REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PLANNING OF TRAIN MOVEMENTS IN SINGLE TRACK RAILWAYS

GÖKÇE AYDIN

PhD THESIS
DEPARTMENT OF CIVIL ENGINEERING
PROGRAM OF TRANSPORTATION

ADVISER
ASSOC.PROF. DR. İSMAİL ŞAHİN

İSTANBUL, 2015

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PLANNING OF TRAIN MOVEMENTS IN SINGLE TRACK RAILWAYS

A thesis submitted byGökçe AYDIN in partial fulfillment of the requirements for the degree of **DOCTOR of PHILOSOPHY** is approved by the committee on18.09.2015 in Department of Civil Engineering, Transportation Program.

Thesis Adviser

Assoc.Prof. Dr. İsmail ŞAHİN

Yıldız Technical University

Approved By the Examining Committee

Assoc.Prof. Dr. İsmail ŞAHİN

Yıldız Technical University ______________________

Prof. Dr. Necati ARAS Member

Boğaziçi University ______________________

Assoc.Prof. Dr. Mustafa GÜRSOY, Member

Yıldız Technical University ______________________

Assoc. Prof. Dr. Halit ÖZEN, Member

Yıldız Technical University ______________________

Assoc. Prof. Dr. Güvenç ŞAHİN, Member

Sabancı University ______________________

ACKNOWLEDGEMENTS

This PhD thesis is another major step in my academic life, which occurred to be very different than I expected ten years ago. Ten years ago, I was struggling to finish my MSc thesis, in environmental engineering. The struggle was mainly because of this fact: I was in love with railway operations, but married with wastewater treatment. As expected, the marriage went wrong. I could not complete the thesis on time and I have been dismissed. To my family, this was a shame and a disaster. However, this disaster saved my life. Now, I am married with the topic I love. This PhD thesis is some kind of proof of my love. Now, it is time to inject this love to others.

Writing a PhD thesis may officially be regarded as an individual work. However, it is not the actual case. Without the support of some others, it is not possible to overcome the difficulty and stress associated with it. Most important thank goes to my advisor, Assoc. Prof. Dr. İsmail ŞAHİN. To be honest, I was hardly the best graduate student in the universe. However, he showed much more tolerance than he was supposed to. He did not abandon this bad student, and now you see the product of this patience. Plus, he directed me to very useful places, where I can grab the necessary background knowledge. Then, the family comes. Unfortunately, my mother could not witness my progress in doctoral education. However, my father, also working as an academic, provided constant guidance throughout all the progress. The next step is, friends and colleagues. Two of them stand out particularly. First, special thanks to Büşra Aktürk. She was the extracurricular factor throughout the process (apart from her putting constant pressure on me to work harder on my thesis), helping me to overcome the stress with her very close friendship. Second, another special thanks to Betül Değer She was not only a very close friend, but also my officemate in the department. So, she was the primary witness of all the process. I thank her not only for her stress-relieving close friendship, but also support and help in some technical details, such as drawings, dealing with the thesis template, etc. I got similar help from my other colleagues, namely, Şeyma Kuşakçı and Abdülsamet Saraçoğlu. I am grateful to them, because they provided me the opportunity to overcome some technical barriers in a much shorter time than I could have done all by myself. Last but not least, special thanks to other member colleagues of our “lunch group”, Hande Aladağ, Didem Yaşar Oktay and Erkan Şenol.

September, 2015

Gökçe AYDIN

TABLE OF CONTENTS

Page

LIST OF SYMBOLS

a_e^s	Arrival time of eastbound train e at station s.
d_e^s	Departure time of eastbound train e from station s.
$\hat{a}_e^s$	Arrival time of westbound train w at station s.
$\hat{d}_e^s$	Departure time of westbound train w from station s
e	Eastbound train index
E	Set of eastbound trains
f	Eastbound train index
F	A monotonically increasing linear function of its operand.
g	Eastbound train index
h	Eastbound train index
$h_{a,a}$	Arrive-arrive headway
$h_{a,d}$	Arrive-depart headway
$h_{d,d}$	Depart-depart headway
M	A sufficiently large positive number.
s	Station index
S	Set of stations
s_F	The first station (western terminus) of the modeled railway line.
s_L	The last station of the (eastern terminus) modeled railway line.
S_1	Set of stations with n = 2 parallel tracks (one mainline track and one secondary track)
S_2	Set of stations with n = 3 parallel tracks (one mainline track and two secondary tracks)
t	Station index
T_e	Vector of minimum travel times of eastbound train e between consecutive stations.
T_w	Vector of minimum travel times of westbound train w between consecutive stations.
W	Set of westbound trains
w	Westbound train index
x	Westbound train index
v	Westbound train index
z	Westbound train index
α_e	Scheduled arrival time of eastbound train e at station s_L as required by the timetable
$\hat{\alpha}_w$	Scheduled arrival time of westbound train w at station s_F as required by the timetable.

$\beta_{e,f}^{s}$ Minimum depart-depart headway when eastbound train e departs from station s after eastbound train f.

$\hat{\beta}_{w,x}^{s}$ Minimum depart-depart headway when westbound train w departs from station s after eastbound train x.

δ_e : Entrance time of eastbound train e into the system.

$\hat{\delta}_w$: Entrance time of westbound train w into the system.

$\eta_{e,f}^{s}$ Minimum arrive-arrive headway when eastbound train e arrives at station s after eastbound train f.

$\hat{\eta}_{w,x}^{s}$ Minimum arrive-arrive headway when westbound train w arrives at station s after eastbound train x.

$\rho_{e,w}^{s}$ Minimum arrive-depart headway when eastbound train e departs from station s after arrival of westbound train w.

$\hat{\rho}_{e,w}^{s}$ Minimum arrive-depart headway when westbound train j departs from station s after arrival of eastbound train e.

σ_e^s Minimum dwell time of eastbound train e at station s as required by timetable

$\hat{\sigma}_w^s$ Minimum dwell time of westbound train w at station s as required by timetable

τ_e Delay (tardiness) of eastbound train e in the final station of its itinerary within the system.

$\hat{\tau}_w$ Delay (tardiness) of westbound train w in the final station of its itinerary within the system.

LIST OF ABBREVIATIONS

AIMMS:	A commercial software for coding optimization problems
CPLEX:	A commercial software for solving linear programming, integer programming, mixed integer programming and quadratic programming problems.
LP:	Linear Programming
MIP:	Mixed-integer programming
MILP:	Mixed-integer linear programming
NP:	Non-deterministically polynomial-time
TCDD:	Turkish state railways

LIST OF FIGURES

Page

LIST OF TABLES

ABSTRACT

PLANNING of TRAIN MOVEMENTS OVER RAILWAY LINES

Gökçe AYDIN

Department of Civil Engineering

PhD Thesis

Adviser: Assoc.Prof. Dr. İsmail ŞAHİN

Railway is known to be the best mode of land transport in terms of energy consumption and land use per passenger-km or ton-km transported; and also in terms of economic efficiency for freight transportation. It is also known to be superior to air transport in terms energy consumption per passenger-km up to some specific distance of travel. Thus, it is of crucial importance to increase the market share of rail transport for economic and environmental sustainability. Customer satisfaction through better punctuality is one of the possible strategies towards this purpose. In reality, most of the railways operate according to a timetable, within which, all trains have predetermined departure times from, arrival times at and / or passing times without stopping through all the reference points (stations, sidings) in their routes. In daily operation, some of the trains may get delayed for various reasons. This creates a knock-on effect, spreading the delay to other trains. Thus, the timetable becomes invalid, and rescheduling of the traffic becomes necessary. Efficient rescheduling helps the railway system be more punctual. In practice, rescheduling is done by human operators (called dispatchers) by manual methods. Human brain has a limited computational ability. This puts an upper limit on the effectiveness of rescheduling solutions produced manually by humans. Fortunately, the computational power of today's modern computers can provide significant improvement. In this thesis, first, a mixed integer programming model for solving the rescheduling problem on a single track railway line to optimality is developed. The model considers most of the real constraints in a real railway operation like deadlock prevention and capacities of stations/sidings and aims to minimize the total weighted delay of the trains. Train scheduling is a strongly NP-Complete problem. This nature of the problem was clearly observed even in the small sized problems, like 4 eastbound trains and 3 westbound trains. This is a big drawback in a rescheduling problem, because rescheduling has to be done in a dynamic environment. Trains are moving and they can get some additional delays during the computation process, if it takes too long. This would make the solution produced worthless. To be specific, any algorithm for train rescheduling has to finish its job in at most 5 minutes, but, preferably, in 3 minutes. Therefore, a plan mixed-integer programming proved to be inadequate for rescheduling. It has to be supported with some additional procedures. We call these procedures as “speed-up routines”. In this thesis, three different speed-up routines were used. The first was using the “lazy constraint” attribute of AIMMS. This attribute enables the user to mark the constraints that are unlikely to be binding. Then, the solver excludes them when computing the linear programming relaxation of the model and checks to solution of the linear programming relaxation against the constraints marked as “lazy”. If it finds that one of the lazy constraints is violated, it adds the violated constraint into the constraint pool and re-solves the linear programming relaxation. In the model, most of the station / siding capacity constraints were marked as lazy. The second speed-up routine was a heuristic solution space restriction algorithm. The algorithm first produces a solution by implementing a greedy algorithm. This algorithm neglects the station / siding capacity constraints and deadlock prevention. Then, it restricts the solution to be not much different from the outcome of the greedy heuristic. This eliminates hundreds of binary variables and thousands of constraints from the model and provides a radical increase in the model’s computational speed. However, the optimality of the solution is no longer guaranteed, although the model produces good solutions. The third speed-up routine was adopting a multiobjective approach. The objective of the main model is to minimize the total weighted delay of all trains. In the multiobjective approach, first, a problem with the same variables, parameters and feasible region, but a different objective function is defined. The objective function is minimizing the maximum weighted delay of all trains. Then, the optimal solution from this model is used as an initial feasible solution for the main problem. Also, in the main problem, weighted delay of each train is constrained not to exceed the maximum weighted delay computed in the first problem. This routine also provided a speed-up. The final model was tested on a hypothetical single track railway line with 18 stations. In the worst cases, the final model with all the speed up routines managed to solve the problems with 6 eastbound trains and 5 westbound trains in less than three minutes.

Key words: Railway traffic control, train scheduling, rescheduling, exact algorithms, integer programming, heuristics

ÖZET

TEK HATLI DEMİRYOLLARINDA TREN HAREKETLERİNİN PLANLANMASI

Gökçe AYDIN

İnşaat Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Doç. Dr. İsmail ŞAHİN

Demiryolu, kara ulaşım türleri arasında, taşınan yolcu-km veya ton-km başına enerji verimiliği en yüksek, arazi kullanımı en düşük tür olarak bilinmektedir. Ayrıca, yük taşımacılığındaki ekonomik verimliliği de karayoluna göre daha yüksektir. Yolcu taşımacılığında, belirli bir mesafeye kadar olan taşımalarda, havayoluna karşı da, enerji verimililiği bakımından üstünlük göstermektedir. Bu nedenlerle, çevresel ve ekonomik sürdürülebilirlik adına, taşımacılıkta demiryolunun pazar payının yükseltilmesi büyük önem taşımaktadır. Daha iyi bir dakiklik marifetiyle müşteri memnuniyetinin arttırılması, bu konuda geliştirilebilecek stratejilerden biridir. Gerçekte, çoğu demiryolu sisteminde, trenler, önceden belirlenmiş bir zaman çizelgesine göre hareket etmektedir. Bu çizelgede, trenlerin, rotaları üzerinde bulunan tüm referans noktalarına (istasyonlar, saydingler) varış, bu noktalardan kalkış ya da bu noktalardan durmadan geçiş zamanları kayıtlıdır. Günlük işletimde, trenlerden bazıları, çeşitli sebeplerden dolayı gecikebilir. Bu gecikmeler, bir yayılma etkisi yaratarak, diğer trenlere de sirayet etmektedir. Sonuç olarak, hazırlanan zaman çizelgesi geçerliliğini yitirmekte, yeniden çizelgeleme gereksinimi ortaya çıkmaktadır. Yeniden çizelgelemeyi verimli bir şekilde yapmak, sistemin dakiklik performansının artmasını sağlayacaktır. Uygulamada, yeniden çizelgeleme, dispeçer adı verilen operatörler tarafından, elle yapılmaktadır. İnsan beyninin hesap yeteneği sınırlıdır. Bu durum, insanlar tarafından elle yapılan yeniden çizelgemenin kalitesi üzerine sınırlar koymaktadır. Günümüzün modern bilgisayarlarının hesap yeteneklerinden yararlanarak, bu verimliliği arttırmak mümkündür. Bu tez çalışmasında, öncelikle, tek hatlı bir demiryolunda yeniden çizelgeleme probleminin optimum çözümünü verecek bir matematiksel model geliştirilmiştir. Bu model, demiryolu işletimi ile ilgili pek çok kısıtı (örn. istasyon ve saydinglerin kapasiteleri) dikkate almakta ve trenlerin ağırlıklandırılmış gecikmelerinin

toplamını en küçüklemeyi amaçlamaktadır. Tren çizelgelemesi problemi, güçlü bir NP-Tam problemdir. Problemin bu doğası, geliştirilen model, 4 adet doğu yönlü ve 3 adet batı yönlü tren gibi küçük boyutlu problemler üzerinde denendiğinde bile kendini açık olarak göstermiştir. Bu, yeniden çizelgeleme için büyük bir handikaptır. Yeniden çizelgeleme problemi, dinamik bir ortamda çözülmek durumundadır. Trenler hareket halindedir ve algoritmanın çözüm üretmesi çok uzun sürecek olursa, çözüm süresi içinde trenlerde ilave gecikmeler meydana gelebilecektir. Bu durum gerçekleşirse, modelin ürettiği çözüm değersiz hale gelecektir. Net olmak gerekirse, bir yeniden çizelgeme algoritmasının, işini en fazla 5 ancak tercihen 3 dakika içinde bitirmesi gerekmektedir. Düz bir karışık tamsayılı programlama modeli, bu iş için yetersiz kalmaktadır. Böyle bir model, mutlaka bazı ilave prosedürler ile desteklenmelidir. Bu tezde, bu prosedürlere "hızlandırma rutini" adı verilmiştir. Çalışmada, üç farklı hızlandırma rutini kullanılmıştır. Bunlardan ilki, AIMMS'in "lazy constraint" özelliğini aktifleştirmektir. AIMMS, kullanıcıya, optimum çözümde bağlayıcı olma olasılığ çok düşük olan kısıtları "lazy" olarak işaretleme olanağı sunmaktadır. "Lazy" olarak işaretlenen kısıtlar, ilk başta problemin doğrusal programlama gevşetmesine dahil edilmemektedir. Bu kısıtlar dahil edilmeden çözülen doğrusal programlama gevşetmesinin çözümünün bu kısıtları ihlal edip etmediği kontrol edilmektedir. İhlal ettiği kısıt varsa, bu kısıt, kısıt havuzuna yeniden dahil edilmekte ve DP gevşetmesiyeniden çözülmektedir. Bu çalışmada, istasyon / sayding kapasitesi kısıtlarının büyük çoğunluğu "lazy" olarak işaretlenmiştir. İkinci hjızlandırmarutini, sezgisel bir çözüm uzayı kısıtlama algoritmasıdır. Bu algoritma, ilk önce, açgözlü bir algoritma kullanarak, problem için bir çözüm üretmektedir. Bu açgözlü algoritma, istasyon ve sayding kapasitelerini, hattın kilitlenmesi durumunu dikkate almamaktadır. Bu çözüm üretildikten sonra, ana modelin çözümü, açgözlü algoritmanın ürettiği çözümden fazla uzaklaşamayacak şekilde kısıtlanmaktadır. Bu algoritma, modelden yüzlerce ikili değişkeni ve binlerce kısıtı attığı için, çözüm hızında radikal bir iyileşme sağlamaktadır. Ancak, bulunan çözümün optimum olduğu artık garanti edilememektedir, lakin, üretilen çözümlerin oldukça iyi olduğu görülmüştür. Üçüncü hızlandırma rutini olarak, bir çok amaçlı optimizasyon uygulaması yapılmıştır. Bu uygulamada, ana problem ile tamamen aynı karar değişkenleri ve uygun çözümler bölgesine sahip, ancak amaç fonksiyonu farklı bir problem çözülmektedir. Bu problemin amaç fonksiyonu, tüm trenlerin ağırlıklandırılmış gecikmelerinin maksimumunu minimize etmektir. Bu problemin optimum çözümü, ana problem için başlangıç uygun çözümü olarak kullanılmaktadır. Ayrıca, ana problemdeki trenlerin ağırlıklandırılmış gecikmeleri, ilk problemde bulunan maksimum değeri geçmeyecek şekilde kısıtlanmıştır. Nihai model, 18 istasyonu bulunan, hayali bir tek hatlı demiryolunda test edilmiştir. En kötü durumda, tüm bu hızlandırma rutinleri ile birlikte, model, 6 doğu yönlü ve 5 batı yönlü tren içeren problemleri 3 dakikanın altında bir süre içinde çözebilmiştir.

Anahtar Kelimeler: Demiryolu trafik kontrolü, tren çizelgeleme, yeniden çizelgeleme, tam çözüm algoritması, tamsayılı programlama, sezgisel yöntemler

CHAPTER 1

INTRODUCTION

1.1 Literature Review

Since train scheduling problem is basically a scheduling problem, the distribution of related literature shows some similarity with the classical scheduling literature. Like any NP-Complete (Non-deterministically polynomial-time-complete) scheduling problem, there are three main solution approaches for train scheduling problem. These are *exact solutions*, *metaheuristic solutions* and *problem-specific heuristic solutions.*

Exact solutions are those obtained using the integer-programming methods to generate the optimal solution. They are difficult to apply in NP-complete problem structures. Nevertheless, they have found some application in the operations research literature. Billionnet[1], Caprara et al.[2], Dessouky et al. [3] are some examples of the studies that use exact solution approaches.

Metaheuristic methods such as simulated annealing, tabu search, ant colony optimization, and genetic algorithms are widely used for solving train scheduling problems. These methods, for most of the time, quickly find good but not necessarily optimal solutions. Although they are generally known as effective methods, getting good results from them requires having a high level of expertise on them. Gorman [4]applied both tabu search and genetic algorithms for solving the problem. Salim and Xiaogiang [5] employed a genetic algorithm model which also aims to reduce the iron dust emissions due to braking. Törnquist and Persson [6] combined tabu search and simulated annealing.

Problem- specific heuristic solutions (also called *tailor-made heuristics*) are developed exclusively for train scheduling problem. They range from simple rule based heuristics to quite complicated branch-and-bound methods. We can further divide this category to two more subcategories: *Greedy Heuristics* and *Heuristics with a look-ahead component.*

Greedy Heuristics: These methods focus on the problem only on the local scale. They solve the conflicts one-by- one and do not consider the effect of the resolution of the current conflict on the later conflicts. These methods focus only on the trains involved in the current conflict and try to optimize the selected performance measure based on the solution of the current conflict. The biggest advantage of these methods is a very short computational time. Since operational scheduling is indeed a real-time scheduling problem, it is very important to reach good feasible solutions in a very short time. Moreover, sometimes, these heuristics provide surprisingly good solutions. One example is the dispatcher's solution in Şahin [7], which is based on decision behaviours of Turkish State Railways' train dispatchers. This greedy heuristic found the optimal solution in some of the cases. However, at least theoretically, a greedy heuristic is not expected to give very good results in all the instances, because of its myopic nature.

Heuristics with a Look-Ahead Component: These methods are not just local methods. They consider the consequences of the local decisions-or they try to improve the solution by trying to choose the alternatives of locally optimal decisions. A good example to the former approach is the heuristic developed by Şahin [7] and to the latter approach is D'Ariano et al. [8].

One problem with the non-exact solutions is that, all train scheduling problems are unique. Railway operation involves a huge number of different operational constraints, related to line topology and physics involved. Up to today, there is no pool of benchmark problems to evaluate the performance of these heuristics. So, problems are generally selected from real-world applications with real-world traffic data. They are then benchmarked with solutions of human planners. If they provide significantly better solutions than human planners, the solutions are considered to be "good enough" solutions.

Understanding the literature about train scheduling requires some understanding of the train scheduling problem itself. Therefore, the rest of the literature review is skipped for now. More comprehensive literature review will be given in Chapter 2, after giving all the necessary and basic understanding about the problem.

1.2 Objective of the Thesis

Railway is known to be the best mode of land transport in terms of energy consumption and land use per passenger-km or ton-km transported; and also in terms of economic efficiency for freight transportation. It is also known to be superior to air transport in terms energy consumption per passenger-km up to some specific distance of travel. Thus, it is of crucial importance to increase the market share of rail transport compared to road and air transport for economic and environmental sustainability. Unfortunately, recent trend is just the opposite: Rail is constantly losing market share against road and air. Indeed, every day, more and more people are gaining consciousness about this subject. However, sheer reliance on the consciousness of the customers is still by no means sufficient to achieve this goal. It is a strict requirement to increase the customer satisfaction related to rail transport. This increase is only possible to improve rail transport in terms of the classical quality measures related to transportation, such as speed, safety, reliability, punctuality, service frequency and price. Developing proper strategies to achieve these improvements is very important for the future of railways.

As the amount of traffic on the rail network increases and approaches to the network's capacity, problems with reliability, punctuality (which also disrupts speed by elongating the travel time) and service frequency start to arise. Moreover, these problems cause an increase in the operational costs and harm the economic edge of rail transport. These problems are serious and it is a must to find a way to solve them. A naive approach is to consider building new railway lines, increasing the number of (parallel) tracks in existing railway lines, building new stations or sidings for trains to meet / overtake each other, adding new tracks to existing stations, etc. Bringing these ideas to reality is, however, often very difficult, if not impossible. Land acquisition costs (especially in residential areas and business districts), political and economic issues about funding, strong opposition by environmental activist groups and some other factors prohibit such infrastructure investments.

When it is impossible or at least very difficult to build new infrastructure, railway infrastructure managers and train operating companies seek other solutions; such as improving the operational procedures to make better use of the existing capacity and thus improving the service quality and cost-effectiveness.

Railway, just like any other transportation system, consists of three major components: The infrastructure component, the vehicles component and the operation component. The operation component includes all the rules, methods and procedures for effective, efficient and safe operation of the system. It can be said that infrastructure and vehicle components are the hardware of the system; whereas the operation component is the software. All these components are in a very close relation and a well-functioning system is possible only when all these components are well-designed and well-functioning.

One of the most important parts of the operation component of a railway system is train scheduling. Train scheduling involves preparing the timetables, to which the train operations are expected to conform; and rescheduling the traffic in case of the perturbations in real-time operations. Improving the performance of these activities would help very much to improve the service quality of rail transport. Hence this thesis is about train rescheduling problem, and especially with the design of computer-based decision support systems for solving the train rescheduling problem. The main purpose and scope of the thesis is rescheduling the train traffic on a single track railway line.

1.3 Hypothesis

In this thesis, a mixed integer programming based algorithm that solves the train rescheduling problem on a single track railway line is developed. Generally, mixed integer programming methods are considered to be incompatible with NP-Complete scheduling problems. It is generally thought that, with such a method, it is not possible to solve such a scheduling problem in a reasonable amount of time, as the size of the problem instance (i.e. number of jobs and/or machines) gets larger and larger. However, it can be also claimed that, by making clever use of the know-how in the problem itself, it may be indeed possible to solve the problems of practical size with a mixed integer programming based algorithm. This approach is made use of in this thesis. It has been inspired by Castillo et al.[9]. Normally, the number of binary variables and constraints show a drastic increase, as the number of trains to be scheduled increase. Nevertheless, it is possible to slow down this increase by applying clever post-modeling tricks to involve only the really required variables and constraints. Therefore, a heuristic-based solution space restriction routine was developed to expedite the mixed integer programming solution. Details of the routine will be given in Chapter 4. Last but not least, practical size limits come into action. Some problems might be theoretically unsolvable in a reasonable amount of time. However, the nature of the problem itself can put practical limits on how large the problem instances can be. This perfectly applies for train scheduling problems. Railway lines have limited capacities. The maximum number of trains can actually run on a single track railway line is much less than the number of jobs to be scheduled on a production line of a factory. This also raises hope for implementing mixed integer programming algorithms in a reasonable amount of time. The latter fact is clearly reflected in the problem instances used for the model. The problem instances are run on a hypothetical single track railway line, which borrows its topology from the TCDD line between Arifiye-Çukurhisar. This line has 18 stations. Among those stations, there is one western terminus, one eastern terminus and there are 16 intermediate stations. In the problem instances run, trains were introduced into this line every 10-20 minutes per direction. This number is already highly unlikely to be observed in practice. In practice, if there would be that much of traffic density on a railway line, railway infrastructure managing companies prefer to have this line double tracked.

CHAPTER 2

GENERAL DEFINITION of the PROBLEM

In this chapter, a wide explanation and definition of the problem will be given. Some basic insight into the railway operations and related complications is an absolute necessity to understand and employ algorithms for train scheduling.

The formal definition of scheduling is “assigning start and end times for certain tasks that share the same resources, in an efficient way” and this definition is perfectly applicable to train scheduling. However, gaining insight about what the “tasks” are can be somewhat tricky. In train scheduling, “tasks” are “train movements over railway lines.” These railway lines have some different components. Therefore, before explaining the train scheduling concepts, it will help very much to provide some background information about train movements over railway lines and their elements.

2.1 Train Movements over Railway Lines

The duty of a train is to travel from one point (*origin* of the train) to another (*destination* of the train); with, if necessary, some intermediate stops. This duty is called the train’s itinerary. Trains perform that duty over railway lines. Railway lines have some different elements and trains move over these elements during their travel. Because of this fact, it may be helpful to describe those elements first.

2.1.1 Elements of a Railway Line

Main elements of a railway line that are relevant to train scheduling are *mainline track sections, stations* and *sidings.*

Mainline Track Sections: Mainline track sections are the sections of railway tracks between adjacent stations or sidings. Over these track sections, trains cover the bulk of the distance. Mainline tracks generally extend through stations and sidings. A typical characterization of railway lines is based on the number of parallel mainline tracks. If there is only one mainline track, the railway line is called a “single track” line. Similarly, if there are two parallel mainline tracks, the line is called a “double track” line, for three or more parallel tracks, it’s called “multiple track” line.

Mainline tracks may be unidirectional or bidirectional. Over unidirectional tracks, trains can only travel in one direction. Unidirectional mainline tracks can be found in railway lines having more than one parallel mainline tracks. Except some special cases (like closure of one of the tracks due to failures, construction works, etc.), movement of trains in the opposite direction is prohibited. An example to unidirectional mainline tracks is the Sirkeci-Halkalı suburban (commuter) train line. On this line, there are two parallel mainline tracks. One of these tracks is allocated for trains from Halkalı to Sirkeci, and the other track is allocated for trains from Sirkeci to Halkalı.

On bidirectional tracks, trains moving in both directions can operate. In single track lines, mainline tracks are always bidirectional. On double or multiple track lines, depending on the rules of adopted by the infrastructure manager, they can be unidirectional or bidirectional. An example to double track lines with bidirectional mainline track sections used to be the Haydarpaşa-Gebze suburban train line, which also used to be serving regional, intercity and freight traffic (Now, for years, the line is abolished in favor of still unstarted construction works). On this line, one of the tracks was mainly allocated for Haydarpaşa bound trains and the other track was mainly allocated for Gebze bound trains. However, to allow fast trains to overtake slow trains, some fast trains could use the line that is mainly allocated for the trains in opposite direction.

Sidings: Sidings are the sections of railway lines, through which, number of parallel tracks is increased for some length of railway line, (say, 1000-2000 meters). Sidings are used for meeting and overtaking operations.

The number of tracks within the siding (including the part of mainline tracks extending through the siding) is the *capacity* of the siding, i.e. the maximum number of trains that can be present within the siding in overlapping time intervals. Sometimes, this capacity can be exceeded by means of a special operation: two or more short trains (total length of them being well below the length of the siding) bound in the same direction can be stopped on the same track. In this case, for departures from the siding, a first-in, first-out rule has to be applied. However, for avoiding overcomplicated models, this special operation to exceed the station capacity is generally not modeled in train scheduling algorithms (More often, station capacity is not modeled at all, for the same purpose).

Stations: Stations are similar to sidings. Within the stations, number of parallel tracks is greater than that in the mainline track sections. Meeting and overtaking operations can be done within the stations. The station capacity concept is the same as the siding capacity concept. However, stations have an additional duty: passengers board and alight the trains there. In addition, freight trains have stations, called *yards*. In yards, the freight cars can be loaded and unloaded. Also, the composition of the freight trains can change within the stations (yards), by freight cars added to or removed from them. These composition changing operations are part of some movements called *shunting*. Shunting operations are not within the scope of this study. The elements of a railway line are illustrated in Figure 2.1. This figure is also an illustration of a single track railway.

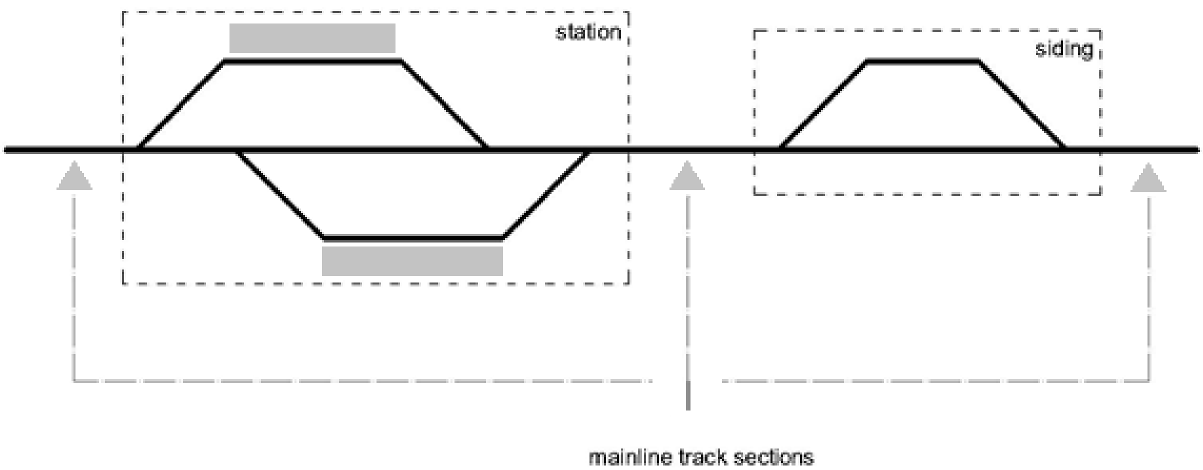

Figure 2.1 Station, siding and mainline track sections

2.1.2 Analogies Between the Elements of Railway Lines and the Elements in Classical Scheduling Problems

Trains-jobs: In classical scheduling problems, there are a number of jobs to be processed. Trains are like jobs. Each train has an origin, a destination and a specific route, which is generally dictated by the alignment of the railway itself. When a train departs from its origin, it is as if the job starts to be processed in the first machine of its route. When a train arrives at its destination, it is as if the job is finished; i.e. its operation on the last machine of its route is finished. The arrival time of the train at its destination is analogous to completion time of a job. There is the "scheduled arrival time" of a train to its destination. This is analogous to the "due date" of a job. Presence of these analogies calls that we can use the same due date and completion time related performance measures for the train scheduling problem; like maximum tardiness, total tardiness, weighted total tardiness, etc. The names of the measures may differ though.

Mainline Tracks-Machines: Mainline tracks are analogous to machines in classical scheduling problems. Each job has to be processed in a number of machines in a particular order. Similarly, each train should pass through a number of mainline track sections to travel from its origin to its destination; simply because these mainline track sections constitute the only path connecting the origin and the destination. Processing of each job in each machine takes some time, called the *processing time*. Analogously, each train covers the length of each mainline track section (remind that mainline track sections extend between two adjacent stations-sidings) in a duration, called *running time.* In this study, a *minimum running time* is used for modeling purposes. In this case, running times are not fixed; they can get any value that is greater than or equal to the minimum running times.

Stations and Sidings-intermediate Storage Areas: In classical flow-shop or job-shop scheduling; each job is processed in one machine and enters an intermediate storage area, then processed in the next machine and enters another intermediate storage area. This goes on until the job is finished. The process is nearly the same in train scheduling: The train covers one mainline track section and enters a station or siding (note that whether it will be stopped in the station or siding for a while or continue without stopping may not be not known without constructing the final schedule), and then enters another mainline track section, then enters another station or siding. This goes on until the train arrives at its destination. The stopping duration of a train in a station or siding is analogous to waiting time of jobs in an intermediate storage area between two machines.

As mentioned earlier, in stations (yards), trains may have to undergo some operations, like passenger boarding-alighting, loading- unloading of freight cars, composition changing, refueling, etc. These operations take some time. So, if any train is required to undergo such operations in a station, trains cannot pass though that station without stopping. In such stations, a minimum required stopping time, called the *minimum dwell time,* has to be enforced. Minimum dwell time constitutes a lower bound for minimum waiting time of a train in a particular station. Minimum dwell times typically do not appear in sidings.

It was also mentioned earlier that, all stations and sidings have finite capacity values. This corresponds to a scheduling problem with limited intermediate storage capacities. Station and siding capacities are hard constraints. In order a study to get any positive value for application to real world train scheduling/rescheduling, it has to take station and siding capacities into account. However, just as scheduling problems with limited intermediate storage capacities, train scheduling problems with limited station and siding capacities are very special and very difficult problems. Consequently, these (actually very important and hard) constraints are relaxed in many studies in the literature. However, it is included in the mathematical model developed in this thesis.

2.2 Analogies Between the Train Scheduling Problem and the General Scheduling Problem

Like the analogies between the elements, some analogies between scheduling problem types can be seen. Existence of an analogy does not necessarily mean that two types of problems are exactly the same. It is rather like a conceptual similarity. Some differences also exist.

Single Track Scheduling-job Shop Scheduling: Example of a single track railway is given in Figure 2.2. Scheduling the trains on a single track railway line is in some sense analogous to job- shop scheduling. As explained in the previous parts, there are mainline track sections (analogous to machines) and stations- sidings (analogous to intermediate storage areas), all connected in series. Since the mainline tracks are bidirectional, trains moving in both directions have to be scheduled. In this sense, it is rather like a scheduling problem where the routings of the jobs may be different from or opposite to each other, like a job-shop. To illustrate, let's imagine a single track railway line with 5 sidings and 4 mainline track sections connecting them. In Table 2.1 there is the one-to-one correspondence between railway line elements and elements of job-shop scheduling problem.

Figure 2.2 A single track railway in UK [10]

As sample explanations, we say that w_{i1} is the waiting time of train (job) i in siding (intermediate storage area) 1, t_{i12} is the running time of train i on mainline track section 1 (or the running time from siding 1 to siding 2), t_{i21} is the running time of train i on mainline track section 1 (or the running time from siding 2 to siding 1) and p_{i1} is the processing time of job i in machine 1.

One difference between job-shop scheduling and single track train scheduling is that, in job- shop scheduling, depending on the routing of the job, some jobs may not visit some machines at all. That means a routing such as machine 1-machine 4-machine 5 is possible. In train scheduling, although trains may start and / or terminate in intermediate stations, they cannot skip any station/siding and/or mainline track section between origin-destination pairs at all (although they can stop or pass without stopping there). That is, a route like siding 2- mainline track section 2- siding 3- mainline track section 3- siding 4 is possible but a route like siding 2- mainline track section 2-siding 4-

mainline track section 4 is not possible. Another difference is hidden within the headway concept.

Table 2.1 One-to-one correspondences

Railway Element	Corresponding Job-Shop Element	Time Parameter in Railway	Corresponding Time Parameter in Job-Shop
Siding 1	Storage Area 1	w_{i1}	w_{i1}
Mainline Track Section 1	Machine 1	t_{i12} or t_{i21}	p_{i1}
Siding 2	Storage Area 2	w_{i2}	w_{i2}
Mainline Track Section 2	Machine 2	t_{i23} or t_{i32}	p_{i2}
Siding 3	Storage Area 3	w_{i3}	w_{i3}
Mainline Track Section 3	Machine 3	t_{i34} ort_{i43}	p_{i3}
Siding 4	Storage Area 4	w_{i4}	w_{i4}
Mainline Track Section 4	Machine 4	t_{i45} ort_{i54}	p_{i4}
Siding 5	Storage Area 5	w_{i5}	w_{i5}

Train scheduling on a line with double (or more) parallel unidirectional tracks-Flow Shop Scheduling: In Figure 2.3, a double track railway is seen, whereas, a multiple track railway is shown in Figure 2.4. If a number of parallel unidirectional tracks are under consideration, this means all trains on a track are moving in the same direction. Since trains moving in opposite directions do not interfere with each other at all (because they will not be sharing any resources), they can be scheduled independently on different computers. Thus, it is sufficient to consider only one of these parallel lines. This makes the problem similar to a flow shop problem, where all the jobs flow in the same direction, without skipping any machine. All the elements and correspondences in Table 2.1 are valid in this case, too. The only difference is that all the trains (jobs) move in the same direction.

Figure 2.3 A double track line [11]

Figure 2.4 A multiple track line from UK [12]

Train scheduling on a line with double (or more) bidirectional tracks-Parallel job shop scheduling: In this case, there are more than one parallel mainline track sections, mostly extending through stations-sidings. A train in any direction can use any one of the parallel tracks, although in general case all mainline tracks have principally allocated directions and trains in the opposite directions are allowed to use them only for overtaking purposes. At some points in the line (including but not limited to beginning and ending points of stations / sidings), parallel tracks are connected to each other through some specific devices called *switches.* Trains can pass from one parallel track to another overa switch. This is similar to a case where there is a job- shop with two or more parallel lines (of machines), and jobs can switch to other line (of machines)in intermediate storage areas. Parameters in Table 2.1 are also valid for this case. However, in the final solution, only the departure and arrival times of trains are not sufficient, the decision variables should also include which track they will use.

2.3 Description of Train Scheduling Activities

Up to now, we have mentioned about lots of similarities between train scheduling activities and classical scheduling activities. Now, it's time to mention about some differences. One of these differences is the representation of schedules. Traditionally, classical machine-based schedules are represented by Gantt diagrams, whereas train schedules are represented by time-distance diagrams-called *Train Graph.* Time-distance diagrams show the movement of a train along its path with respect to time.

There are two different basic orientations in railway time-distance diagrams. In the first orientation, the horizontal axis is the time axis and the vertical axis is the distance axis. In the other orientation, this scheme is reversed. Both of these orientations are widely used among different railway operating companies and infrastructure managers in the world. Neither of them has any specific advantage or disadvantage over the other one. Selection of one of them is just a matter of tradition. In Figure 2.5 and Figure 2.6, a couple of time-distance diagram examples are given.

At this point, it may be useful to define *scheduling at tactical level* and *scheduling at operational level* concepts used in train scheduling. Scheduling at tactical level is part of preparing timetables, which are valid for a pre-determined period (e.g., seasonal). These timetables indicate the arrival, departure and transit times of trains over various infrastructure elements, generally stations and sidings. Train operations are conducted according to these timetables and required to adhere to them as much as possible. For this reason, scheduling at tactical level is also called *timetabling*. Scheduling at operational level, also called rescheduling or traffic control, is a daily scheduling task, involving generation of modified schedules in case of deviations from the timetables in practical operation. Both types of problems can have completion time based performance measures. But, due date based performance measures are specific to scheduling at operational level. All operational scheduling activities are based on an underlying tactical schedule, because the due dates are input from there. Time-distance diagrams generally represent the tactical schedules (timetables). But, because of this tradition, operational schedules are plotted in the same format as well. Also, traffic control systems plot a diagram called “train graph”, which shows the actual movements of trains. This diagram uses the same time-distance format. In Figure 2.5, a sample train- graph for a section of railway line with 4 stations and 4 trains is seen. In this train graph, the horizontal axis is the time axis and the vertical axis is the distance axis, which shows the stations. In this figure, stations are numbered. The general practice is writing the name of the stations there.

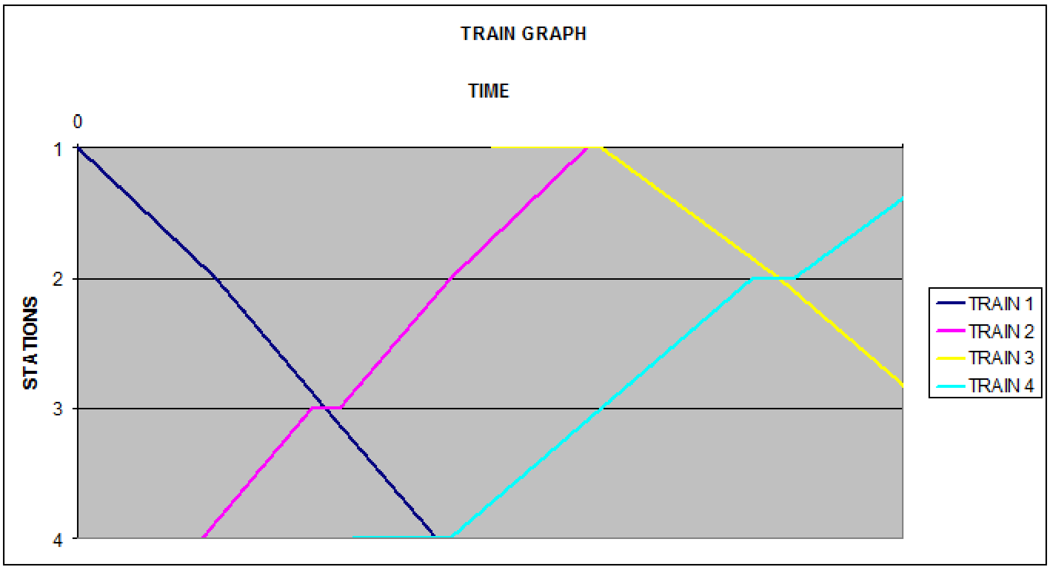

Figure 2.5 A sample railway time-distance train- graph

In Figure 2.6, the same line and train movements are depicted, but in the reverse form of orientation. As there is no rule for the choice between them and it's only a matter of preference and tradition, from now on, Turkish State Railways' tradition will be adopted; distance for the vertical axis and time for the horizontal axis. Note that, in both of the graphs, sometimes the paths of trains become parallel to the time axis, and this occurs only when trains are within the stations. This means, train stops at this station for some time.

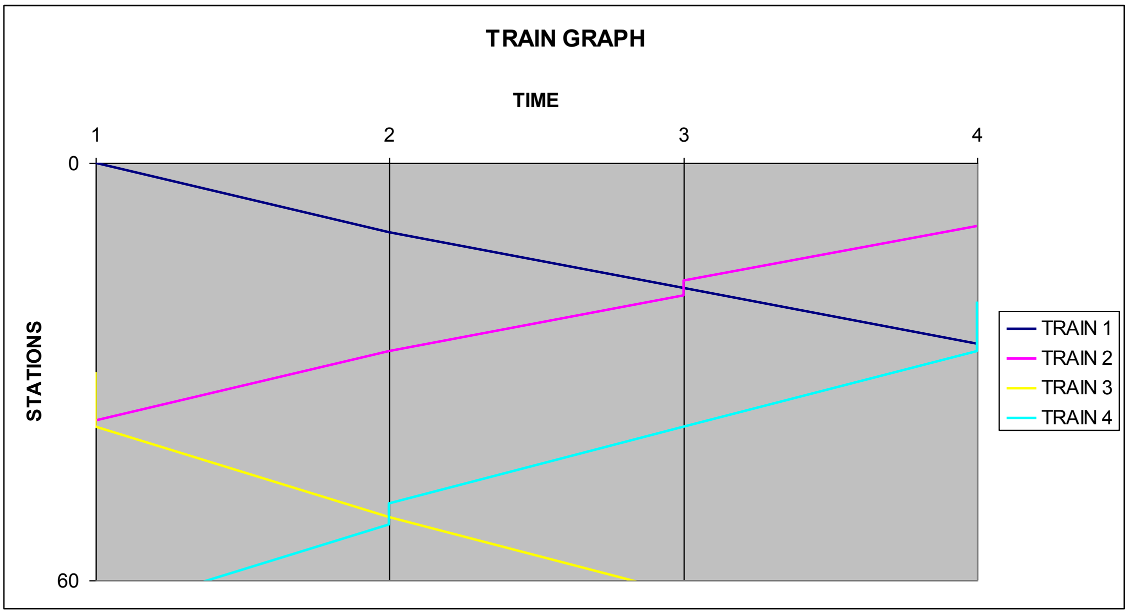

Figure 2.6 The same sample railway time-distance graph as Figure 2.1 in reversed orientation

2.3.1 Conflict Resolution

The backbone of train scheduling is conflict resolution. When two trains try to use the same infrastructure element in overlapping time intervals, a conflict occurs. Conflict means infeasibility and has to be resolved. There are various solutions to conflicts, which will be discussed soon, shortly after defining the types of conflicts.

2.3.1.1 Conflicts Between Trains in the Same Direction

There are two basic types of conflicts between trains running in the same direction. These are *headway conflicts* and *overtaking conflicts*. Some background information is needed to describe them.

In railway operations, safety is the primary concern. All the procedures are designed with safety in mind. Thanks to this fact, railway is one of the safest modes of transport. With safety in mind, a lot of rules and operational procedures are designed to avoid collisions between trains.

The big problem is that trains have much longer braking distances than road vehicles. A train travelling at 160 km/h on a flat (zero- gradient) track section can stop within more than 1000 meters. This dictates that, unlike the case for road vehicles, visual sight of the driver (like he/she sees the other, realizes that the other train is on the same track and applies brakes) is not sufficient to avoid collisions between rail vehicles. A minimum safety distance should always be maintained between the trains and the requirement for a stop to avoid collision should be acknowledged to train driver well before he/she actually sees the other train. When a train follows another train in the same direction with speed V, the minimum safety distance between these trains is the complete stopping (braking) distance of the succeeding (following) train, plus an additional safety distance. For succeeding Train 2 to be allowed to get any closer to preceding Train 1, it has to reduce its speed. Figure 2.7 illustrates this rule.

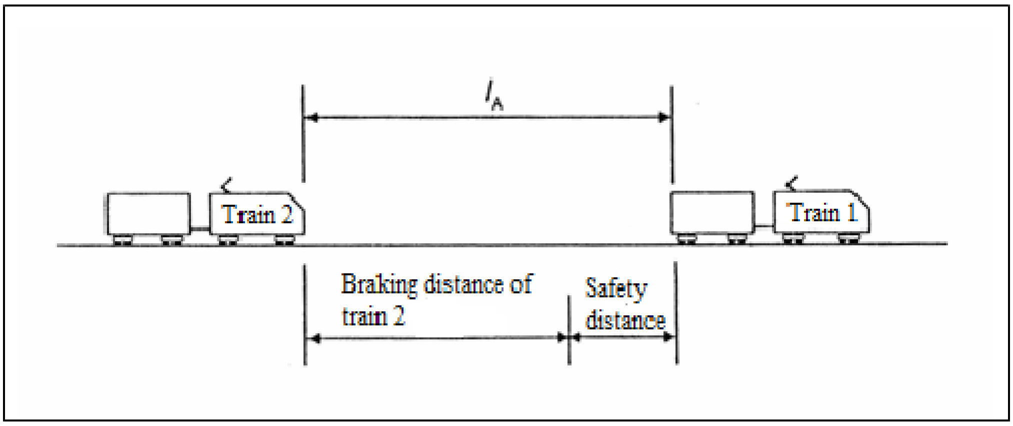

Figure 2.7 Minimum distance rule [13]

This rule is useful but, for most of the times, too complicated to apply. Stopping distance from V to 0 is dependent upon a number of factors, like the gradient (slope) of the line section, braking capability of the train, value of V itself, etc. Moreover, to apply this rule, all trains should have on board computer systems, which know the exact position of the train itself, exact position of all other (preceding/leading) trains to continuously calculate the braking distance. This is applicable in some railway systems like urban metro lines, but it's difficult to apply for intercity railways, where a number of different train types (passenger, freight) operate. To overcome this difficulty, a system called "block signaling system" is used. In this system, mainline track sections are divided into "blocks". Length of a block can range from one to several kilometers, or even the entire mainline track section (between two adjacent stations / sidings) can itself be a block. This choice depends on the conditions and requirements. No matter how long it is, the function of the block is the same: Only one train at a time is allowed to occupy the block. In order to allow another train into the block, block should be entirely cleared by the preceding train. The word "entirely cleared" means, even one axle of the train must not be within the block. All blocks are protected with signals. If there is a train within the block, the signal protecting the block will order any other train to stop before entering the block. But, because of long stopping distances of trains, this order should be given to drivers well before the entrance of the block. It can be given by the signal at the entrance of the previous block or by a special signal, called *distant signal*. Distant signal informs the driver about the status of the block signal ahead. When block lengths are carefully selected, block system guarantees that the minimum distance between two trains will always be maintained. Figure 2.8 illustrates the block system. In the figure, the block signals in the boundaries of the block sections are visible. The warning distance here is either the entire previous block or the distance between the distant signal and the block signal.

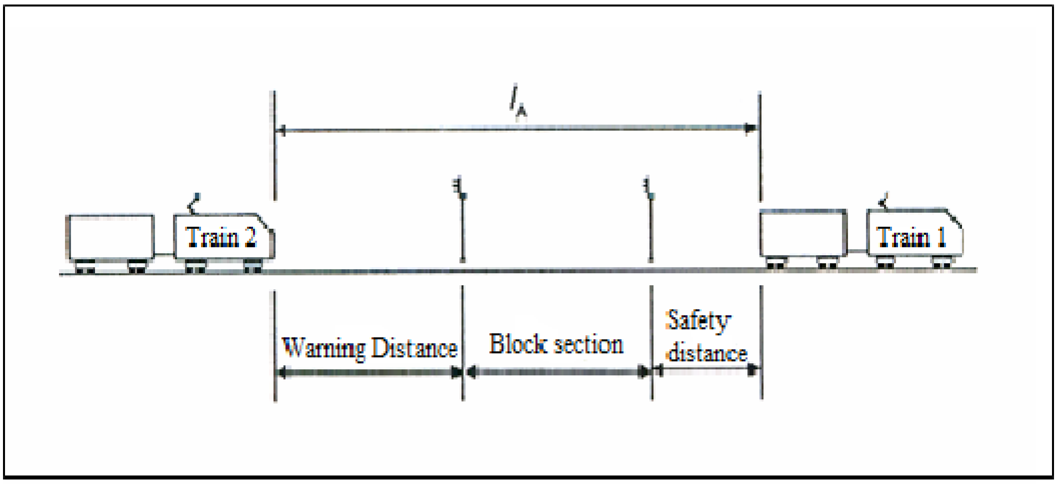

Figure 2.8 Illustration of a block system [13]

Up to here, we have seen that there should be a minimum distance between the trains. This is quite relevant for scheduling activities. When a train leaves a station / siding (i.e., enters the associated mainline track section), any train following it can be scheduled to enter the same mainline track section only after the preceding train is at least a minimum safety distance apart. However, incorporating this safety distance as a variable or a parameter would make the model extremely complicated. For simplicity, it is generally assumed that this distance constraint can be converted to a time constraint. So, when one train follows another one in the same direction, there should be at least a predetermined amount of time between those two trains. This amount of time is called *minimum headway*. Minimum headway is similar to sequence dependent setup time. An example of a minimum headway constraint that is expressed verbally is as follows: "If eastbound (westbound) train e (w) will enter the mainline section s immediately after eastbound (westbound) train f (x), it should enter the section at least h_d minutes later than this preceding train does." From now on, the word “minimum” will not be used. Only headway will be used and this will indeed refer to minimum headway.

These minimum headway constraints are in action not only when entering the mainline track sections. They are in action all the way along the mainline track sections. For simplicity, it is assumed that this fact can be simulated using two different headway constraints, *depart-depart headway* and *arrive-arrive headway*. The former is the headway constraint in the entrance of the section, and the latter is the one in the exit of the section. The verbal expression that was given in the previous paragraph was for depart-depart headway. Arrive-arrive headway version is as follows: "If eastbound (westbound) train e (w) will leave the mainline section s immediately after eastbound (westbound) train f (x), it should leave the section at least h_a minutes later than this preceding train does." Now, we will illustrate some headway conflicts on train graphs. In Figure 2.9, a depart-depart headway conflict is shown. Train 1 and Train 3 are trying to depart from Station 1 (and thus enter the related mainline track section) at the same time. This is an infeasible schedule. At least the depart-depart headway of time should pass between the entrance of them.

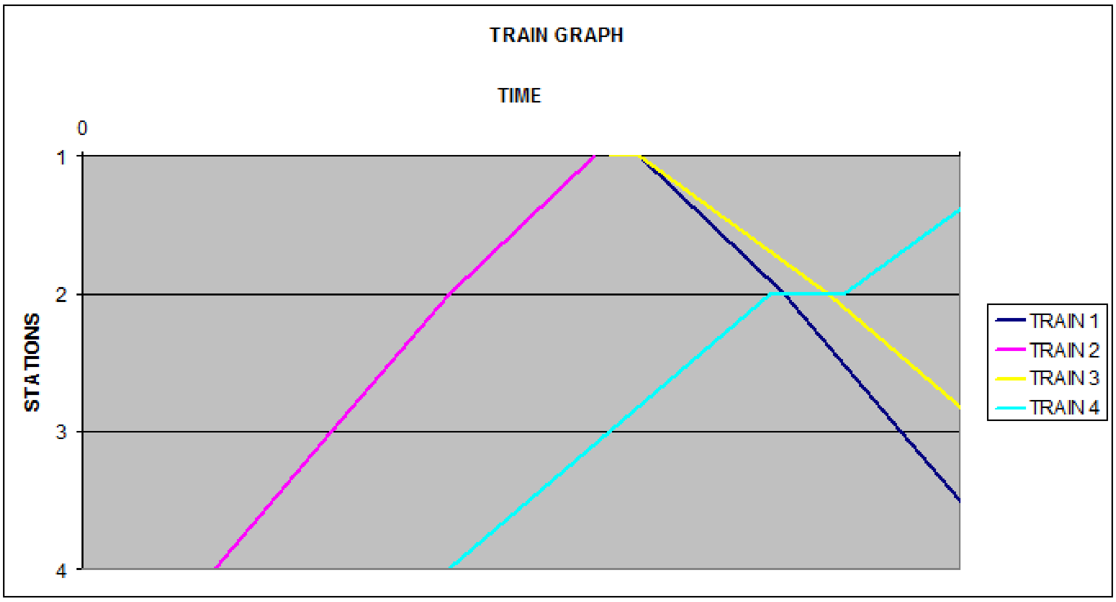

Figure 2.9 A depart-depart headway conflict between Train 1 and Train 3 at Station 1

In Figure 2.10, this conflict is resolved. Departure of Train 3 from Station 1 is delayed until the minimum depart-depart headway ($h_{d,d}$) amount of time passes. This schedule is feasible. It may or may not be optimal.

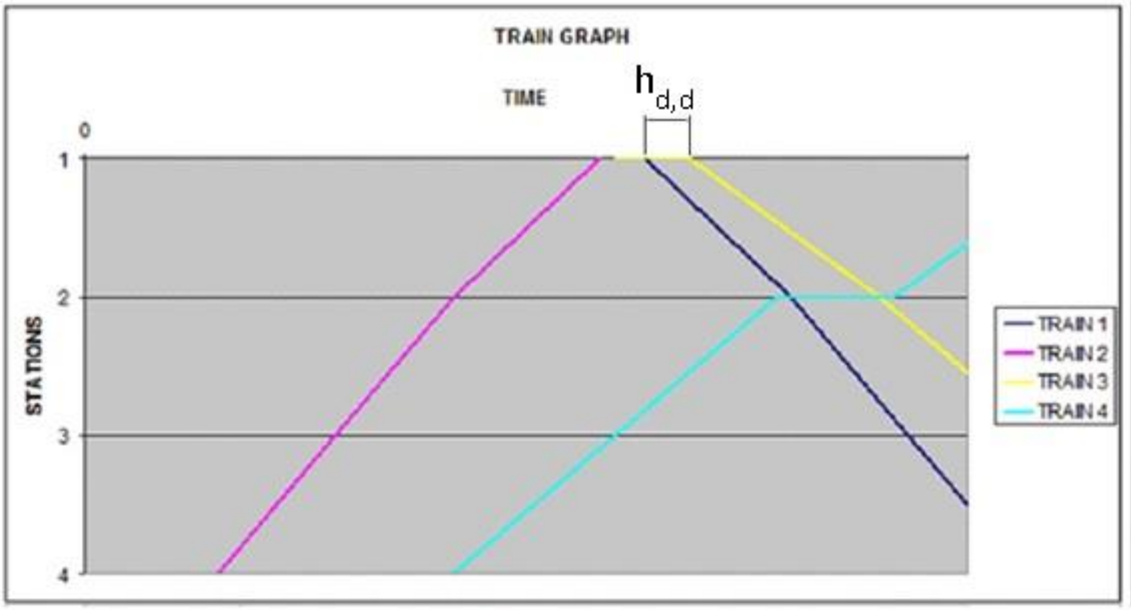

Figure 2.10 Depart-depart headway conflict resolved between Train 1 and Train 3 at Station 1

In Figure 2.11, an arrive-arrive headway conflict is depicted. Train 1 departs station 1 and enters the mainline track section 1 after Train 3. Depart-depart headway constraint is satisfied for station 1. However, Train 1 is faster than Train 3, and catches it up on the way. Two trains try to leave mainline track section 1 and enter Station 2 at the same time. This is an infeasible schedule. There is a conflict and this conflict has to be resolved.

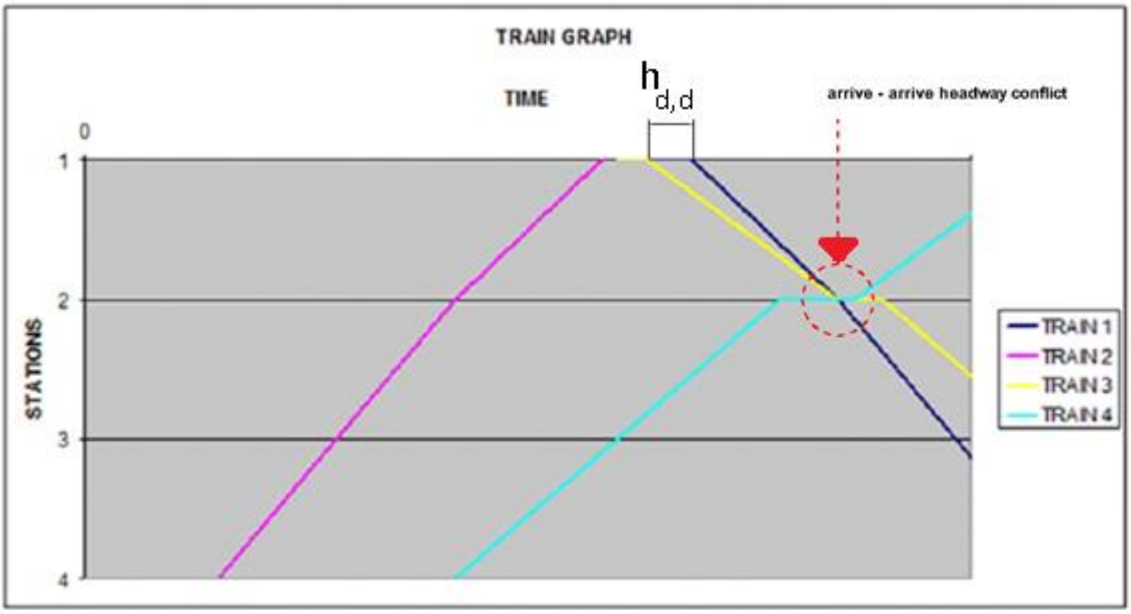

Figure 2.11 Arrive-arrive headway conflict between Train 1 and Train 3 at Station 2

There are two possible solutions to this conflict. Train 1 can be delayed to exit mainline track section 1 after Train 3. Figure 2.12 depicts this resolution. This resolution considers only the conflict that was in mention, not other conflicts.

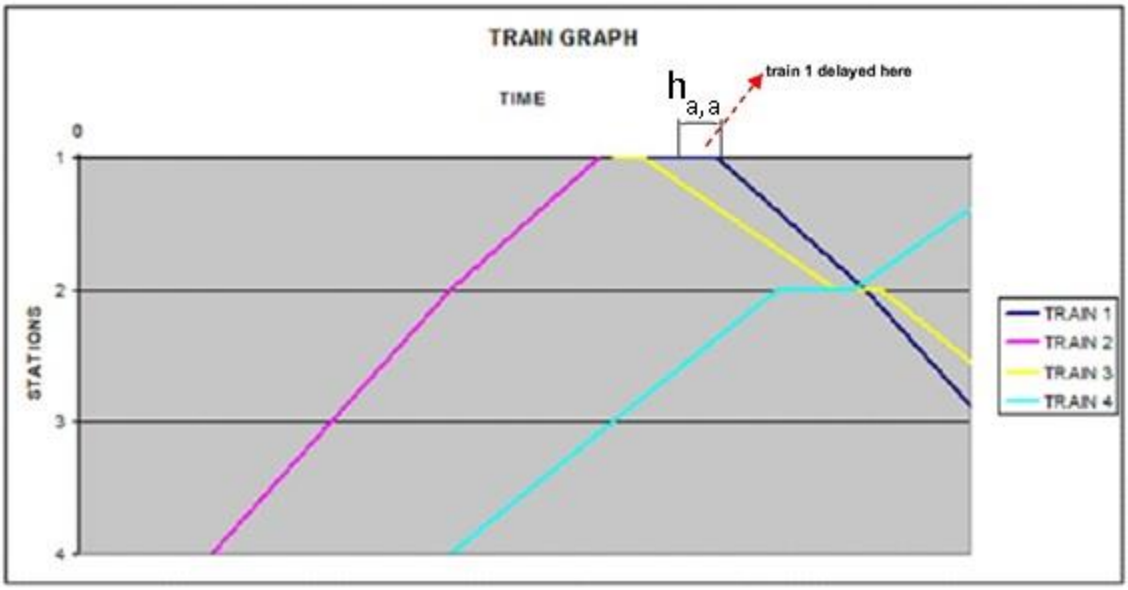

Figure 2.12 Resolution of the arrive-arrive headway conflict between Train 1 and Train 3 at Station 2 by further delaying Train 1 at Station 1

The other solution is, reversing the order of departure from station 1. In other words, allowing Train 1 to *overtake* Train 3 at station 1. Figure 2.13 depicts this resolution. Note that, it can be generally (except the case of double or more parallel bidirectional tracks) assumed that a train can overtake another train only at stations-sidings (just like order reversing cannot be made within the machines, only in intermediate storage areas). The train that will be overtaken has to stop and wait at the station until the overtaking train passes and depart-depart headway amount of time passes after the overtaking train had passed.

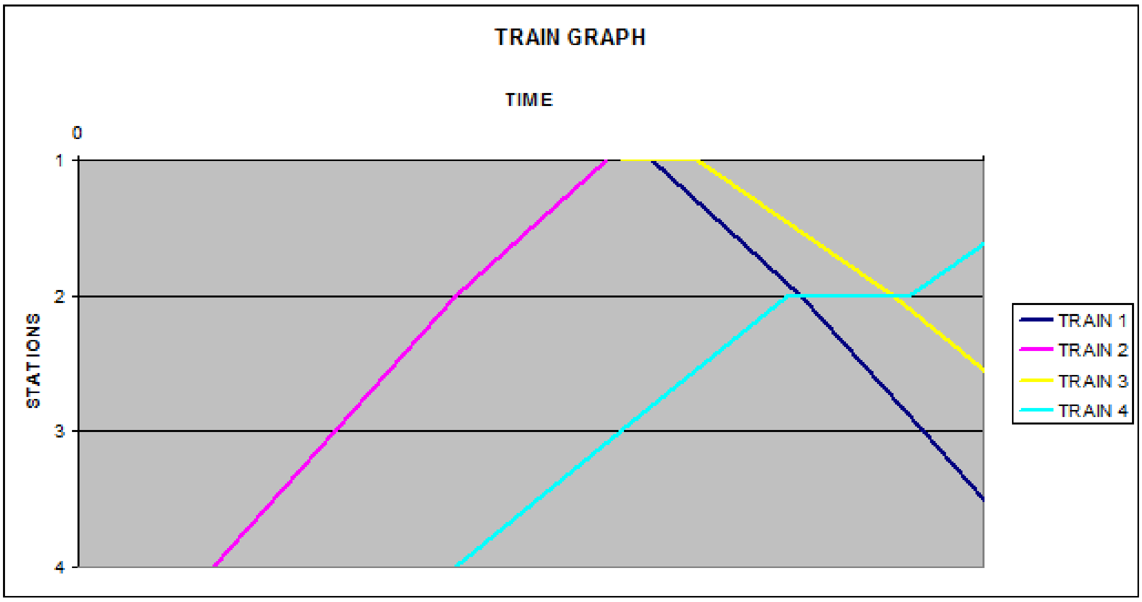

Figure 2.13 Conflict resolution by allowing Train 1 to overtake Train 3 at station 1

Another type of conflict that can arise between two trains in the same direction is the *overtaking conflict.* Figure 2.14 shows an example. Here, arrive-arrive and depart-depart headway constraints are satisfied at Stations 1-2 and 3. However, Train 1 is faster than Train 3. As shown in the figure, Train 1 tries to overtake Train 3 within the mainline track section 2. Both trains use the same mainline track, overtaking is not possible. Overtaking can only be done at the stations. This schedule is not feasible.

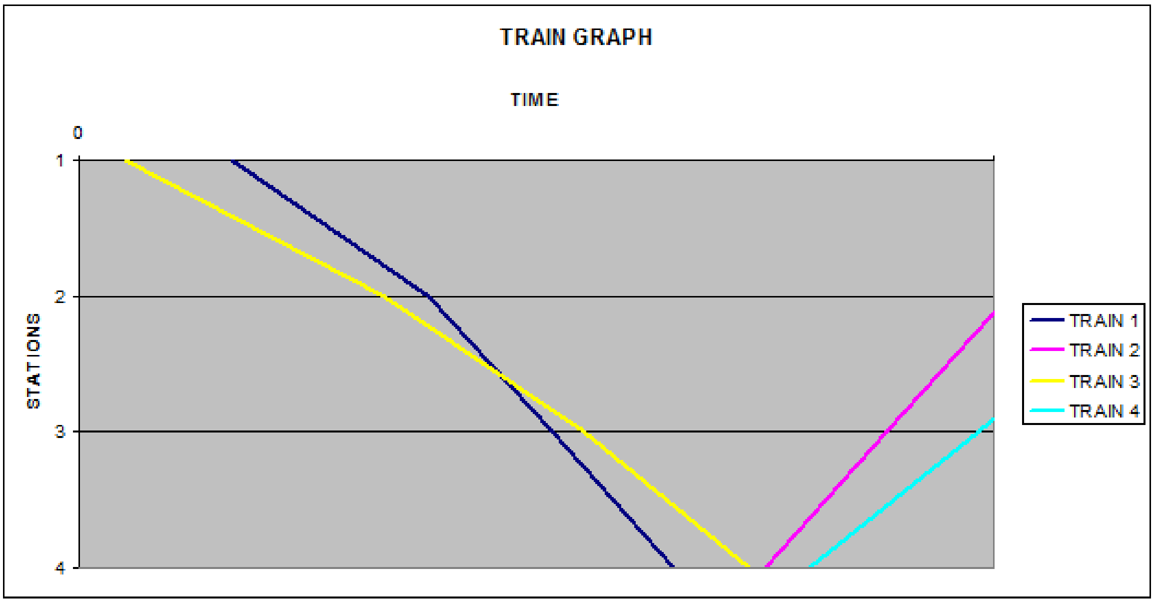

Figure 2.14 An overtaking conflict between Train 1 and Train 3 at mainline track section 2

There are two possible solutions to this overtaking conflict. Train 1 may be allowed to overtake Train 3 at Station 2. This is depicted in Figure 2.15.

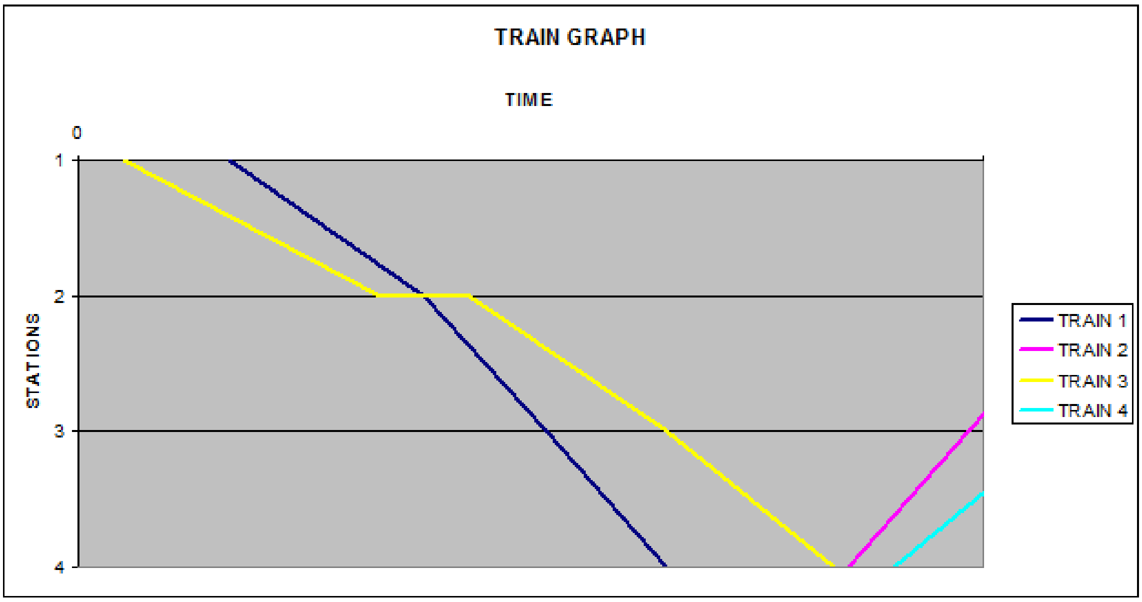

Figure 2.15Overtaking Conflict Resolution by allowing Train 1 to overtake Train 3

Alternatively, Train 1 can be forced to follow Train 3 until Station 3, with taking the headway constraints into account. This solution is depicted in Figure 2.16. It can be observed that, this solution creates another overtaking conflict between the same trains ahead in time.

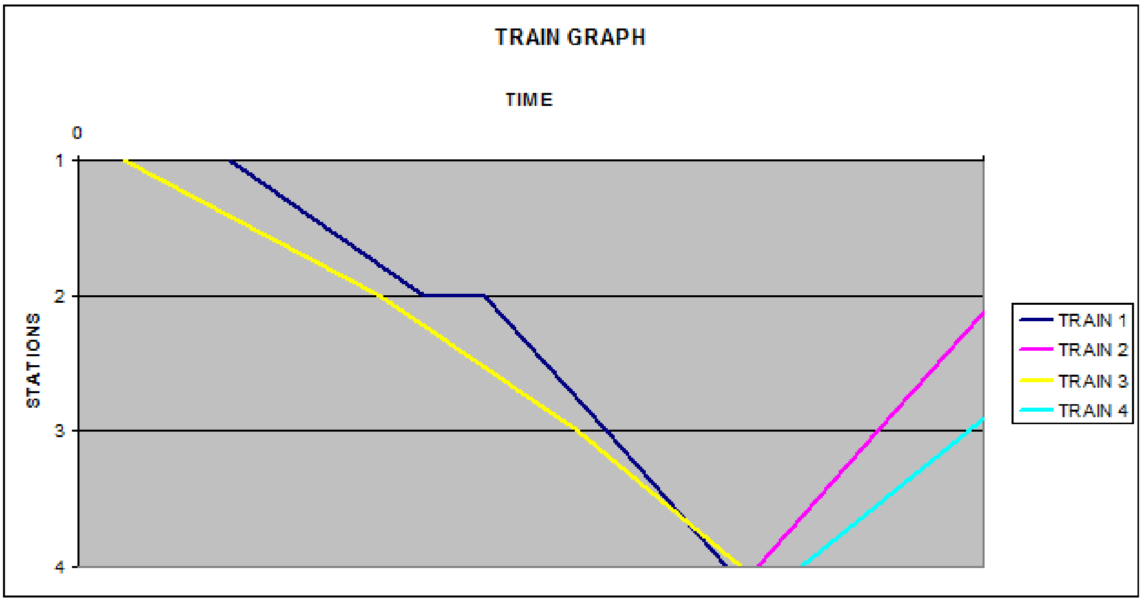

Figure 2.16Overtaking Conflict Resolution by delaying Train 1 at Station 2 so that it follows Train 3 with proper headway

2.3.1.2 Conflicts Between Trains in Opposite Directions on Bidirectional Mainline Tracks

There are mainly two types of conflicts: *Meeting conflicts* and *arrive-depart headway* conflicts.

Meeting Conflicts: Two trains moving in opposite directions on bidirectional tracks cannot cross each other within the mainline track sections. This crossing can only occur within stations and sidings. Figure 2.17 depicts a violation of this rule. We assume that the given train-graph belongs to a single track railway.

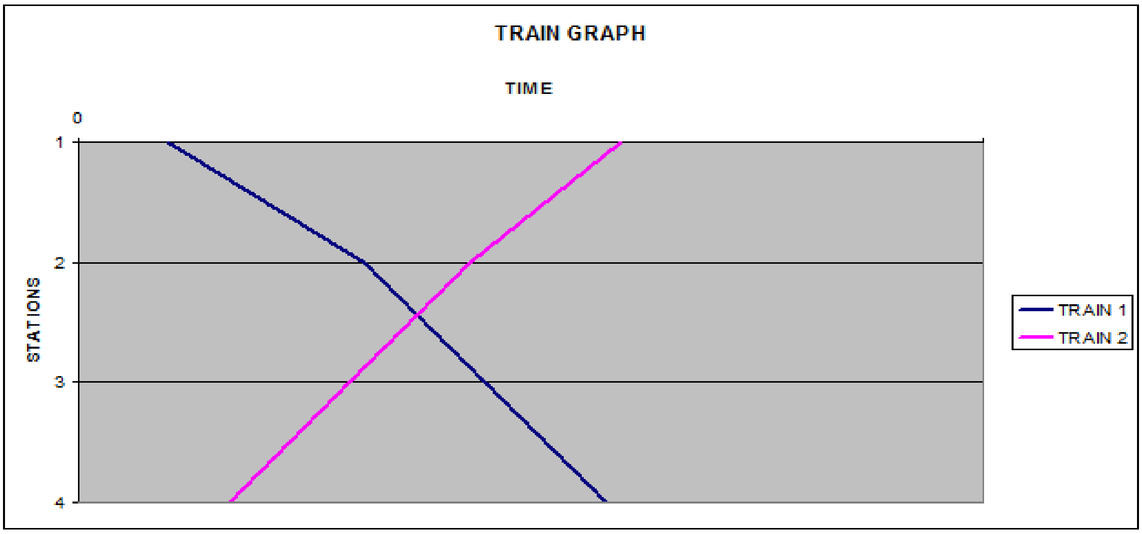

Figure 2.17Meeting conflict between Train 1 and Train 2 in the mainline track section between Stations 2 and 3

As seen in the figure, two trains try to cross each other on the mainline track section. This is an infeasible schedule. The problem is analogous to a job-shop scheduling problem with a single production line. The machine cannot process more than one job at a time. So, one of the jobs should be given the priority and the other job should wait in the intermediate storage area until the first job finishes. In train scheduling, one of the trains should be given the priority and the other train has to wait in the station until the train with the priority arrives. Figure 2.18 shows the resolution with Train 2 taking the priority. Figure 2.19 shows the opposite, Train 2 is forced to stop and wait.

Now, a difference between train scheduling and classical job- shop scheduling shows up. We defined mainline track sections as machines. In job- shop scheduling, one machine can process one job at a time. However, more than one train can enter the mainline track sections, provided that they are bound for the same direction and they satisfy all the headway constraints. Nevertheless, two trains traveling in opposite directions cannot be present on the same mainline track section in overlapping time intervals, since there is no possibility for them to cross each other. They can cross only at stations/sidings.

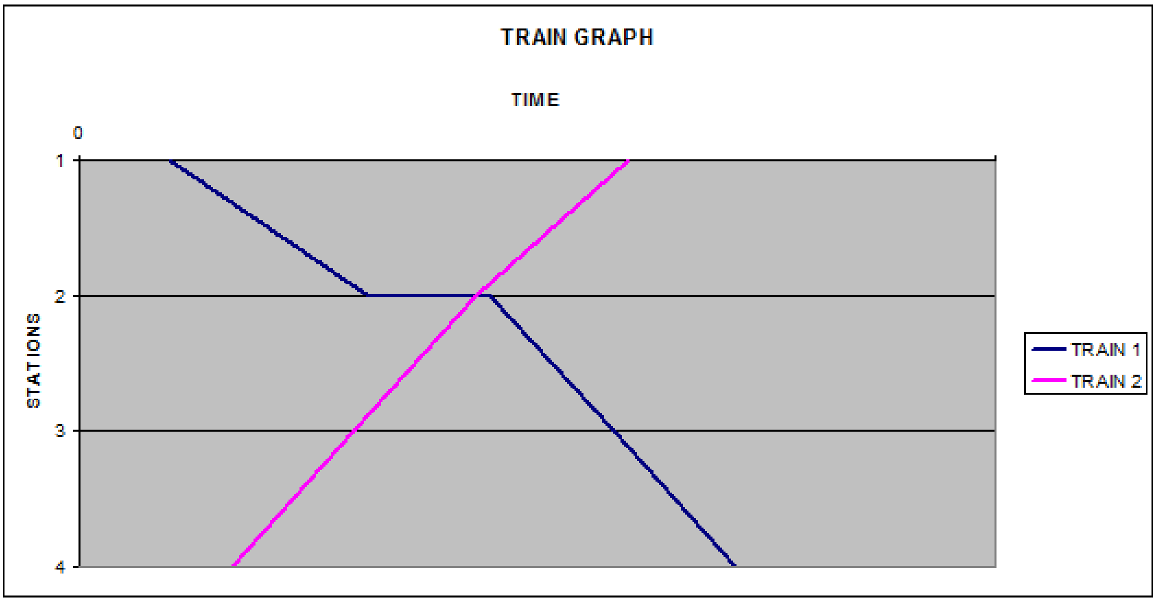

Figure 2.18Train 2 takes the priority.

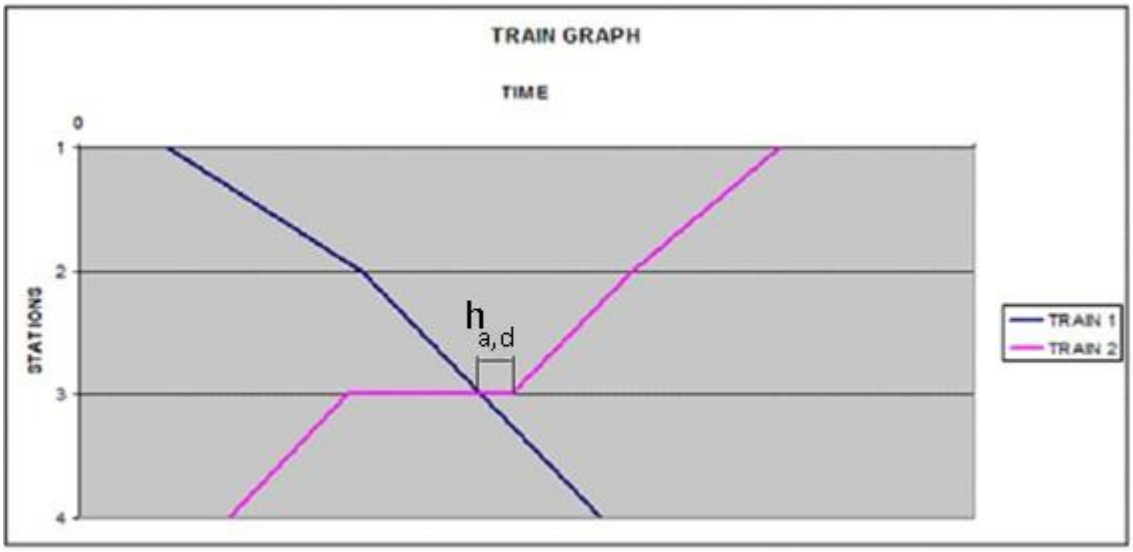

Figure 2.19 Train 1 takes the priority.

Arrive-depart Headway Constraints: In Figures 2.18 and 2.19, it can be observed that waiting trains do not depart just as the arriving trains arrive. This is logical. It takes some time to arrange the mainline track section for a change of traffic direction (like throwing and locking the switches, clearing the signals, etc.). This imposes an arrive-depart headway constraint. Figure 2.20 shows the infeasible solution to the last conflict. The solution is the same as the one in Figure 2.18, Train 1 takes the priority. However, arrive-depart headway constraint is violated.

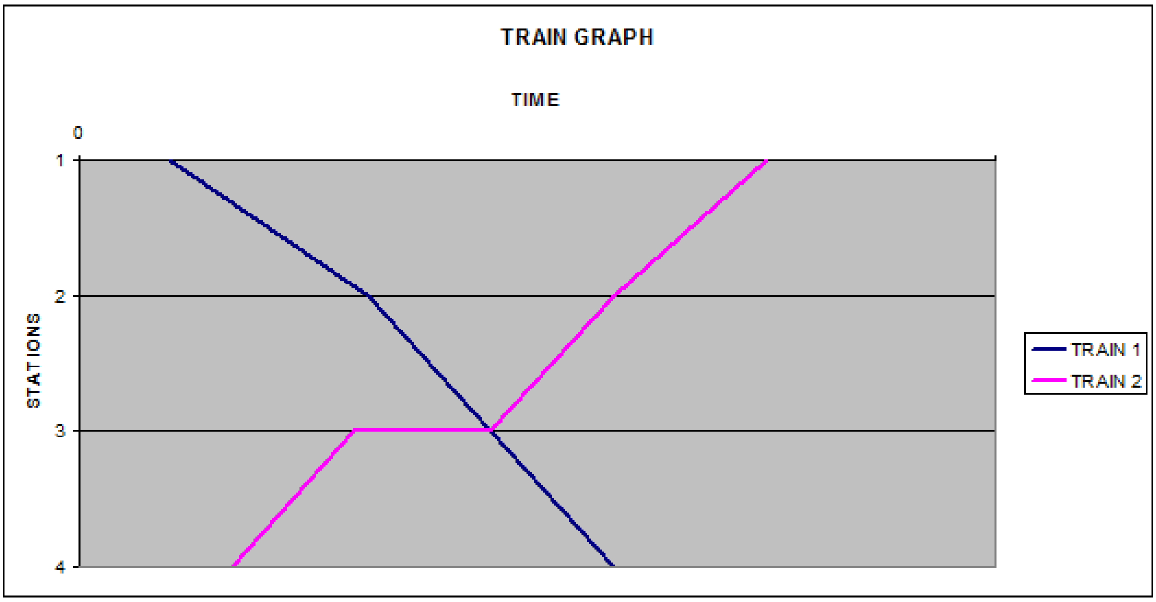

Figure 2.20 Violation of arrive-depart headway constraint between Train 1 and Train 2 at Station 3.

2.4 Literature Review (Cont'd)

Train scheduling has been increasingly popular among researchers during the last decade. This is partly due to the increasing train traffic and inability to make infrastructure investments to meet this demand. This inability is due to several reasons such as lack of finance, unavailability of land, opposition from environmentalist groups and intense lobbying activities of the shareholders of road transportation. Since it is not very possible to build new railway lines or increase the physical capacities of existing lines, railway industry now thinks of clever ways to exploit the existing infrastructure in a best efficient way. Making dispatching decisions in a better way is one of the possible methods to achieve this, thus, researchers show great interest to the subject. It has been drawing the attention of not only railway engineers, civil engineers, transportation engineers, etc., but also industrial engineers, mathematicians, computer engineers, computers scientists, etc. One can find dozens of research articles and theses related to the subject.

In our study, we focus on a hypothetical single track line. Although hypothetical, network structure is indeed derived from the TCDD line between Çukurhisar and

Arifiye. However, train types are hypothetical and minimum running times between adjacent stations are not derived from TCDD data, they are generated randomly. To our knowledge, our definition for variables and traffic constraints is unique.

Jovanovic [14] provided an enumeration scheme, similar to our enumeration scheme for station / siding capacities. However, his work included stations with one secondary line. Our work includes stations with two secondary lines as well.

The literature about train scheduling problem can be reviewed regarding a variety of classifications: The line topology for which the study is designed (single track line, double track line with unidirectional tracks, double track line with bidirectional tracks, multiple track line with unidirectional tracks, multiple track line with bidirectional tracks), simplifying assumptions for speeding up the solution, the modeling and solution approach (exact, metaheuristic, problem-specific heuristic, hybrid), the objective function used and other clever methods for speeding up the solution. Also, different authors study different aspects of the problem. Some focus on platform assignment in stations. Some study on urban railway networks, where no meeting or overtaking operations are really taking place, the main focus is modeling the consequences of disruptions, where vehicle scheduling is the main problem. Also, passenger connections at transfer stations are considered. Some papers concentrate solely on connections, the main decision to be made is whether a connecting train will wait for the late running train or not. Regarding the line topology and problem type, a simplifying decision is made now: Since this study is about the train scheduling problem on a single track line, only the studies for a single track line will be included here. These are generally either specifically designed for a single track line or for a multiple track line with bidirectional tracks. The latter will be included in the review because such algorithms can easily be adopted for a single track line by defining the number of parallel bidirectional tracks as one.

The objective function used is a very important consideration evaluating the success of the algorithm. The problem considered in our study can be classified as "A special-structured job- shop scheduling problem with limited intermediate storage, ready times, and sequence-dependent setup times, *'Total Weighted Tardiness'* being the objective function." Most of the job-shop scheduling problems studied in the literature focus on C_{max}, the maximum completion time, which makes the problem easier[15]. A similar situation could be observed in the train scheduling case, too. There are also some problems in the train scheduling literature that focus on minimizing the maximum delay of the trains. Without being able to provide a formal proof, our observation is that, if the maximum (weighted) delay is used as the objective function, then, the solver finds the optimal solution very quickly, compared to the case with total (weighted) delay as the objective function (This observed difference led us the apply a speed- up routine and adopt a multiobjective approach to increase the quality of solutions. Please see Chapter 4). The problem with this objective function is that it will not always provide good quality solutions. Table 2.2 provides a good example for this problem. In this problem, delays for all the trains are assumed to have equal weights, so that the reader can directly compare the delay values.

Table 2.2 An example of a bad outcome of a solution with maximum delay chosen as the objective function

	Delays in Solution 1 (minutes)	**Delays in Solution 2 (minutes)**
Train 1	9	10
Train 2	9	1
Train 3	9	0
Train 4	9	0
Train 5	9	1
Train 6	9	0

In the table, two different (supposed to be) feasible solutions are outlined. Any sensible planner would choose Solution 2 over Solution 1. However, when the objective function is Maximum Delay, the algorithm would choose Solution 1 over Solution 2. Such an objective function may pay off for the railway companies who pay compensation to their passengers if the train gets a large delay. For example, German Railways (Deutsche Bahn) reimburses 25% of the ticket price if a train is more than one hour late. However, if the compensation limit is not as large (for example, five minutes like the Spanish High Speed Rail case), an objective like “number of delayed trains” would be better.

Yalçınkaya and Bayhan [16] studied a problem very similar to ours, a single track line. They considered nearly all operational constraints like acceleration, braking, siding capacities and deadlock prevention. However, their study is mainly focused on generating a feasible solution in a short time. It does not contain any routines to generate a good or optimal solution with respect to a defined objective. To our observation, generating a feasible solution quickly is not a very big deal, because, in our trial runs, the solver can do this in a reasonable time. Indeed, for most of the cases, it finds either the optimal solution or a near- optimal solution in a relatively short amount of time. What takes time is, to prove that this solution is optimal or near optimal.

Törnquist and Persson [17], Acuna-Agost et al. [18], and Krasemann [19] based their algorithms on the same MIP formulation. These studies are for multiple bidirectional tracks. They divide the railway line into longitudinal segments and at each segment; number of parallel tracks is among the input parameters. Thus, which train will use which track is to be determined by the values of binary variables. Theoretically, by setting the number of parallel tracks in the relevant segments to one, it can directly be used for single track lines. These models, along with some other properties that won't be detailed here, generate a huge number of binary variables and it is not possible to solve them within a reasonable time. Hence, each team developed their own heuristic to solve the problem. Törnquist and Persson [17]classify the solution approaches into track swap, order swap and both combined. They then put a heuristically determined upper bound on the number of swaps with respect to the original timetable and pathing. This is indeed a very clever and applicable approach for problem instances within which the disruption is minor. However, in case of major disruptions where a number of trains are running well ahead of time, the trade-off between solution quality and computation time would get sharper. Acuna-Agost et al. [18] develop three approaches to obtain suboptimal solutions: Right-shift rescheduling, MIP based local search method and iterative MIP based local search method. MIP based local search method is based on the method developed by Törnquist and Persson [17], but they iteratively upgrade the upper bound on swaps. Note that, both approaches require an initial timetable with all choices of lines and platforms, and they are not applicable to single track railways operating without a timetable, like the US freight railways case. Our solution approach can be easily adopted to railways operated without a timetable. Törnquist Krasemann [19] utilizes a greedy algorithm to solve the problem.

Şahin [7], studying on the same problem (single track), totally avoids MIP formulations and goes with the heuristics. He proposed two algorithms. Both are similar to dispatcher's conflict resolution style. Dispatchers typically solve the conflicts as they occur, starting from the first conflict. The computer algorithm behaves similarly: Generates trains' free paths, finds the first conflict in the time axis and solves it. Each conflict resolution needs a decision to be solved. He constructed two algorithms for this decision making. The first one is the dispatcher's solution and imitates the Turkish Dispatcher's decision behavior. This solution is totally myopic, considers only the current conflict and does not take the future consequences of the current decision into account. The other heuristic method contains a look ahead routine, to guess each alternative decision's consequences. It is generally focused on minimizing the total delay without any weights imposed on them. This paper assumes that all the intermediate stations have infinite capacity.

Lee and Chen [20] consider "timetabling" and "pathing" together. The study is primarily aimed at double track lines, but, to our observation, it can easily be extended into single track line. This paper works as a combination of a probabilistic improvement heuristic with mathematical programming. It heuristically determines the order of the trains in the mainline track sections. This heuristic randomly picks up some mainline track sections and swaps the order of the trains over them according to some criteria. Then, it determines the assignment of trains to the tracks / platforms in the stations via binary integer programming. Making use of the information generated in the first two phases, it solves for the order of the trains that are assigned to the same track in the stations and finally the timetable itself. It is guaranteed to find a feasible solution in the initialization phase, since it allows trains to transverse the network one at a time. However, this solution will typically be a very bad one and we are suspicious that it will find a better feasible solution within the short time allowed for a rescheduling problem. Their objective is to minimize the associated cost with delays.

Higgins et al.[21] also study on a single track network. They relax the number of secondary lines in the formulation of the model and use the classical job- shop based MIP models. Then, they comparatively employ different heuristic approaches (local search, tabu search, genetic algorithm and hybrid algorithms).

Zhou and Zong [22] worked specifically on single track lines. They produced an MIP model of the train timetabling problem. They approach the station / siding capacity issue in a different way than ours. They discretize time into small enough intervals and they take all different junctions or segments of the line explicitly. They have binary variables which mean “This variable takes value 1 if train i occupies segment j at time interval k.” Usual precedence variables also remain in place.

Şahin [23] considered multiple aspects of a railroad planning problem, such as maintenance planning, scheduling from scratch, inserting additional trains into an existing schedule, planning new sidings, etc. He also went with discretization of time to consider the capacities of the stations / sidings. He modeled the railway network as a time- space network, which then translates into an integer programming formulation. He then combined a number of heuristic supports to boost the performance of the integer programming formulation.

Dessouky et al. [3] developed an algorithm that is applicable but not limited to single track lines. First, they obtain an explicit integer programming formulation of the train dispatching problem. Then, they employ a branch-and-bound procedure to solve the system. The MIP formulation does not explicitly involve any station / siding capacity. However, they employ a sequence fixing approach within the B&B procedure and they check the “safety” of the sequences and the resulting schedule against “deadlocks”. Of course, deadlock avoidance is not the only infeasibility that can occur due to exceeding the station / siding capacity.

Gafarov et al. [24] put on a different approach to train scheduling problem. They formulated and solved a train scheduling problem on a line with only two stations. In this case, there is only one mainline track section. They divided the track section into segments (blocks in railway operation) and used the entrance and exit times of trains to/from the blocks. This problem has a number of different objective functions. The authors describe a formal way of converting the train scheduling problem to a single-machine scheduling problem and then apply polynomial reductions.

Dündar and Şahin [25] developed a genetic algorithm to solve train rescheduling problem on a single track railway line. They also developed an artificial neural network model that imitates the decision procedures of train dispatchers, to benchmark the solutions obtained by the genetic algorithm. The genetic algorithm generates good feasible solutions in a short time. However, in their study, station / siding capacity constraints are neglected.

Espinoza-Aranda and Garcia- Rodenas [26] use the similar formulation of the train scheduling problem to that D'Ariano et al. [8] used, which was mainly for a double track line. However, they also emphasized the importance of using a proper objective function for delays. They also took into account the delays in intermediate stations. Their claim is such that their modeling approach can be used for any network configuration. This implies that it can be applied for single track lines. However, explicit consideration of station / siding capacities for limiting the number of simultaneous passing overtaking operations were not included. They employed an "Avoid Most Delayed Alternative Arc (AMDAA)" heuristic to obtain good results for the total weighted delay problem.

Albrecht et al. [27]handle the single track railway rescheduling problem with maintenance disruptions. In their study, they can close some segments of the railway line for maintenance works by introducing pseudo trains. Their objective function is minimizing the sum of train delays and maintenance delays. The heuristic they use for solving the problem is the "Problem Space Search" metaheuristic. They generated feasible solutions by means of some constructive processes and each movement of the constructive process was checked against deadlocks. However, apart from deadlocks, an explicit consideration of station/siding capacity is not present in the model.

Narayanaswami and Rangaraj [28] used a multi-agent system architecture for rescheduling on a single track railway line. Along with many publications, their algorithm includes a safety mechanism against deadlock situation, but not an explicit consideration of station / siding capacities. This model requires an initial feasible schedule (i.e. timetable) as an input and therefore not suited for railway lines operated without any timetable. Their objective function is to minimize the sum of all train durations. Their system architecture detects the conflicts and produces good solutions using some kind of auctioning and bidding operations carried out among the agents.

Li et al. [29] developed a multi-objective optimization model for train scheduling on a railway line. Their objectives are minimizing the total passenger times and carbon emission. Total passenger time is equivalent to assigning weights to the train delays in proportion to the number of passengers on the trains. The model is not particularly intended for single track railway lines, but, since it allows the user to define a number of different segments between adjacent stations, by setting this number to one, it can be adopted for single track lines. They employed a fuzzy mathematical modeling scheme for producing a good solution. No emphasis was put on any way to impose station / siding capacity constraints and deadlock avoidance.

Xu et al.[30] developed a scheduling model, mainly intended for double track lines. However, with some adjustments, it can be adopted for a single track line, based on a previous work by Mu and Dessouky. It involves a “switchable” procedure, where a fast train is allowed to traverse into the line intended for the opposite direction. They then developed a rule based solution heuristic, constructed over discrete event simulation architecture.

Narayanaswami and Rangaraj [31] developed an MILP model for rescheduling the traffic on a single track line, after one of the mainline track sections is closed to traffic for a duration. This model is intended to operate in stations which are in close proximity of the disruption location and applicable for small problem instances. Moreover, an explicit formulation for station / siding capacity constraints and/or deadlock avoidance was not provided.

Louwerse and Huisman [32] developed a model formulation for recovery from partial or complete blockades of railway lines. This formulation takes vehicle scheduling into account and provides cancelling of some trains as decision options. The model can be adopted to be used for single track railways, but it does not provide explicit information about station / siding capacity constraints and / or deadlock avoidance.

Fabris et al. [33] developed a tailor-made heuristic for the train timetabling problem over a complicated network. It can be adopted for timetabling on a single track railway line by defining the network accordingly and properly. Since it also includes platform assignment within the stations, it can also be adopted for modeling the station / siding capacities and avoiding deadlocks. However, it is a plain heuristic with no provision of proving near-optimality.

Furini [34] provided a fast heuristic for train timetabling on a railway network, which can be adopted the single track operation, since all arcs into and out from all nodes are defined separately. This study is mainly intended for timetabling, where different rail operators set out their desired timetable and the algorithm then tries to reduce the total deviation from the ideal timetables. Note that, this algorithm needs a predetermined order of running trains.

Xu etal. [35] developed a combined algorithm for a single track railway line, taking station / siding capacities and deadlocks into account. This algorithm is the combination of a tailormade heuristic and the genetic algorithm. Genetic algorithm is used to provide an effective velocity profile for the trains to keep up with the schedule. The tailormade heuristic, which is a rule based one, is to create a generic timetable for the trains.

Cacchiani et al. [36] initially work on a classical train scheduling model for a single track railway line. They forced the running times of trains between the adjacent stations to be constant, claiming that this creates a stronger LP relaxation. However, they are not taking station / siding capacities into account and if they are taken into account, solution quality, even the feasibility, may be threatened. They developed column generation algorithms to solve the LP relaxation of the model quicker than that can be done on a traditional solver.

Shafia et al. [37] worked on a comprehensive version of a timetabling problem on a single track railway line. It involves station capacities as well as acceleration and deceleration losses. They also paid attention to periodicity and robustness aspects of railway timetabling. Since the resulting problem is too different to tackle as an MILP, they approached the problem as a Periodic Event Scheduling Problem and used a simulated annealing approach to solve the instances.

Meng and Zhou [38] developed a stochastic programming model to produce efficient dispatch plan under different scenarios for capacity breakdowns. They employed a two-stage stochastic programming model, which is then solved by a branch-and-bound procedure. Precautions against exceeding the station / siding capacity and deadlock were not explicitly mentioned.

Burdett and Kozan [39] attacked the problem using a hybrid job- shop scheduling formulation. They pointed out the differences between the classical job- shop scheduling and train scheduling, and their approach is to eliminate problems created by these differences. They also take into account the station and siding capacity constraints by treating them as capacitated buffers in production scheduling. They then employ a constructive algorithm to create feasible train schedules in a short time. They state that, their algorithm can constitute a base for a metaheuristic or be used as a standalone approach. It is important to note that, their algorithm is mainly intended for makespan type of objective function.

Kuo et al. [40]proposed a freight train timetabling model with elastic demand, choosing total operational cost as the objective function. Their model is at tactical level, but they are also taking train movement feasibility into account. Modeling of the effect of station / siding capacities and/or deadlock avoidance were not explicitly mentioned. They used construction and improvement heuristics to create efficient train slots, and then a mixed-integer programming approach to select slots to operate, which is then solved by a column generation algorithm.

Corman et al. [41]proposed a model based on the alternative graph formulation and branch-and-bound algorithm for railway scheduling of D'Ariano [8]. This study is a further generalization of the former study by D'Ariano, over multi class train traffic, where delays of different trains are of different importance.

Liu and Kozan [42] focus on a coal carrying railway network, which comprises of single and double track sections. They treat double track sections as parallel-machine sections as in a job- shop, so they are implicitly considering station / siding capacity constraints and also deadlock constraints. They solve the problem using a constructive algorithm in conjunction with a best- insertion heuristic and tabu search.

Arangu et al. [43]developed a general-purpose arc-consistency algorithm for constraint satisfaction problems. To test their algorithm, rather than using common benchmark problems, they used a train scheduling model. The algorithm can be used for any type of network topology.

Luethi et al. [44] developed a simulation based rescheduling strategy for Swiss Railways. They did so, because Swiss Railways believe that, an effective rescheduling strategy helps to reduce the buffer times in the timetabling process. This means an increase in capacity.

Caimi et al. [45] proposed a network decomposition approach for a railway scheduling problem. They decompose the network into two groups: station areas and mainline track sections. By some adjustments, this algorithm can be used for the version of the problem in this thesis. Then, they used different heuristics and formulations (including an MILP formulation) to solve the problem.

Mu and Dessouky [46] proposed an optimization model for the U.S freight network. Since this model takes into account all the possible tracks as nodes, theoretically, it can be used to model the station and siding capacities. However, apart from a verbal statement like “no deadlock should occur”, they did not mention about such a procedure. They generated two different approaches: In the first the path each train will take is fixed and in the second it is relaxed. This clearly demonstrates a tradeoff between solution quality and solution speed. For the flexible path model, they developed several heuristics to overcome the intractability of the problem.

Cordone and Redaelli [47] proposed a different approach for railway timetabling profession. They claimed that there is a reciprocal type relation between the quality of the timetable and the actual passenger demand exists and this should be incorporated into the model. This makes the model nonlinear and they solve the model with their own branch-and-bound algorithm.

Cacchiani et al. [48] studied a different partial rescheduling problem. In their problem, they are not scheduling the trains from scratch. Instead, they are considering the problem of adding new freight trains to a line where all of the passenger trains and possibly some freight trains have already been scheduled and it is forbidden to alter the passenger trains’ timetables. Their model and algorithm is intended for any topology of railway line. Their objective function is to minimize the deviation of the additional train(s) from their respective ideal timetable(s). They employed a Lagrangian heuristic to solve the problem.

Burdett and Kozan [49] also study on the problem of adding new trains to an existing timetable. They compared three approaches to solve this problem: the first one is fixing all previously timetabled trains and inserting the new ones by means of constructive algorithms, with possible improvement by metaheuristics. The second one is fixing some previously timetabled trains and inserting the new ones by means of constructive algorithms, with possible improvement by metaheuristics. The third is employing a metaheuristics solution with none of the trains fixed. They used makespan as the objective function. They used their previously developed hybrid job-shop scheduling approach and simulated annealing for modeling and solution.

Ghoseiri and Morshedsolouk [50] applied ant colony optimization for scheduling trains on a single track railway line. The model they worked on was classical train scheduling model and no special precaution against exceeding station/siding capacities and/or deadlocks was mentioned.

Parkes and Ungar [51]studied on a mixed single and double track line with different territories of dispatching. Their model is based on the classical train scheduling model, where there is not explicit enforcement of station / siding capacity constraints and/or deadlock avoidance. They treat trains as agents that are aimed at maximizing their own profit and employ an auction based solution procedure to schedule trains across multiple dispatching territories.

Oliveira and Smith [52]also used constraint programming for modeling the feasible solutions for single track railway traffic. They then developed their own three- stage optimization algorithm. The three stages are simple rule-based dispatching, hill climbing and branch-and-bound. No special precaution against exceeding station/siding capacities and/or deadlocks was mentioned. Minimizing the total delay was the objective function.

Medanic and Dorfman [53], along with a few other authors, tackled the scheduling problem and energy minimization problem together. Since the resultant problem is too complicated to solve, they decomposed the problem into two subproblems: scheduling and energy optimization. Scheduling problem is solved by a greedy algorithm called "Travel Advance Strategy" and then velocity optimization comes. They assumed that all trains will run at constant velocity.

Higgins et al. [54]started their way with a classical train scheduling model and then they coupled it with energy consumption minimization. In this first model, locations of sidings were input parameters. They solved the problem with their own branch-and-bound algorithm. In the second stage, they used this model as a basis for determining the locations of the sidings. Then, they employed a decomposition algorithm, where the two subproblems were siding locations and detailed schedules. In the second problem, they also took siding capacity into account, although without giving details explicitly.

Carey [55] treated the scheduling and pathing problem as a combined problem and developed a model that can also be applied for a single track railway line. Since his work included pathing, it also worked fine for station / siding capacity constraints. To overcome the intractability of such a complex model, he employed a variety of strategies, including scheduling the trains one at a time and/or fixing the paths / schedules of already scheduled trains, speed fleeting or direction fleeting of the trains, etc.

Cai et al.[56] proposed greedy algorithms for train scheduling on a single track railway line. They relaxed the station / siding capacity, claiming that feasible solutions may be arrived by postsolution repair mechanisms. The greedy heuristics are simulation-based, focused on finding locally optimal solutions.

It is not quite easy to compare the studies in the subject. This is due to two facts: The first one is, since train scheduling problem is a strongly NP-Complete one, it is indeed a very hard one to solve. Many studies then relax the problem with some (sometimes very unrealistic) assumptions. Finding a basis for comparing studies, which have different assumptions, is absolutely not an easy task. Because, in principle, any two studies are solving two different problems. Also, infrastructure type is a very important factor. The type of infrastructure affects the primary structure of the problem. Generally, authors are studying on specific networks present in their country and their algorithms are developed respecting the special structure of the network that may be irrelevant for other networks.

There are also some additional barriers for benchmarking the solutions obtained. To compare to algorithms, it is necessary to compare the difficulties of the instances solved. Number of trains and number of stations play an important role in the difficulty of an instance, but they are not sufficient. Also, the order and the frequency of the trains running in the same directions are very powerful indicators of the difficulty of an instance. We clearly show this effect in Chapter 5.2.1. Unfortunately, very few (if any) authors share this data explicitly on their numerical experiments part.

With all the factors above combined, it is not surprising to see that, direct benchmarking is not a strong tradition in train scheduling literature. Very few of the authors directly benchmark their studies with others. When they do benchmark, this is not still a very exhaustive one. They select one or two studies among the previously published one and benchmark their studies. However, this is not very meaningful, given the large number of studies in the literature. Until some global pool of benchmark problems (like the ones for one machine scheduling problems) is created, direct benchmarking will be a deficit in the train scheduling literature.

Since direct benchmarking is not possible, the literature review is provided with an indirect benchmarking point of view. When explaining the published work, as the reader can observe, an implicit conversion is provided (like what we do and they don't, what they do and we don’t). In the light of these indirect comparisons, we reckon that, this study is unique enough to be classified as a solution methodology in the literature.

CHAPTER 3

THE MATHEMATICAL MODEL

Here, an MIP (Mixed Integer Programming) model, which considers station / siding capacities (i.e. number of parallel tracks within the stations / sidings), will be presented. The reason for doing so was an attempt to obtain a more realistic model. Since train scheduling is a strongly NP-complete problem, it is most frequently relaxed with some assumptions. Some of these assumptions are realistic, but some others are not. Neglecting station / siding capacities (i.e. assuming that intermediate stations / sidings can hold an infinite number of trains simultaneously) is a very unrealistic one. So, even at the cost of increasing complexity and greater sub optimality, we believe that it should be incorporated.

Before presenting the actual model, a generalization of station / siding capacity constraints will be given.

Suppose that a station / siding has n parallel tracks. Clearly, this capacity is exceeded when n+1 trains try to be present within the station / siding in overlapping periods. All the cases where the number of eastbound and westbound trains add up to n+1 should be enumerated for a complete modeling of the station / siding capacities. This will add an absolutely intractable amount of computational burden into the process; however, as it will be explained in Chapter 4, there is a way of handling them, with cleverly designed algorithms.

In general, station / siding capacity (total of n parallel tracks) can be exceeded when i westbound trains (i = 0,….,n+1) and $n+1-i$ eastbound trains try to occupy the station within overlapping time periods. We will deal explicitly with two cases: (i = 0) and (i = 1) and also the general case that arises when n = 2. Afterwards, we will implicitly deal with the general case when n > 2 and i > 1. Note that, the problem is symmetric. The case with *i* westbound trains and *n +1- i* eastbound trains is the same as the case with *i* eastbound trains and *n +1- i* westbound trains, the only difference is that the number of trains in two directions are swapped, creating a case which is symmetric with the former. So, here, we include the cases where $i \leq (n + 1) / 2$ into the explanation. That is, there are more eastbound trains than westbound trains in consideration. Of course, the symmetric cases are also included when the model is coded into the solver.

Case 1: *i* = 0 for any n.

This means, station / siding capacity can be exceeded by *n + 1* eastbound trains. The constraint that first comes to mind can be verbally expressed as follows: "Any eastbound train cannot overtake n trains at the same station, which has n parallel tracks." This is clearly true, because, if it tries to do so, n tracks will be occupied with n trains. The last eastbound train will need an $(n+1)^{st}$ track to overtake all of them, which does not exist. However, although true, this is not a sufficient approach. In Table 3.1, an example is given. Suppose that Table 3.1 shows the arrival times (A_j) and departure times (D_j) of four different trains at (from) station 5, which has 3 parallel tracks (n = 3). In this case, these trains are bound for the same direction.

Table 3.1 Sample arrival and departure times with respect to time 0

Train j	**A_j(minutes)**	**D_j (minutes)**
Train 1	120	128
Train 3	122	131
Train 5	124	134
Train 7	126	137

It is clearly seen from table that, although there is no overtaking between any two of these four trains, between t = 126 and t = 128, all of the four trains are within the station. This is an infeasible movement and it must not be given as an output by the model.

Feasibility can be guaranteed by the disjunctive constraint, whose verbal expression is as follows: “If any *n + 1* trains in the same direction will use the same station with n parallel tracks, then, the last train into the station may enter the station only after the departure of first train to leave the station.” Such a constraint may yield the arrival and departure times shown in Table 3.2. This is feasible. The values in Table 3.2 are NOT related with the values in Table 3.1, they are just sample values.

Table 3.2 Sample (feasible) arrival and departure times
with respect to time 0

Train j	**A_j(minutes)**	**D_j (minutes)**
Train 1	120	128
Train 3	122	134
Train 5	124	137
Train 7	130	131

Case 2: $i = n$ for any n.

This is the case that n eastbound trains and one westbound train (or vice versa) demand to pass through the same station, which has n parallel tracks. However, all of those n + 1 trains cannot be present within the station simultaneously. Furthermore, if such a meeting would occur, one of those n trains have to be the last one to enter the station. Figure 3.1 illustrates this situation for $n = 2$. Two trains already arrived at the station and occupied all of the available tracks. Train 2 cannot enter and any of the odd-numbered trains cannot exit. This is a kind of deadlock situation. In this situation, the only way of getting the traffic flow again is to reverse Train 1 or Train 3, send it out of the station, take Train 2 into the vacated line of the station, sending the eastbound train that was not reversed, and taking the eastbound train that war reversed into the station. This is a big time loss and such case should not be allowed to happen.

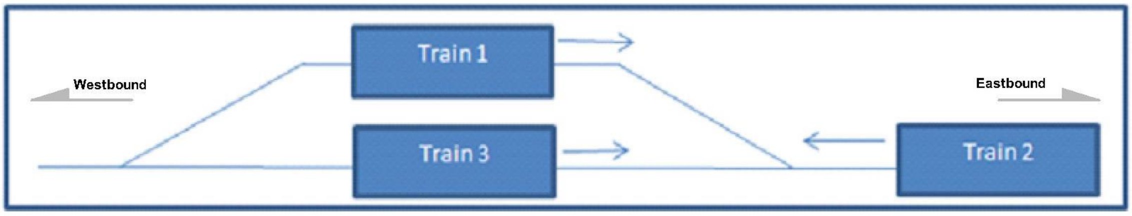

Figure 3.1Train 2 cannot enter the station.

Feasibility can be guaranteed by the disjunctive constraint, whose verbal expression is as follows: "If any n eastbound (westbound) trains and one westbound (eastbound) train are to meet at station s, then, the last eastbound (westbound) train into station can enter only after the departure of the first eastbound (westbound) train from the station." Figure 3.2 illustrates this situation for $n = 2$. In this figure, Train 3 is the last eastbound train moving into the station. Train 1 itself has to wait for departure until Train 2 arrives, thanks to the meeting conflict resolution constraints (which will be explained shortly). So, after Train 2 arrives, all of the tracks within the stations will be occupied. The disjunctive constraint will then keep Train 3 from entering the station until Train 1 clears the station, so there will be one empty track to accommodate Train 3.

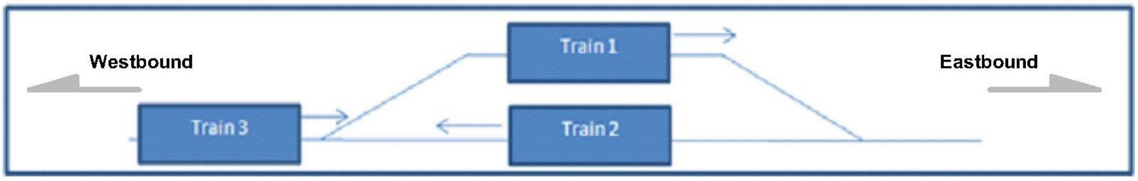

Figure 3.2Feasibility will be guaranteedby forcing Train 3 to wait without entering the station until Train 1 leaves the station.

Now, there will come a proposition, which will define the general case for any n and i = 1.

Proposition 1: All station / siding capacity and deadlock conflicts for this case will be resolved by one standard and two disjunctive constraint groups. Moreover, these constraints do not prohibit any valuable solutions. Below, verbal expressions for these constraints are provided.

1) Any train cannot overtake more than n-1 trains at station s, which has n parallel tracks.

2) If n eastbound trains will pass through station s and if the last train overtakes all of the remaining trains at station s, then, it cannot meet with any trains at station s.

3) If n eastbound trains will pass through station s and if the last train does not overtake all of the remaining trains at station s (i.e. at least one of the remaining trains depart s before the last train into s), then, the last eastbound train into station s can enter only after the departure of the first eastbound train from station s.

Now, we will *explain* the proposition, since it will be too simple to call it a proof. The first part is very straightforward. Any train cannot overtake n trains at the same station, because n trains to be overtaken will occupy all of the n tracks in the station and the overtaking train will not find any track to pass through. The second part can easily be understood by examining Figure 3.1. Suppose that, in the figure, Train 1 is trying to overtake Train 3 at the station. At the same time, it tries to meet with Train 2. However, it cannot depart before Train 2's arrival into the station and if it is within the station then Train 2 has nowhere to arrive. The same situation will arise if Train 2 enters the station before Train 3, which can be observed in Figure 3.2. In Figure 3.2, Train 1 cannot overtake Train 3, because station does not have an additional track to facilitate this. Part 3 is a little trickier. The situation is illustrated in Figure 3.3, again for $n = 2$. In this figure, Train 1 and Train 3 will pass through the same station. If Train 1 will not overtake Train 3, then, if it will meet with another train, it should enter the station after Train 3 departs. If it will not meet with any train, it may actually be within the station simultaneously with Train 3. However, preventing it from doing so will not give any harm to the objective function. Since it will not overtake Train 3, it has to wait until departure time of Train 3 + depart-depart headway to leave the station. Typically, depart-depart headway will not be smaller than arrive-arrive headway, so postponing the arrival of Train 1will not cause any additional delay to Train 1.

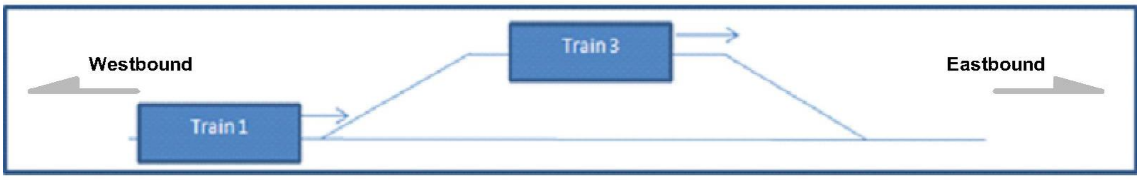

Figure 3.3Illustration of part 3 of proposition 1.

Now, we will illustrate how these rules will also prevent deadlocks. In Figure 3.4, a typical deadlock situation is illustrated. In this situation, Train 1will not overtake Train 3, because, if it did, it would not be allowed to meet with Train 2 at this station (This follows from part 2 of proposition 1). So, following from part 3, it will have to wait until the departure of Train 3 from the station. Conflict resolution constraints dictate that, departure of Train 3 from the station requires Train 4 to be held at the adjacent station (i.e. previous station in the itinerary of Train 4), so, a situation like in the Figure 3.4 will never arise.

Figure 3.4Illustration of deadlock prevention

Please note that, the constraints that arise when i = 1 (i.e. there is only one eastbound train in conflict) need the order of entrances of eastbound trains into the station, within themselves. However, these do not need the entrance orders of trains in both directions. That is, they do not need to know if one eastbound train and one westbound train will meet at any station, which one will enter the station first. These constraints only care that whether meeting between any particular train couple will occur or not.

In the general case where $i \geq 2$,such binary variables are needed:

$$c_{ij}^{s} = \begin{cases} 1, & \text{if eastbound train i and westbound train j meet at station s AND train i enters the station before train j,} \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{ij}^{s} = \begin{cases} 1, & \text{if eastbound train i and westbound train j meet at station s AND train j enters the station before train i,} \\ 0, & \text{otherwise.} \end{cases}$$

These meeting variables explicitly state which train waits which one. Now, we will proceed with showing that, such explicit statement is not necessary if $n = 2$ (i.e. for a station that has one through mainline track and one secondary line). It is quite straightforward. We already illustrated that, when $i = 0$, only eastbound trains are included and when i = 1, such an explicit statement is not necessary, because these constraints do not care about the precedence of westbound trains with respect to eastbound trains. If $n = 2$, then, i would be either 0 or 1. If $i = 2$, then it only represents the symmetric case where there are 2 westbound and 1 eastbound trains, whereas, *if* $i = 3$, then it only represents the symmetric case where there are 3 westbound and no eastbound trains. Thus, now, the binary variables for stations with 2 parallel tracks are reduced to:

$$c_{ij}^{s} = \begin{cases} 1, \text{ if eastbound train i meets with westbound train j at station s,} \\ \\ 0, \text{ otherwise.} \end{cases}$$

The former binary variables are still required for stations having strictly greater than 2 parallel tracks. We will see this shortly.

If $n \geq 3$ and $i \geq 2$, situation gets more complicated. Figure 3.5 and Figure 3.6 are illustrations of this situation for $n = 3$ and $i = 2$.

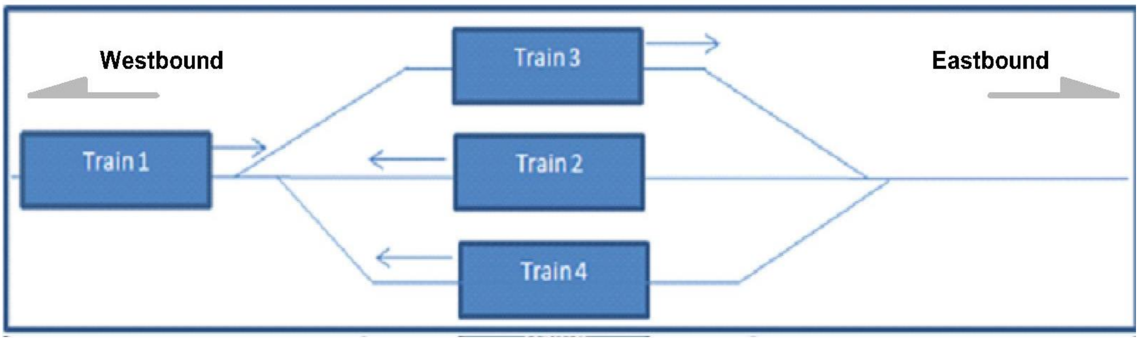

Figure 3.5An illustration for n = 3 and i = 2

Here, Train 1 is the last to arrive at the station among these four trains (not only the eastbound ones). The following are the valid limitations and possibilities when such an instance occurs:

1) Train 1 cannot overtake Train 3. Indeed, it has to postpone its arrival into the station to the time after the departure of Train 3.

2) Train 2 could have arrived later than Train 4 and in this case it may or may not overtake Train 4.

3) Train 4 could have arrived later than Train 2 and in this case it may or may not overtake Train 2.

In Figure 3.6, another illustration for n ≥ 3 and $i \geq 2$ (i.e. there are at least 3 parallel tracks in the station/siding and there are at least two trains in each direction) is depicted.

Here, Train 2 is the last to arrive at the stations among all the four trains (not only the westbound ones). The following are the limitations and possibilities when such an instance occurs:

1) Train 2 cannot overtake Train 4. Indeed, it has to postpone its arrival into the station to the time after the departure of Train 4.

2) Train 1 could have arrived later than Train 3 and in this case it may or may not overtake Train 3.

3) Train 3 could have arrived later than Train 1 and in this case it may or may not overtake Train 1.

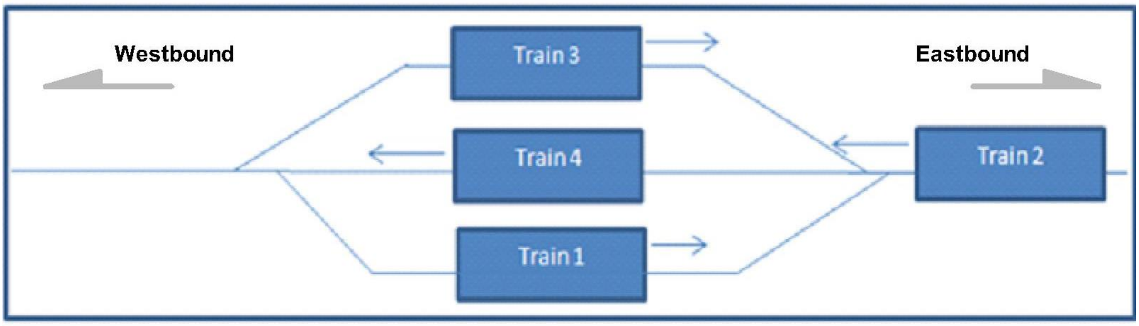

Figure 3.6Another illustration for n = 3 and i = 2

As seen in the figures, both instances involve exactly the same trains and exactly the same station. However, who is the last to enter the station (among all eastbound and westbound trains) affects the limitations and possibilities. This calls for the requirement that, binary variables should not only tell where the meeting will occur, but also, they should tell which train will arrive at the station last.

Hence, for stations which have n > 2, the following decision variables should still apply (now they are renamed):

$$c_{ij}^{s} = \begin{cases} 1, \text{ if eastbound train i and westbound train j meet at station s and train i enters the station before train j,} \\ 0, \text{ otherwise.} \end{cases}$$

$$f_{ij}^{s} = \begin{cases} 1, & \text{if eastbound train i and westbound train j meet at station s and train j enters the station before train i,} \\ 0, & \text{otherwise.} \end{cases}$$

Also, we revert to the classical train scheduling model for the trains in the same direction, regardless of the number of stations and sidings:

$$b_{ik}^{s} = \begin{cases} 1, & \text{if eastbound train i enters the mainline track section between stations s and s + 1 before eastbound train k,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\hat{b}_{jl}^{s} = \begin{cases} 1, & \text{if westbound train j enters the mainline track section between stations s and s- 1 before westbound train l,} \\ 0, & \text{otherwise.} \end{cases}$$

In Figure 3.7, and Figure 3.8, related indices are depicted.

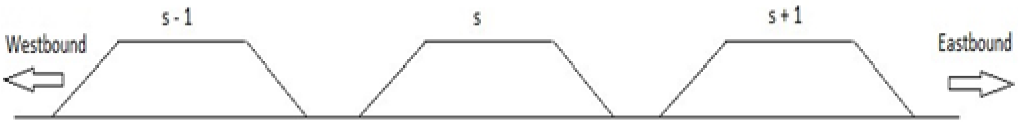

Figure 3.7Station indices

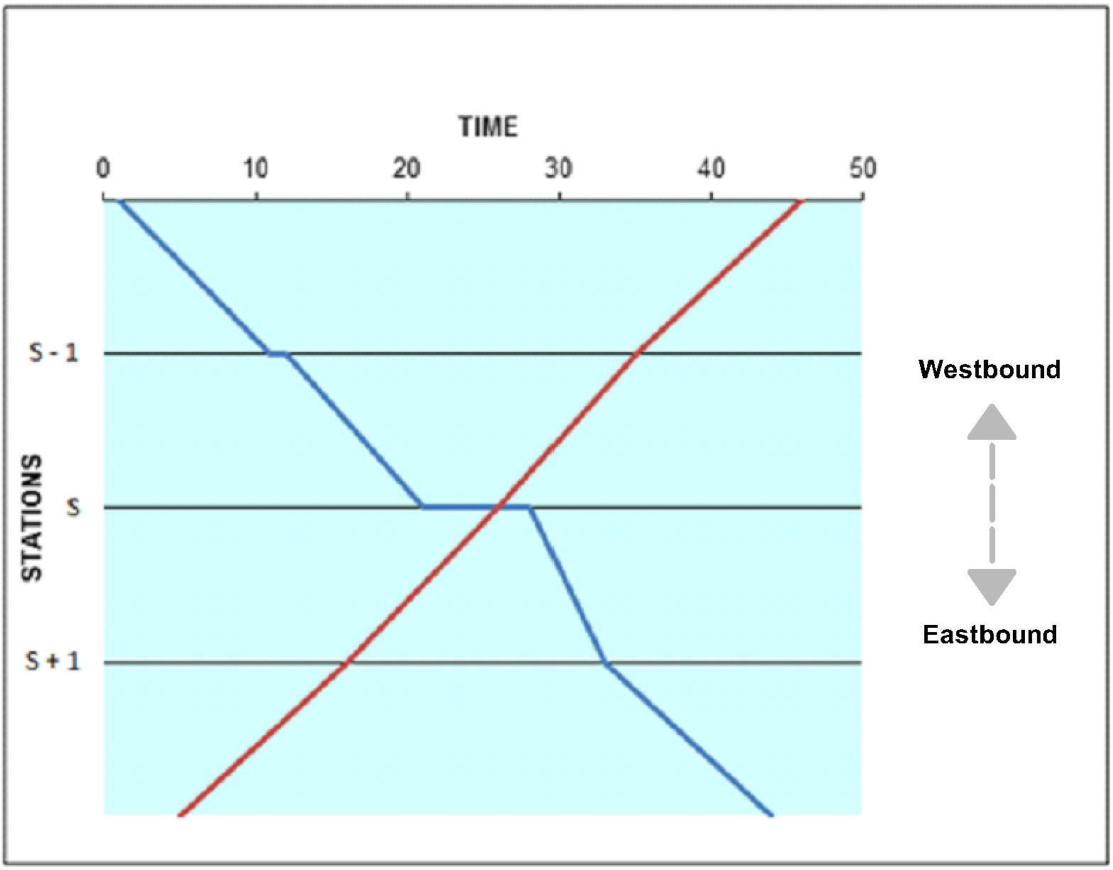

Figure 3.8Station indices on a time-distance graph.

In Table 3.3, we give the values of decision variables for the situation in Figure 3.5, which is replicated in Figure 3.6. In order such a position to arise, the bold values should be strictly true. The other ones can have the alternative value, too.

Table 3.3 Values of the binary variables in the instance shown in Figure 3.5.

$c_{1,2}^{s}$	$f_{1,2}^{s}$	$c_{1,4}^{s}$	$f_{1,4}^{s}$	$c_{3,2}^{s}$	$f_{3,2}^{s}$	$c_{3,4}^{s}$	$f_{3,4}^{s}$	$b_{1,3}^{s}$	$b_{1,3}^{s+1}$	$b_{2,4}^{s+1}$	$b_{2,4}^{s}$
0	**1**	**0**	**1**	0	1	1	0	**0**	**0**	0	1

In Figure 3.5, there are four meetings in the station. The meeting between Train 1 and Train 2, The meeting between Train 1 and Train 4, The meeting between Train 3 and Train 2 and The meeting between Train 3 and Train 4. For the meeting between Train 1 and Train 2, we can observe that, meeting occurs at this station and Train 2 arrived at the station before Train 1. So, $c_{1,2}^{s} = 0$ and $f_{1,2}^{s} = 1$. Exactly the same applies for Train 1 and Train 4. However, for the meeting between Train 3 and Train 2 and the meeting between Train 3 and Train 4, we don't know whether the eastbound or the westbound train arrived at the station before the other. So, their respective values are not written in bold. $c_{3,4}^{s} = 0$ and $f_{3,4}^{s} = 1$ may also be true here.

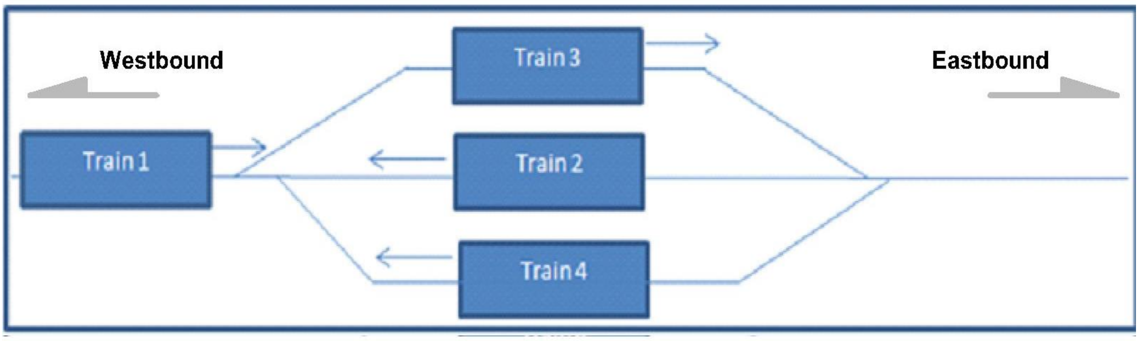

Figure 3.9 Replication of Figure 3.5

The general rule for such cases is as follows:

"If i westbound trains and $n + 1$-$i$eastbound trains will meet each other at station s, which has n parallel tracks, such that $2 \leq i \leq n$-1 and $n \geq 3$, then, the last train to arrive at station s should postpone its arrival to the time after the departure of the first train to leave the station in the same direction."

These constraints have to be written for all combinations of i eastbound and $n + 1$- i westbound trains and for every station having n parallel tracks. Please note that, indeed there are two nested if's within the statement. The less obvious one is related to "the first train in the same direction", which can be deduced from the values of the following and overtaking variables. Then, i eastbound and $n + 1 - i$ westbound trains and for every combination of first entering and first exiting trains.

3.1 Formal Mathematical Model for Scheduling Train Movements on a Single Track Railway Line

Before presenting the model, it is necessary to state some definitions and assumptions.

In practice, some trains may have different itineraries. For example, when the system under consideration consists of 20 stations, some trains may traverse all the system. Some other may start at (for example) Station 5 within the system and terminate at (for example) Station 12 within the system. Some other trains may start outside the system and terminate within the system, or vice versa. In this study, it is assumed that all trains traverse all stations of the system. This is just to simplify the model's on- paper representation and the model can easily be modified to consider different itineraries. It is also assumed that, trains in two opposite directions are categorized as “eastbound trains” and “westbound trains”. In the indexing scheme, stations are numbered from one to the total number of stations (cardinality of the set of stations) and numbers increase in eastbound direction. The western terminus (s_F) and eastern terminus (s_L) are assumed to have an infinite number of secondary tracks. **In the notations, if a symbol denotes a vector, it is written in bold.** The model is coded in AIMMS, a commercial software for coding optimization problems and feeding them into the solvers.

S: Set of stations

S_1: Set of stations with n = 2 parallel tracks (one mainline track and one secondary track)

S_2: Set of stations with n = 3 parallel tracks (one mainline track and two secondary tracks)

E: Set of eastbound trains

W: Set of westbound trains

e,f,g,h: Eastbound train indices

w,x,v,z: Westbound train indices

s, t: Station indices

s_F: The first station (western terminus) of the modeled railway line.

s_L: The last station of the (eastern terminus) modeled railway line.

δ_e :Entrance time of eastbound train e into the system. This is also its minimum possible departure time from the first station within the system. It is named as “ready time” in AIMMS model.

$\hat{\delta}_w$:Entrance time of westbound train w into the system. This is also its minimum possible departure time from the first station within the system. It is named as “ready time” in AIMMS model.

$\mathbf{T_e}$: Vector of minimum travel times of eastbound train e between consecutive stations.

$\mathbf{T_e}$: (T_e^{sF}, T_e^{2}, …, T_e^{sL-1}), such as T_e^{s} = Minimum travel time of train e from station s to station s + 1. $\forall$ e $\in$ E , $\forall$ s $\in$ S \ {s_L}

$\mathbf{T_w}$: Vector of minimum travel times of westbound train w between consecutive stations.

$\mathbf{T_w}$: $(T_w^{sL}, T_w^{sL-1}, \ldots, T_w^{sF+1})$, such as T_w^s = Minimum travel time of westbound train w from station s +1 to station s. $\forall$ e $\in$ E , $\forall$ s $\in$ S \ {s_F}

σ_e^s: Minimum dwell time of eastbound train e at station s as required by timetable (equal to 0 if train e is not required to stop at station s by the timetable or there is no timetable at all). $\forall$ e $\in$ E , $\forall$ s $\in$ S \ {s_F, s_L}

$\hat{\sigma}_w^s$: Minimum dwell time of westbound train w at station s as required by timetable (equal to 0 if train w is not required to stop at station s by the timetable or there is no timetable). $\forall$ w $\in$ W , $\forall$ s $\in$ S \ {s_F, s_L}

$\eta_{e,f}^s$: Minimum arrive-arrive headway when eastbound train e arrives at station s after eastbound train f. $\forall$ e, f $\in$ E e < f , $\forall$ s $\in$ S \ {s_F}

$\hat{\eta}_{w,x}^s$: Minimum arrive-arrive headway when westbound train w arrives at station s after eastbound train x$\forall$ w, x $\in$ W , w < x,$\forall$ s $\in$ S \ {s_L}

$\beta_{e,f}^s$: Minimum depart-depart headway when eastbound train e departs from station s after eastbound train f. $\forall$ e, f $\in$ E , e < f, $\forall$ s $\in$ S \ {s_L}

$\hat{\beta}_{w,x}^s$: Minimum depart-depart headway when westbound train w departs from station s after eastbound train x. $\forall$ w, x $\in$ W , w < x,$\forall$ s $\in$ S \ {s_F}

$\rho_{e,w}^s$: Minimum arrive-depart headway when eastbound train e departs from station s after arrival of westbound train w. $\forall$ e $\in$ E, w $\in$ W , $\forall$ s $\in$ S \ {s_L}

$\hat{\rho}_{e,w}^s$: Minimum arrive-depart headway when westbound train j departs from station s after arrival of eastbound train e. $\forall$ e $\in$ E, w $\in$ W , $\forall$ s $\in$ S \ {s_F}

In the AIMMS model, all depart-depart headways, arrive-arrive headways and arrive-depart headways are assumed to be equal among themselves.

M: A sufficiently large positive number.

α_e : Scheduled arrival time of eastbound train e at station $s_L$as required by the timetable. If there is no timetable, minimum possible arrival time may be written here.

$\hat{\alpha}_w$: Scheduled arrival time of westbound train w at station s_F as required by the timetable. If there is no timetable, minimum possible arrival time may be written here.

F: A monotonically increasing linear function of its operand. In AIMMS model, the total weighted delay (tardiness) is used. Final delay of every train is multiplied by a pre-assigned weight and all the weighted delays are summed up.

Up to here, all of the presented quantities were constants and parameters. Now, it is time to present the decision variables.

Continuous variables:

a_e^s: Arrival time of eastbound train e at station s. Not defined for s = s_F, where train e enters the system. Arrival time of train e into s_F is a parameter, not a decision variable.

d_e^s: Departure time of eastbound train e from station s. Not defined for station s = $s_{L,}$ the final station of train e within the system. As soon as train e arrives at the final station, we are done with it; we do not care what it does afterwards.

$\hat{a}_w^s$: Arrival time of westbound train w at station s. Not defined for station s = s_L, where train w enters the system. Arrival time of train w into station s_L is a parameter, not a decision variable.

$\hat{d}_w^s$: Departure time of westbound train w from station s. Not defined for station s_F, the final station of train w within the system. As soon as train w reaches this final station, we are done with it; we don't care what it does afterwards.

τ_e : Delay (tardiness) of eastbound train e in the final station of its itinerary within the system.

$\hat{\tau}_w$: Delay (tardiness) of westbound train w in the final station of its itinerary within the system.

Binary variables:

$$b_{e,f}^s = \begin{cases} 1, & \text{if eastbound train e departs from station s before eastbound train f,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that this variable is defined for s ≠ $s_L$and e < f

$$\hat{b}^{s}_{w,x} = \begin{cases} 1, & \text{if westbound train w departs from station s before westbound train x} \\ 0, & \text{otherwise.} \end{cases}$$

Note that this variable is defined for $s \neq s_F$. and $w < x$

Meeting variables defined for stations with one secondary line ($s \in S_1$):

$$\overline{c^{s}_{e,w}} = \begin{cases} 1, & \text{if eastbound train e meets with westbound train w at station s,} \\ 0, & \text{otherwise.} \end{cases}$$

Meeting variables defined for stations with more than one secondary line ($s \in S_2+\{s_F\}$):

$$c^{s}_{e,w} = \begin{cases} 1, & \text{if eastbound train e and westbound train w meet at station s AND train e enters the station before train w,} \\ 0, & \text{otherwise.} \end{cases}$$

($s \in S_2 +\{s_L\}$):

$$\hat{c}^{s}_{e,w} = \begin{cases} 1, & \text{if eastbound train e and westbound train w meet at station s AND train w enters the station before train e,} \\ 0, & \text{otherwise.} \end{cases}$$

Objective Function:

$$\text{Min} \sum_{e \in E} F(\tau_e) + \sum_{w \in W} F(\hat{\tau}_w) \tag{3.1}$$

Objective function (3.1) is a linear combination of delays of each (eastbound and westbound) train.

Minimum Departure and Arrival Time Constraints:

$$d^{s_F}_{e} \geq \delta_e \quad \forall \text{ e} \in \text{E} \tag{3.2}$$

$$\hat{d}_w^{s_L} \geq \hat{\delta}_w \;\; \forall \text{ w} \in \text{W} \tag{3.3}$$

These constraints prevent any train from departing from the first station of its itinerary within the system before it actually enters the system (i.e. arrives at the first station).

Delay Definition Constraints:

$$\tau_e \geq a_e^{s_L} - \alpha_e, \quad \forall \text{ e} \in \text{E} \tag{3.4}$$

$$\hat{\tau}_w \geq \hat{a}_w^{s_F} - \hat{\alpha}_w, \quad \forall \text{ w} \in \text{W} \tag{3.5}$$

These constraints define the delay. Since the variables representing delay are nonnegative, ifany train arrives earlier than its scheduled arrival time, delay takes the value 0. By this definition, we are allowing trains to arrive earlier than their scheduled arrival times, but an early arriving train does not contribute to the objective function. Typically, model will send the trains early if it helps to reduce the delays of other trains.

Minimum Running Time Constraints:

$$a_e^{s+1} \geq d_e^s + T_e^s \;\; \forall \text{ e} \in \text{E, s} \in \text{S} \backslash \text{s}_\text{L} \tag{3.6}$$

$$\hat{a}_w^s \geq \hat{d}_w^{s+1} + \hat{T}_w^s \;\; \forall \text{ w} \in \text{W, s} \in \text{S} \backslash \text{s}_\text{L} \tag{3.7}$$

These constraints enforce the minimum running times between the adjacent stations.

Minimum Dwell Time Constraints:

$$d_e^s \geq a_e^s + \sigma_e^s \;\; \forall \text{ e} \in \text{E, s} \in \text{S} \backslash \{\text{s}_\text{F},\text{s}_\text{L}\} \tag{3.8}$$

$$\hat{d}_w^s \geq \hat{a}_w^s + \hat{\sigma}_w^s \;\; \forall \text{ w} \in \text{W, s} \in \text{S} \backslash \{\text{s}_\text{F},\text{s}_\text{L}\} \tag{3.9}$$

These constraints enforce the minimum dwell times of the trains as required by the timetable. If no dwell is required at a particular station or there is no timetable at all, these constraints are still needed. This time, they enforce the train continuities. Any train cannot depart from any station before arriving at it. Departure times are not defined for the last stations of the respective trains' itineraries, nor dwell time constraints.

Following and Overtaking Constraints for Eastbound Trains:

$$d_e^s \geq d_f^s + \beta_{e,f}^s - M * b_{e,f}^s \;\; \forall \text{ e, f} \in \text{E, e} < \text{f, s} \in \text{S} \backslash \text{s}_\text{L} \tag{3.10}$$

$$d_f^s \geq d_e^s + \beta_{f,e}^s - M * (1 - b_{e,f}^s) \ \forall \text{ e, f} \in \text{E, e < f, s} \in \text{S}\backslash\text{s}_\text{L} \tag{3.11}$$

$$a_e^{s+1} \geq a_f^{s+1} + \eta_{e,f}^{s+1} - M * b_{e,f}^s \ \forall \text{ e, f} \in \text{E, e < f s} \in \text{S}\backslash\text{s}_\text{L} \tag{3.12}$$

$$a_f^{s+1} \geq a_e^{s+1} + \eta_{f,e}^{s+1} - M * (1 - b_{e,f}^s) \ \forall \text{ e, f} \in \text{E, e < f s} \in \text{S}\backslash\text{s}_\text{L} \tag{3.13}$$

Constraints 3.10 and 3.12 are active if eastbound train f enters the mainline track section between stations s and s + 1 before eastbound train e. Constraint 3.10 ensures that there is sufficient headway between two trains in departure. Constraint 3.12 ensures that train e does not overtake train f in the mainline track section and also there is sufficient headway upon arrival to the next station. Constraints 3.11 and 3.13 are active if eastbound train e enters the mainline track section between stations s and s + 1 before eastbound train f. Constraint 3.11 ensures that there is sufficient headway between two trains in departure. Constraint 3.13 ensures that train f does not overtake train e in the mainline track section and also there is sufficient headway upon arrival to the next station.

Following and Overtaking Constraints for Westbound Trains:

$$\hat{d}_w^s \geq \hat{d}_x^s + \hat{\beta}_{w,x}^s - M * \hat{b}_{w,x}^s \ \forall \text{ w, x} \in \text{W, w < x, s} \in \text{S}\backslash\text{s}_\text{F} \tag{3.14}$$

$$\hat{d}_x^s \geq \hat{d}_w^s + \hat{\beta}_{x,w}^s - M * (1 - \hat{b}_{w,x}^s) \ \forall \text{ w, x} \in \text{W, w < x, s} \in \text{S}\backslash\text{s}_\text{F} \tag{3.15}$$

$$\hat{a}_w^s \geq \hat{a}_x^s + \hat{\eta}_{w,x}^s - M * \hat{b}_{w,x}^{s+1} \ \forall \text{ w, x} \in \text{W, w < x, s} \in \text{S}\backslash\text{s}_\text{F} \tag{3.16}$$

$$\hat{a}_x^s \geq \hat{a}_w^s + \hat{\eta}_{x,w}^s - M * (1 - \hat{b}_{w,x}^{s+1}) \ \forall \text{ w, x} \in \text{W, w < x, s} \in \text{S}\backslash\text{s}_\text{F} \tag{3.17}$$

Constraints 3.14 and 3.16 are active if westbound train x enters the mainline track section between stations s +1 and s before westbound train w. Constraint 3.14 ensures that there is sufficient headway between two trains in departure. Constraint 3.16 ensures that train w does not overtake train x in the mainline track section and also there is sufficient headway upon arrival to the next station. Constraints 3.15 and 3.17 are active if westbound train w enters the mainline track section between stations s + 1 and s before westbound train x. Constraint 3.15 ensures that there is sufficient headway between two trains in departure. Constraint 3.17 ensures that train x does not overtake train w in the mainline track section and also there is sufficient headway upon arrival to the next station.

Preventing a slow train to overtake a fast train:

$$b_{e,f}^{s} = 1 \ \forall \text{ e, f} \in \text{E, s} \in \text{S}\backslash \text{s}_\text{L} \mid \text{e} < \text{f} \ \& \ \delta_e \leq \delta_f \tag{3.18}$$

$$\hat{b}_{w,x}^{s} = 1 \ \forall \text{ w, x} \in \text{W, s} \in \text{S}\backslash \text{s}_\text{F} \mid \text{w} < \text{x} \ \& \ \hat{\delta}_w \leq \hat{\delta}_x \tag{3.19}$$

In railway operations practice, fast trains have priority over slow trains and thus a slow train is not allowed to overtake a faster train. This fact is utilized to reduce the size of the solution space. In the model, it is assumed that trains are indexed according to their speeds (priorities), faster trains having a smaller index. This constraint tells that, if a faster train enters the system before a slower train, than it will always precede the slower train.

Preventing alternate overtakings:

$$b_{e,f}^{s+1} \geq b_{e,f}^{s} \ \forall \text{ e, f} \in \text{E, s} \in \text{S}\backslash \text{s}_\text{L} \mid \text{e} < \text{f} \ \& \ \delta_e > \delta_f \tag{3.20}$$

$$\hat{b}_{w,x}^{s} \geq \hat{b}_{w,x}^{s+1} \ \forall \text{ w, x} \in \text{W, s} \in \text{S}\backslash \text{s}_\text{F} \mid \text{w} < \text{x} \ \& \ \hat{\delta}_w > \hat{\delta}_x \tag{3.21}$$

Same principle as preventing a slow train to overtake a fast train: This time, if a fast train overtakes a slower train, than it will always precede the slower train afterwards.

Forcing Meeting Constraints:

$$\sum_{s \in s_1} \overline{c_{e,w}^{s}} + \sum_{s \in S_2 + \{s_F\}} c_{e,w}^{s} + \sum_{s \in S_2 + \{s_L\}} \hat{c}_{e,w}^{s} = 1 \ \forall \text{ e} \in \text{E, w} \in \text{W} \tag{3.22}$$

These constraints ensure that, each train couple will meet once and only once.

Meeting in s_F:

$$d_e^{s_F} \geq \hat{a}_w^{s_F} + \rho_{e,w}^{s_F} - M*(1 - c_{e,w}^{s_F}) \ \forall \text{ e} \in \text{E, w} \in \text{W} \tag{3.23}$$

To understand what this constraint does, the reader should remember that, this model considers trains that have already arrived at their respective origin in the system (for eastbound trains, that is s_F) and trains that have not yet arrived at their respective destination (for westbound trains, that is s_F). Before arriving at the origin (entering the system) or after arriving at the destination (leaving the system), the trains simply do not exist for the model. So, the only way of a meeting between of eastbound train e and westbound train w at Station s_F that can be recognized by the model, is the specific case where eastbound train e arrives at s_F (i.e. enters the system) before westbound train w at

s_F (i.e. leaves the system). In that case, eastbound train e has to wait until the westbound train w arrives at station s_F. This constraint is active if such a meeting occurs at s_F.

Meeting in s_L:

$$\hat{d}_w^{|S|} \geq a_e^{s_L} + \hat{\rho}_{e,w}^{s_L} - M*(1-\hat{c}_{e,w}^{s_L}) \ \forall \text{ e} \in \text{E, w} \in \text{W} \tag{3.24}$$

This constraint is active if eastbound train e and westbound train w meet at station s_L. The explanation is similar to that of constraint 3.23.

Meeting constraints for intermediate stations with one secondary line- 1:

$$d_e^s \geq \hat{a}_w^s + \rho_{e,w}^s - M*(1-\overline{c_{e,w}^s}) \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_1 \tag{3.25}$$

These constraints are active if eastbound train e and westbound train w meet at intermediate station s, which has one siding. It forces train e to depart at least arrive-depart headway later than the arrival time of train w.

Meeting constraints for intermediate stations with one secondary line- 2:

$$\hat{d}_w^s \geq a_e^s + \hat{\rho}_{e,w}^s - M*(1-\overline{c_{e,w}^s}) \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_1 \tag{3.26}$$

These constraints are active if eastbound train e and westbound train w meet at intermediate station s, which has one siding. It forces train w to depart at least arrive-depart headway later than the arrival time of train e.

Meeting constraints for intermediate stations with two secondary lines- 1:

$$d_e^s \geq \hat{a}_w^s + \rho_{e,w}^s - M*[1-(c_{e,w}^s + \hat{c}_{e,w}^s)] \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_2 \tag{3.27}$$

These constraints are active if eastbound train e and westbound train w meet at intermediate station s, which has one siding. It forces train e to depart at least arrive-depart headway later than the arrival time of train w. Note that, constraints 3.22 enforce that at most one of variables in constraint 3.27 can take the value one.

Meeting constraints for intermediate stations with two secondary lines- 2:

$$\hat{d}_w^s \geq a_e^s + \hat{\rho}_{e,w}^s - M*[1-(c_{e,w}^s + \hat{c}_{e,w}^s)] \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_2 \tag{3.28}$$

These constraints are active if eastbound train e and westbound train w meet at intermediate station s, which has one secondary line. It forces train w to depart at least arrive-depart headway later than the arrival time of train e. Note that, constraints 3.22 enforce that at most one of variables in constraint 3.28 can take the value one.

Meeting constraints for intermediate stations with two secondary lines- 3:

$$\hat{a}_w^s \geq a_e^s + 1 - M * (1 - c_{e,w}^s) \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_2 \tag{3.29}$$

Meeting constraints for intermediate stations with two secondary lines- 4:

$$a_e^s \geq \hat{a}_w^s - M * (1 - \hat{c}_{e,w}^s) \ \forall \text{ e} \in \text{E, w} \in \text{W, s} \in \text{S}_2 \tag{3.30}$$

Constraints 3.29 and 3.30 enforce the definition of binary meeting variables for the stations with two secondary lines. 3.29 is active when $c_{e,w}^s = 1$, which means eastbound train e arrives at the meeting station before westbound train w. 3.30 is active when $\hat{c}_{e,w}^s = 1$, which means westbound train w arrives at the meeting station before eastbound train e. The adopted convention is that, if both trains arrive at the meeting station exactly the same time, the case is treated as westbound train arrives first. This is the explanation of + 1 present in constraint 3.29. Now, the station/siding capacity constraints will be given. These constraints are based on the enumeration of the possible situations such that the station / siding capacity is exceeded. At the end of the chapter, a general example is given to understand the mechanism of enumeration.

Capacity constraints for stations with one secondary line-1:

$$\sum_{f \in E | e < f} (b_{e,f}^s - b_{e,f}^{s-1}) \leq 1 \ \forall \text{ e} \in \text{E, s} \in \text{S}_1 \tag{3.31}$$

These constraints prevent any eastbound train to overtake more than one eastbound train at the same station.

Capacity constraints for stations with one secondary line-2:

$$\sum_{w,x \in W | w < x} \left(\hat{b}_{w,x}^s - \hat{b}_{w,x}^{s+1}\right) \leq 1 \ \forall \text{ w} \in \text{W, s} \in \text{S}_1 \tag{3.32}$$

These constraints prevent any westbound train to overtake more than one westbound train at the same station with one secondary line.

Capacity constraints for stations with one secondary line-3:

$$\left(b_{e,f}^{s} - b_{e,f}^{s+1}\right) + \overline{c_{e,w}^{s}} \leq 1 \;\forall\; e,f \in E \mid e < f,\; w \in W,\; s \in S_1 \tag{3.33}$$

These constraints prevent any eastbound train to overtake an eastbound train and meet with a westbound train at the same station with one secondary line.

Capacity constraints for stations with one secondary line-4:

$$\left(\hat{b}_{w,x}^{s} - \hat{b}_{w,x}^{s+1}\right) + \overline{c_{e,w}^{s}} \leq 1 \;\forall\; e \in E,\; w, x \in W \mid w < x,\; s \in S_1 \tag{3.34}$$

The same case with constraint 3.31, this time with 2 westbound and 1 eastbound trains.

Capacity constraints for stations with one secondary line-5:

$$a_{f}^{s} \geq d_{e}^{s} + \eta_{e,f}^{s} - M * (1 - b_{e,f}^{s-1}) \;\forall\; e,f \in E \mid e < f,\; s \in S_1 \tag{3.35}$$

This constraint is active when eastbound train f (slower than e) enters station s (with one secondary line) after eastbound train e and dictates that train f can enter the station only after train e clears the station.

Capacity constraints for stations with one secondary line-6:

$$\hat{a}_{x}^{s} \geq \hat{d}_{w}^{s} + \hat{\eta}_{e,f}^{s} - M * (1 - \hat{b}_{w,x}^{s+1}) \;\forall\; w,x \in W \mid w < x,\; s \in S_1 \tag{3.36}$$

Same as constraint 3.35 for two westbound trains.

Capacity constraints for stations with one secondary line-7:

$$a_{e}^{s} \geq d_{f}^{s} + \eta_{e,f}^{s} - M * (b_{e,f}^{s} + b_{e,f}^{s-1}) \;\forall\; e,f \in E \mid e < f,\; s \in S_1 \tag{3.37}$$

This constraint is active if eastbound train e enters station s (with one secondary line after (slower) eastbound train f and will not overtake eastbound train f at station s and it dictates that eastbound train e can enter the station only after eastbound train f clears the station.

Capacity constraints for stations with one secondary line-8:

$$\hat{a}_{w}^{s} \geq \hat{d}_{x}^{s} + \hat{\eta}_{w,x}^{s} - M * (\hat{b}_{w,x}^{s} + \hat{b}_{w,x}^{s+1}) \;\forall\; w,x \in W \mid w < x,\; s \in S_1 \tag{3.38}$$

Same as constraint 3.37 for two westbound trains.

Capacity constraints for stations with two secondary lines-1:

$$\sum_{f \in E | e < f} (b_{e,f}^{s} - b_{e,f}^{s-1}) \leq 2 \quad \forall \text{ e} \in \text{E, s} \in \text{S}_2 \tag{3.39}$$

Capacity constraints for stations with two secondary lines-2:

$$\sum_{w,x \in W | w < x} \left(\hat{b}_{w,x}^{s} - \hat{b}_{w,x}^{s+1} \right) \leq 2 \quad \forall \text{ w} \in \text{W, s} \in \text{S}_2 \tag{3.40}$$

Constraints (3.39) and (3.40) prevent any train to overtake more than two trains at the same station with two secondary lines.

Capacity constraints for stations with two secondary lines-3:

$$\hat{c}_{e,w}^{s} + \hat{c}_{e,x}^{s} + \hat{c}_{e,y}^{s} \leq 2 \quad \forall \text{ e} \in \text{E, w,x,y} \in \text{W | w < x < y} \tag{3.41}$$

These constraints prevent three westbound trains to "wait" one eastbound train at the same station. These constraints are not absolutely required, as the general rule for n > 2 and i = 1 already prevents this. However, these constraints are added to reduce the solution space.

Capacity constraints for stations with two secondary lines-4:

$$c_{e,w}^{s} + c_{f,w}^{s} + c_{g,w}^{s} \leq 2, \quad \forall \text{ w} \in \text{W, e,f,g} \in \text{E | e < f < g} \tag{3.42}$$

Same as constraint 3.39, this time for three eastbound trains and one westbound train.

Capacity constraints for stations with two secondary lines 5-13:

$$a_{g}^{s} \geq d_{e}^{s} + \eta_{e,g}^{s} - M*[1-(b_{e,f}^{s} + b_{e,g}^{s} + b_{f,g}^{s})/3] \; \forall \text{ e,f,g} \in \text{E | e < f < g} \tag{3.43}$$

$$a_{f}^{s} \geq d_{e}^{s} + \eta_{e,f}^{s} - M*[1-(b_{e,f}^{s-1} + b_{e,g}^{s-1} - b_{f,g}^{s-1} + b_{e,f}^{s} + b_{e,g}^{s})/4] \; \forall \text{ e,f,g} \in \text{E | e < f < g} \tag{3.44}$$

$$a_{f}^{s} \geq d_{g}^{s} + \eta_{f,g}^{s} - M*[1-(b_{e,f}^{s-1} - b_{e,g}^{s-1} - b_{f,g}^{s-1} + b_{e,f}^{s} - b_{f,g}^{s})/2] \; \forall \text{ e,f,g} \in \text{E| e < f < g} \tag{3.45}$$

$$a_{f}^{s} \geq d_{e}^{s} + \eta_{e,f}^{s} - M*[1-(b_{e,f}^{s-1} - b_{e,g}^{s-1} - b_{f,g}^{s-1} + b_{e,f}^{s} + b_{e,g}^{s})/3] \; \forall \text{ e,f,g} \in \text{E | e < f < g} \tag{3.46}$$

$$a_{g}^{s} \geq d_{e}^{s} + \eta_{e,g}^{s} - M*[1-(b_{e,g}^{s-1} - b_{e,f}^{s-1} + b_{f,g}^{s-1} + b_{e,f}^{s} + b_{e,g}^{s} + b_{f,g}^{s})/5] \; \forall \text{ e,f,g |e<f<g} \tag{3.47}$$

$$a_{g}^{s} \geq d_{f}^{s} + \eta_{f,g}^{s} - M*[1-(b_{e,g}^{s-1} - b_{e,f}^{s-1} + b_{f,g}^{s-1} - b_{e,f}^{s} + b_{e,g}^{s} + b_{f,g}^{s})/4] \; \forall \text{ e,f,g} \in \text{E | e< f < g} \tag{3.48}$$

$$a_{e}^{s} \geq d_{f}^{s} + \eta_{e,f}^{s} - M*[1-(b_{f,g}^{s-1} - b_{e,g}^{s-1} - b_{e,f}^{s-1} + b_{f,g}^{s} - b_{e,f}^{s})/2] \; \forall \text{ e,f,g} \in \text{E | e < f < g} \tag{3.49}$$

$$a_e^s \geq d_g^s + \eta_{e,g}^s - M*(b_{e,f}^{s-1} + b_{e,g}^{s-1} + b_{f,g}^{s-1} + b_{e,g}^s + b_{f,g}^s) \; \forall \; \text{e,f,g} \in \text{E} \; | \; \text{e} < \text{f} < \text{g} \qquad (3.50)$$

$$a_e^s \geq d_f^s + \eta_{e,f}^s - M*[1-(b_{f,g}^s - b_{e,f}^s - b_{f,g}^{s-1} - b_{e,g}^{s-1} - b_{e,f}^{s-1})] \; \forall \; \text{e,f,g} \in \text{E} \; | \; \text{e} < \text{f} < \text{g} \qquad (3.51)$$

These constraints are related to part 3 of Proposition 1. To implement this rule, it is needed to determine who enters the station first and who leaves the station first. Values of the binary variables tell that. Any constraint will be active only when the summation within the bracket divided by the respective integer number gives either zero or one, depending on the case. Then, the constraint enforces that the last train to enter the station would depart later than the first departing train.

Capacity constraints for stations with two secondary lines 14-22:

$$\hat{a}_y^s \geq \hat{d}_w^s + \hat{\eta}_{w,y}^s - M*[1-(\hat{b}_{w,x}^s + \hat{b}_{w,y}^s + \hat{b}_{x,y}^s)/3] \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.52)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{b}_{w,x}^{s+1} + \hat{b}_{w,y}^{s+1} - \hat{b}_{x,y}^{s-1} + \hat{b}_{w,x}^s + \hat{b}_{w,y}^s)/4] \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.53)$$

$$\hat{a}_x^s \geq \hat{d}_y^s + \hat{\eta}_{x,y}^s - M*[1-(\hat{b}_{w,x}^{s+1} - \hat{b}_{w,y}^{s+1} - \hat{b}_{x,y}^{s+1} + \hat{b}_{w,x}^s - \hat{b}_{x,y}^s)/2] \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.54)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{b}_{w,x}^{s+1} - \hat{b}_{w,y}^{s+1} - \hat{b}_{x,y}^{s+1} + \hat{b}_{w,x}^s + \hat{b}_{w,y}^s)/3] \quad \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.55)$$

$$\hat{a}_y^s \geq \hat{d}_w^s + \hat{\eta}_{w,y}^s - M*[1-(\hat{b}_{w,y}^{s+1} - \hat{b}_{w,x}^{s+1} + \hat{b}_{x,y}^{s+1} + \hat{b}_{w,x}^s + \hat{b}_{w,y}^s + \hat{b}_{x,y}^s)/5] \; \forall \; \text{w,x,y} \in \text{W} | \text{w} < \text{x} < \text{y} \qquad (3.56)$$

$$\hat{a}_y^s \geq \hat{d}_x^s + \hat{\eta}_{x,y}^s - M*[1-(\hat{b}_{w,y}^{s+1} - \hat{b}_{w,x}^{s+1} + \hat{b}_{x,y}^{s+1} - \hat{b}_{w,x}^s + \hat{b}_{w,y}^s + \hat{b}_{x,y}^s)/4] \; \forall \; \text{w,x,y} \in \text{W} | \; \text{w} < \text{x} < \text{y} \qquad (3.57)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{b}_{x,y}^{s+1} - \hat{b}_{w,y}^{s+1} - \hat{b}_{w,x}^{s+1} + \hat{b}_{x,y}^s - \hat{b}_{w,x}^s)/2] \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.58)$$

$$\hat{a}_w^s \geq \hat{d}_y^s + \hat{\eta}_{w,y}^s - M*(\hat{b}_{w,x}^{s+1} + \hat{b}_{w,y}^{s+1} + \hat{b}_{x,y}^{s+1} + \hat{b}_{w,y}^s + \hat{b}_{x,y}^s) \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.59)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{b}_{x,y}^s - \hat{b}_{w,x}^s - \hat{b}_{x,y}^{s+1} - \hat{b}_{w,y}^{s+1} - \hat{b}_{w,x}^{s+1})] \; \forall \; \text{w,x,y} \in \text{W} \; | \; \text{w} < \text{x} < \text{y} \qquad (3.60)$$

These are exactly the same as constraints (3.43)-(3.51), except being for the symmetrical case.

Capacity constraints for stations with two secondary lines 23-46:

$$c_{e,w}^s + c_{e,x}^s + c_{f,w}^s + c_{f,x}^s + (b_{w,x}^s - b_{w,x}^{s+1}) \leq 4, \; \forall \; \text{e,f} \in \text{E, w,x} \in \text{W} \; | \; \text{e} < \text{f} \; \& \; \text{w} < \text{x} \qquad (3.61)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + c_{e,x}^s + c_{f,w}^s + c_{f,x}^s - \hat{b}_{w,x}^{s+1})/4] \; \forall \; \text{e,f} \in \text{E, w,x} \in \text{W} | \text{e} < \text{f} \& \text{w} < \text{x} \qquad (3.62)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + c_{e,x}^s + c_{f,w}^s + c_{f,x}^s + \hat{b}_{w,x}^{s+1})/5] \ \forall \text{ e,f}\in\text{E, w,x}\in\text{W}|\text{e<f\&w<x} \quad (3.63)$$

$$(c_{e,w}^s + c_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s - \hat{b}_{w,x}^{s+1}) \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.64)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + c_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e < f \& w<x} \quad (3.65)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + c_{e,x}^s + \hat{c}_{f,w}^s + c_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.66)$$

$$a_f^s \geq d_e^s + \eta_{e,f}^s - M*[1-(c_{e,w}^s + c_{e,x}^s + \hat{c}_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.67)$$

$$c_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s - \hat{b}_{w,x}^{s+1} \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.68)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + c_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e < f \& w<x} \quad (3.69)$$

$$c_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s - (\hat{b}_{w,x}^s - \hat{b}_{w,x}^{s+1}) \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.70)$$

$$\hat{a}_w^s \geq \hat{d}_x^s + \hat{\eta}_{w,x}^s - M*[1-(c_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s - \hat{b}_{w,x}^{s+1})/4] \ \forall \text{ e,f}\in\text{E,w,x}\in\text{W}|\text{e<f\&w<x} \quad (3.71)$$

$$a_f^s \geq d_e^s + \eta_{e,f}^s - M*[1-(c_{e,w}^s + \hat{c}_{e,x}^s + \hat{c}_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.72)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{c}_{e,w}^s + c_{e,x}^s + c_{f,w}^s + c_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.73)$$

$$\hat{a}_x^s \geq \hat{d}_w^s + \hat{\eta}_{w,x}^s - M*[1-(\hat{c}_{e,w}^s + c_{e,x}^s + \hat{c}_{f,w}^s + c_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.74)$$

$$a_f^s \geq d_e^s + \eta_{e,f}^s - M*[1-(\hat{c}_{e,w}^s + c_{e,x}^s + \hat{c}_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in\text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.75)$$

$$\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + c_{f,x}^s + (b_{e,f}^s - b_{e,f}^{s-1}) \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.76)$$

$$a_e^s \geq d_f^s + \eta_{e,f}^s - M*[1-(\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + \hat{c}_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.77)$$

$$\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s + (b_{e,f}^s - b_{e,f}^{s-1}) \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.78)$$

$$a_e^s \geq d_f^s + \eta_{e,f}^s - M*[1-(\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + c_{f,w}^s + \hat{c}_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.79)$$

$$\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + \hat{c}_{f,w}^s + c_{f,x}^s + (b_{e,f}^s - b_{e,f}^{s-1}) \leq 4 \ \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e < f \& w < x} \quad (3.80)$$

$$a_e^s \geq d_f^s + \eta_{e,f}^s - M*[1-(\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + \hat{c}_{f,w}^s + c_{f,x}^s)/4] \ \forall \text{ e,f} \in \text{E, w,x}\in\text{W} \mid \text{e<f \& w < x} \quad (3.81)$$

$$a_f^s \geq d_e^s + \eta_{e,f}^s - M*[1-(\hat{c}_{e,w}^s + \hat{c}_{e,x}^s + \hat{c}_{f,w}^s + \hat{c}_{f,x}^s + b_{e,f}^{s-1})/5] \ \forall \text{ e,f}\in\text{E,w,x}\in\text{W}|\text{e<f\&w<x} \quad (3.82)$$

$$\hat{c}^{s}_{e,w} + \hat{c}^{s}_{e,x} + \hat{c}^{s}_{f,w} + \hat{c}^{s}_{f,x} + (b^{s}_{e,f} - b^{s-1}_{e,f}) \leq 4 \; \forall \text{ e,f} \in \text{E, w,x} \in \text{W} \mid \text{e} < \text{f \& w} < \text{x} \qquad (3.83)$$

$$a^{s}_{e} \geq d^{s}_{f} + \eta^{s}_{e,f} - M * [1 - (\hat{c}^{s}_{e,w} + \hat{c}^{s}_{e,x} + \hat{c}^{s}_{f,w} + \hat{c}^{s}_{f,x} - b^{s-1}_{e,f}) / 4] \text{ e,f} \in \text{E, w,x} \in \text{W} | \text{e} < \text{f \& w x} \qquad (3.84)$$

These constraints constitute the heaviest part of the model. They are generally the enforcement of the following rule: "If *i* westbound trains and n + 1 – i eastbound trains will meet each other at station s, which has n tracks, such that $2 \leq i \leq$ n-1 and n ≥ 3, then, the last train to arrive at station s should postpone its arrival to after the departure of the first train in the same direction." So, we have to determine who enters last and who is the departing train in its direction first. It may be useful to explain how these constraints work by an example. For constraint 3.84 to be active, $\hat{c}^{s}_{e,w} + \hat{c}^{s}_{e,x} + \hat{c}^{s}_{f,w} + \hat{c}^{s}_{f,x} - b^{s-1}_{e,f} = 4$ should hold. This is possible if and only if $\hat{c}^{s}_{e,w} = \hat{c}^{s}_{e,x} = \hat{c}^{s}_{f,w} = \hat{c}^{s}_{f,x} = 1$ and $b^{s-1}_{e,f} = 0$ hold. The last equality tells that, eastbound train e arrives at station s after eastbound train f. The first one tells that westbound train w arrives at station s before eastbound train e. The second one tells that westbound train x arrives at station s before eastbound train e. The third one tells that westbound train w arrives at station s before eastbound train f. The fourth one tells that westbound train x arrives at station s before eastbound train f. All these add up to the fact that, train e is the last train among those for to arrive at the station. Therefore, it cannot enter the station unless eastbound train f clears one track. Of course, it is not possible for this last entering train to overtake the other train, so there are some constraints (such as 3.80) to prevent this. Also, some constraints were written to make eliminations of combinations of binary variables which leads to the inconsistent cases (like e comes before w, w comes before x, x comes before e), mainly to reduce search space. In Figure 3.10, the situation represented by constraint 3.80 is depicted. Constraint 3.80 holds Train e outside the station until train f clears the station. This situation further affects the solution. Because, at this point, no westbound train can leave the station until Train e enters the station.

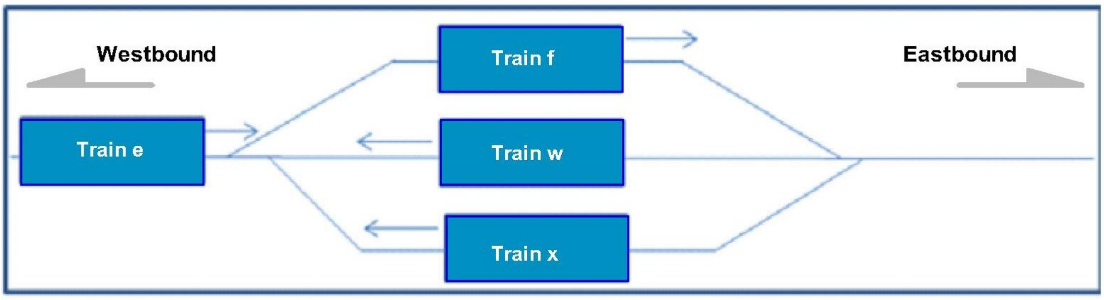

Figure 3. 10 Situation represented by constraint 3.80.

CHAPTER 4

SPEED-UP METHODS

Although the proposed mathematical model represents the train movements on a single track railway line quite accurately (except neglecting the time losses of a stopping train because of braking and acceleration), it has some serious drawbacks. It models the meetings at stations with two secondary lines with two binary variables per train couple per station. This causes a significant increase in the number of binary variables. Also, complete enumeration of all situations in which station / siding capacities are exceeded generates a "more than huge" number of constraints. For a system with$|S_1|$ intermediate stations with one secondary line,$|S_2|$ intermediate stations with two secondary lines, |E| eastbound trains and |W| westbound trains, the number of such constraints is given by the formula written below:

Number of capacity constraints for stations with two secondary lines =

$$\binom{|E|}{3} \times |W| \times 9 \times |S_2| + \binom{|W|}{3} \times |E| \times 9 \times |S_2| + \binom{|E|}{2} \times \binom{|W|}{2} \times 24 \times |S_2| + |E| \times |S_2| + |W| \times |S_2| \quad (4.1)$$

In the formula, the first term refers to constraints 3.43-3.51, the second term refers to constraints 3.52-3.60, the third term refers to constraints 3.61-3.84, the fourth term refers to constraints 3.41 and the fifth term refers to constraints 3.42. In this formula, the constraints written for reducing search space are not included. If |E| = |W| = 15and $|S_2|$ = 16, total number of such constraints is 6,199,680. This makes the problem impossible to tackle without any external interferences. However, very few of these constraints will actually be binding. Figure 4.1 depicts an example solution provided by the model. The model was coded in AIMMS and solved by CPLEX 12.6. AIMMS is a commercial programming language and interface, specifically designed for coding and solving optimization problems. It provides interfaces to several commercial solvers for several types of optimization problems. CPLEX, is a commercial solver for linear programming, integer programming, mixed integer programming and quadratic programing problems. CPLEX, along with Gurobi, is globally accepted as the state- of art solver for such problems. In the considered railway line, stations 4, 6, 8, 11 and 14 has one secondary line, whereas, the others have two secondary lines. The termini at both ends are assumed to have infinite capacity. As mentioned previously, at least three trains should be there to exceed the capacity of a station with one secondary line and at least four trains should be there to exceed the capacity of a station with two secondary lines. In Figure 4.1, it can be seen that, only three of the stations with one secondary line (4, 6, 8) were "under threat" of siding capacity violation, each from only one trio of trains.

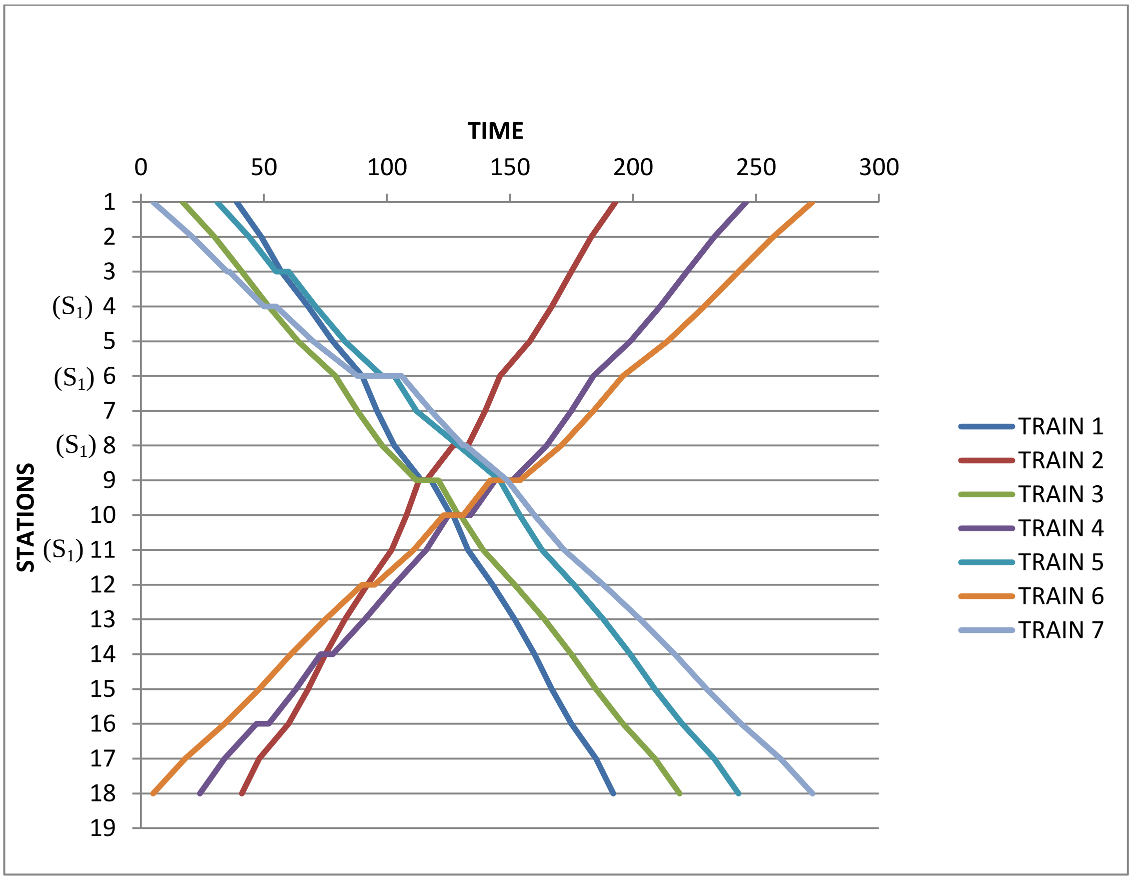

Figure 4.1Outcome of an example instance

Similarly, only two of the stations with two secondary lines (9, 10) were "under threat" of capacity violation, each from only one quartet of trains. This clearly indicates that, only a handful of these constraints were binding. The others were just there to make the model unnecessarily complicated and intractable. Indeed, there is one more fact that makes most of these constraints actually unnecessary: <u>Any quartet of trains, unless they</u>

are all travelling in the same direction, cannot exceed the capacity of more than one station. This is due to the nature of train traffic. Two trains in the opposite directions cannot meet more than once.

4.1 Lazy Constraint Attribute

AIMMS provides an opportunity to label some of the constraints as "lazy". If the solver recognizes lazy constraints, which CPLEX 12.6 does, it does not directly include them in the model. Instead, it solves a "relaxed" model, with these lazy constraints omitted. Each time it finds a feasible solution with respect to this "relaxed" model, it checks the solution against the lazy constraints. If it finds that one of the lazy constraints is violated, it creates a new "relaxed" model by adding the violated row into the constraint matrix and repeats the procedure. In the model, all of the station and siding capacity constraints except the ones involving only binary variables are defined as lazy constraints.

4.2 Solution Space Restriction Algorithm for Reducing the Number of Binary Variables and Constraints

As previously stated, the model has an unnecessarily large number of binary variables and constraints. The effect of this can be easily felt when the size of the problem instance gets larger. However, very few of the constraints for conflict resolution will actually be binding in the final solution. Moreover, most of the binary variables defined for meetings will simply be zero. This immediately bears the thought: If we can eliminate the constraints are very unlikely to be binding and the binary variables which are very unlikely to be basic, a significant reduction in the problem size and thus the computation time may be provided. Note that, such a progress is already done by defining the station / siding capacity constraints as "lazy constraints", and CPLEX eliminates the redundant constraints and binary variables in the presolve stage of the algorithm. However, it may still be possible to take this process one step further. Therefore, a heuristic solution space restriction algorithm has been developed to aid CPLEX in the solution procedure. It involves implementing a very basic greedy heuristic that does not take station / siding capacity constraints into account, to see how actually a solution may "look like." Then, the actual solution will be restricted to be "not very far away" from the outcome of this greedy heuristic. The greedy heuristic is quite simple: It is a construction heuristic, which starts from scratch, detects all the conflicts and resolves them one at a time. Finally, it arrives at a feasible solution when there is no conflict left. The heuristic tackles the constraints in a “time order”: It finds the first conflict with respect to time, resolves it, then finds the first conflict (among the remaining ones), resolves it, until there is no conflict left. Since it does not take the station / siding capacity constraints into account, there is no need for it to recognize the conflicts within a “group” of trains. It assumes that all constraints are between two trains only. Of course, if there are actually more than two trains in the same conflict, the heuristic senses all the conflicts between all the train couples as separate conflicts and resolves them one by one.

The decision logic of the heuristic is very simple and myopic; therefore it is a greedy heuristic. When it detects the conflict, firstly, it solves the conflict with delaying one of the trains. It then computes the total weighted unrecoverable delay of two trains in conflict, after the resolution, locally. Then, it reverses the solution, resolves the conflict by delaying the other train. After that, again, it computes the total weighted unrecoverable delay of two trains in conflict, after the resolution. The alternative that gives the lower total weighted unrecoverable delay (locally) will be the selected resolution for the current conflict. The heuristic will then update the departure and arrival times of the delayed train and find the next conflict. It computes the unrecoverable delay for one train according to the following formula:

UNRECOVERABLE DELAY = CURRENT TIME IN THE TRAIN’S CURRENT POSITION-THE EARLIEST TIME THAT THE TRAIN COULD EVER BE IN ITS CURRENT POSITION-RECOVERY TIME OF THAT TRAIN.

The way the heuristic works is outlined below:

STEP 1-initialization: For all trains, compute the departure times from and arrival times at all stations, as if all trains can move freely, without touching each other.

STEP 2: Find all the (meeting, headway, overtaking, etc.) conflicts in the system. If there is no conflict, terminate and report the final solution (All departure and arrival times for all trains plus the locations of meetings and overtakings). If there is at least one conflict, go to STEP 3.

STEP 3: Determine the exact occurrence time of the conflicts. This is done by treating the train paths (as seen on a train graph) as lines and making use of formulas to find intersection points of two lines in a two-dimensional coordinate axis.

STEP 4: Take the “earliest” conflict and resolve it.

STEP 5: Update the departure and arrival times of the train selected to be delayed in the conflict resolution, as if it can move freely, without touching any other train after the current conflict resolution and go to STEP 2.

This algorithm is guaranteed to give a feasible solution (for the relaxed problem where all the stations are assumed to have infinite capacity).

4.2.1 Restriction of the Solution Space for Meetings

Several trial runs were made and it was experienced that, for the meetings, there are never radical differences between the outcome of the heuristic and the actual optimal solution for the original problem (with station / siding capacities taken into account). For a significant number of cases, the meetings between any two trains occurred exactly at the same station in the heuristic and the optimal solutions. In some other cases, it differed by only one station, i.e. the meeting occurred in some station in the heuristic’s outcome and in one of the adjacent stations in the actual optimal solution. In none of the trials, it differed by two stations. Of course, this is not a scientific proof and it is indeed not possible to provide such a proof. This constitutes the rule of thumb for the solution space restriction. Regardless of the total number of stations, meetings between any tuple one eastbound- one westbound train were restricted to be happen in one of the five stations only: The station in which the meeting occurs in the heuristic and until two stations away in each direction. Note that, meeting constraints 3.22 to 3.30 are defined in such a way that, feasibility is always guaranteed. Constraint 3.22 forces the meeting to happen in one of the stations where meeting variables are defined. The other constraints guarantee satisfaction of the arrive-depart headway constraints in that station and also prevent meeting on the mainline track section. Thus, after making the finding the restricted region for meetings, all remaining constraints and binary variables that allow meetings in other stations can be and indeed are simply omitted from the model. The heuristic is coded in AIMMS and run as *MainInitialization* routine. Afterwards, through proper index domain definitions of the variables and constraints, the model with the constraints eliminated is automatically generated by AIMMS.

4.2.2 Restriction of the solution Space for Followings and Overtakings

When it comes to the following and overtakings, the situation is nowhere as simple as in the meeting case. In this case, radical differences may be observed between the heuristic outcome and the actual optimal solution. For example, in the heuristic outcome, one train may overtake the other in the center region of the route, whereas, in the optimal solution, no overtaking may occur between these two trains. The opposite may also happen-no overtaking in the heuristic but overtaking in the actual optimal solution. The problem is, it is not possible to know where the fast train will "catch" the slow one (if will it ever catch) without actually obtaining the solution, because, this depends on the solution itself. If the fast train is delayed in conflict resolutions before catching the slow train, the "gap" between them will widen. Alternatively, if the slow train is delayed in conflict resolutions before being caught up by the fast train, the "gap" between them will tighten. Therefore, it is not possible to reduce the model as much as it was done in the meeting case. However, there is still room for improvement. Both the fast train and the slow train may experience conflicts before the fast train catches up the slow train. In such conflicts, if the slow train is stopped and forced to wait, the fast train can catch it up earlier. On the contrary, if the fast train is stopped and forced to wait, it can catch the slow train up later. This is a good clue for determining the earliest possible place that a fast train can catch up the slow train. The fast train can catch up the slow train in earliest possible place if the fast train is never stopped in the conflicts before catching up the slow train, whereas, the slow train is stopped and forced to wait in all conflicts before being caught up by the fast train. Such a solution can be obtained by temporarily assigning a very high weight on the fast train's delay (remember that the solution takes place over the weighted delays) and very low weight on the slow train's delay. The same can be done on recovery times. If the slow train has a very large recovery time, whereas the fast train has a negative and big in magnitude recovery time (i.e. this train is already severely delayed when it enters the system), coupled with the difference in weights on delays, the heuristic will always give the priority to the fast train and never give the priority to the slow train. Thus, it will be possible to determine the earliest place that the fast train can catch and overtake the slow train. Table 4.1 shows an example to this.

Table 4.1An example to determine the earliest station that train 1 can overtake train 3

	Recovery time (mins)	Temporary weight on Delay
Train 1	-500	5000
Train 3	500	0,0001
Train 5	6	3
Train 7	9	2
Train 9	9	2

In such a case, the heuristic will always force train 3 to stop and wait in its all conflicts. Whereas, train 1 will never stop and wait. This will give us the earliest station that train 1 can catch train 3. For all train couples, for which the slow one enters the system before the fast one, the weights and recovery times are temporarily changed to such extreme values and the heuristic is run once. Then, it is assumed that, the overtaking can happen in one of the four stations in the upstream (for those trains' direction) of the heuristic outcome or later. All binary variables for the stations upstream of this earliest possible station are fixed (not eliminated to be on the safe side for modeling the station / siding capacities).

If any fast train enters the system "well before" a slow train in the same direction, it is assumed that the gap will widen and these two trains will never interact, so all the related variables and constraints are eliminated.

4.3 Use of a "Min-Max" Problem for the Initial Feasible Solution

As previously mentioned, the objective function chosen affects the solution time. Our observation is that, if the maximum (weighted) delay is used as the objective function, then, CPLEX finds the optimal solution very quickly, compared to the case with total (weighted) delay as the objective function. Optimization engines can accept initial feasible solutions and use these solutions to shrink the solution space, i.e. by providing good upper bounds. In all of the problems, before solving the actual problem with total weighted delay, a problem with the maximum weighted delay objective is solved, coupled with the heuristic problem reduction algorithm mentioned above. This application provided quite a good boost in the solution speed of the actual model. This also provided the opportunity to obtain better- quality solutions (from a railway operation perspective). This is shown in Table 4.2. Among the two alternative solutions given in the table, if the objective is total delay, the algorithm would choose solution 2. However, from railway operation perspective, solution 1 is better.

Table 4.2 Sample bad solution obtained from a total delay approach

	Delays in Solution 1 (min)	**Delays in Solution 2 (min)**
Train 1	5	29
Train 2	5	0
Train 3	5	0
Train 4	5	0
Train 5	5	0
Train 6	5	0

So, a multiobjective optimization approach is adopted. The final objective will still be the total weighted delay. But, the maximum weighted delay will be constrained not to be higher than the maximum weighted delay obtained in the initial feasible solution with the maximum weighted delay objective.

The reason for the first problem (maximum weighted delay) being quicker to solve than the second problem (total weighted delay) is probably that, the first one is a *relaxation* of the second one. To illustrate this, we start with a definition, adopted from Wolsey [57].

Definition: Let P_1 be a problem, defined as Min $\{(f_1(x) : x \in X_1\}$ and P_2 be another problem, defined as Min $\{(f_2(x) : x \in X_2\}$. Then, P_2 is a relaxation of P_1, if:

i) $X_1 \subseteq X_2$

ii) $f_2(x) \leq f_1(x)$ for every $x \in X_1$.

Let P_1 be the original problem (with total weighted delay objective) and let P_2 be another problem, with exactly the same domain of decision variables and constraints, but a different objective function (maximum weighted delay).

Proposition: P_2 is a relaxation of P_1

Proof: Condition i is trivial, since feasible solutions are defined exactly on the same variable and constraint domain. For condition ii, definition of the delay comes in. Since delay is defined to be nonnegative (trains arriving earlier than their respective scheduled arrival times have a delay of zero), total weighted delay cannot be smaller than the maximum weighted delay.

CHAPTER 5

NUMERICAL EXPERIMENTS

In this chapter, the properties of the problem instances solved in the numerical experiments and the results will be given.

5.1 Railway Line

The hypothetical railway line on which the problem instances are generated is derived from a real TCDD line. Its topology is based on the original topology of Arifiye-Çukurhisar section of İstanbul-Ankara conventional railway line. Arifiye- Çukurhisar section is a single track line. Arifiye is the western terminus and Çukurhisar is the eastern terminus of the line. Starting from Çukurhisar, the line is double track until Eskişehir. Note that only the topology (names of the stations / sidings and numbers of secondary tracks in the stations / sidings) are real. The minimum running times and train types (which will be presented soon) are all hypothetical. In Table 5.1, the topology of the line is presented. Names of the stations are given only for informative purposes. From now on, stations will be denoted by their respective station number. Remember that, station numbers increase in the eastbound direction.

Table 5. 1 Topology of the hypothetical railway line

Station name	Station or siding	Station / siding number	Number of secondary lines
Arifiye	Station	1	∞
Doğançay	Station	2	2
Alifuatpaşa	Station	3	2
Pamukova	Station	4	2
Hayrettin	Siding	5	1
Mekece	Station	6	2
Osmaneli	Station	7	2
Sarmaşık	Siding	8	1
Bayırköy	Station	9	2
Vezirhan	Station	10	2
Pelitözü	Siding	11	1
Bilecik	Station	12	2
Yayla	Siding	13	1
Karaköy	Station	14	2
Ayvalı	Siding	15	1
Bozüyük	Station	16	2
İnönü	Siding	17	2
Çukurhisar	Siding	18	∞

5.2 Trains

In the line, three different types of trains are assumed to run. These are named as fast, semi-fast and slow. All trains have their respective minimum running times defined in each mainline track segment. The problem instances are generally 6 + 5, that is, there are 6 eastbound trains and 5 westbound trains in the instance. This is the upper limit of the model, under the conditions described below. Trains are numbered from 1 to11. Eastbound trains took odd numbers and westbound trains took even numbers. Numbering is done according to a priority system. In the system, a smaller number stands for a higher priority train. In railway operations, faster trains get higher priorities and this practice is also adopted here. If two trains are of the same type (which will certainly happen, because there are three types and eleven trains), then, another priority scheme has to be used among the trains bound for the same direction. In this case, the train that enters the system first, gets the priority, hence the smaller number. Remind that, this is a weighted delay problem, so, a weight has to be assigned to each train. Unsurprisingly, the trains with a higher priority have a higher weight. Thus, their delays worsen the objective function more than the lower priority trains' delays. Trains are summarized in Table 5.2.

Table 5. 2 Trains in the problem instances

Train numbers	Directions	Train types	Weights
1	Eastbound	Fast	6
2	Westbound	Fast	6
3	Eastbound	Fast	6
4	Westbound	Semi-fast	4
5	Eastbound	Semi-fast	4
6	Westbound	Semi-fast	4
7	Eastbound	Semi-fast	4
8	Westbound	Slow	3
9	Eastbound	Slow	3
10	Westbound	Slow	3
11	Eastbound	Slow	3

5.2.1 Orders and Frequencies of the Trains

In the instances, the order of entrance time of the trains has a great effect on the actual difficulty of the instance. As previously mentioned, when a faster (higher priority) train enters the system before a slower (lower priority) train in the same direction, no overtaking will be allowed between those two trains. Thus, all binary variables related to following and overtaking of these two trains can be either fixed or omitted from the model (see Chapter 4.1). Furthermore, since the gap between these two trains can be safely assumed to widen, practically, no headway constraints would be required. This leads to the fact that, one can easily "cheat" by selecting the problem instances properly. This is not the case for this thesis. Admittedly, some cheating is done by limiting the number of different train types by three, but, among these types, no cheating is done. In all the instances, trains enter the system from the slowest to the fastest. In Table 5.3, the entrance orders are given. The order is from the left to the right. Different instances are defined by the entrance times of the trains plus the entrance delays. However, the entrances orders do not differ. All of them follow the same scheme. In the table, "s" stands for "slow", "s-f" stands for "semi-fast" and "f" stands for "fast."

Table 5.3 Train entrance orders

Eastbound	9 (s)	11 (s)	5 (s-f)	7 (s-f)	1 (f)	3 (f)
Westbound	8 (s)	10 (s)	4 (s-f)	6 (s-f)	2 (f)	-

Also, entrance times of the trains would produce an effect. If the trains are too infrequent, most (if not all) of the following and overtaking constraints would be nonbinding. Similar would be valid for the respective binary variables, solution space restriction algorithm will fix most of them. An example of how the train order can affect the problem complexity is given in Table 5.4. Note that solution space restriction algorithm does not eliminate any continuous variables, this value will not differ. It can be clearly seen that, such a “cheating” modifies the combinatorial structure of the problem. As the number of trains increases, if that cheating is done, the accompanying increase in the real problem size is not that great. So, in the instances, trains are introduced into the system in small intervals and from slowest to fastest.

Table 5.4Effect of train order and number of trains on problem size

Problem type	**# of trains**	**# of binary variables**	**# of constraints**
Fastest to slowest, big intervals	3 + 3	63	662
Fastest to slowest, big intervals	6 + 5	167	3667
Slowest to fastest, small intervals	3 + 3	119	1571
Slowest to fastest, small intervals	6 + 5	462	>= 41000

To further demonstrate the effect of train order and frequency on difficulty, a special instance with 8 eastbound and 8 westbound trains is prepared. In this instance, trains are introduced into the system from the fastest to the slowest. Also, the intervals between train entrances are one hour. The unsolved state of this special instance is given in Figure 5.1.

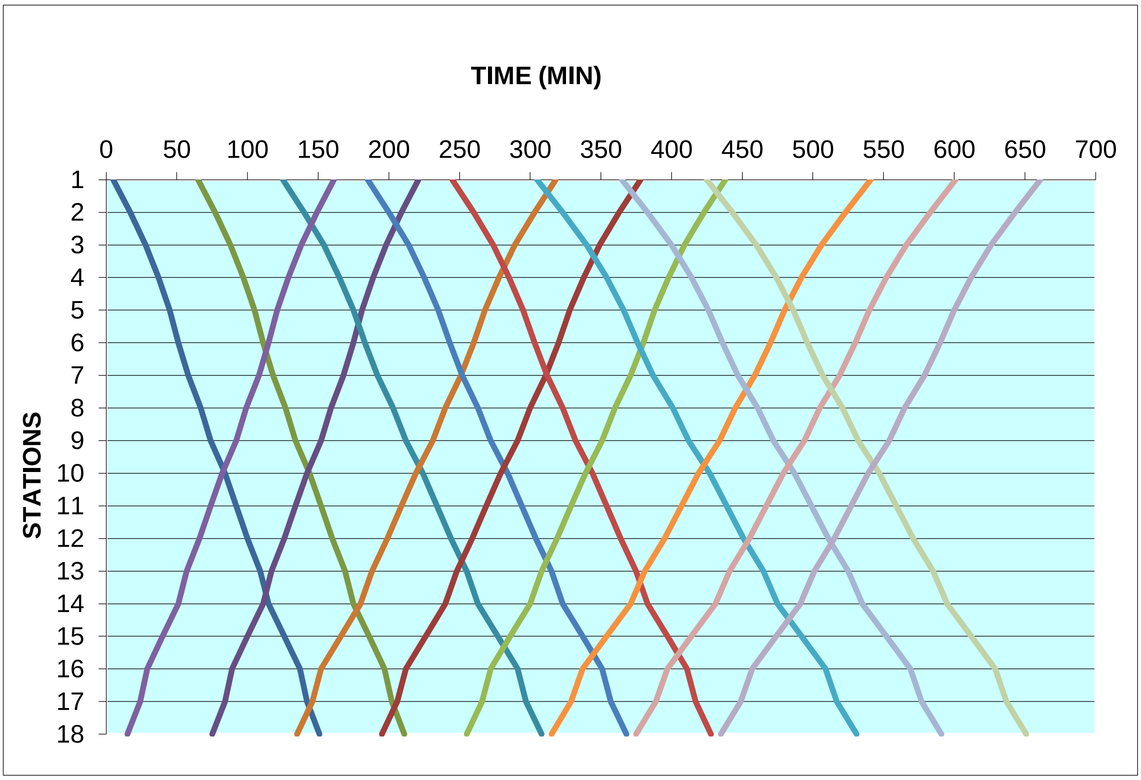

Figure 5.1 Unsolved state of special 8 + 8 instance

Even from the unsolved state, it can be observed that this is not a difficult problem. Gaps between the trains in the same direction are never shrinking, but sometimes enlarging. Even in their smallest form, the gap is 60 minutes. This implies that, no overtaking constraints and no following constraints will be active. Moreover, station capacity constraints will not be active at all. Lastly, some trains will never meet within the system. Problem space restriction algorithm and lazy constraints attributes will recognize such special structures and make the instance even easier to solve. These are all forecasts at this stage, but, in Figure 5.2, it can be seen that they indeed held. This special is instance is solved **within only 7 seconds.** Therefore, despite having more trains, this special instance is much easier than the 6 + 5 instances that will be presented very soon.

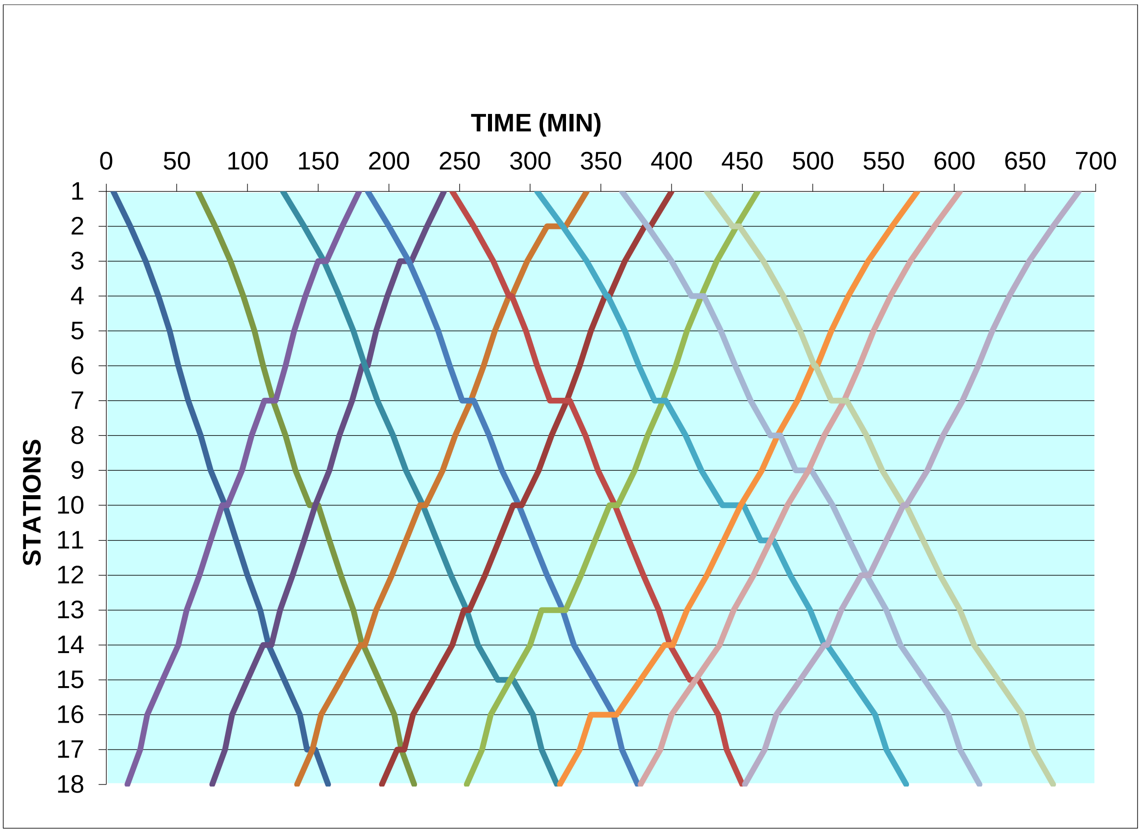

Figure 5.2 The solved state of special 8 + 8 instance

Encouraged by the success in solving the 8 + 8 special instance, **we even tried a 13 + 13problem, formed with the same logic.** Even the 12 + 12 problem is solved in 80 seconds. This implies that, a 13 + 13 instance can be as easy as a 6 + 5 instance if it is specially formed. We again argue that, number of trains and stations are not powerful enough indicators of the difficulty of the problem instance. The frequency and order of the trains are also required to define this.

5.2.2 Minimum Running Times of the Trains

A single minimum running time scheme is used for all the ten instances. It is given in Table 5.5. In the table, the number in the row of station s corresponds to minimum running time between stations s and s + 1. Hence, the row of station 18 is empty.

Table 5.5 Minimum running times (minutes)

	TRAIN TYPE		
Station number	**Fast**	**Semi-fast**	**Slow**
1	12	15	18
2	11	14	17
3	9	11	14
4	8	10	12
5	6	8	10
6	7	9	11
7	9	11	14
8	7	9	11
9	10	12	15
10	8	10	12
11	8	10	12
12	9	11	14
13	6	8	10
14	11	14	17
15	11	14	17
16	5	6	8
17	9	11	14
18	-	-	-

5.2.3 Recovery Times and Entrance Delays of the Trains

In daily operational practice, when a train enters the system, it comes either with some recovery time or with some already experienced delay. Therefore, the generic model accepts recovery times and initial delays as inputs, either explicitly or implicitly, hidden within the relation between the scheduled arrival time and the earliest possible arrival time. Earliest possible arrival time of a train is simply the sum of all minimum running times and the minimum dwell times of that train added into the entrance time of that train. However, generating consistent data for entrance delays and / or recovery times requires the existence of a timetable. Since the model is run on a hypothetical railway line, no timetable is in hand and generation of one would be outside the scope of this study.

At this point, it is useful to provide some information about the way rescheduling works. In this rescheduling model, the only objective is minimizing the total weighted delay. Timetable fidelity is not an objective (There are some studies in the literature, which have such objectives). Timetables have all the conflicts resolved in the planning stage. When, even one of the trains deviate from the timetable, these conflict resolutions are no longer entirely valid. Some of them can be kept fixed. However, if no timetable fidelity objective is present, minimizing delay is the only objective, when even one of the trains get a delay, it is practical to cancel all the conflict resolutions in the timetable and solve the conflict resolution problem from scratch. Therefore, without loss of generalization, we can assume that, all trains have zero entrance delays and zero recovery times. This is similar to the cases in the railway lines operated without a timetable. These lines do not have a timetable at all, and our line's timetable is no longer valid because of a delay train disrupting the previous conflict resolutions. When all trains have zero recovery times and zero entrance delays, their scheduled arrival times are assumed to be equal to their earliest possible arrival times and their delays are calculated accordingly. In this case, no train could arrive earlier than its scheduled arrival time.

5.2.4 Entrance Times of the Trains

Trains are introduced into the system at randomly generated times. However, the same order given in Table 5.3 are always respected. The entrance times are given in Table 5.6.

Table 5.6 Entrance times of the trains into the system, with respect to time 0 (minutes)

	Train 1	**Train 2**	**Train 3**	**Train 4**	**Train 5**	**Train 6**	**Train 7**	**Train 8**	**Train 9**	**Train 10**	**Train 11**
Instance 1	79	60	95	39	40	49	59	6	9	23	26
Instance 2	82	57	92	42	43	46	56	9	12	26	23
Instance 3	46	66	56	37	22	56	33	7	2	19	12
Instance 4	67	72	81	34	33	54	51	1	4	16	19
Instance 5	73	70	85	41	43	56	56	9	5	26	25
Instance 6	72	54	88	33	39	44	58	1	9	14	19
Instance 7	76	67	91	37	37	48	57	2	8	18	27
Instance 8	59	69	75	38	31	54	44	9	1	27	19
Instance 9	82	72	100	41	44	53	62	9	1	23	26
Instance 10	67	54	83	28	33	44	53	5	2	16	13

5.2.5 Minimum Headways of the Trains

As mentioned before, one set of parameters that define the problem instance is the headways. There are three different headways required: Arrive-arrive headway, depart-depart headway and arrive-depart headway. In reality, these take different values for each train couple and each station. However, for simplicity, they are assumed to be constant in all the instances. In Table 5.7, headway values are provided.

Table 5. 7 Headways

Arrive-arrive headway (min)	**Depart-depart headway (min)**	**Arrive-depart headway (min)**
2	3	2

5.3 Numerical Results of the Runs

The constructed instances are run on a computer, with Intel Core i5- 2400 processor, 3.1 GHz processor speed and 6.0 GB RAM. In Figure 5.1, the initial state of Instance 1 is given. In order to make the figure large enough, legend is given in a separate figure (Figure 5.2). The figure is a picture of what happens if the trains try to run with their maximum speeds (minimum running times). There are 6 x 5 = 30 unresolved meeting conflicts. Also, there are an unknown number of following and overtaking conflicts. It is unknown, because, if a fast train catches up a slow train, if it is not allowed to overtake, the conflict will occur repeatedly. This is not the case for meeting conflicts. Once the meeting conflict between an eastbound train and a westbound train is resolved, it will not occur again. Hence, the number of meeting conflicts is known, but the number of following- overtaking conflicts is not. Long story short, with its initial state, the system is a real mess. Reader is strongly encouraged to compare this situation with the one given in Figure 5.1.

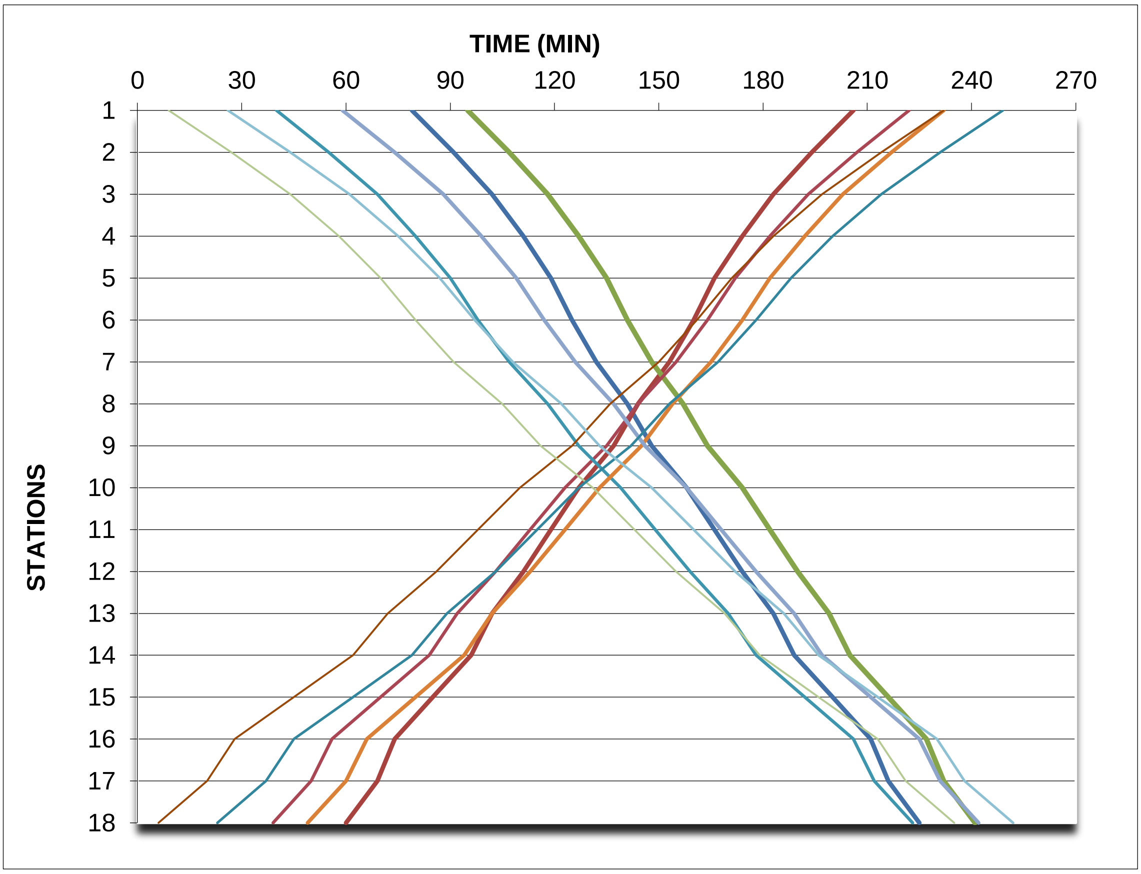

Figure 5.3 Initial (unsolved) state of instance 1

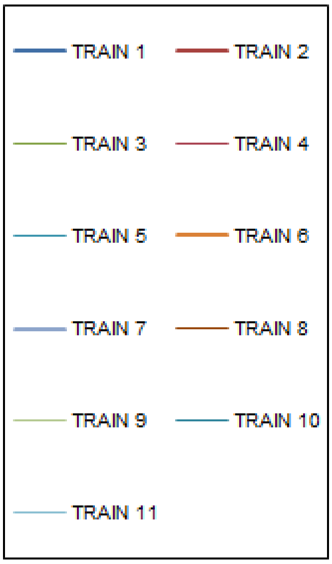

Figure 5.4 Legend of Figure 5.1

In Figure 5.3, the final solution generated by the algorithm is presented.

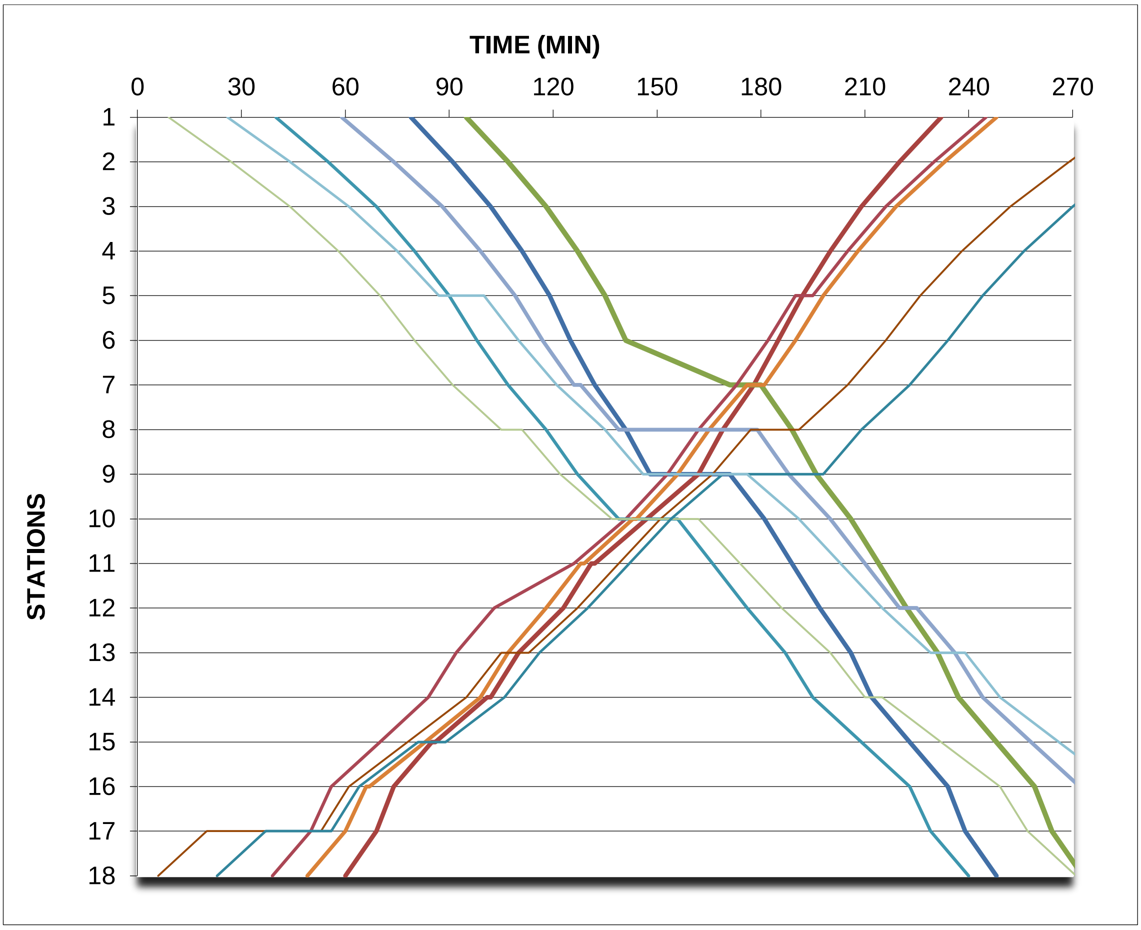

Figure 5.5Final solution of instance 1

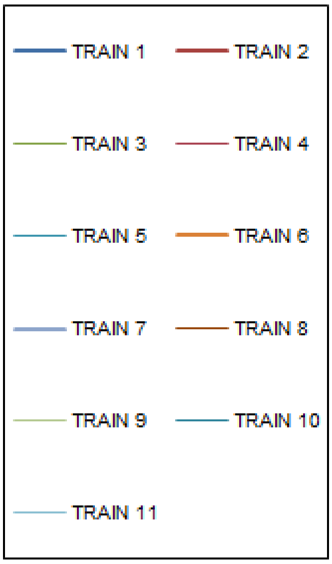

Figure 5.6 Legend of Figure 5.3

As seen in the figure, all conflicts are resolved. In this instance, there is a congested area between stations 7- 10. None of the station / siding capacities are violated in the solution, but, it is not clearly visible in Figure 5.3. In Figure 5.5, we zoom into this congested area. In the figure, the numbers of secondary lines in the stations are also given. This is clearly a feasible solution, no station capacity is violated.

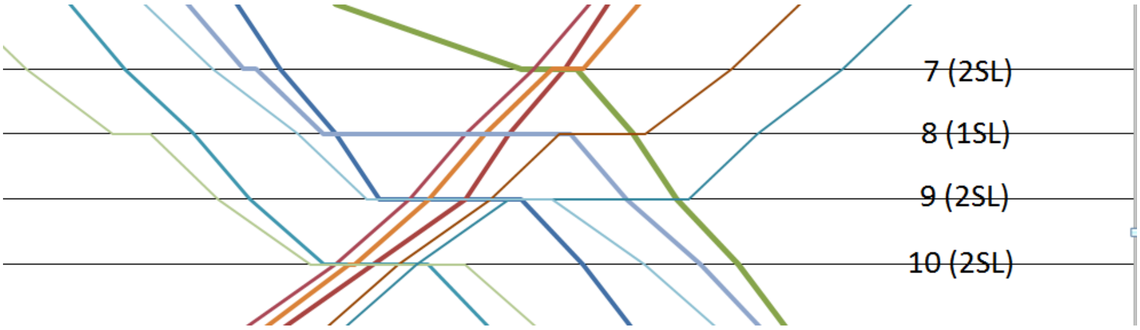

Figure 5.7 Congested area of Instance 1 is zoomed in.

In Table 5.8, the summary of the solution of Instance 1 is provided.

Graphical solution outcomes of the other instances are not provided here, to avoid excessive crowd in the document. The results of all other instances (together with instance 1) are summarized in Table 5.8. It can be observed that, there are slight differences in the number of binary variables. This is due to the solution space restriction algorithm, described in Chapter 4.2. Once a decision variable is fixed, AIMMS treats that variable as a parameter and does not count it towards the total number of variables. Lastly, the really huge level of the number of constraints should not fool the reader. Very few of these constraints are indeed effective at any iteration of the algorithm. Most of them are lazy constraints as defined in Chapter 4.1. They are explicitly considered only if they are violated in the LP relaxation.

Table 5.8 Summary of the optimal solution of instance 1

Train Number	**Entrance time (min)**	**Arrival time (min)**	**Desired arrival time (min)**	**Arrival Delay (min)**	**Weight**	**Weighted delay (min)**
1	79	248	225	23	6	138
2	60	232	206	26	6	156
3	95	273	241	32	6	192
4	39	245	222	23	4	92
5	40	240	223	17	4	68
6	49	248	232	16	4	64
7	59	289	242	47	4	188
8	6	287	232	55	3	165
9	9	271	235	36	3	108
10	23	305	249	56	3	168
11	26	305	252	53	3	159

Table 5.9 Summary of all runs

Instance Number	**# of Binary Variables**	**# of Continuous Variables**	**# of Constraints**	**Max. Weighted Delay (min)**	**Total Weighted Delay (min)**	**Solution time (s)**
1	457	387	38501	192	1498	84
2	463	387	38462	192	1383	96
3	501	387	38694	198	1562	101
4	445	387	37828	195	1593	87
5	493	387	38694	204	1384	108
6	498	387	39052	198	1496	116
7	461	387	38069	198	1408	119
8	515	387	39049	201	1466	164
9	462	387	38349	192	1453	130
10	467	387	38479	204	1370	105

5.4 Experiments on a Small Sized Problem Instance to Demonstrate the Effects of Speed- Up Methods

Bigger sized problem instances, such as the ones solved in the previous parts of this chapter, cannot be solved in a reasonable time without applying speed- up routines. Because of this fact, it is not possible to observe the effects of the individual speed- up routines. For this purpose, a smaller (4 + 4) problem instance was set up. In this instance, the train entrance orders and frequencies follow the same schemes as the experimental runs in this chapter. Minimum running times, weights and headways are also the same.

Table 5.10 Train orders in the 4 + 4 instance

Train numbers	**Directions**	**Train types**	**Weights**
1	Eastbound	Fast	6
2	Westbound	Fast	6
3	Eastbound	Semi-fast	6
4	Westbound	Semi-fast	4
5	Eastbound	Slow	4
6	Westbound	Slow	4
7	Eastbound	Slow	4
8	Westbound	Slow	3

Table 5.11 Entrance times of the trains into the system in the 4 + 4 instance with respect to time 0

	Train 1	**Train 2**	**Train 3**	**Train 4**	**Train 5**	**Train 6**	**Train 7**	**Train 8**
Entrance times (min)	50	70	35	30	5	50	20	30

The instance is run using the same hardware and software as the previous instances. The same instance is solved with four different approaches: No speed- up (base), using only lazy constraint attributes (Lazy), using lazy constraint attributes with the solution space restriction algorithm (Lazy + Restr) and finally the multiobjective added onto the top of them (Lazy + Restr + Multi). At every minute, the optimality gap values are recorded and the results are plotted in Figure 5.8. The performance difference is clearly visible in the figure.

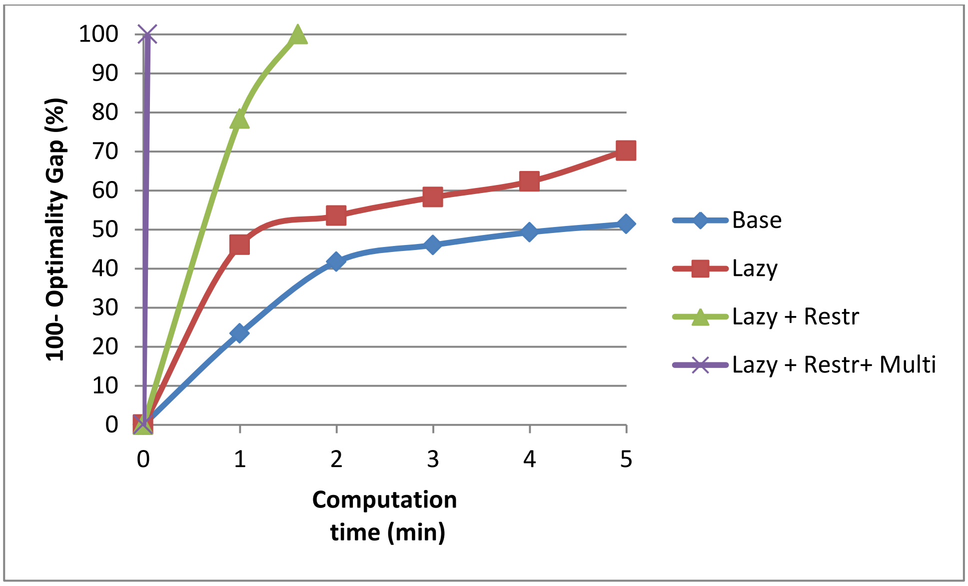

Figure 5.8 Comparison of optimality gaps within five minutes of running

In Table 5.12, the reader can see what happens if the solution is not interrupted until the end. This table also demonstrates the effect of the speed- up routines. Also, for this instance, it can be seen that, the solution restriction algorithm did not harm optimality of the solution. Although this is not a proof, depending on our personal knowledge in railway operations, this algorithm will find the optimal solution for most of the times. In Table 5.13, the arrival time and delay data of trains in the final solution is given. The versions with and without the solution space restriction algorithm produced exactly the same solution as each other. Of course, the last one produced a different objective value for the total weighted delay, because it is practically a multiobjective problem. It also aims to reduce the maximum weighted delay.

Table 5.12 Outcomes of the instance with different approaches

Instance label	# of Binary Variables	# of Continuous Variables	# of Constraints	Total Weighted Delay (min)	Computation time (s)
Base	634	281	12164	707	1156
Lazy	634	281	12164	707	711
Lazy + Restr.	298	281	11073	707	93
Lazy + restr + multi	298	281	11081	797	3

Table 5.13 Summary of the optimal solution of the 4 + 4 instance

Train Number	Entrance time (min)	Arrival time (min)	Desired arrival time (min)	Arrival Delay (min)	Weight	Weighted delay (min)
1	50	208	188	20	6	120
2	10	228	211	17	6	102
3	35	252	213	39	4	156
4	30	245	231	14	4	56
5	5	248	229	19	3	57
6	50	263	235	28	3	84
7	20	278	245	33	3	99
8	70	266	255	11	3	33

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

The aim of this study is to develop a mathematical model for rescheduling the traffic on a single track rail line. The proposed model is a mixed integer linear programming (MILP) model. This model takes into account the station / siding capacity constraints by enumeration. Train scheduling problem is an NP complete problem, analogous to some extend to job shop scheduling problem, albeit an even more difficult one. The combinatorial structure of this NP complete problem clearly demonstrated itself in all of the trial runs. Without any heuristic improvement, all it could solve was a 3 + 3 problem (3 eastbound plus 3 westbound trains). However, the heuristic improvements proposed increased this number to, for now, 6 + 5. The algorithm can solve a 6 + 5 problem within a duration of less than three minutes. This is very important. In real operations, rescheduling problem has to be solved in a very short time, because it has to be solved in a dynamic environment. Trains are not at a static state as the model runs, they are moving. Note that, we did not "cheat" while choosing the instances in the trial runs. Normally, railway authorities tend to "speed-fleet" the trains in timetabling. In this scheme, trains in the same direction enter the system in the order of decreasing speed, thus minimizing the number of potential following and overtaking conflicts. However, in our instances, the scheme is just the opposite. **In order to observe the reaction of the model in the worst cases**, trains are introduced into the system in the order of increasing speed, thus maximizing the potential following and overtaking conflicts. Trains are introduced into the system in the intervals of 10 to 20 min / train / direction, which is indeed an unrealistically high train frequency for such a long (18 stations, approximately 160 km) single track line. **This unrealistically high frequency of trains**

is again (like the difficulty created in choosing the order of trains) chosen for observing some kind of worst case performance.

The proposed speed- up methods are heuristics for solution space restriction,, using lazy constraint attribute to reduce the effect of enumerated constraints and adopting a multiobjective approach to first find the Maximum Weighted Delay and then Total Weighted Delay, using the feasible solution from the former. All of these algorithms provided significant improvements of computation time for the specific instances.

The solution space restriction algorithm shortened the solution time of the model quite radically, as shown in Chapter 5.4. Indeed, in its presolve stage, CPLEX applies its own algorithms to eliminate some of the constraints and variables. However, we can be somewhat more aggressive in elimination, because, our algorithm is indeed based on a solution of the problem, albeit a relaxed version (station / siding capacities). Since this algorithm is restricting the places of meetings, it is not possible to provide a formal proof of the optimality of the solutions. However, based on our own knowledge and experience in railway operations, we can argue that, in the optimal solution, it is highly unlikely that the meeting will need to be outside the region defined by this restriction algorithm. The region is assumed to contain five stations. Carrying the meeting point outside this region creates an excessive waiting time for one of the trains and this most probably contradicts with the maximum weighted delay objective.

The multiobjective approach provided a surprisingly good boost for solution speed. It provided in the order of 90% improvement in the computation time. In the trial runs, it was observed that, the majority of time is consumed by the first problem, Minimum_Maximum_Weighted_Delay. After the first problem is solved, the second and major problem is indeed very quickly solved, in only a few seconds. This is most probably due to the properties and capabilities of CPLEX. Both problems have the same feasibility domains, thus, any combinations of decision variables feasible for one will be feasible for the other. Most probably, CPLEX can improve the solution with simple left-shifts and quickly arrives at the optimum solution. Since the problem is solved in a heuristically restricted solution space, it is not possible to present a formal proof of optimality. However, at least the modeling approach is designed towards obtaining a pareto- optimal solution, because the problem is now a multi- objective one. However, it is not a very typical multi- objective problem. There is a slight contradiction between the objectives (see Table 4.2), but, also, a positive correlation exists. The positive correlation between two objectives may also help for increase in the solution speed.

6 + 5 trains may seem to be a small problem instance, and from timetabling point of view, it indeed is. However, this is not a timetabling problem. This is a rescheduling problem. Rescheduling algorithms have to be run in dynamic environments. Trains are moving and they can always gain additional primary delays. This necessitates the model to be rerun. This puts a practical limit to the time window of a rescheduling problem. Time window is a preset limit to the scope of the rescheduling problem. Two hours is quite a reasonable limit, because, the larger the time window is, the more likely the solution generated by the model is to become invalid soon. 6 + 5 is quite a high number of trains to be scheduled on a single track railway line. In most of the single track railways, such a big traffic is not expected. It can be observed in exceptional emergency cases.

Future work on this model will include adopting clever decomposition algorithms. In this model, meetings in stations with more than one secondary line are defined with two decision variables per station per train couple. This puts a huge computational workload onto the model and it is unnecessary for most of the cases. Such variables are needed only when siding capacity constraints for more than one eastbound trains and more than one westbound trains will be binding at a particular station. Lazy constraint attribute prevents some of the workload generated, but, in this case, the variables are also “lazy.” Thus, future work will include such decomposition algorithms.

REFERENCES

[1] Billionnet, A. (2003). "Using Integer Programming to Solve the Train-Platforming Problem", Transportation Science, 37(2):213-222.

[2] Caprara A, Fischetti, M and Toth, P. (2002)."Modeling and Solving the Train Timetabling Problem", Operations Research, 50(5):851-861.

[3] Dessouky, M.M., Lu, Q., Zhao, J. and Leachman, R.C. (2006)."An exact solution procedure to determine the optimal dispatching times for complex rail networks. IEE Transactions, 38(2):141-152.

[4] Gorman, F.(1998)."An application of genetic and tabu searches to the freight railroad operating plan problem", Annals in Operations Research 78:51–69.

[5] Salim, V. and Cai, X.(1997)."A genetic algorithm for railway scheduling with environmental considerations", Environmental Modeling Software 12(4):301-309.

[6] Törnquist, J. and Persson, J. A.(2005)."Train Traffic Deviation Handling Using Tabu Search and Simulated Annealing", Proceedings of 38th Annual Hawaii Int Conf Syst Sci,00(C):1–10.

[7] Şahin, İ.(1999)."Railway traffic control and train scheduling based on inter-train conflict management", Transportation Research Part B: Methodological, 33(7):511-534.

[8] D'Ariano, A., Pacciarelli, D. and Pranzo, M.(2007)."A branch and bound algorithm for scheduling trains in a railway network", European Journal of Operational Research,183(2):643-657.

[9] Castillo, E., Gallego, I., Ureña, J.M. and Coronado, J.M.(2011)."Timetabling optimization of a mixed double-and single tracked railway network", Applied Mathematical Modelling, 35(2):859–78.

[10] Wikipedia, Single-track railway, https://upload.wikimedia.org/wikipedia/commons/f/f1/The_train_to_Kyle_of_Lochalsh_-_geograph.org.uk_-_467835.jpg, Date of access: September 30, 2015.

[11] Wikipedia, Double track, https://upload.wikimedia.org/wikipedia/commons/thumb/4/41/Twin_track_of_train_rails_in_a_wooded_area.JPG/1024px-Twin_track_of_train_rails_in_a_wooded_area.JPG, Date of access: September 30, 2015.

[12] Wikipedia,Quadruple track, https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Otterington_railway_station_MMB_01.jpg/1280px-Otterington_railway_station_MMB_01.jpg., Date of access: September 30, 2015

[13] Pachl, J. (2002). Railway Operation and Control, VTD Rail Publishing, Mountlake Terrace WA

[14] Jovanovic, D. (1989).Improving railroad on-time performance: models, algorithms and applications., PHd thesis, University of Pennsylvania.

[15] Pinedo, M and Singer, M.(1999)."Shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop", Naval Research and Logistics, 46(1):1–17.

[16] Yalçınkaya, Ö.and Bayhan, M. G.(2012)."A feasible timetable generator simulation modelling framework for train scheduling problem". Simulation Modelling Practice and Theory,20(1):124-141.

[17] Törnquist, J. and Persson, J. A.(2007)."N-tracked railway traffic re-scheduling during disturbances". Transportation Research Part B: Methodological,41(3):342-362.

[18] Acuna-Agost, R., Michelon, P., Feillet, D. and Serigne, G. A.(2011)."MIP-based Local Search Method for the Railway Rescheduling Problem". Networks,DOI 10.100:70–86.

[19] Törnquist Krasemann, J.(2012)."Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances". Transportation Research Part C: Emerging Technologies,20(1):62–78.

[20] Lee, Y. and Chen, C.Y.(2009)."A heuristic for the train pathing and timetabling problem", Transportation Research Part B: Methodological, 43(8-9):837-851.

[21] Higgins,A., Kozan, E. and Ferreira, L.(1997)."Heuristic techniques for single line train scheduling", Journal of Heuristics, 3(1):43–62.

[22] Zhou, X and Zhong, M.(2007)."Single track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bound"Transportation Research Part B: Methodological,41(3):320-341.

[23] Şahin, G. (2006).New combinatorial approaches forsolving railroad planning and scheduling problems, PHd thesis, University of Florida, 2006.

[24] Gafarov, E.R., Dolgui, A. and Lazarev, A. A.(2015)."Two-station single track railway scheduling problem with trains of equal speed", Computers and Industrial Engineering 85:260–7.

[25] Dündar, S. and Şahin,İ.(2013)."Train re-scheduling with genetic algorithms and artificial neural networks for single track railways", Transportation Research Part C: Emerging Technologies,27:1–15.

[26] Espinosa-Aranda, J.L. and García-Ródenas, R.(2013)."A demand-based weighted train delay approach for rescheduling railway networks in real tim"e. Journal of Rail Transport Planning and Management, 3(1-2):1–13.

[27] Albrecht,A. R., Panton, D.M. and Lee, D.H.(2013)."Rescheduling rail networks with maintenance disruptions using Problem Space Search", Computers and Operations Research, 40(3):703–712.

[28] Narayanaswami, S. and Rangaraj, N.(2015)."A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation", Expert Systems and Applications, 42(5):2638–2656.

[29] Li, X., Wang, D., Li,K. and Gao, Z.(2013)."A green train scheduling model and fuzzy multi-objective optimization algorithm", Applied Mathematical Modelling 37(4):2063-2073.

[30] Xu, X., Li, K., and Yang, L.(2015)."Scheduling heterogeneous train traffic on double tracks with efficient dispatching rules",Transportation Research Part B: Methodological, 78:364–384.

[31] Narayanaswami, S. and Rangaraj, N.(2013)."Modelling disruptions and resolving conflicts optimally in a railway schedule", Computers and Industrial Engineering 64(1):469–481.

[32] Louwerse, I. and Huisman, D.(2014)."Adjusting a railway timetable in case of partial or complete blockades", European Journal of Operational Research, 235(3):583–593.

[33] Fabris, S. D., Longo, G., Medeossi, G. and Pesenti, R.(2014)."Automatic generation of railway timetables based on a mesoscopic infrastructure model". Journal of Rail Transport Planning and Management,4(1-2):1–12.

[34] Furini, F., Kidd, M.P.(2013)."A fast heuristic approach for train timetabling in a railway node", Electronic Notes in Discrete Mathematics 41:205–212.

[35] Xu, X., Li, K.,Yang, L.and Ye, J.(2014)."Balanced train timetabling on a single-line railway with optimized velocity." Applied Mathematical Modelling 38(3):894–909.

[36] Cacchiani, V., Caprara A. and Toth, P.(2008)."A column generation approach to train timetabling on a corridor", 4OR, 6(2):125–142.

[37] Shafia, M.A, Sadjadi, S.J., Jamili, A, Tavakkoli-Moghaddam, R. and Pourseyed-Aghaee, M.(2012)."The periodicity and robustness in a single track train scheduling problem", Applied Soft Computing 12(1):440–452.

[38] Meng, L. and Zhou, X.(2011)."Robust single track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach",Transportation Research Part B: Methodological 45(7):1080–1102.

[39] Burdett, R.L. and Kozan, E.(2014)."Performance profiling for predictive train schedules", Journal of Rail Transport Planning and Management, 4(4):98–114.

[40] Kuo, A., Miller-hooks, E. and Mahmassani, H.S.(2010)."Freight train scheduling with elastic demand", Transportation Research Part E: Logistics and Transportation Review 46(6):1057–1070.

[41] Corman, F., D’Ariano, A., Hansen, I.A., and Pacciarelli, D.(2011)."Optimal multi-class rescheduling of railway traffic", Journal of Rail Transport Planning and Management, 1(1):14–24.

[42] Liu, S.Q. and Kozan, E.(2011)."Optimising a coal rail network under capacity constraints", Flexible Services and Manufacturing Journal,23(2):90–110.

[43] Arangú, M., Salido, M.A. and Barber, F. (2010). "AC2001- OP: An arc-consistency algorithm for constraint satisfaction problems", Lecture Notes in Computer Science, 6098 LNAI(PART 3):219–228.

[44] Luethi, M., Medeossi, G. and Nash, A.(2009)."Structure and simulation evaluation of an integrated real-time rescheduling system for railway networks" Networks and Spatial Economics,9(1):103–121.

[45] Caimi, G., Burkolter, D., Herrmann, T., Chudak, F., and Laumanns, M.(2009)."Design of a railway scheduling model for dense services". Networks and Spatial Economics,9(1):25–46.

[46] Mu, S. and Dessouky, M.(2011)."Scheduling freight trains traveling on complex networks",Transportation Research Part B: Methodological, 45(7):1103–1123.

[47] Cordone, R. and Redaelli, F.(2010)."Optimizing the demand captured by a railway system with a regular timetable". Transportation Research Part B: Methodological, 45(2):430–446.

[48] Cacchiani, V., Caprara, A. and Toth, P.(2010) "Scheduling extra freight trains on railway networks",Transportation Research Part B: Methodological 44(2):215–231.

[49] Burdett, R.L. and Kozan, E. (2009)."Techniques for inserting additional trains into existing timetables",Transportation Research Part B: Methodological 43(8-9):821–836.

[50] Ghoseiri, K. and Morshedsolouk, F.(2006). "ACS- TS: Train scheduling using ant colony system", Journal of Applied Mathematics and Decision Sciences (January):1–28.

[51] Parkes, D. C. and Ungar, L.H.(2001)."An auction-based method for decentralized train scheduling", Proceedings ofFifth International Conference on Autonomus Agents AGENTS 01, May 28- June 1, 2001, 43–50.

[52] Oliveira, E. and Smith, B.M.(2001)."A Combined Constraint-Based Search Method for Single track Railway Scheduling Problem", Lecture Notes in Artificial Inteligence,371–378.

[53] Medanic, J. and Dorfman, M.J.(2002)."Efficient scheduling of traffic on a railway line". Journal of Optimization Theory and Applications,115(3):587–602.

[54] Higgins, A., Kozan, E. and Ferreira, L.(1997)."Modelling the number and location of sidings on a single line railway", Computers and Operations Research ,24(3):209–220.

[55] Carey, M.(1994)."A model and strategy for train pathing with choice of lines, platforms, and routes", Transportation Research Part B: Methodological,28(5):333–353.

[56] Cai, X., Goh, C.J., Mees, A.(1998)."Greedy heuristics for rapid scheduling of trains on a single track", IIE Transactions,30(5):481–493.

[57] Wolsey, L. A. (1998). Integer Programming, John Wiley and Sons Inc, New York.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Gökçe AYDIN

Date of birth and place :24.01.1980-ANKARA

Foreign Languages : ENGLISH

E-mail : elbe80@gmail.com

EDUCATION

Degree	Department	University	Date of Graduation
Master	Civil Engineering	Yıldız Technical University.	September 2009
Undergraduate	Environmental Engineering	Middle East Technical University	June 2002

WORK EXPERIENCE

Year	Corporation/Institute	Enrollment
2009 - Now	Yıldız Technical University	Research Assistant
2008 - 2009	Aksaray University	Research Assistant