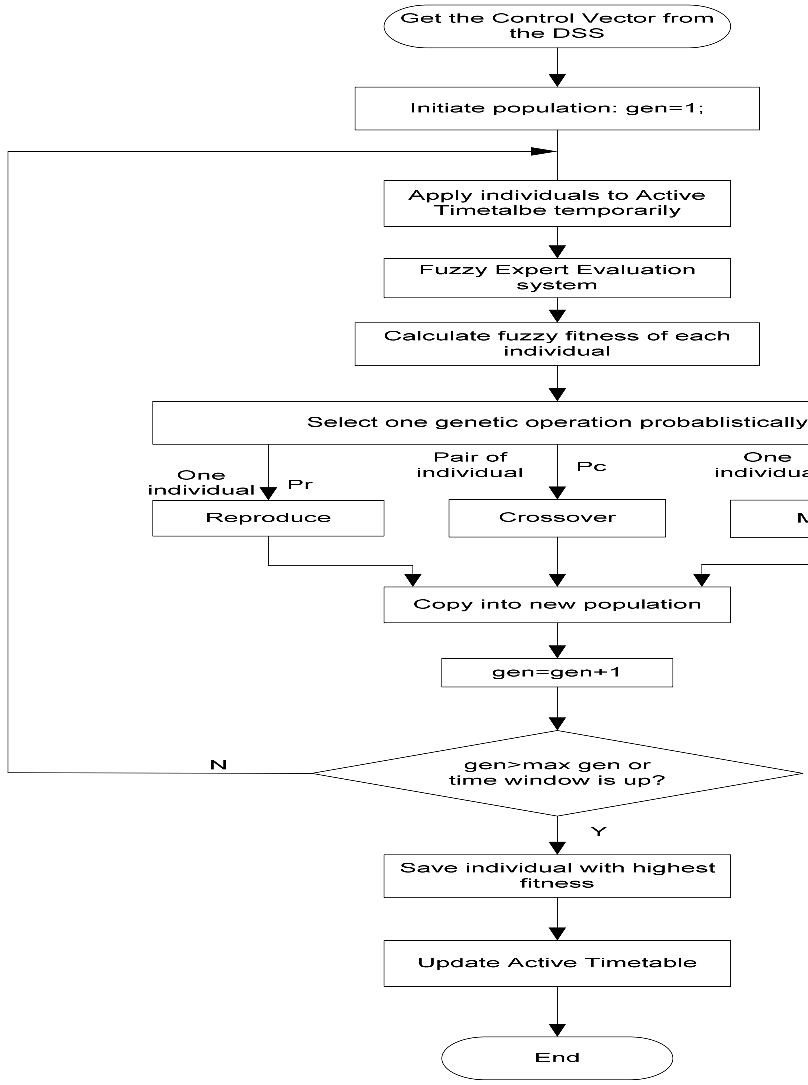


Fig 5.4 Block diagram of the GA-based method for timetable rescheduling

5.5 Implementation of Differential Evolution in Online Rescheduling

Although GA can achieve better performance than the traditional optimization technique for its robustness, reliability, and flexibility in many cases, in some online applications GA cannot exploit its own merits fully within strict time constraints because of its complicated process of genetic operations. Among all the EAs, DE is not only significantly faster at numerical optimization than GA, it is also much more likely to find a function’s true global optimum. DE is exceptionally simple in that its main search engine can be easily equipped into the applications.

Differential Evolution is a parallel direct search method which basically employs the difference of two randomly selected parameter vectors as the source of random variations for a third parameter vector. The crucial idea behind DE is the scheme for generating trial parameter vectors. DE generates new parameter vectors by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population number, the newly generated vector replaces the vector with which it was compared. In addition, the best parameter vector is determined for every generation in order to keep track of the progress made during the optimisation process.

The following steps are implemented in DE's mechanism:

Step 1 After scenario analysis, a dynamic control vector will form the input $\underline{P} = (S_1, S_2, S_3 ... S_n)$ for each time window, where S_i, the combination of Δt_i and l_i is the possible solution for dwell time and coast level at station *i*.

Step 2 Generating an initial population

The initial population is often generated by adding normally distributed random deviations to the nominal solution that is the combination of nominal dwell times and coast levels for the corresponding stations. A population is described in equation:

$\{ \underline{P}_{1,G}, \underline{P}_{2,G}, \underline{P}_{3,G} \dots \underline{P}_{NP,G} \}$

where *NP* is the individual's index number and *G* is the generation number.

As the individuals of the population, $\underline{P}_{1,G}, \underline{P}_{2,G} \dots \underline{P}_{NP,G}$ are described in equation:

$\underline{P}_{1,G} = \{ \quad S_{1,1}, S_{1,2}, S_{1,3} \ldots \; S_{1,N} \quad \}$

(5.7)

$\underline{P}_{2,G} = \{ \quad S_{2,1}, S_{2,2}, S_{2,3} \ldots \; S_{2,N} \quad \}$

(5.8)

……

$\underline{P}_{i,G} = \{ \quad S_{i,1}, S_{i,2}, S_{i,3} \ldots \; S_{i,N} \quad \}$

(5.9)

……

$\underline{P}_{NP,G} = \{ \quad S_{NP,1}, S_{NP,2}, S_{NP,3} \ldots S_{NP,N} \quad \}$

(5.10)

where $S_{i,1}, S_{i,2}, S_{i,3} \ldots \; S_{i,n}$ are the solutions (dwell time and coast level) of the i_{th} individual and N is the number of stations included in the current time window.

Step 3 Calculating the fitness value for each individual, finding the fittest individual $\underline{P}_{Best,G}$ and saving it.

Step 4 Creating a new trail vector

For each vector $\underline{P}_{i,G}$, i=0, 1, 2, …, NP-1, a new vector $\underline{V}_{i,\ G+1}$ is generated according to following mutation scheme:

$$\underline{V}_{i,\ G+1} = \underline{P}_{i,G} + F \times (\underline{P}_{r1,G} - \underline{P}_{r2,G}) + \lambda \times (\underline{P}_{Best,G} - \underline{P}_{i,G}) \tag{5.11}$$

The integer $r1$ and $r2$ are chosen randomly over [0, NP-1]. F is a real and constant factor, which controls the amplification of the differential variation $(\underline{P}_{r1,G} - \underline{P}_{r2,G})$. λ is to provide a mean to enhance the greediness of the scheme by incorporating the current best vector $\underline{P}_{Best,G}$.

Step 5 Determining the new vector for the next generation

Step 5.1 Recombination

To increase the diversity of the new parameter vectors, discrete recombination is introduced, the vector $\underline{u} = (u_1, u_2, ..., u_N)^T$

with $$\underline{u} = \begin{cases} \underline{V}_j & \textit{for } j = < n >_N, < n+1 >_N, ..., < n+L-1 >_N \\ (\underline{P}_{i,G})_j & \textit{otherwise} \end{cases} \quad (5.12)$$

The starting index n is a randomly chosen integer from the interval [0, N-1]. The integer L is drawn from the interval [0, N-1] with the probability $\Pr(L = v) = (CR)^v$. $CR \in [0,1]$ is the crossover probability. The random decisions for both n and L are made anew for each trail vector, so a certain sequence of the vector elements of $\underline{u}$ are identical to elements of $\underline{V}_{i,\ G+1}$, and the other elements of $\underline{u}$ acquire the original values of $\underline{P}_{i,\ G}$

Step 5.2 Selection

In order to decide whether the new vector $\underline{u}$ shall become a population member of generation G+1, it will compete against one population member $\underline{P}_{i,\ G}$. If vector $\underline{u}$ yields a higher fitness value than $\underline{P}_{i,\ G}$, then $\underline{P}_{i,\ G+1}$ is set to $\underline{u}$; otherwise, the old value $\underline{P}_{i,\ G}$ is retained.

Step 6 Repeating steps 4 and 5 for all individuals in this generation and finding the one with highest fitness value.

Step 7 Repeating steps 4, 5 and 6 until a certain number of iterations have been executed or the time window comes to its end, the process will be terminated and the fittest solution from the current generation will be preserved.

5.6 Selection of Control Parameters

There have been a variety of studies on determining the best control parameter values for EAs to optimally balance exploration and exploitation. If crossover and mutation rates are very high, much of the space will be explored, but there is a high probability of losing good solutions and failing to exploit existing schema.

Operational Performances	**GA**	**DE**
Population size	30	30
Coding Type	Integer	Integer
Selection and Reproduction mechanisms	Roulette wheel Crossover (single point, 0.8) Mutation rate Linear decrease from 0.01 to 0)	λ :Linear decrease from 0.1 to 0 F :Linear decrease from 0.35 to 0 Crossover rate:1

Table 5.1 Control parameters for GA and DE

Simple forms of dynamic tuning on control parameters were used in the two EAs because of its simplicity and fast implementation on real time system. This variation allows for a large exploration of the search space at the initial stage (global exploration) and a faster convergence after sufficient iterations (local fine tuning).

In order to be accepted as a reliable optimization technique in the industry, EAs under discussion need to demonstrate reliable performance within a given number of generations with the selected parameters including population size, mutation and crossover rate. In the process of selecting the optimal population size, 20 trial runs of simulations were performed for each population size in a range from 10 to 80 (in steps of 10) with termination condition of 800 generations. The results recorded from each population size were the average of the best fitness value and the average time (the number of generation) at which it reaches the best fitness. The result shows that with a population size of 30 and above, the fitness value achieved from EAs averages at approximately 0.750 without any significant improvement with larger population sizes. However, the larger the population size is, the more time consuming the algorithm would be. Hence, a population size of 30 was selected as it can achieve satisfactory fitness value within time constraints.

Further trial simulations were run for the selection of initial value of mutation and crossover rate for GA or initial value of λ and F for DE. Each simulation was performed using different parameter values. For GA, a mutation rate was chosen from interval [0.001, 0.1] and a crossover rate was chosen from interval [0.1, 1]. For DE, λ was chosen from interval [0.5, 1] and F was chosen from interval [0.1, 1]. The simulation results yielded an array of fitness levels for each combination of parameters. Finally the optimum initial values that presented the highest fitness levels as indicated in Table 5.1 were selected for all simulations in this thesis.

Another important control parameter for the online optimization is the time window size. As discussed, starting from an active timetable, ATR derives a series of control variables from the timing and other operational states of monitored trains within each time window, typically set at 80secs after many tests for striking a balance between computation efficiency and optimisation effort. Before comparing the results of the two algorithms, the selection of the length of time window needs to be explained first.

The experimental tests are conducted under DE with control parameters listed in Table 5.1.

Table 5.2 shows the performances of the ATR under various time window lengths. The selected time window should have three features: 1). The higher the number of trains whose arriving times are within the selected window, the higher will be the quality of optimization. 2) Evolutionary optimization done within the time constraints could get the nearest result to the optimal one, which largely depends on the selection of algorithms and the performance of computer in use. 3) The maximum time-window length is limited by the capacity of information to be monitored promptly.

The optimal fitness is the fitness value to which EAs converge without a time constraint. Average fitness value is the average fitness for a number of experiments that are conducted within a specific time constraint. From the Table 5.2, the maximum number of operational parameters derived from the 80sec time window could reach 22.

Time windows (s)	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Maximum No. of parameters	6	8	11	15	18	20	22	26	30
Average fitness value	0.723	0.842	0.828	0.805	0.769	0.689	0.702	0.602	0.539
Optimal	0.723	0.842	0.846	0.858	0.824	0.758	0.789	0.756	0.720
Percentage (%)	100	100	97.9	93.8	93.3	90.8	88.9	79.6	74.8

Table 5.2 Choice of best look-ahead control window

The average fitness value is also satisfactory, 88.9% to the optimal fitness. To keep the DSS updated with real time operational information, 80sec would be an appropriate time span according to experimental experiences. Therefore, the proposed ATR adopted 80sec as the length of the time window on which the rest of experiments were conducted.

CHAPTER 6

OPTIMIZATION RESULTS FOR ONLINE TRAIN RESCHEDULING AND CONFLICT RESOLUTION OF MASS RAPID TRANSIT SYSTEM

This chapter documents the simulation test of the ATR online rescheduling algorithms. The optimization results of two evolutionary algorithms, GA and DE, which are applied in ATR, are compared against each other in terms of convergence pattern and overall system performances improvement. The robustness and efficiency of the proposed on-line rescheduling strategy under disturbances are validated through six case studies.

6.1 Introduction

Simulations are conducted on the Intel Pentium IV 2.5GMHZ, 512 RAM computer using the 'Automatic Train Regulator' (ATR) simulation package to investigate the performance of railway operation on the Singapore MRT East-West Line (EWL) from Boon Lay (BL) to Changi Airport (CA), with a total of 29 stations accommodating 30 trains operating with or without automatic train regulation in various scenarios. In Section 6.2.1, two evolutionary algorithms, DE and GA, are implemented for comparison in the DSS of the ATR with 8 conflicts involving headway encroachment, to test the effectiveness and efficiency of the proposed online rescheduling strategy under disturbance. Section 6.2.2 presents 6 case studies involving sudden increases in passenger flows. The control parameters that govern the performance of the GA- and DE- algorithms are presented in the Table 5.1. In sections 6.3 and 6.4, four case studies are performed using the preferred evolutionary algorithm to test the robustness of ATR in preventing the system from breakdown under sharp disturbances.

6.2 Study Cases and Test Situations

In all case studies, pre-scheduling is performed to simulate the early-hour operation of MRT system by starting trains from terminal stations and running them according to a pre-defined off-peak timetable. After reaching the steady state, passenger flows at the trains and platforms remain unchanged before the next change in running condition, which can take the form of headway encroachment due to run-time delay of certain train, distributed increases in passenger flow at all stations (peak conditions 1 and 2), or sharp passenger rise at specific station. Data of these running conditions are provided in Appendix C. With either change in running condition, ATR fine-tunes the predefined timetable. ATR divides the study time period into a series of time windows, within which potential conflicts are detected and resolved with near-optimal performance. The online rescheduling algorithm in the DSS is triggered as soon as any potential conflict is detected. A time window having a potential conflict is called a critical time window. In a critical time window, the active timetable is analyzed along with the current state of the railway (regularity, congestion or energy consumption degree etc.) in order to generate a solution with satisfactory overall performance.

6.3 Comparison of Results of Two Evolutionary Algorithms

6.3.1 Resolution of Train Headway Encroachment

Table 6.1 compares the performance of the DE-, GA- and RS (random search)-based optimizations for resolving the eight (8) conflicts created within a critical time window. As detailed in Appendix C, Conflicts 1-6 arise from headway encroachment of trains traveling between two stations S_i and S_{i+1} (equations 3.9 and 3.10), and can potentially cause head-tail collision. Conflicts 7 and 8 arise from headway encroachment at the terminal (equations 3.11-3.13). The second and third columns in Table 6.1 define the initial location and time of occurrence of these conflicts. The RS algorithm is added in Table 6.1 as a reference of comparison. It shows that both GA and DE always achieve the better solution than RS, which reflects their intelligence in searching for the optimal solution. Results clearly favor the DE for achieving the best solution after optimization within 80sec operational time window. The calculation the fitness value of individual solutions is described in the Section 5.2.

Conflict No.	Between stations	Time of Occurrence (s)	Fitness Value (DE)		Fitness Value (GA)		Fitness Value (RS)
			Within 80secs	After convergence	Within 80 seconds	After convergence	
1	**S5↔S6**	**1530**	**0.801**	**0.858**	**0.732**	**0.858**	**0.415**
2	**S6↔S7**	**1737**	**0.776**	**0.776**	**0.731**	**0.776**	**0.523**
3	**S9↔S10**	**2055**	**0.495**	**0.684**	**0.486**	**0.681**	**0.433**
4	**S8↔S9**	**2407**	**0.689**	**0.798**	**0.543**	**0.794**	**0.321**
5	**S4↔S5**	**2806**	**0.884**	**0.917**	**0.661**	**0.917**	**0.316**
6	**S8↔S9**	**3659**	**0.928**	**0.958**	**0.743**	**0.956**	**0.621**
7	**Terminal area**	**4022**	**0.824**	**0.854**	**0.712**	**0.854**	**0.516**
8	**Terminal area**	**4159**	**0.734**	**0.852**	**0.545**	**0.845**	**0.478**

Table 6.1 Comparison of fitness values for DE, GA and RS

Figures 6.1-6.3 illustrate the convergence patterns of the DE and GA algorithm for the 2nd, 5th, and 7th conflicts as in Table 6.1. The best fitness in each generation is captured in these figures. Convergence is assumed when the variation in the best fitness is less than 0.01% within the first 300 generations. The calculation will however be terminated after 1000 generations.

From Figs 6.1- 6.3, both GA and DE converge quickly during the early generations, and smoothly after about 120 generations. Though both algorithms converge to the same values after about 750 generations, GA performs inferiorly to DE. The latter takes 7.5 minutes in average to converge, whereas the former takes 9 minutes. Both algorithms have taken more the size of each time window (80secs) to converge before entering the next time window. Both algorithms are therefore not fast enough for online optimization. DE provides in general better fitness values after a computation time of 80ses.

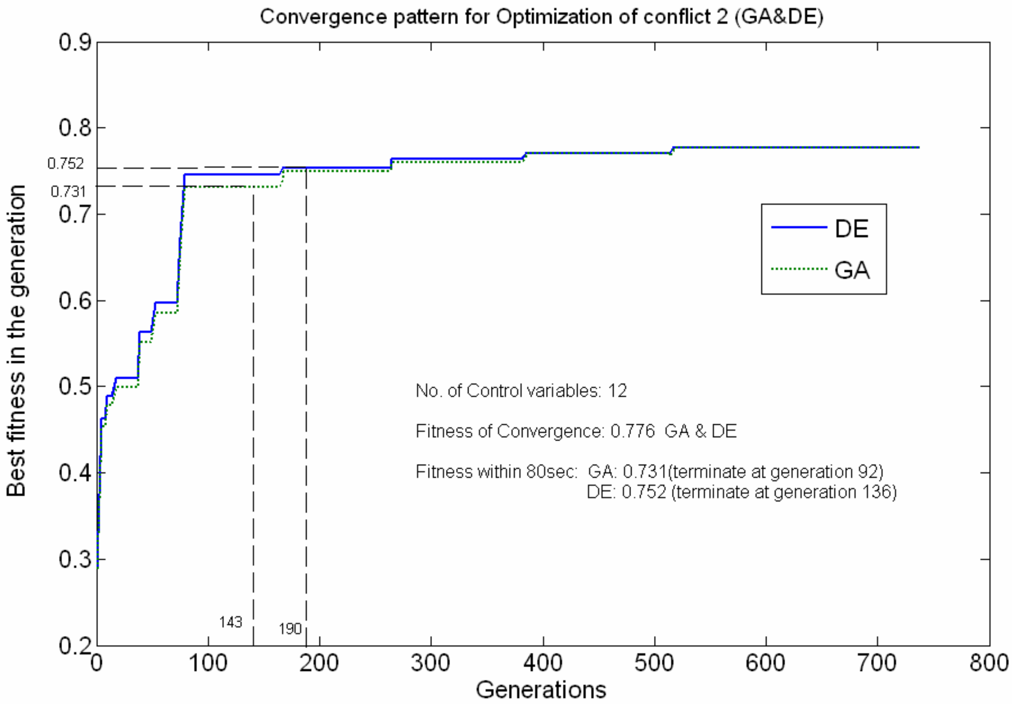


Fig 6.1 Convergence pattern of GA and DE optimization results for resolving conflict 2

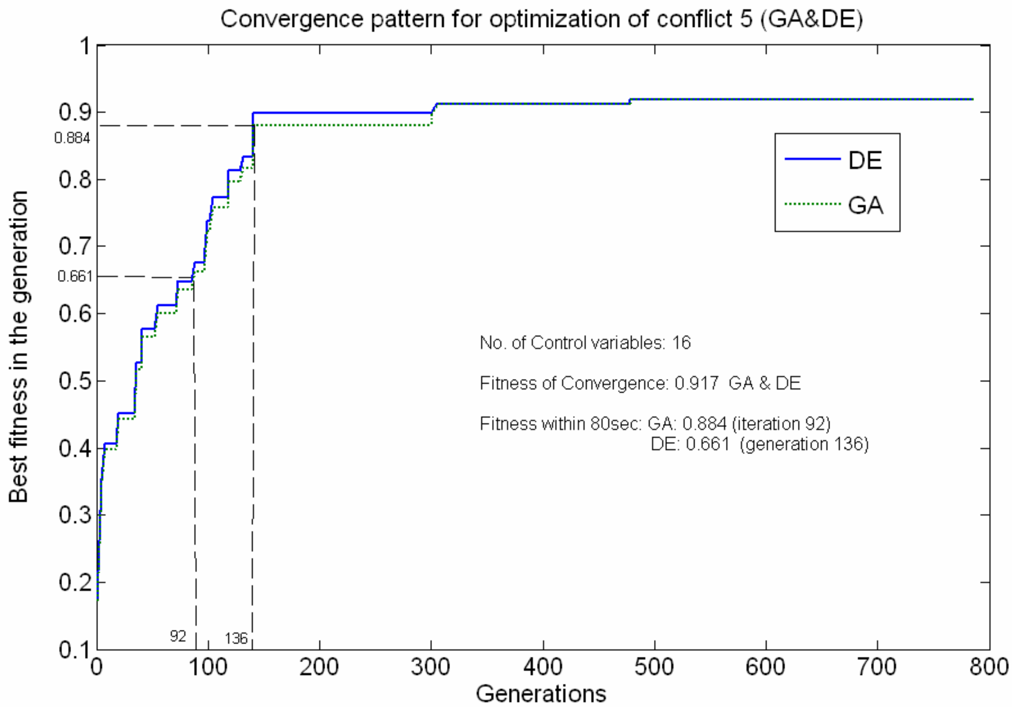


Fig 6.2 Convergence pattern of GA and DE optimization results for resolving conflict 5

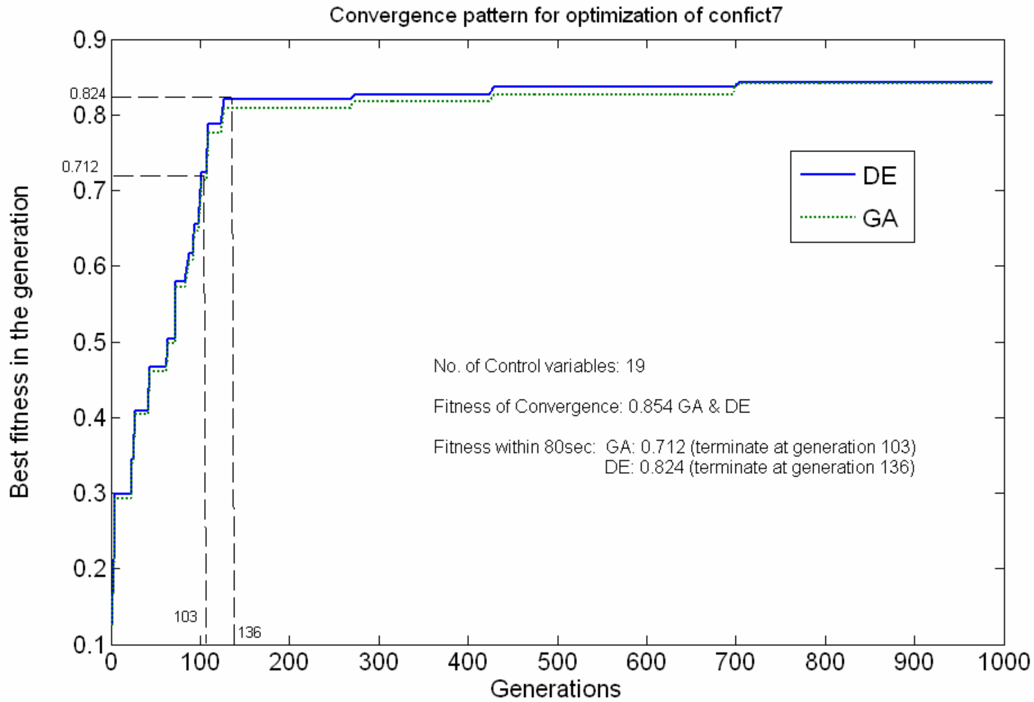


Fig 6.3 Convergence pattern of GA and DE optimization result for resolving conflict 7

In addition to better computational performance as above, DE also enjoys being simpler than GA. Other reasons for favoring DE are:

1) In DE, selection is done after mutation and crossover. So DE does not need the elitism operator as widely used in GA in order to maintain the fittest in each generation. The fittest candidate automatically survives in the subsequent generation. When a certain number of iterations have been executed or the time window comes to its end, the process will be terminated and the fittest solution will be preserved from the current generation.

2) For each crossover and mutation, GA most commonly uses three kinds of crossover namely: single point, two-point, template crossover with a certain probability determined by crossover rate. Two children are born at a time. DE, on the other

hand, adopts uniform crossover. Less than one child is born each time, as the child whose fitness is lower than its parent may be aborted.

3) The GA employs randomly bit flipping in the procedure of mutation, while the DE algorithm adds a scaled differential vector to the target solution. The later is the more convenient source of zero mean Gaussian noise used in the EA as a mutation operator.

6.3.2 Performance of ATR under Different Passenger Flows

Table 6.2 presents the 6 cases of increases in passenger flows, whose flow rates are provided in Appendix C.

Case	Is DE-based ATR scheme incorporated?	Prescheduling (off peak)	Passenger flow data used (duration time)
1	Yes	0 - 15 min.	Peak 1(15-45 min)
2	No	0 - 15 min.	Peak 1(15-45 min)
3	No	0 - 35 min.	Peak 2 (35-60 min)
4	Yes	0 - 35 min.	Peak 2 (35-60 min)
5	No	0-28min.	Sharp Passenger Rise at Specific Station (28-60 min)
6	Yes	0-28min	Sharp Passenger Rise at Specific Station (28-60 min)

Table 6.2 Summary of study cases of increases in passenger flows

ATR is applied with either GA or DE for a study period of 4800 seconds after prescheduling under Peak 1. As seen in Table 6.3, both GA and DE have improved over the performance indices by appropriately degrading train regularity, as compared with the case of no regulation. Both train and platform congestion levels have consistently been moved closer to the desired operating levels with the overall energy consumption reduced.

Operational Performances	**Without ATR Optimization**	**With GA-based ATR Optimization**	**With DE-based ATR Optimization**
Average delay per train (second)	0	55	52
Energy consumption (kwh)	7620	6782	6798
Maximum platform congestion	520	423	406
Maximum train congestion	1480	1270	1256
Average platform congestion performance index	0.424	0.602	0.605
Average train congestion performance index	0.621	0.788	0.782

Table 6.3 Summary and comparison of operational performances for ATR without optimization, with GA-based or DE-based optimization under Peak 1

Figs 6.4 – 6.7 show the comparison between the fuzzy performance indices before and after GA- and DE-based ATR optimization. The simulation results from the previous Fuzzy-control based ATR [5] are illustrated together with GA- and DE-based ATR to show the improvement of the proposed EA-based algorithms. The Fuzzy control-based ATR employs a dwell time and dispatch frequency controller in local stations to determine the optimal schedules for train dwell time and dispatch intervals for individual trains based on the four performance indices.

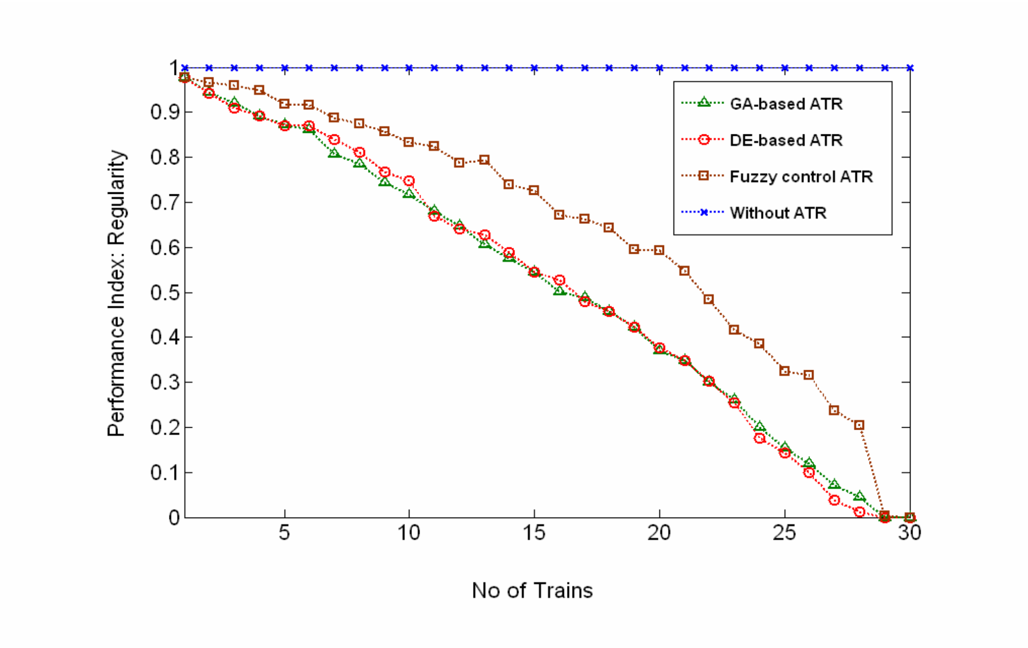


Fig 6.4 Performance index of regularity of 30 trains

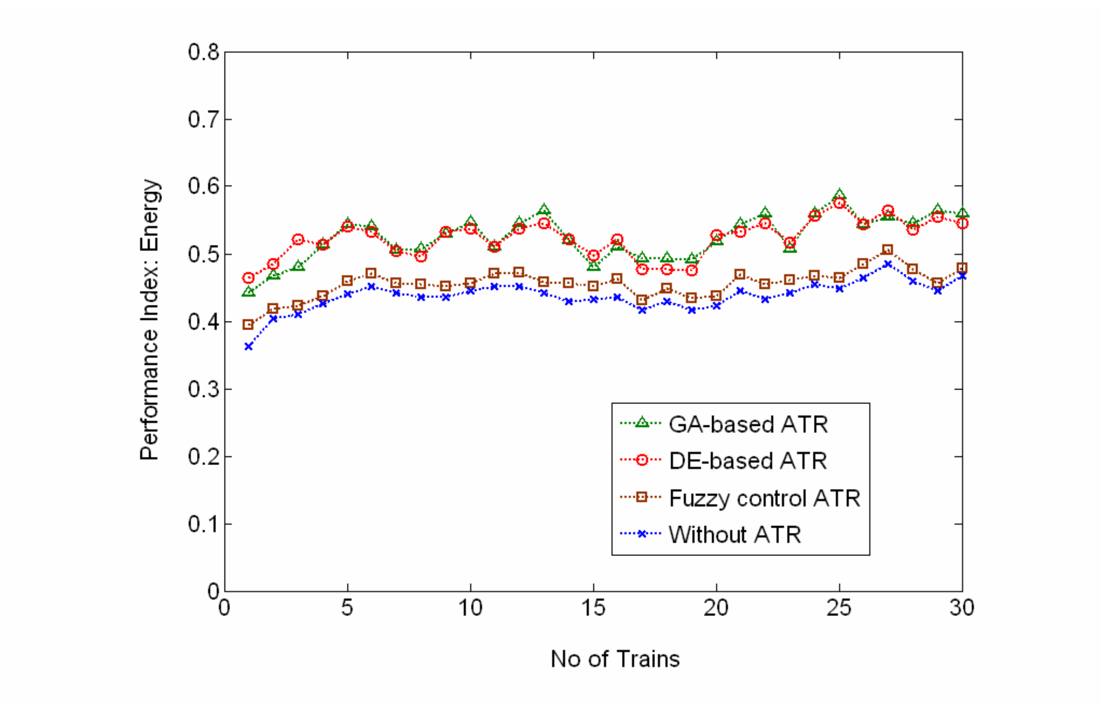


Fig 6.5 Performance index of energy consumption for 30 trains

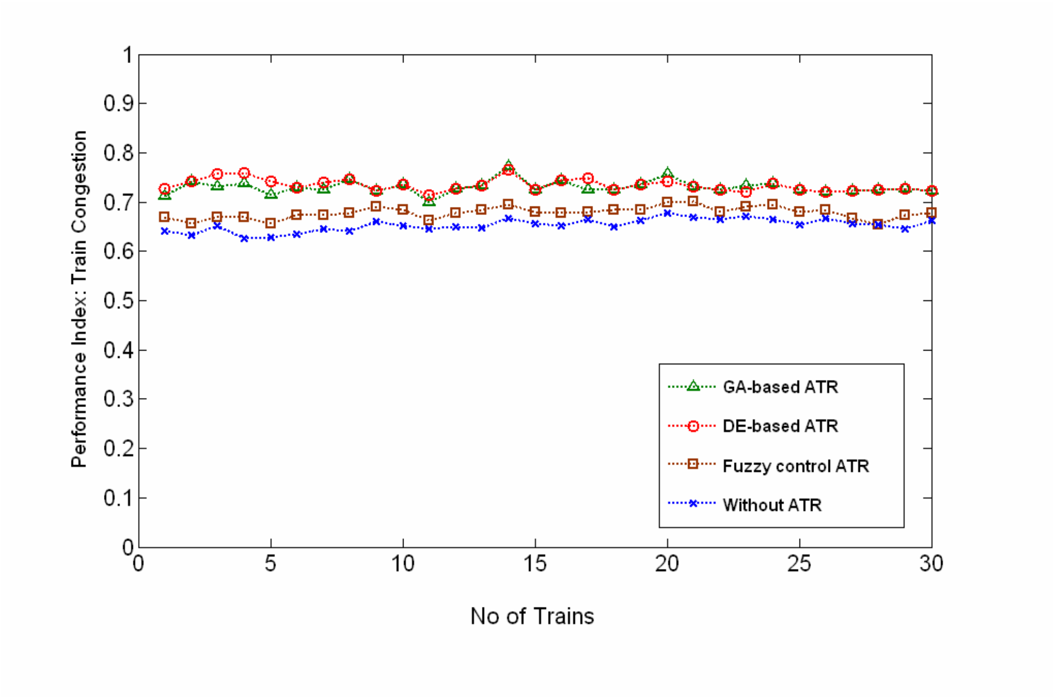


Fig 6.6 Average performance of train congestion for 30 trains

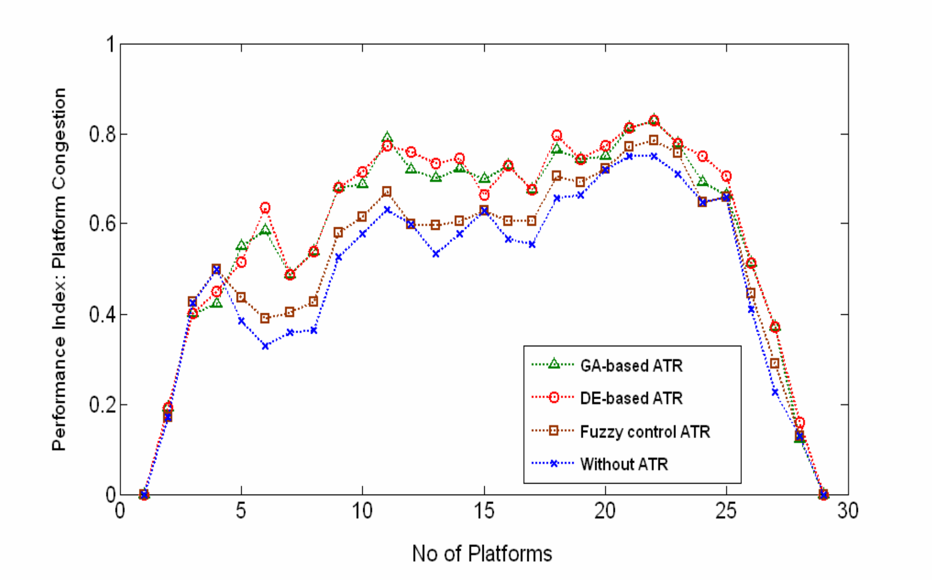


Fig 6.7 Average performance index of platform congestion for 29 platforms

Figure 6.4 shows average performance index of regularity for trains that presents the degree to which the trains deviate from their predefined schedule. The first train always gains better regularity performance than the subsequent ones because the deviation from the predefined schedule accumulates from every train to the next train. Figure 6.5 shows performance indices of the energy consumption for 30 trains after one journey between two depots. Figure 6.6 and Fig 6.7 shows average performance indices of congestion level in 30 trains and 29 platforms respectively.

Results show clearly that in all three optimization methods performance index of regularity is degraded to achieve improvement over the other three performance indices: energy consumption, train congestion and platform congestion. However, since the fuzzy control-based ATR locally optimize the individual train without considering inter-train parameter coupling, the performance improvement made by the previous fuzzy-control method as shown in Figures 6.4-6.7 is significantly less than the ones made by the GA and DE based ATR, which take into account the whole line and optimize operational parameters of the various trains at the same time. Results also show that both DE and GA based ATR reach the same level of effectiveness in improving energy cost and congestion. As DE is preferred to GA for optimizing the ATR (see Section 6.2.1), four case studies (cases 3-6 in Table 6.2) are simulated in the following sections to validate the robustness of the proposed on-line rescheduling strategy.

6.4 Performance of the On-line Rescheduling Strategy after Sharp Rise in Overall Passenger Demand

After off-peak prescheduling (Figs. 3.1-3.5), Peak 2 (Appendix C) is applied in the 35th minute causing huge surges in passenger flows as below.

6.4.1 Performance of Mass Rapid Transit system without DE-optimized ATR Control

Using ATR without DE-based train rescheduling, trains are dispatched according to the pre-defined timetable from terminal station at fixed 180-second intervals. Although being adequate for prescheduling, the 180-second time interval is seen to be unsuitable to Peak 2. Among the four performance indices, train congestion and platform congestion are most severely affected as shown in Fig 6.8 and 6.9.

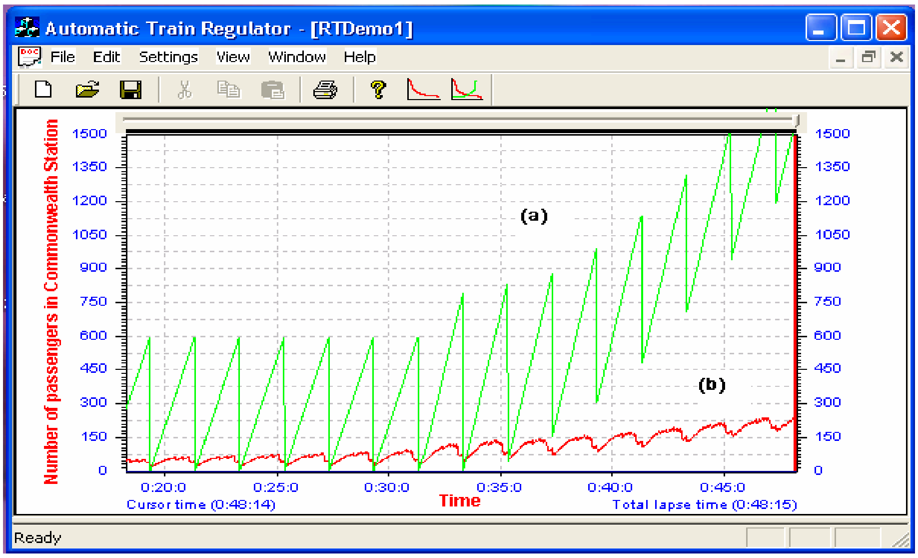


Fig 6.8 Platform congestion after a rise of passenger load across the network in the 35th min with no DE-optimized ATR control

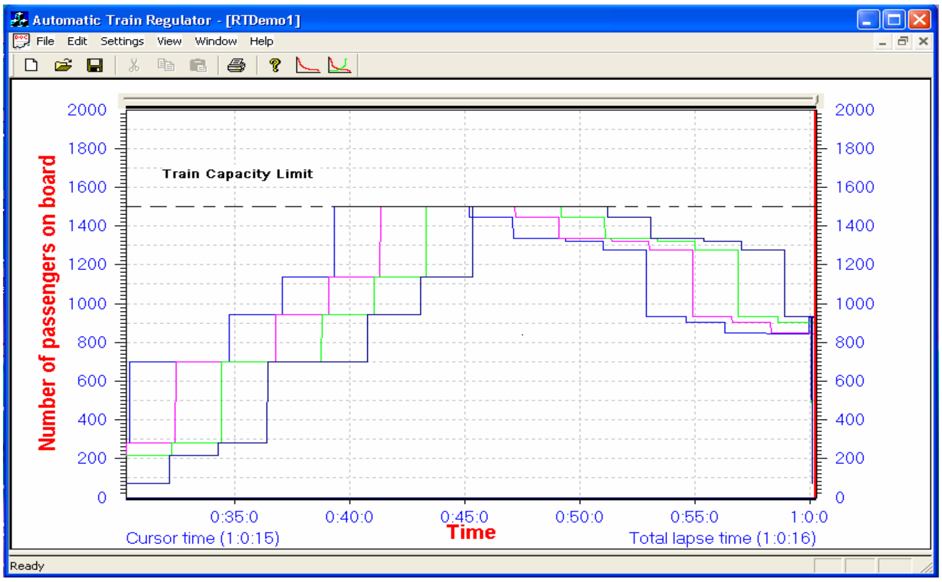


Fig 6.9 Train congestion after a rise of passenger load across the network with no DE-optimized ATR control

Plots (a) and (b) in Fig. 6.8 represent the passenger congestion level at the most congested station and the average platform congestion for the whole MRT line. Fig 6.9 illustrates the passenger on-board for four successive trains under Peak2, which hit the train capacity for part of their journey. This is clearly not desirable as the passenger congestion level in the train builds up.

6.4.2 Performance of Mass Rapid Transit System with DE-optimized ATR Control

DE-optimized ATR fine-tunes the dispatch interval from each terminal station, dwell times at immediate stations and coast levels at each inter-station run. This has effectively prevented the rapid buildups in congestion at both platforms (Fig.6.10) and trains (Fig.6.11) arising from drastic rises in passenger flow (Peak 2) from the 35th minute. Fig. 6.12 shows the variation in dispatch interval, which has been gradually decreased from 180 seconds to 90 seconds.

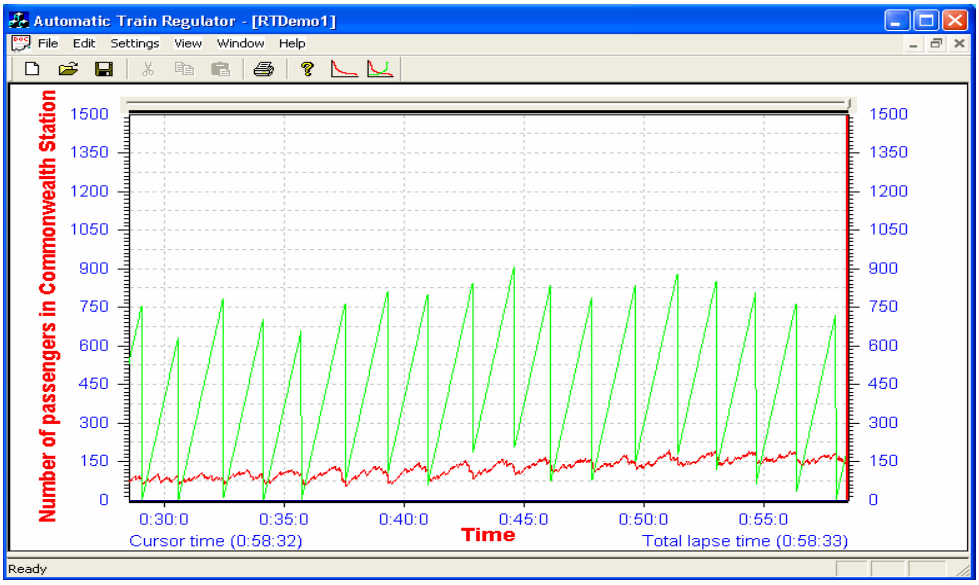


Fig 6.10 Platform congestion after a rise of passenger load across the network with DE-optimized ATR

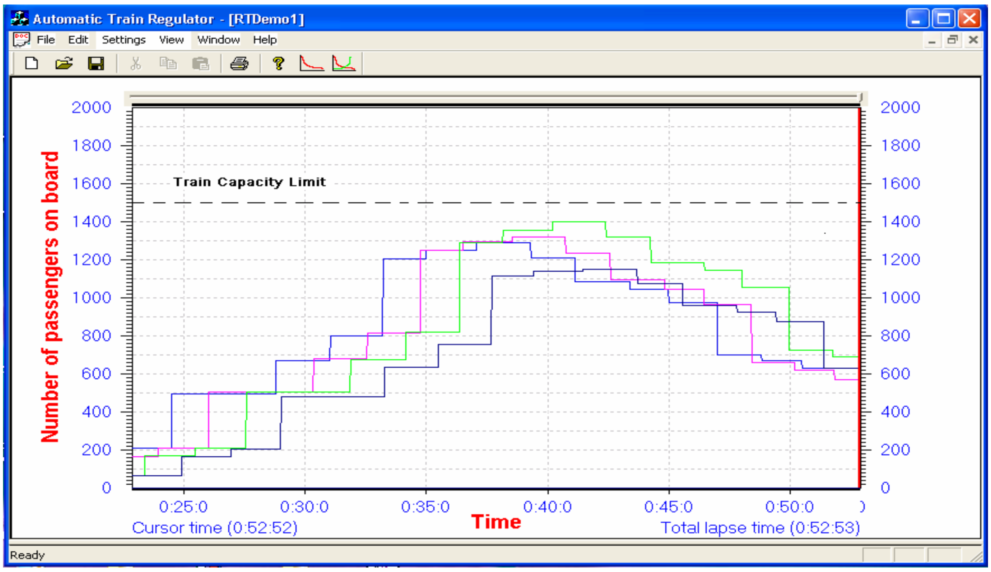


Fig 6.11 Train congestion after a rise of passenger loads across the network with DE-optimized ATR Control

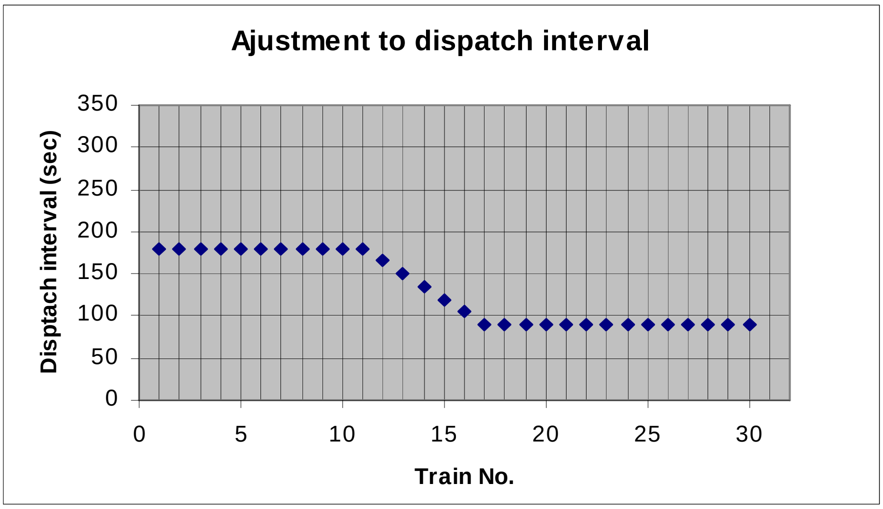


Fig 6.12 Dispatch interval of trains before and during disturbance across the network

Fig 6.13 indicates the corresponding variations in dwell times at intermediate stations showing a well coordinated solution for improving platform and train congestions, regularity and energy consumption.

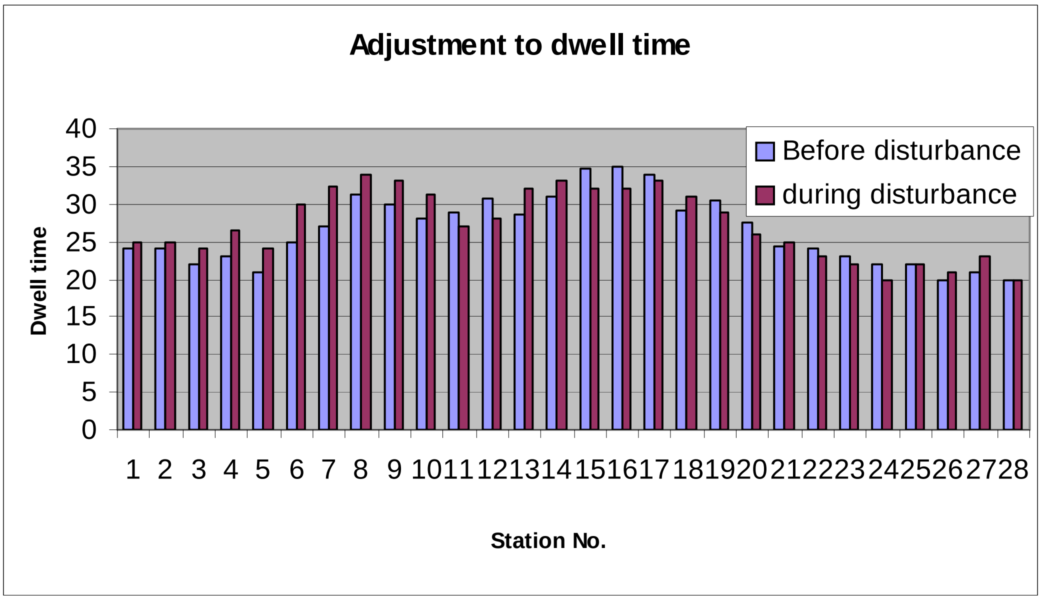


Fig 6.13 Dwell times of stations before and during disturbance across the network

6.5 Performance of the On-line Rescheduling Strategy after Sharp Rise in Passenger Demand at Specific Station

Such scenario can happen in MRT system with large-scale activities, such as exhibitions or sports matches, which .is studied by applying a sharp rise in passenger demand (7.2 persons per sec) at JUR station at the 28 minutes after a period of off-peak operation (2.45 persons per sec). The platform congestion levels at JUR station without DE-optimization are shown in Fig. 6.14. The DE-optimized platform congestion levels are shown in Fig. 6.15, showing remarkable improvements.

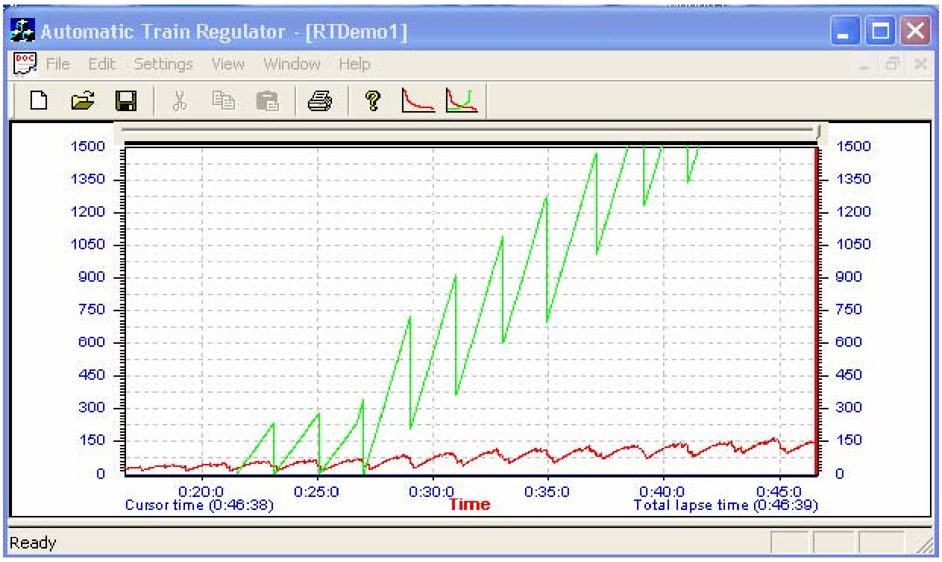


Fig 6.14 Number of passengers on JUR station platform during a sudden disturbance without DE-based ATR

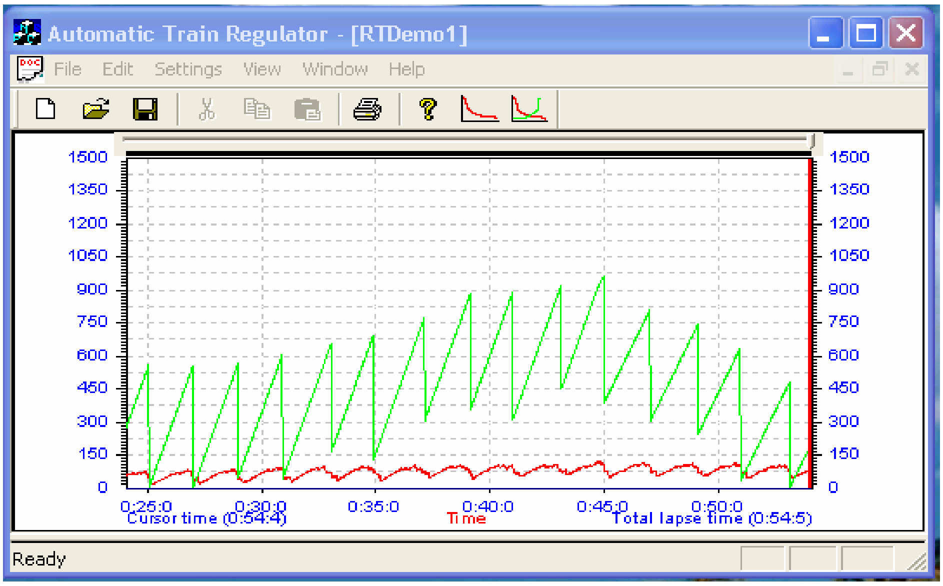


Fig 6.15 Number of passengers on platform of JUR station during a sudden disturbance with DE-based ATR

Correspondingly, the DE-optimized dispatch intervals and dwell times are included in Figs. 6.16 and 6.17.

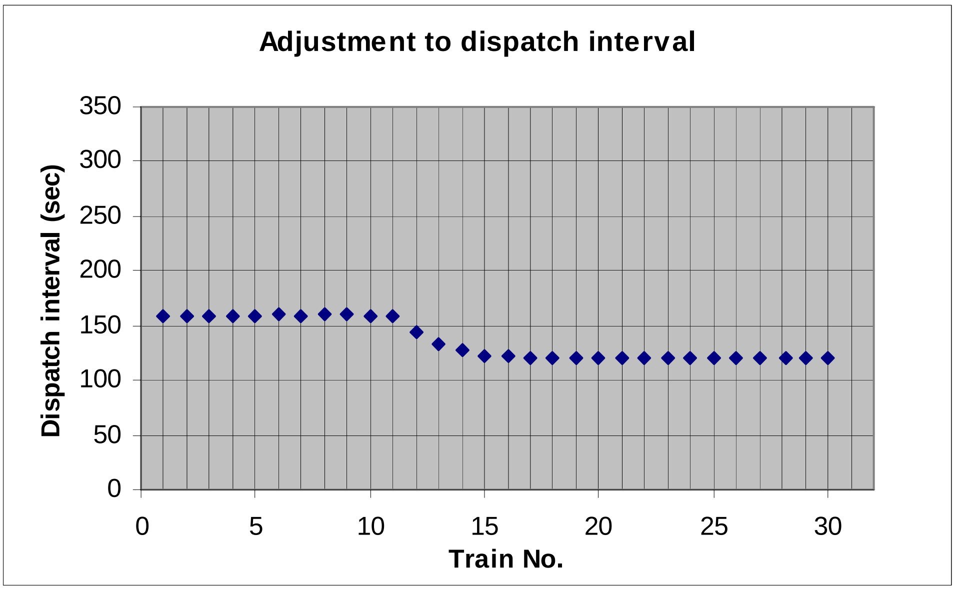


Fig 6.16 Dispatch interval of trains before and during disturbance at JUR station

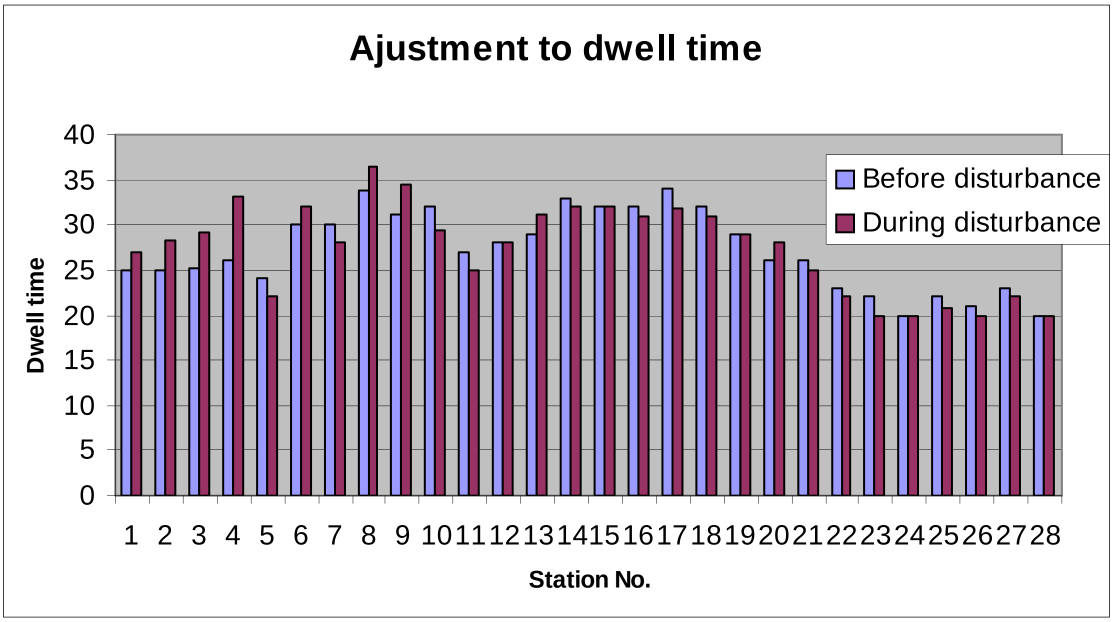


Fig 6.17 Dwell time of stations before and during disturbance at JUR station

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, a summary of the work done and the achievements made are given. Possible areas of future work are described.

7.1 Conclusions

This research project is based on our substantial exposure to the knowledge of practical industry operations through our many years collaboration with SMRT. We met regularly with engineers and operators from SMRT to learn about the practical MRT control systems, especially the software system running on the control computers which monitor the key operational variables including speed, energy consumption, and passenger flow. These key operational variables are the basis for all train scheduling optimization techniques and have been incorporated into the numerical optimization algorithm in this thesis. In SMRT, various reports of operational performances indices were generated for the purpose of monitoring and analyzing. The method in which SMRT adjusts the timetable manually to construct the daily schedule was introduced and clarified in the discussion meeting. This method originates the idea of using a working timetable as the online optimization object in this thesis. SMRT also provided the sampled statistical data such as passenger flow data, normal schedule timing and energy consumption on different coast levels as the inputs to our simulated railway system. This ensures that the simulations closely match real world situations. From the large dataset available, selected frequently recurring cases studies were performed on the ATR simulation software to prove the effectiveness of the proposed numerical optimization algorithm in the reduction of power consumption, which is the major component of actual operational cost. Justification of using the optimization algorithm in this thesis also resides in the discussion on limitations and weaknesses of some other optimization techniques applied to solve the online train-rescheduling problem.

The project objectives have been accomplished with the development of an innovative online rescheduling strategy to solve the multiobjective optimization problem for train rescheduling. In the process, a comprehensive Automatic Train Regulator has been developed to serve as a platform for implementing various control strategies. Two Evolutionary Algorithms (EA), Differential Evolution (DE) and Genetic Algorithms (GA), have been applied to optimize the online Active Timetable based on a fuzzy expert fitness evaluation system.

Simulated tests on ATR have demonstrated the feasibility and effectiveness of the proposed online rescheduling strategy for several real-world scenarios. By optimizing the Active Timetable, the MRT system equipped with the DE-based controller has shown significant reduction or removal of congestions while maintaining relatively low loss in energy efficiency and regularity. Since safety was another major concern in this project, the proposed strategy has included "hard" safety constraints for conflict detection and resolution, which have been tested in 8 scenario studies.

The main contributions of this project are as follows:

1. Multithreaded and object-oriented programming techniques are adopted to simulate the MRT system in real time, which makes the system simulated closer to the real life system. The simulation of train movement is programmed as a thread while a worker thread created for Decision Support System runs in the background to perform scenario analysis, conflict resolution and optimisation. The thread communication mechanism simulates real-time communications between the trains, the Local Process Unit (LPU) and the Operation Control Centre (OCC), which reflects the actual communication process in a real-life system. Object-oriented techniques are used in

the construction of the ATR to achieve reusability, extendibility, and efficiency for future development. The proposed ATR allows the simulation of different scenarios and synthesizing of many strategies.

2. In this project, the proposed strategy evaluates the MRT line as a whole to ensure a global optimization in the system level unlike the online rescheduling strategies applied in the previous works that optimize the individual trains at a time. The Active Timetable mechanism is the key to the realization of the new methods. By analyzing and optimizing the Active Timetable, potential conflicts are detected and removed with a near-optimum solution.

3. The multiobjective optimization problem in this project is solved through the combination of EAs and a fuzzy expert evaluation system. To optimize the train operations, this method utilizes the excellent performance of EAs with the aid of expert experiences, which are mainly represented in terms of sets of fuzzy performance indices and a rule-base. This combination makes the strategy very flexible to include new constraints or rules to be introduced for new optimization problems. The simulation results verify the simplicity, effectiveness and efficiency of DE over GA for solving the online train rescheduling problems.

7.2 Recommendations for Further Work

In this thesis, the train rescheduling problem was explored on a single-line system, which has established a foundation for the extension into a full-fledged multi-track MRT system, all track lines are interconnected by several interchange stations. For example, in the current MRT system, the East West line is connected to the North

South line at interchange stations: Jurong East, Dhoby Ghuant, City Hall, and Raffles Place. Hence, the timetables for the two lines will overlap at these points. This situation potentially introduces a challenging new area of research, known as the line-to-line coordination, for automatic train regulators. This will involve close matching between arrival, departure and dwell times for several train as well as coast levels around these interchange stations for minimizing the passenger's time taken to change from one MRT line to another, and other objectives of the MRT operation.

REFERENCES

[1] T. Taskin, J.Allan, “Automatic Service Regulation for London Underground’s Central Line”, Computers in Railway IV, Vol.2, 1994, pp63-71.

[2] S. Yasunobu., Phoa, S. Miyamoto, and H, Ihara, “Fuzzy control for automatic train operation system”, 4th IFAC/IFIP/IFORS international conference on Control in transportation systems, Baden, 20-22 Apr. 1983.

[3] S. Araya, S. Sone. “Traffic dynamic of Automated Transit Systems with Preestablished Schedules”, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-14, No. 4, 1984. pp 677-687.

[4] C. K. Chua, “Disturbed running on Metro systems: a simulation study”, M.Phil. thesis, December 1986, University of Birmingham, UK.

[5] H.B. Quek, “Development of Automatic Train Regulator for On-line Rescheduling of Train Dwell Time and Dispatch Frequency Using Tabu-Fuzzy Optimization Techniques ”, National University of Singapore, 1998.

[6] A. Higgins, E. Kozan, and L. Ferreira, “Optimal Scheduling of Trains on a Single Line Track”, Transportation Research B 30, 147-161 (1996).

[7] X. Cai and C.J. Goh, “A Fast Heuristic for the Train Scheduling Problem”, Computers and Operations Research 21, 499-510 (1994).

[8] S.K. Raymond Kwan and P. Mistry, “A co-evolutionary algorithm for train timetabling”, Congress on Evolutionary Computation, pp. 2142-2148 IEEE Press. 2003.

[9] D. Kraay, and P. Harker, “Real-Time Scheudling of Feight Networks”, Transportation Research B 29(3), 213-229 (1994).

[10] T.K. Ho, J.P. Norton and C.J. Goodman, “Optimal traffic control at railway junction”, IEE Proc-Electr. Power Appl., Vol. 144, No.2, March 1997.

[11] U. Brannlund, P.O. Lindberg, A. Nou, and J-E. Nilsson, “Railway Timetabling Using Lagrangian Relaxation”, Transportation Science 32(4), 358-369 (1998).

[12] C.S. Chang, Y.H. Phoa, “Economy/regularity fuzzy logic control of DC railway systems using event-driven approach”, IEE, Proc. B, Nov. 1995, 142, (6), pp.1-9.

[13] C.S. Chang, C.S.Chua, “Development of Train Movement Simulator for Analysis and Optimizatiion of Railway Signalling Systems”, IEE International Conference on Developments in Mass Transit Systems, Conference Publication No.453, 1998, pp243-248.

[14] R.J. Hill, and L.J. Bond, 1995, “Modelling Moving-Block Railway Signalling Systems Using Discrete-event Simulation”, ASME/IEEE, 105-111.

[15] R. James, B. Michael, P. Williwam, E. Frederick, and L. William, “Object-oriented Modeling and Design”, Prentice-Hall, 1991, Englewood Cliffs, NJ 07632.

[16] C.S. Chang, B.S. Thia, “Online rescheduling of mass rapid transit systems: fuzzy expert system approach”, IEE Proceedings, UK.

[17] B.D. Fogel, “An introduction to simulated evolutionary optimization”, IEEE Transaction on Neural Network, Vol.5, No. 1 January 1994.

[18] A.Augugliaro, L. Dusonchet, S.Favuzza, E. Riva Sanseverino, “A Fuzzy-Logic based Evolutionary Multiobjective Approach for Automated Distribution Networks Management”, Evolutionary Computation, 2004, CEC2004, Congress on Volume 1, 19-23, June 2004 Page(s):847-854. Vo.1

[19] I.G. Damousis, A.G.Bakirtzis, "A solution to the Unit-Commitment Problem Using Integer-Coded Genetic Algorithm", IEEE Transactions on Power System, Vol. 19. No.2.May 2004.

[20] S.Baskar, P.Subbaraj, M.V.C. Rao, S.Tamilselvi, "Genetic algorithms solution to generator maintenance scheduling with modified genetic operators", IEE Proc-Gener.Transm.Distrib. Vol.150, No.1, January, 2003.

APPENDICES

APPENDIX A: FUZZY LOGIC SYSTEM

A.1 Introduction to Fuzzy Logic

The term "fuzzy logic" emerged in the development of the theory of fuzzy sets by Lotfi Zadeh. Fuzzy logic is used in system control and analysis design, because it shortens the time for engineering development and sometimes, in the case of highly complex systems, is the only way to solve the problem.

In effect, fuzzy logic may be viewed as a methodology for computing with words (linguistic variables) rather than numbers. Although words are inherently less precise than numbers, their use is closer to human intuition. In this project, fuzzy perception is an assessment of a physical condition that is not measured with precision, but is assigned an intuitive value.

A.2 Fuzzy Sets

The very basic notion of a fuzzy system is a fuzzy set. In crisp sets, a given element is either wholly included or wholly excluded from the set. In contrast, fuzzy set is a set without a crisp, clearly defined boundary. It contains elements that have varying degrees of membership in the set, as the following statement that lays the foundation of logic: *In the world of fuzzy logic, the truth of any statement becomes only a matter of degree.*

A.3 Membership Function

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The most commonly used membership functions include the triangular MF and the trapezoidal MF, which is really a truncated triangular MF. The linguistic variables are usually labeled LN, MN, SN, ZE, SP, MP, and LP which symbolizes Large Negative, Medium Negative, Small Negative, Zero, Small Positive, Medium Positive, and Large Positive respectively.

A.4 Fuzzification

Fuzzification is the process of assigning a degree of truth to statements about the input variables via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable and the output is a fuzzy degree of membership (always in the interval between 0 and 1) corresponding to one or more membership functions. Fig A.1 illustrates the process of fuzzification:

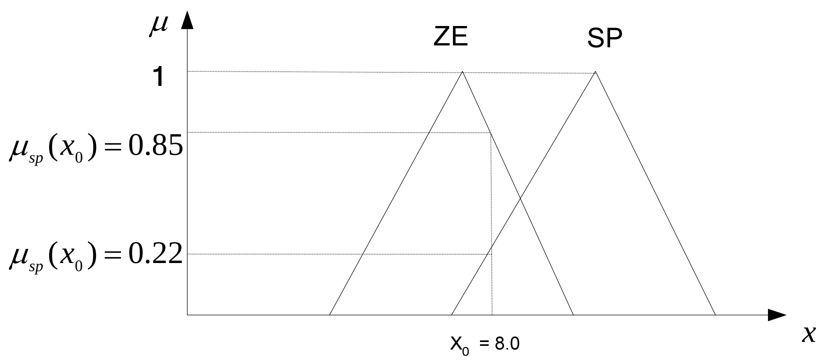


Fig A.1 Fuzzification: The point $x_0 = 8.0$ is a member of ZE to a degree of 0.85 and a member of SP to a degree of 0.22

A.5 Fuzzy Inference

Approximate reasoning is made based on linguistic variables and their values. Fuzzy rules are generated based on previous experience. The rules combine two or more input fuzzy sets, called the *antecedents* sets. The inputs are combined logically using the AND operator to produce output response values for all expected inputs. The active conclusions are then combined into a logical sum for each membership function. A firing strength for each output membership function is computed. Due to the partial matching attribute of fuzzy control rules and the fact that the preconditions of the rules do overlap, usually more than one fuzzy rule will fire at one time. The fuzzy reasoning process of Mamdani's minimum operation that is one of the most commonly used fuzzy reasoning methodologies is illustrated in Fig A.2.

Given: Rule 1: if X is SP and Y is MN Then Z is SN

Rule 2: if X is ZE and Y is LP Then Z is ZE

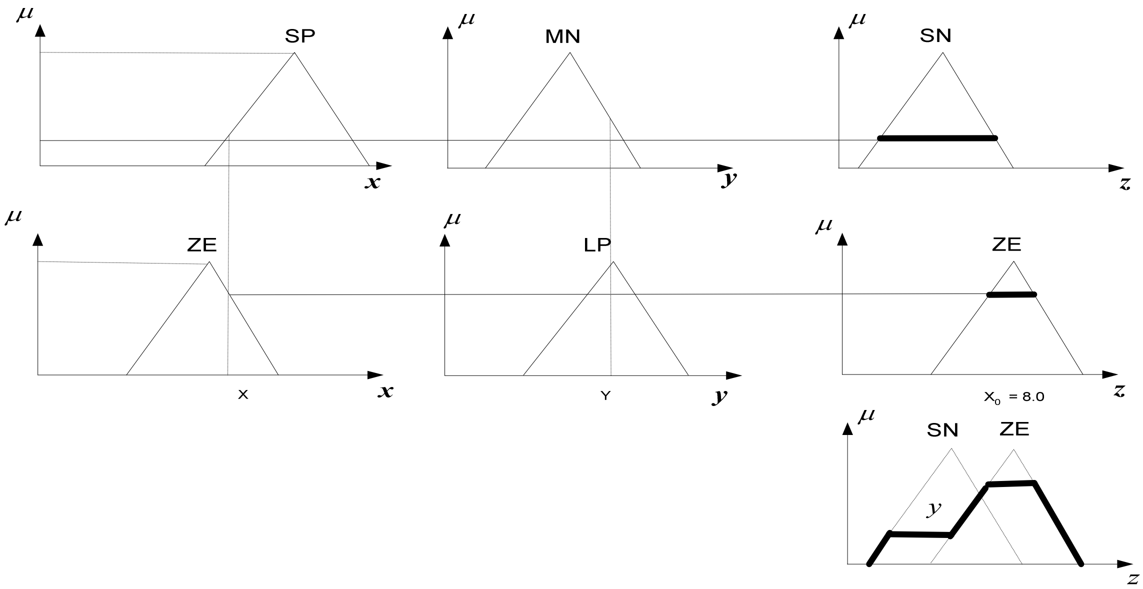


Fig A.2 Graphical illustration of Mamdani's minimum operation rule

One method of storing and representing fuzzy rules is through the use of a fuzzy associative memory (FAM) matrix. The number of inputs, or antecedents, to the fuzzy rules determines the dimension. Three inputs would result in a FAM matrix that looks like a 3-dimensional cube. Each FAM matrix entry is an output fuzzy set or linguistic variable that is the consequent of the fuzzy rule corresponding to the input variables associated with it.

A.6 Defuzzification

The fuzzy outputs for all rules are finally aggregated to one fuzzy set. To obtain a crisp decision from this fuzzy output, we have to defuzzify the fuzzy set. This necessary operation produces a crisp value that best represents the aggregate membership function obtain from the fuzzy inferencing process. The two most common defuzzification methods are the Center-of-Area method (COA) and the Center-of –Sum method (COS). The COA method attempts to determine the centroid or centre of area of the aggregate MF and assign it as the defuzzified output. The COS method is a simplified version of the former and is formally given by:

$$u^* = \frac{\sum_{i=1}^{m} \mu_{z,i} \cdot \overline{z_i}}{\sum_{i=1}^{m} \mu_{z,i}}$$

where:

m: number of overlapped rules that are fired simultaneously

$\mu_{z,i}$: membership value of the output for the i th fired rule

$\overline{z}_i$: specific crisp value assigned to each linguistic variable

APPENDIX B: GENETIC ALGORITHM AND DIFFERENTIAL EVOLUTION

B.1 The Method of GA

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 60s, Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real-world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs. The pseudo-code for GA is presented:

```
Procedure GA:
Begin
        t=0;
        Initialize Population(t);
        Evaluate Population(t);
        While not finished do
        Begin
                t=t+1;
                                Select P(t) from P(t-1);
                                Mate pairs at random;
```

Apply crossover and mutation operators;

Evaluate each individual's fitness;

End

End

B.1.1 Reproduction

The production operator involves choosing a number of individuals according to fitness that will be used for breeding. The purpose of the reproduction is to give more reproductive chances, on the whole, to those individuals that have high fitness values. There are many different techniques, which a genetic algorithm can use to select the individuals to be copied over into the next generation, such as the tournament selection and the roulette wheel selection. In the tournament selection, subgroups of individuals are chosen from the larger population, and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce. As for the roulette wheel selection, a form of fitness-proportionate selection in which the chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. Conceptually, this can be represented as a game of roulette - each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones. The wheel is then spun, and whichever individual "owns" the section on which it lands each time is chosen. The roulette wheel selection is adopted in the study. The integer-coded chromosomes are used in this project to perform the operations of crossover and mutation [19-20].

B.1.2 Crossover

Crossover is performed upon the selected chromosome. It takes two such strings (parents) and exchanges portions of the strings to produce two new strings (children) with probability determined by the crossover rate. Each child incorporates information from the two parents. The effect of crossover is to produce new individuals, which contain genetic material from two parents. There are several different crossover operators, and the following two operators are used most commonly.

1. Single point crossover. A point of exchange is set at a random location in the two individuals' genomes, and one individual contributes all its code from before that point and the other contributes all its code from after that point to produce an offspring. For example, given two initial strings

A = 25   30 | 38 13   B = 20 14 | 36 19

and the randomly selected cross point indicated between the third and forth gene (indicated by the |). In this case, crossover generate the following two new strings

A’ = 25 30 | 36 19   B’ = 20 14 | 38 13

(2) Two-point crossover. Two distinct cross points along the string are chosen uniformly at random. The segments between the two points on these two strings are exchanged. For example, consider the two strings:

C = 25 | 30 38 | 13   D = 20 | 14   36 | 19

And the randomly selected two cross points indicate between genes 1 and 2 and between genes 3 and 4. In this case, the two-point crossover operator generates the following two new strings:

C' = 25 | 14 36 | 19 D' = 20 | 30 38 | 13

B. 1.3 Mutation

In natural evolution, mutation is a random process, where one gene of a chromosome is replaced by another to produce a new genetic structure. In GA, mutation involves a change to any particular gene of an individual. Each gene is considered in turn, and is considered with probability determined by the mutation rate. The effect of mutation on an integer string of 4 genes is illustrated

E = 25 <u>30</u> 38 13

Then, the following string will be obtained

E' = 25 <u>15</u> 38 13 (the 2^{nd} gene is "mutated" to a different value between the upper bound and lower bound of the variable)

Mutation can introduce new genetic material into the population. When the values in the same bits of all individuals are the same, this value cannot change if only the crossover operator is used. However, mutation operator can change the bit value thereby introduce new material.

Typical crossover rates are between 0.6 and 0.9. Typical mutation rates are of the order one in a hundred to one in a thousand bits. Much higher rates tend to disrupt the action of crossover and lead to a more random type of search.

B.2 The Method of Differential Evolution (DE)

Differential evolution is a novel parallel direct search method, which utilizes NP (number of population) n-dimensional parameter vectors as the population for each generation G for each iteration of the minimization:

$X_{i,G}$, i = 0,1, 2,…, NP-1

NP is fixed during the minimization process. The initial population is usually achieved by generating the required number of individuals using a random number generator that uniformly distributes numbers in the desired range. The DE algorithm generates new parameter vectors by adding the weighted different vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, then the newly generated vector will replace the old vector. Otherwise, the old vector is retained.

There are several variants of DE algorithms. Among them, Strategy DE/best and Strategy DE/rand as well as Strategy DE/rand-to-best are the most promising strategies. In the three strategies, the trail vector is generated by following equations B 3.1, B3.2, B3.3 respectively.

$$\underline{V}_{i,\ G+1} = \underline{P}_{i,G} + F \times (\underline{P}_{r1,G} - \underline{P}_{r2,G})$$

$$\underline{V}_{i,\ G+1} = \underline{P}_{best,G} + F \times (\underline{P}_{r1,G} - \underline{P}_{r2,G})$$

$$\underline{V}_{i,\ G+1} = \underline{P}_{i,G} + F \times (\underline{P}_{r1,G} - \underline{P}_{r2,G}) + \lambda \times (\underline{P}_{Best,G} - \underline{P}_{i,G})$$

In equations B2.1 and B3.3, $\underline{P}_{best,G}$ is the best candidate in each generation for the constant dimensioned problem. In equation B3.3, F is a real and constant factor within the range of [0,2], which amplifies the differential, variation $(\underline{P}_{r1,G} - \underline{P}_{r2,G})$. λ controls the greediness of the scheme.

To increase the potential diversity of the perturbed parameter vectors, a crossover probability CR is introduced. To this end, the new vector becomes:

$\underline{u} = (u_1, u_2, ..., u_N)^T$

$\underline{u}$ is an n-dimensional parameter vector and

$$\underline{u} = \begin{cases} \underline{V}_j & \textit{for } j = \langle n \rangle_N, \langle n+1 \rangle_N, ..., \langle n+L-1 \rangle_N \\ (\underline{P}_{i,G})_j & \textit{otherwise} \end{cases}$$

The starting index n is a randomly chosen integer from the interval [0, N-1]. The integer L, which denotes the number of parameters that are going to be exchanged, I is drawn from the interval [1, n].is drawn from the interval [0, N-1] with the probability $\Pr(L = v) = (CR)^v$. $CR \in [0,1]$ is the crossover probability.

APPENDIX C: POTENTIAL CONFLICTING POINTS IN SINGAPORE MASS RAPID SYSTEM AND OPERATIONAL DATA FOR CASE STUDIES

Appendix C-1:

The Automatic Train Regulator is built based on the Singapore MRT network as its simulation platform. The track layout of MRT systems determines the natures of potential conflicts, which provide relevant information for detection and resolution. Hence, for the knowledge of the conflicts, it is necessary to understand the infrastructure of SMRT. The Singapore MRT system consists of the following tracks:

1. North track
2. South track
3. East track
4. West Track
5. North-East Track (North-bound)
6. North-East Track (South-bound)
7. Changi (East-bound)
8. Changi (West-bound)
9. Marina (South-bound)
10. Marina (North-bound)

The overview of the MRT network is illustrated in the Fig C.1:

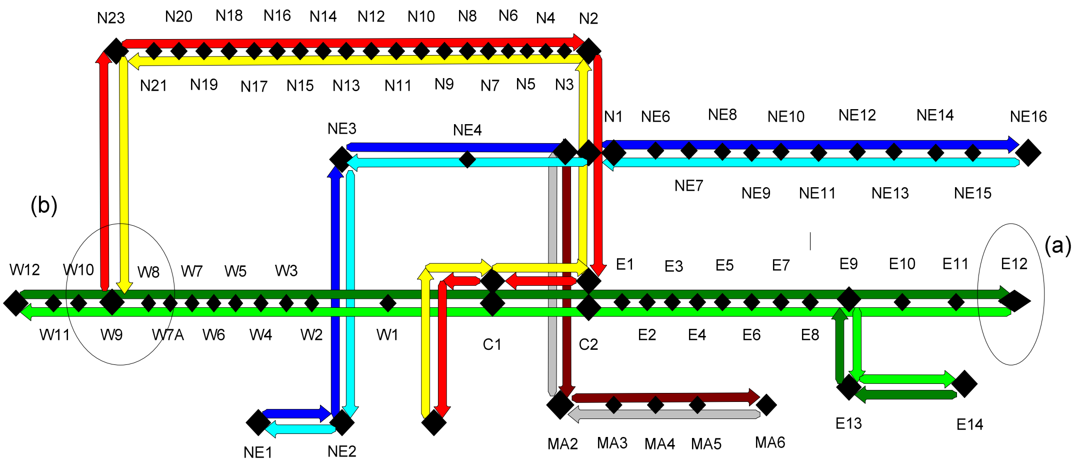


Fig C.1: Overview of the MRT Network

Three types of conflicts may occur at specific spots distributed in the existing network of SMRTS: passing conflict, junction conflict, and terminal conflict.

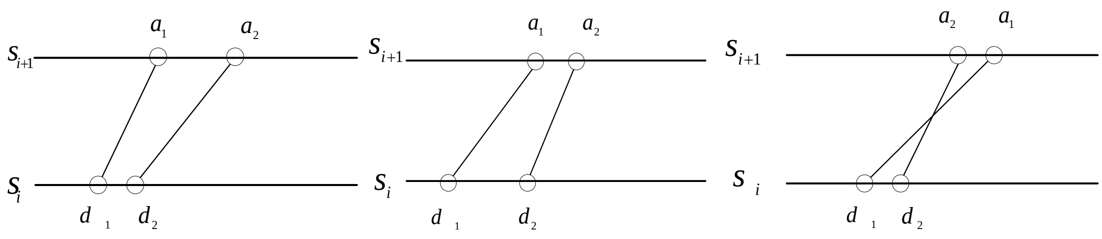
Passing conflict

Passing conflict arises whenever, on the basis of their timings, two trains, running on the same track, in the same direction, are expected to arrive in a station on an inverse order with respect to their departure from the previous station. Practically, this means that a train is leaving a station before another train with a time margin not sufficient to reach the next station before the latter with the safety time interval. Passing conflict arises if one of the following conditions (a), (b), or (c) holds:

a_1 and a_2 represent the time instants in which the train can arrive at station s_{i+1};

d_1 and d_2 represent the time instants in which the train can depart at station s_i;

st represents the safety time interval between two consecutive arrivals and two consecutive departures.



(a) (b) (c)

Fig C.2 Three conditions of passing conflict

(a) the time interval between the time instants associated with a_1 and a_2 is smaller than st , $d_1 < d_2$ and $\Delta(d_1, d_2) < st$, see Fig A3-2 (a);

(b) the time interval between the time instants associated with d_1 and d_2 is smaller than st , $a_1 < a_2$ and $\Delta(a_1, a_2) < st$, see Fig A3-2 (b);

(c) the two line $a_1 d_1$ and $a_2 d_2$ cross each other, see Fig A3-2 (c);

Terminal conflict

Terminal conflict occurs due to common route, where 2 trains cannot occupy the same route at the same time. As illustrated in Fig C.3, E1, E2, W3, W4 are four conflicting points along the tracks within the terminal area illustrated in Fig 8.

For train t1 departing from Terminal to east, there are two possible routes:

a) $Platform\,1->E1->E2->Eastline$: the train departs from platform 1, passes E1, E2, and then runs along the rest of east line;

b): $Platform\ 2->W3->W4->E2->Eastline$: the train departs from platform2, passes W3, W4, through a branch to E2, and then runs along the rest of east line;

.

Similarly, for train t2 arriving at Terminal from east to west, there are two optional pathings:

c): $Westline->W4->W3->Platform2$

d): $Westline->W4->W3->E1->Platform1$

The trains are driven according to scheduled routes and timings in order to avoid the potential conflicts. This kind of conflict occurs in most terminal stations, like Marina Bay which is highlighted with circle (a) in Fig C.3, as well as Boonlay, Pasir Ris, etc.

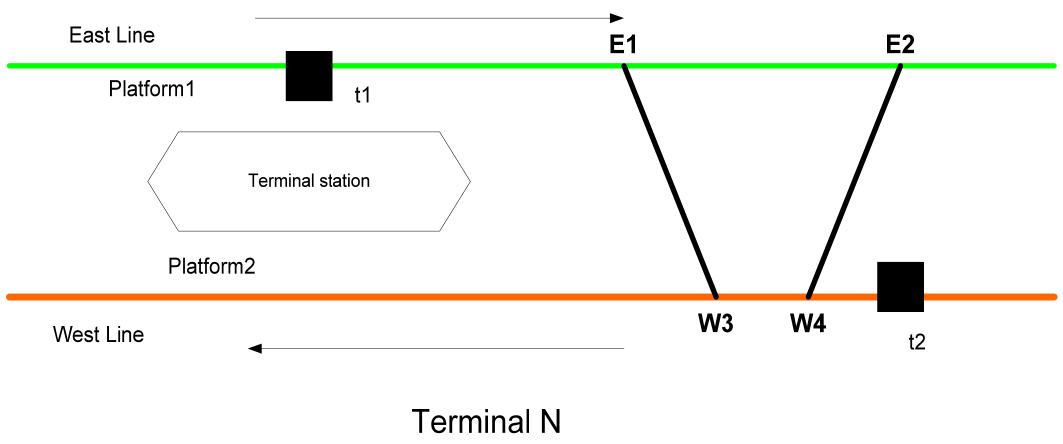


Fig C.3 Conflict scenario arising at terminal stations

Junction conflicts

Junction conflict arises as 2 trains cannot occupy the same station platform at the same time. It occurs at some interchange station like Jurong East highlighted in Fig C.1 with circle (b), where one platform needs to be shared by trains. As illustrated in Fig C.4, the trains from North track dwell on the platform 2 and then depart from it. Hence, the competition for track of platform 2 between the dwelling train t1 and the incoming train t2 create the potential conflicts for schedulers to resolve.

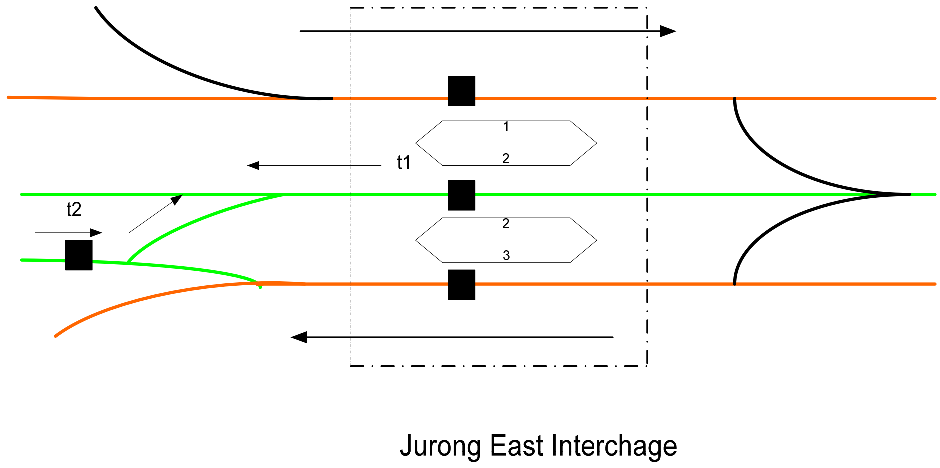


Fig C.4 Conflict scenario arising at Jurong East interchange

In this research, the former two types of conflicts which exist in the Eastwest line are discussed. The conditions of conflicts are formulated into the corresponding constraints for conflict detection and resolution.

Appendix C-2

Table A1 provide the study data for this thesis on passenger-flow rates at stations during the running condition Off peak, Peak 1 and Peak 2.

Station_name	Passenger rate (off-peak)	Passenger rate (Peak 1)	Passenger rate (peak 2)
BNL	0.60	0.72	1.04
LKS	1.42	1.70	2.30
CNG	0.80	0.96	1.52
JUR	2.45	2.95	4.90
CLE	4.00	4.80	5.16
SGP	2.58	3.09	4.32
BNV	6.40	7.04	8.12
COM	3.47	4.06	5.34
QUE	3.15	3.78	6.36
RDH	1.23	1.47	2.56
TIB	1.75	2.10	3.45
OTP	2.30	2.76	3.21
TPG	2.76	3.13	5.13
RFP	2.60	3.01	5.36
CTH	1.56	1.87	3.12
BGS	1.20	1.44	2.23
LVR	0.78	0.91	1.16
KAL	1.66	1.98	2.35
ALJ	0.70	0.84	1.03
PYL	1.20	1.44	2.21
EUN	1.34	1.61	2.45
KEM	1.56	1.87	3.30
BDK	1.46	1.75	3.29
PSR	1.08	1.30	1.16
TAM	2.78	3.33	5.38
SIM	1.76	2.10	3.53
TNM	1.03	1.21	2.11
EXP	0.50	0.62	1.06
CHA	0	0	0

Table A.1 Rate of passenger flow of Off peak, Peak1, and Peak 2

APPENDIX D: SMRT COST FUNCTIONS

One basic objective of the Singapore MRT system is to operate the railway at low cost while maintaining maximum passenger satisfaction. The cost of running the MRT system is affected by factors like the electrical energy usage, the salary paid to the crews and the maintenance costs of tracks and other facilities etc.

The MRT has a set of cost functions that are based on the following set of idea in mind:

1. Total Cost = fixed cost + variable cost
2. Fixed cost = rolling stock + infrastructure + track maintenance + rolling stock maintenance + system maintenance + station operations + overhead
3. Variable cost = Power consumption (Traction energy + Air-conditioning) + Train Ops manpower (TO salary + shift allowances + transport allowances).

The cost functions provided by SMRT are listed below:

1. Total Driver Cost

$$\$TD = 1.33 \times \begin{pmatrix} Average\_salary \\ of\_train\_driver/day \end{pmatrix} \times$$

$$\left(\sum_{n=am,\,pmpeak} \left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h} \right)^m \right)^n + 1.3 \left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h} \right)^m \right)_{offpeak} + \begin{pmatrix} No\_of\_spare \\ duties/day \end{pmatrix} + \begin{pmatrix} No\_of\_double- \\ ending\_duties/day \end{pmatrix} \right)$$

2. Cost of Transport Allowance

$$\$TA = \sum_{\substack{n=before\ 0600\ hrs, \\ 0700-0800\ hrs, \\ after\ 2300\ hrs}} \left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h} \right)^m \times \begin{pmatrix} \%\_of\_trains \\ operated\_at\_that \\ time-frame \end{pmatrix} \times \begin{pmatrix} Transport\_allowance \\ for\_that \\ time\_frame \end{pmatrix} \right)^n$$

3. Cost of Shift Allowance

$$\begin{aligned}\$SA = {} & \left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h}\right)^{m}_{offpeak} \times \begin{pmatrix} \%\_of\_trains \\ launched\_during \\ morning\_off-peak \end{pmatrix} \times \begin{pmatrix} Early\_morning \\ shift \\ allowance \end{pmatrix} \right) + \left(\left(\begin{pmatrix} No\_of \\ double-ending \\ duties/day \end{pmatrix} + 10 \right) \times \begin{pmatrix} Split \\ shift \\ allowance \end{pmatrix} \right) \\ & + \left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h}\right)^{m}_{offpeak\ m} \times \begin{pmatrix} \%\_of\_trains \\ operated\_during \\ off-peak\_lateshift \end{pmatrix} \times \begin{pmatrix} Late \\ shift \\ allowance \end{pmatrix} \right) + \left(\left(\begin{pmatrix} No\_of \\ 3rd\_shift \\ duties/day \end{pmatrix} + 1 \right) \times \begin{pmatrix} Night \\ shift \\ allowance \end{pmatrix} \right) \\ & + \left(\left(\left(\sum_{m=d1,d2} \left(\frac{dt+rt+lt}{h}\right)^{m}_{offpeakm} \right) - \begin{pmatrix} No\_split \\ shift \\ duties/day \end{pmatrix} - \begin{pmatrix} No\_of\_trains \\ operated \\ during\_lateshift \end{pmatrix} + \begin{pmatrix} No\_of\_spares, \\ depots,\_mainline, \\ duties/day \end{pmatrix} \right) \times \begin{pmatrix} 2nd\_shift \\ allowance \end{pmatrix} \right)\end{aligned}$$

4. Cost of electrical energy for air-conditioning

$$\$AC = \sum_{n=am,pm,offpeak} \left(\sum_{m=d1,d2} \left((dt+rt+lt) \times (AC\ kWh/s) \times \left(\frac{dt+rt+lt}{h} \right) \right)^{m} \right)^{n}$$

5. Cost of electrical energy for traction power

$$\$TC = \sum_{n=am,pm,offpeak} \left(\sum_{m=d1,d2} \left(\left(\frac{dt+rt+lt}{h} \right) \times (\$/kWh) \times \left(\frac{dt+rt+lt}{h} \right) \right)^{m} \right)^{n}$$

In summary:

Total variable cost of train service per day,

$X (variable components only) = $TO + $ EG + $RS + $TM

where,

$TO = total cost of train operations manpower per day

$EG = total cost of energy consumption per day

$RS = total variable cost of rolling stock maintenance per day

$TM = total variable cost of track maintenance per day

and

$TO = $TD + $SA + $TA

$TD = Train driver cost =

(No. of train in each time period + No. of double-end duty + No. of spare duty) * Cover factor * Salary of train driver (a function of cycle time & headway)

$SA = Cost of shift allowance (a function of no. of off-peak train)

$TA = Cost of transport allowance (a function of no. of off-peak train)

$EG = $AC + $TC

$AC = Cost of electrical energy for air-conditioning (a function of runtime and headway)

$TC = Cost of electrical energy for traction power (a function of train KM)

$TM = Indirect cost + Direct variable cost (a function of headway)

NB: The direct variable costs are small when compared to the indirect cost

In the above equations, dt denotes dwell time, rt denotes run time, lt denotes layover time, and h denotes headway.