RAILWAY CREW CAPACITY PLANNING PROBLEM WITH CONNECTIVITY CONSIDERATIONS IN PAIRINGS

by

Ali Çetin Suyabatmaz

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
Spring 2012

RAILWAY CREW CAPACITY PLANNING PROBLEM WITH CONNECTIVITY CONSIDERATIONS IN PAIRINGS

Approved by:

Assist. Prof. Dr. Güvenç Şahin ....
(Thesis Supervisor)

Assist. Prof. Dr. Tevhide Altekin ....

Assoc. Prof. Dr. Barış Balcıoğlu ....

Assoc. Prof. Dr. Kerem Bülbül ....

Assoc. Prof. Dr. Bülent Çatay ....

to my family...

Acknowledgements

I owe my deepest gratitude to my advisor, Dr. Güvenç Şahin, for his enthusiasm, encouragement and guidance throughout my graduate studies. It has been and always will be a great pleasure to work with him. Besides Dr. Şahin, I would like to thank Dr. Tevhide Altekin, Dr. İlker Birbil and Dr. Kerem Bülbül for influencing me to higher education, to research, and to teaching. I could not have asked for better role models, each inspirational, supportive and patient.

One of the joys of completion is to look over the past and remember all the friends and family who have helped and supported me along this short but fulfilling journey. I am indebted to my many student colleagues for providing a stimulating and fun environment in which it is pleasure to learn and grow. I would like to express my heartiest thanks to Mahir Yıldırım, İbrahim Muter, Nurşen Aydın, Ömer Özkırımlı, Halil Şen, Anıl Can, Nükte Şahin, Mustafa Şahin, Semih Atakan, Birce Tezel, Özge Arabacı, Recep Koca, Utku Olgun and every single member of FENS-1021. I am also grateful to Ece Nalbantoğlu and Serhat Çeşme for their endless and sincere friendship since high school. I would like to thank Bahar Mete for always being by my side patiently throughout this adventure and bringing joy to every second we shared together. Finally and above all, I would like to thank my family for their love, patience and support at all times.

EŞLEŞMELERDE BAĞLANILABİLİRLİĞİN DİKKATE ALINDIĞI DEMİRYOLLARI EKİP KAPASİTE PLANLAMA PROBLEMİ

Ali Çetin Suyabatmaz

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2012

Tez Danışmanı: Güvenç Şahin

Anahtar Kelimeler: Ekip Planlama, Taktik Seviyede Planlama, Demiryolu, Kolon-ve-satir Turetme, Uzay-zaman Çizgesi, Çizge Akışı.

Özet

Demiryolu planlamasinda ekip, stratejik, taktik ve operasyonel seviyede ele alınması gereken en önemli kaynaklardan birisidir. Birçok demiryoulu işletmesinde personel giderleri yakıt ve enerji giderlerini dahi geride bırakmış ve toplam giderlerin üçte birinden fazlasını oluşturmuştur. Bu bağlamda etkin ve verimli ekip yönetimi önemli bir planlama problemine dönüşmektedir. Bu çalışmada, bölge ekip kapasitesini (ihtiyaç duyulan ekip üyesi sayısını) belirlemeyi amaçlayan taktik seviyede demiryolu ekip kapasite planlama problemi ele alınmaktadır. Genişletilmiş problem tanımı, operasyonel bir gereklilik olan ekip görev çizelgelerinin bir çizelge döneminden diğer çizelge dönemine geçişte birbirlerine bağlanabilir olması, dolayısıyla birbirini takip edecek çizelge dönemlerinde ekiplerin olurlu ve kurallara uygun birer görev çizelgesiyle çalışıyor olmasını da kapsar. Problem için bir küme-kaplama tipi model geliştirilmiş ve çözüm yöntemi olarak bir eşzamanlı kolon-ve-satır türetme algoritması önerilmiştir. Buna ek olarak, çok katmanlı bir ağ gösterimi geliştirilmiş ve çözüm için bir ağ akışı modeli önerilmiştir. Türkiye Cumhuriyeti Devlet Demiryolları'ndan alınan veri üzerinde yapılan bilgisayısal çalışma sonuçları sunulmaktadır.

RAILWAY CREW CAPACITY PLANNING PROBLEM WITH CONNECTIVITY CONSIDERATIONS IN PAIRINGS

Ali Çetin Suyabatmaz

Industrial Engineering, Master's Thesis, 2012

Thesis Supervisor: Asst. Prof. Dr. Güvenç Şahin

Keywords: Crew Planning, Tactical Planning, Railway, Column-and-row Generation, Space-time Network, Network Flow.

Abstract

Crew is one of the most crucial resources in railway planning that needs to be considered at strategic, tactical and operational planning levels. During the last decade, crew-related costs outweigh energy expenditures and constitute more than one third of general expenditures in most railways. Therefore, sufficient but effective crew management is a critical planning problem which may lead to important savings. In this study, we deal with the tactical crew capacity planning problem which determines the minimum required number of crew members. In our setting, the feasibility of crew schedules and the connectivity of rosters are integrated to find a repeatable set of schedules that satisfy the operational rules and regulations. We develop a set-covering type formulation and propose a simultaneous column-and-row generation algorithm. We also propose a network representation of the problem and develop a corresponding network flow formulation. In order to compare efficiency and effectiveness of the two solution methods, we perform a comprehensive computational study with data sets acquired from Turkish State Railways and present the results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Crew is one of the most crucial resources in railway planning that needs to be considered at strategic, tactical and operational planning levels. In the strategic level, company-wide decisions that would vastly affect the level and allocation of crew resources over all regions are made. Some of these problems are related to physical infrastructural changes such as establishment of new crew regions or new crew exchange stations for trains operating between crew regions; as well as high level adjustment of company practice such as re-distribution of inter-regional workload that may result from modifications in the duty-crew region assignments. At the tactical level, individual capacities of crew regions are determined. Given the train timetable that is under the responsibility of a particular crew region, a tactical level planning problem determines the minimum required number of crew members that would operate these train schedules. With respect to various company-wide and/or region-specific objectives, train duties are paired into feasible crew schedules that honor several rules and regulations. Based on the crew capacities of the regions, decisions related to daily practice are made at the operational level. These decisions are based on the assignment of individual crew members to train duties that would result from rostering of the crew schedules that are determined at the tactical level.

As an example, the system in Turkish State Railways (TCDD) is composed of multiple crew regions. Each region has a central home base station and each region is responsible for effectively planning and managing their crew; however, the operations should be executed in coordination with other regions. During the last decade, crew-related costs outweigh energy expenditures and constitute more than one third of general expenditures of the Turkish State Railways [19, 20, 21]. Therefore, sufficient but effective crew management is a critical planning problem. Since the fixed crew salaries constitute a major component in crew related costs, minimizing crew capacities of regions (at the tactical level) may lead to significant savings.

Tactical decisions should take into account the operational level considerations as much as possible. Besides the cost-related concerns, balancing workload amongst individual crew members and feasible rostering are some of the vital ones. In the planning process, tactical decisions are made by considering a finite planning period and the recurrence of the planning period is overlooked, which may lead to crew schedules that are not implementable in practice. We may exemplify such a drawback for a base station where we have only two crew members to operate the weekly duties under the responsibility of this station. According to feasible crew schedules, one crew member returns back to the base at 11 pm while the other member returns back at 11:30 pm on Sunday having performed their last duties at the end of the week. The earliest duty at the beginning of the week starts at 8 am on Monday morning. As a result, none of these crew members may perform this train duty as the minimum home rest should last at least 16 hours. Although we have feasible schedules for the planning period of one week; we fail to honor some of the regulations in the second week of operations.

In railways, crew resource is critical and needs to be scheduled with respect to strict rules and regulations. When infeasibilities regarding the continuity of the schedules are faced at the operational level, either the managers resort to patching or the schedules are manually modified to guarantee continuity of the crew schedules so that the pairing plans can be repeated from one period to the next. Thus, integrating rostering related (operational level) concerns into the pairing (tactical level) decisions while determining periodically repetitive schedules is a challenging one as the crew should be able to follow duty schedules not only in a single period, but also the availability of crew should be guaranteed with respect to the periodic recurrence of the planning horizon. In this study, we deal with the tactical crew capacity planning problem which determines the minimum required number of crew members. In our setting, the feasibility of crew schedules and the connectivity of rosters are integrated to find a repeatable set of schedules that satisfy the operational rules and regulations.

From a methodological point of view, set-covering type formulations and network flow formulations have been competing with each other in this research area. Note that this is also true for other resource planning problems in other transportation modes such as airlines and freight transportation. Therefore, we look into both type of formulation approaches and aim to understand the efficiency and effectiveness of solution methods based on these types of formulations.

Our contributions in this study can be summarized as follows:

- We first focus on the tactical level crew capacity planning problem where:
  - we suggest improvements for the network flow formulation of the problem and improve the results of an earlier study, and
  - we develop a set-covering type formulation of the problem and propose a column generation algorithm.
- We define the tactical level crew capacity planning problem with connectivity considerations in pairings;
  - we develop a set-covering type formulation and propose a simultaneous column-and-row generation algorithm, and
  - we discuss a network representation of the problem and give the corresponding network flow formulation.
- In order to compare the efficiency and effectiveness of the two solution methods devised for each problem, we perform a computational study with data sets acquired from TCDD.
- We show that the decisions on regional crew capacities without connectivity of the schedules might significantly differ from those where connectivity of schedules are integrated into the problem.
- We propose an algorithm for an alternative rostering strategy for the tactical level crew capacity planning problem with connectivity considerations in pairings where the long-term workload balancing is to be achieved.

Following a review of the literature on crew-related railway problems in Chapter 2, we present our study on the tactical level crew capacity planning problem in detail in Chapter 3. We define the tactical level crew capacity planning problem with connectivity considerations in pairings, propose two modeling and solution approaches, and present our computational study in Chapter 4. In Chapter 5, we present alternative rostering strategies for an enhanced version of the problem. Finally in Chapter 6, we conclude with a summary and some remarks on future research.

Chapter 2

Literature Review

Crew planning problems at railways have been studied several times for various environments considering particular railway companies, and nation-wide or region-wide systems. For instance, Caprara et al. [4, 5, 6] focus on the Italian case; Freling et al. [11], Morgado and Martins [16], Kroon and Fischetti [13, 14] and Abbink et al. [1, 2] focus on Dutch railways; Vaidyanathan et al. [22] focus on the North American railways; and Şahin and Yüceoğlu [8] focus on Turkish railways. Although the problem environment is different from one system to the other, several features are shared including universally accepted rules as well as company and legislative regulations. For more information on scheduling and rostering problems in railway optimization, including crew scheduling, we refer the interested reader to the extensive review of general scheduling problems in passenger railway optimization Caprara et al. [7]. In addition, Ernst et al. [9] and Kumar et al. [15] survey the literature on crew scheduling applications in railways and airlines, and propose new research directions related to railway crew scheduling problems.

From a mathematical modeling point of view, the crew planning and associated problems are generally studied with two mainstream approaches: network flow formulations and set covering/partitioning type formulations. While research on crew planning problems with a network flow formulation is limited, set partitioning and set covering formulations are more frequently used. The network flow formulations usually depend on a spacetime network representation of the problem, and solution methods are often based on relaxations of the problem. Set covering type formulations of the problem lead to developing decomposition-based methods and column generation.

With respect to the problem environment we consider, our study is close to Ernst et al.

[10] and Şahin and Yüceoğlu [8]. Both studies consider minimizing the crew size (i.e. number of crew members) required to operate the trains under the responsibility of the region. Ernst et al. [10] consider the problem in two phases: the planing stage where the number of crew members is determined and the operational (rostering) stage where the connectivity of the crew schedules is maintained. Ernst et al. [10] develop a two-stage solution methodology that does not guarantee optimal solutions. Their heuristic two-stage approach minimizes the number of crew members in the first stage and tries to satisfy the connectivity of rosters in the second stage. Şahin and Yüceoğlu [8] study the planning stage and represent the operational stage problem as the tactical level counterpart of the planning stage problem. Şahin and Yüceoğlu [8] focus on optimally minimizing the number of crew members required in the region to cover the duties. They develop a network representation of the problem and solve the corresponding network flow problem with a solver.

From a methodological point of view, our study is inspired by Caprara et al. [4, 5, 6]. These studies tackle the operational level crew planning problem in Italian State railways by dividing it into three subproblems: pairing generation, pairing optimization, and rostering optimization. These sequential problems are modeled with set covering/partitioning type formulations and solved with various optimization techniques including column generation algorithm. Chronologically in Caprara et al. [4, 5, 6], improvements are observed on the quality of solutions obtained with the use of Lagrangian relaxation techniques and feedback mechanism between pairing optimization and rostering optimization phases. Although the overall cost based objective includes minimizing the number of required workforce, the connectivity of the rosters are not addressed.

In this study, we deal with the tactical crew capacity planning problem where the aim is to determine the minimum required number of crew members. In our setting, the feasibility of crew schedules and the connectivity of rosters are integrated to find a repeatable set of schedules that satisfy the operational rules and regulations. We use simultaneous column-and-row generation algorithm to tackle with the tactical level crew capacity planning problem when formulated as a set covering problem with additional constraints. This novel methodology [17, 18] can be used successfully for solving large scale linear programming (LP) formulations with exponentially many variables. The distinctive feature of these set covering type formulations is an additional set of linking constraints. These constraints are either too many to be included in the formulation directly, or the full set of linking constraints can only be identified if all variables are generated explicitly.

Chapter 3

Railway Crew Capacity Planning

Railway crew capacity planning problem (RCCP) determines the minimum number of crew members required to cover all duties in the region. Each crew-base region of TCDD has a central *home station* which is responsible for providing and managing the crew resources to operate a predetermined set of trains. Also for each crew-base region, there is a predetermined set of stations called *away stations* which are either the home stations of other crew-base regions or *intermediate stations* located between two home stations.

The crew-base is responsible for operating the trains that are starting at the home station and ending at an away station and vice versa. These predetermined list of trains correspond to the duties to be covered by a crew-base region. We focus on crew-base regions with a single home station and multiple away stations and consider a finite-length planning horizon that repeats itself periodically with respect to the schedules of trains. We assume that all crew members are at their home station at the beginning of the planning horizon, and each crew member has to end its duties at the home station at the end of the planning horizon.

RCCP is solved with respect to different rules and policies governed by the company and the labor unions. Parameters of the rules and policies applied by TCDD are shown in Table 3.1. *Minimum* and *maximum home rest* times limit the duration of a *rest* that a crew member reaching the home station after completing a duty must take. Likewise, a crew member reaching the away station after covering a duty must take an *away rest* which is between a predetermined *minimum* and *maximum away rest* times. Instead of taking an away rest, the crew member reaching at an away station can be transfered to the home station for *deadheading*. Following a short rest defined by *minimum* and *maximum*

Parameter	Value at TCDD (hours)
Minimum home rest time	16
Maximum home rest time	48
Minimum away rest time	8
Maximum away rest time	24
Minimum deadhead start time	4
Maximum deadhead start time	24
Excess duty time	12
On-duty time	1
Off-duty time	1/2
Double manning time	8

Table 3.1: The environmental parameters of the problem at TCDD [8]

deadhead start times, the crew member reaching an away station can be transferred with a train destined to his home station without covering a duty. Alternatively, the crew member can return back to home by covering a second train duty if the total duration of both duties is less than the *excess duty* time.

According to the policies of TCDD, the actual duty duration corresponds to the train duty time extended by *on-duty* and *off-duty* times which are scheduled for filling paperwork and debriefs on the trip. If a train duty takes more than *double manning time*, then to cover this duty, at least two crew members are required instead of one. Lastly, TCDD applies a day-off policy which imposes that every crew member must take a certain number of days off during the finite planning horizon (i.e. a day-off each week). The day-off should be spent at the home station of the crew, and it should include a complete day (from 00:00 to 23:59) and not any 24 hours.

Şahin and Yüceoğlu [8] develop a network representation of the problem and solve a minimum flow problem to determine the minimum crew capacity. We focus on an extension of RCCP but we also revisit their problem. For the sake of flow and clarity, we first introduce the original RCCP problem; next, we summarize the developments in Şahin and Yüceoğlu [8]; we, then, discuss our suggested improvements and a new modeling approach along with a proposed solution method. In Section 3.1, problem definition and the network representation in Şahin and Yüceoğlu [8] is summarized along with the improvements that we suggest on the network representation and the corresponding mathematical model. In Section 3.2, a set covering based formulation of the problem and the overview of the proposed column generation algorithm is given. Finally in Section 3.3, we present the results of our computational study which shows the improvements on the network representation and compares the efficiency and effectiveness of the two solution methods devised for the RCCP.

3.1 Network Flow Formulation for RCCP

Şahin and Yüceoğlu [8] model the tactical level crew capacity planning problem with a space-time network representation where different policies and practical considerations that also include the ones as applied in TCDD are considered. The nodes represent events in the space-time network and have two attributes: space and time, respectively representing the place (i.e. the station) and the time of the event. Each duty is defined with two nodes (on-duty and tie-up) that represent the beginning and the end of that duty in each layer of the network. The arcs in the space-time network represent the engagement of crew with the activity represented by the arc. According to the type of events represented by the nodes and corresponding activities, various type of arcs are included:

- A source arc emanates from the source node and enters an on-duty node at home station. The flow on this arc represents the beginning of a crew schedule.
- A sink arc emanates from a tie-up node at home station and enters the sink node. A flow on this arc represents the end of a crew schedule.
- A duty arc emanates from the on-duty node of a duty and enters its corresponding tie-up node. The flow on this arc represents the crew member engaging with the duty.
- A rest arc goes from a tie-up node to an on-duty node representing the rest activity between the two duties as dictated.
- A deadhead arc goes from a tie-up node at an away station to a tie-up node at home station representing deadhead activity.
- A direct connection arc goes from a tie-up node at an away station to an on-duty node at the same station, representing the coverage of an excess duty.

Figure 3.1 is an illustration of the space-time network. On the space-time network, a source-sink ($s - t$) path corresponds to a crew schedule representing a sequence of events

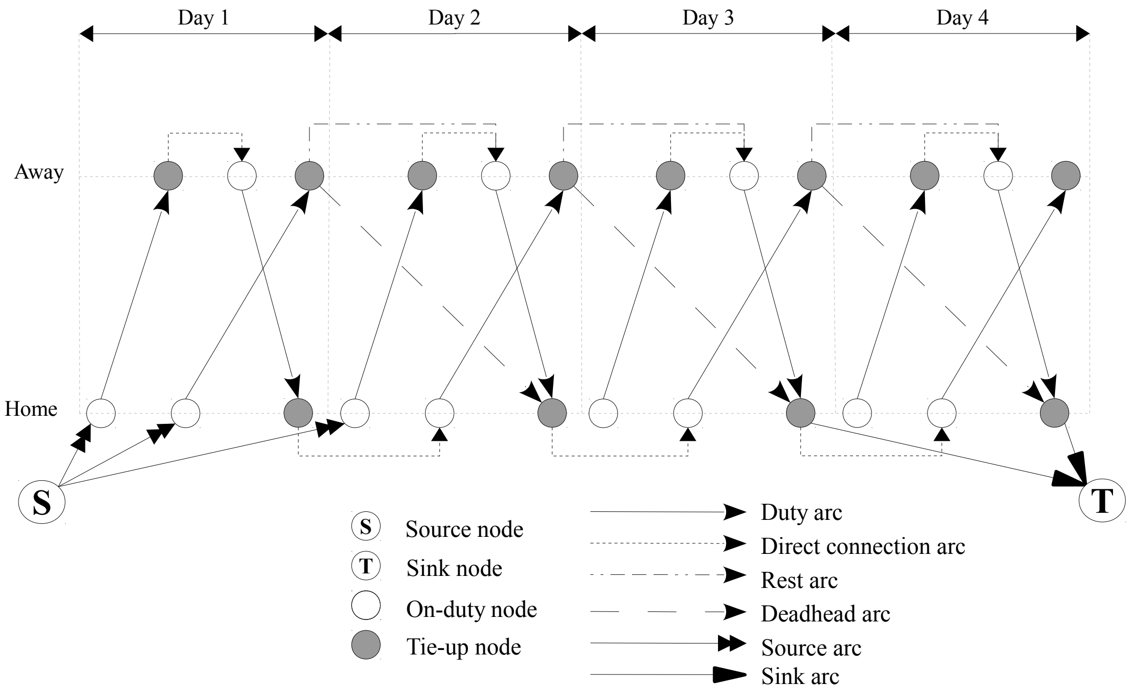

Figure 3.1: An illustration of the space-time network [8]

(i.e. duties) and activities (i.e. rest, deadhead, direct connection) the crew is engaged with during a finite planning horizon whose beginning and end is marked with the source node and sink node, respectively. Since the space-time network is constructed according to the rules and regulations, the flow on an $s - t$ path corresponds to a feasible schedule. Beginning of a schedule is marked with a flow on a source arc which is connected to an on-duty node (i.e. first duty in the schedule). After covering a set of on-duty nodes and tie-up nodes (i.e. set of duties) and using different types of arcs between them, the flow reaches the sink node. The last tie-up node in the path before reaching the sink node indicates the end of the crew schedule.

One challenging requirement of TCDD for crew schedules is the day-off policy. In order to represent the day-off requirement, Şahin and Yüceoğlu [8] enhance their network representation into a multi-layer network. For a generalized problem with g days-off, the network consists of $g + 1$ layers from $Layer_0$ to $Layer_g$, with identical nodes. Day-off activities are represented with day-off arcs between a tie-up node at $Layer_h$ and on-duty node at $Layer_{h+1}$ $(0 \leq h \leq g - 1)$ at home station for a one-day-off requirement. In their multi-layered network representation, Şahin and Yüceoğlu [8] preserve the feasibility of $s - t$ paths so that a flow on any $s - t$ path corresponds to a feasible crew schedule in which the day-off rules are also honored. We refer readers to Şahin and Yüceoğlu [8] for the details of the space-time network representation.

3.1.1 Improvements in the Network Representation

Şahin and Yüceoğlu [8] have discussed two special cases regarding the availability of crew to cover at the beginning and at the end of the planning horizon; they propose to modify the construction of a special set of duty arcs as well as additional sink arcs.

It is assumed that the crew is located at the home station at the beginning of the planning horizon and she has to return to the home station at the end. However, this assumption may not hold for two cases, and representing these duties in the network requires special handling according to Şahin and Yüceoğlu [8]. In the first case, the challenge is to represent the duties at an away station that are too close to the beginning of the planning horizon as there may not be any incoming arcs from the source node to the on-duty nodes of such duties. An on-duty, *away early on-duty*, is moved to the end of the planning horizon with an updated time attribute without losing the generality. The second case is concerned with the duties ending at an away station that are too close to the end of the planning horizon; in this case, the crew member performing such a duty cannot return back to the home station. *Pseudo sink arcs* are introduced to connect these duties to the sink node supposing that the crew member comes back home through deadheading or performing a duty at the beginning of the next planning horizon.

Şahin and Yüceoğlu [8] have not discussed any particular order for creating the *away early on-duty* nodes and the *pseudo sink arcs*. In this respect, we observe the following:

- If the *away early on-duty* nodes are moved at the end of the planning horizon before introducing *pseudo sink arcs* from tie-up nodes at an away station, there might exist other types of arcs between these two nodes since the time attributes of the *away early on-duty* nodes will be updated close to the very end of the planning horizon. If those tie-up nodes can be connected to these *away early on-duty* nodes, that would allow a crew member who engages with those duties to come back to home station at the end of the planning horizon; this eliminates the need for introducing pseudo sink arcs. By processing *away early on-duty* nodes before processing *pseudo sink arcs*, we introduce less number of *pseudo sink arcs* than Şahin and Yüceoğlu [8] when constructing the network with same input parameters while we also create more opportunities for the same schedule to cover more duties.

Our second modification is concerned with a more practical issue. In practice, TCDD does not favor the schedules for crew members that begin or end in midweek days. This is also due to the fact that a crew member might use too long of a rest period or too many days as the weekly day-off period. In the original network representation, such cases are not prohibited since source arcs are from the source node to every on-duty node at home station and sink arcs are from every tie-up node at home station to the sink node. With our modification, we remove the "unnecessary" source and sink arcs from the network regarding these practical considerations as follows:

- Source arcs determine the first duty of a schedule; source arcs are created from the source node to an on-duty node if the time attribute of that on-duty node is within the maximum home rest period that starts at the beginning of the planning horizon. In a multi-layer network (where the day-off requirement is considered) since the first layer represents the duties before taking a day-off, same rule should be applied.
- Sink arcs determine the last duty of a schedule; a sink arc can be created from the tie-up node to a sink node if time attribute of that tie-up node node is within the maximum home rest period that ends at the end of the planning horizon. In a multi-layer network (where the day-off requirement is considered) since the last layer represents the duties after taking all necessary day-offs, same procedure should be applied.

Figure 3.2 is an illustration of the space-time network with source and sink arcs.

3.1.2 Mathematical Model

Based on the improved layered network representation of the problem, we revisit the corresponding mathematical programming formulation in [8]. Notation for the space-time network and mathematical formulation of the minimum flow problem is given in Table 3.2. $\bar{x}_a$ denotes the amount of flow on arc $a \in A$ and $\bar{x}_a^l$ is the amount of flow on copy of duty arc $a \in A_d$ on layer $l \in L$. Then the integer programming formulation of the corresponding network flow problem is as follows:

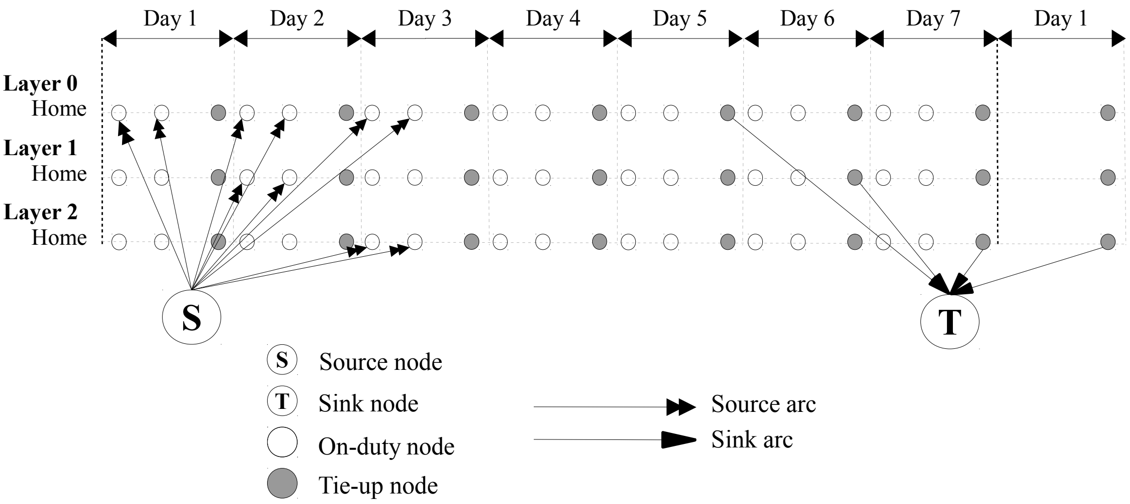

Figure 3.2: An illustration of the source and sink arcs

N	set of nodes in the network
L	set of layers in the network
s	source node of the network
t	sink node of the network
A	set of all arcs in the network
A_d	set of duty arcs in the network
A_s	set of source arcs
A_t	set of sink arcs
A_{n^+}	set of outgoing arcs at node n
A_{n^-}	set of incoming arcs at node n
c_a	required number of crew members to cover the duty represented by arc a

Table 3.2: Notation for the space-time network and minimum flow problem [8]

$$\text{minimize} \quad \sum_{a \in A_s} \bar{x}_a \tag{3.1}$$

$$\text{subject to} \quad \sum_{a \in A_s} \bar{x}_a = \sum_{a \in A_t} \bar{x}_a, \tag{3.2}$$

$$\sum_{a \in A_{n^+}} \bar{x}_a = \sum_{a \in A_{n^-}} \bar{x}_a, \qquad \forall n \in N \setminus \{s, t\}, \tag{3.3}$$

$$\sum_{l \in L} \bar{x}^l_a \geq c_a, \qquad \forall a \in A_d, \tag{3.4}$$

$$\bar{x}_a \in \mathbb{Z}_+, \qquad \forall a \in A. \tag{3.5}$$

The objective function (3.1) minimizes the total amount of flow leaving the source node, which corresponds to minimizing the number of crew members required to operate the given duties in the region. Constraints (3.2) and (3.3) are flow-balance constraints for the source, the sink, and other nodes. The coverage constraint (3.4) guarantees that the total amount of flow on all copies of a duty arc is at least as much as the required amount, c_a, to ensure that duties are covered by the required number of crew members. Constraints (3.5) represent the domain of the decision variable.

Şahin and Yüceoğlu [8] solve the minimum flow problem (3.1)-(3.5) to determine the minimum crew capacity.

3.2 A Column Generation Algorithm for RCCP

An alternative approach is to formulate the tactical RCCP problem as a pure set covering problem and solve it with a column generation algorithm, where a feasible crew schedule corresponds to a column in the formulation.

We let the set of duties in the finite planning horizon be denoted by I and the set of feasible schedules covering the horizon be denoted by J. A binary parameter a_{ij} indicates that duty $i \in I$ is included in schedule $j \in J$ when $a_{ij} = 1$. Minimum required crew member to cover duty i is represented by c_i. The decision variable x_j is defined as

$$x_j = \begin{cases} 1, & \text{if schedule } j \text{ is selected/included in solution;} \\ 0, & \text{otherwise.} \end{cases}$$

Then, the mathematical programming formulation of the tactical crew capacity planning problem studied in Şahin and Yüceoğlu [8] as a set-covering problem is as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in J} x_j & & (3.6) \\ \text{subject to} \quad & \sum_{j \in J} a_{ij} x_j \geq c_i, & i \in I, \quad & (3.7) \\ & x_j \in \{0, 1\}, & j \in J. \quad & (3.8) \end{aligned}$$

Objective function (3.6) minimizes the number of selected schedules (i.e. number of crew members as each schedule may be associated with one crew). Constraints (3.7) ensure that each duty is covered by at least the required number of crew members. Constraints (3.8) represent the domain of the decision variable.

A traditional set covering problem can be solved with the column generation algorithm. Initially, the problem with only a small subset of the decision variables is solved; this problem is called the restricted master problem (RMP). Using the dual information of the optimal solution to RMP, the pricing sub-problem (PSP) is employed to generate a new decision variable (that does not yet exist in RMP) that is expected to improve the solution when added to RMP. RMP is expanded iteratively while PSP is solved with updated dual information at each iteration.

In the column generation algorithm, the set of constraints in RMP is fixed, and complete dual information is supplied from RMP to PSP. In each iteration, PSP generates a new column (i.e. decision variable) to be added to RMP, by computing the reduced cost of the column using the retrieved dual information. If the computed reduced cost of this column is negative, we add the column to RMP. Otherwise, the solution of RMP cannot be improved; optimal solution to the LP relaxation of the original problem is found.

In order to mathematically describe PSP that generates a new schedule for RMP of problem (3.6)-(3.8), let J_c be the set of existing schedules in RMP, and the set of remaining feasible schedules be $\overline{J_c}$. Let u_i be the dual variables corresponding to, respectively, constraints $i \in I$ in (3.7) for only $j \in J_c$. The dual of RMP can then be formulated as

$$\begin{aligned} &\text{maximize} && \sum_{i \in I} c_i u_i && && (3.9) \\ &\text{subject to} && \sum_{i \in I} a_{ij} u_i \leq 1, && j \in J_c, && (3.10) \\ & && u_i \geq 0, && i \in I. && (3.11) \end{aligned}$$

PSP is to find a schedule $j \in \overline{J_c}$ that has a negative reduced cost (i.e. violates the corresponding dual constraint in (3.10)). If we search for the schedule that is expected to make the most improvement in the objective function value (3.6) of RMP, the pricing problem becomes

$$j^* = \arg\min_{j \in \overline{J_c}} \left\{ 1 - \sum_{i \in I} a_{ij} u_i \right\}. \tag{3.12}$$

The new schedule, represented by x_{j^*}, is expected to enter the optimal basis of RMP (i.e. the new schedule is included in the optimal solution) at the next iteration of the column generation algorithm. Consequently, if $1 - \sum_{i \in I} a_{ij^*} u_i < 0$, x_{j^*} is added to RMP and the correct termination condition for the column generation algorithm that optimally solves the LP relaxation of (3.6)-(3.8) is formalized with the following theorem.

Theorem 1 *Let $u_i, i \in I$ be the optimal dual solution corresponding to the optimal basis $\mathcal{B}$ of the current RMP. The primal solution associated with $\mathcal{B}$ is optimal for the LP relaxation of (3.6)-(3.8) if*

$$1 - \sum_{i \in I} a_{ij} u_i \geq 0 \tag{3.13}$$

for every $j \in \overline{J_c}$.

The LP relaxation of (3.6)-(3.8) can be solved to optimality with a column generation algorithm when the termination condition in Theorem 1 is reached. We develop a column generation algorithm to solve the LP relaxation of (3.6)-(3.8) where both the initial solution procedure and computationally efficient solution method for PSP are developed based on the network representation of the original problem as discussed in Section 3.1.1.

3.2.1 Initial Solution Procedure

To start the column generation algorithm, a feasible solution to (3.6)-(3.8) is required to construct the initial RMP. A trivial procedure would be to produce schedules each of which cover a single train duty. Another approach would be as follows: since any $s - t$ path on the network representation corresponds to a feasible schedule (i.e. column), initial RMP is constructed by finding a set of paths on the network. A feasible solution to (3.6)-(3.8) requires that all the duties are covered (i.e. corresponding duty arcs in the network are included in paths). And the overall objective is to attain the minimum number of schedules (i.e. $s - t$ paths). In order to find such a set of schedules, we develop a greedy search algorithm that finds a path at every iteration.

Our greedy algorithm is developed for the network representation by setting the parameters of the network accordingly and updating the parameters at each iteration. It is based on finding $s-t$ paths (i.e. schedules) that cover the most number of uncovered duties. In each iteration, to find an $s-t$ path that covers the most number of the uncovered duties, a shortest path problem is solved on the network where the (arc) length of a duty arc that is uncovered is set to a negative value. Initially, all the duty arc lengths are set to $-c_i$ (number of crew members required to cover the corresponding duty). After finding an $s-t$ path, the length of the arcs on the path is increased by one; and, eventually they are set to 0 in order to exclude the covered arcs in the following iterations. In each iteration, the optimal path would correspond to a column that covers at least one uncovered duty. We terminate when the optimal path of the shortest path problem on the network does not contain any uncovered duty; this indicates that no new path would cover an uncovered duty, i.e. no uncovered duty exists. We construct the initial RMP with the columns corresponding to the paths found until the terminating iteration.

3.2.2 Solution Method for PSP

PSP as stated in (3.12) can be solved by using the network representation of the problem; it can be formulated to find a path on the network where the arc costs are arranged in such a way that the total length of an $s-t$ path represents the reduced cost of the corresponding column (crew schedule). Recall that a solution to PSP at any iteration of the column generation algorithm is found using the dual values obtained from the optimal solution of the most recent RMP. In order to achieve this, at each iteration of the algorithm arc lengths are arranged as follows: the dual price of each duty constraint (3.7) ($-u_i$) is introduced as the arc length of the corresponding duty arc (in a multi-layer network all copies of the same duty arc are updated). In this respect, the schedule with the most negative reduced cost should correspond to the shortest $s-t$ path on the network.

A solution to the shortest $s-t$ path problem on the corresponding network corresponds to a schedule with a total cost of $-\sum_{i\in I} a_{ij^*}u_i$ where j^* represents the schedule corresponding to the shortest $s-t$ path. Whenever $1-\sum_{i\in I} a_{ij^*}u_i < 0$ (i.e. negative reduced cost) that schedule (column) is added to RMP; otherwise, the termination criterion is satisfied as no schedule with negative reduced cost is found.

3.2.3 Procedure for the Column Generation Algorithm

The column generation algorithm that solves the LP relaxation of (3.6)-(3.8) has three main components: initial solution, RMP and PSP. The iterative mechanism of the algorithm is depicted in Figure 3.3. The initial solution procedure corresponds to the algorithm (presented in Section 3.2.1) on the network representation of the problem. Then, with the schedules in the initial solution, RMP is solved to optimality with an LP solver. Resulting optimal dual values of the constraints are used in solving PSPs as explained in Section 3.2.2. In order to solve PSP, the arc lengths are updated accordingly with the dual values of the constraints; then, a shortest path problem is solved on this network, and the schedule with minimum reduced cost is constructed from the resulting $s - t$ path. If the reduced cost of the schedule is negative, it is added to RMP. The RMP is solved to optimality iteratively repeating the same procedure until no schedule with negative reduced cost is found.

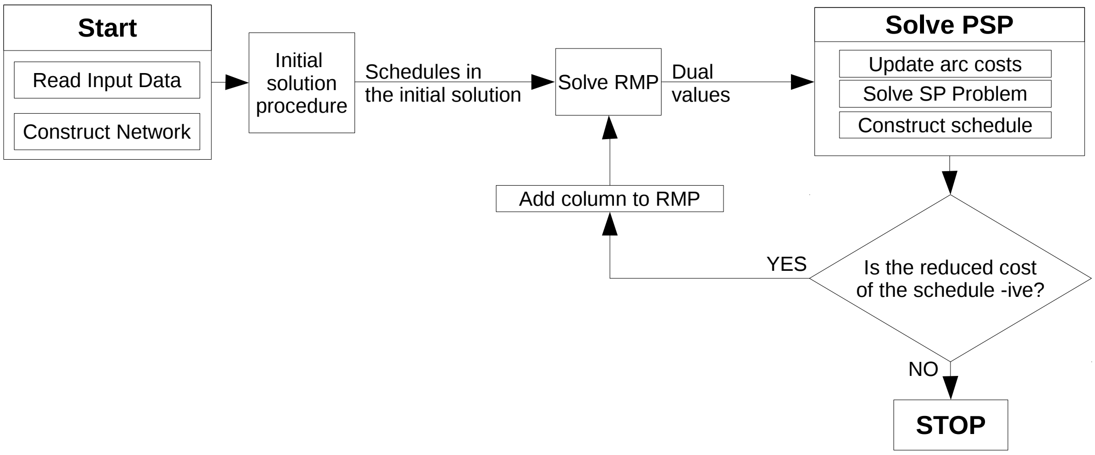

Figure 3.3: The flow of the column generation algorithm

The optimal solution to LP relaxation of (3.6)-(3.8) may not be integer feasible; we use a heuristic idea to find an integer feasible solution with the information obtained from the terminating iteration of the column generation algorithm. Column generation algorithm terminates when the optimal solution to the LP relaxation of (3.6)-(3.8) is found. To obtain an integer feasible solution, we solve (3.6)-(3.8) as an integer programming problem with the schedules generated by the column generation algorithm until the termination. Since the columns generated by the algorithm represents a limited solution space, the optimal solution of (3.6)-(3.8) with the columns in terminating RMP generates an integer feasible upper-bounding solution to the problem.

3.3 Computational Study

The computational study is performed on a data set that is representative of three different crew regions in the Turkish State Railways system, namely Ankara, Haydarpasa and Eskisehir. We implement the network flow formulation and the column generation algorithm in C++ using ILOG Optimization Studio 12.2 as the LP and IP solver on a PC with Intel Core i7 @3.20 GHz CPU and 24GB RAM.

We present results for three TCDD crew regions where we create problem instances with a planning horizon of one week and two weeks and different day-off requirements. Complete results are reported in Table 3.3 where OPT column corresponds to the results of Şahin and Yüceoğlu [8]. LB, OPT(*) and Time1 columns correspond to the optimal objective function value of the LP relaxation, optimal integer objective function value of (3.1)-(3.5) and computational time required in seconds, respectively, for the network flow formulation. Number of iterations required by the column generation algorithm is reported in the Itr and the LP optimal solution value is reported in the LP-LB column. Time2 indicates total computational time (in seconds) required by the column generation algorithm. IP-UB indicates the number of feasible schedules found by the initial solution procedure, thus integer upper bound. IP-FS indicates the integer solution found by the heuristic idea, solving (3.1)-(3.5) with columns in the terminating iteration of column generation algorithm, and finally Time3 indicates time required by CPLEX solver (bounded by one hour).

With the computational study, we first want to observe the impact of the improvements made on the network representation. This has two dimensions:

- possible improvement in the solution quality
- reduction in computational effort

For this purpose, we solved the problem (3.1)-(3.5) with the improvements by applying the network flow formulation of Şahin and Yüceoğlu [8]. We note that in Şahin and Yüceoğlu [8] the existence of so-called “unnecessary” source arcs and sink arcs does not impact the solution quality. As the objective function minimizes the required number of crew members, the optimal solution implicitly prohibits such unnecessary home rest periods. Yet, we should also note that removing these arcs decreases the size of the network and may impact the computational effort to solve problems using this network representation. The results indicate the decrease in the number of required crew members (i.e. decrease in the objective function value). As expected, we observed % 2-5 improvement on average in Ankara and Haydarpasa data sets, but the most striking ones are in Eskisehir data set where improvements vary between % 15-25. For example, with a planning horizon of 1-week and 2-day-off requirement, we have found the optimal number of crew members as 76 which is 18 less than that reported in Şahin and Yüceoğlu [8].

We also want to understand the quality of the solutions obtained with the column generation algorithm and the heuristic idea to obtain integer feasible solutions. Although we see that good integer feasible solutions are obtained by our heuristic procedure, the column generation performs poorly by not only solution quality but also required computational time when compared against the network flow formulation. We observe that integer feasible solutions obtained by our heuristic idea are % 5-15 worse on average, indicating a mediocre performance. On the other hand, in all of our data sets, computational effort required by the network flow formulation to find the optimal integer solutions is less than the computational effort required by the column generation algorithm to find the optimal solution to the LP relaxation to the problem. Furthermore, in more than half of the cases, CPLEX is not able to solve the IP problem to optimality in one hour of computational time (which is considered as a reasonable time limit and indicated in column Time3). In conclusion, our computational study shows that the network flow formulation is a preferred solution approach to the RCCP problem under these circumstances.

				Network Flow			Column Generation					
Region	Weeks	Days-Off	OPT	LB	OPT*	Time1(sec)	Itr	LP-LB	Time2(sec)	IP-UB	IP-FS	Time3(sec)
Ankara	1	0	-	50.00	50	1	239	50.00	5	56	55	210
	1	1	57	54.80	55	1	285	54.80	6	61	60	3600
	1	2	67	65.20	66	1	280	65.20	5	74	69	150
	2	0	-	50.00	50	1	790	50.00	151	56	56	3600
	2	1	54	52.00	52	8	1058	52.00	264	61	61	3600
	2	2	58	56.33	57	49	934	56.33	190	65	65	3600
Haydarpasa	1	0	-	38.00	38	1	376	38.00	15	41	41	3
	1	1	46	42.66	43	6	356	42.66	13	46	46	4
	1	2	53	51.20	52	8	325	51.20	10	57	57	665
	2	0	-	38.00	38	1	1033	38.00	584	41	41	68
	2	1	46	40.15	41	52	1024	40.15	586	46	46	3600
	2	2	46	43.50	44	77	1010	43.50	535	49	49	3600
Eskisehir	1	0	-	65.00	65	1	361	65.00	12	69	69	134
	1	1	79	66.20	67	2	509	66.20	22	72	72	3600
	1	2	94	75.33	76	3	401	75.33	14	80	80	2601
	2	0	-	65.00	65	1	1233	65.00	637	70	70	3600
	2	1	76	65.00	65	35	1369	65.00	813	70	70	3600
	2	2	81	67.07	68	112	1654	67.07	1186	73	73	3600

Table 3.3: Results for Railway Crew Capacity Planning

Chapter 4

RCCP with Connectivity Considerations in Pairings

We deal with an extension of RCCP where the periodic repeatability of crew schedules is considered as well. This issue arises from a particular deficient supposition in the railway crew planning that crew schedules can be repeated over the periodic recurrence of the planning horizon. When it is left unconsidered, the continuity of schedules over the recurrence of the schedule period may lead to crew schedules that are not implementable in practice. We extend RCCP to find a set of feasible crew schedules that can be connected to other schedules from one planning period to the other; thus, we name our new RCCP problem with connectivity considerations in pairings as RCCP-C.

For this extension of the problem, we follow the footsteps of the solution methods developed for the original RCCP: (i) a set-covering type formulation and a simultaneous column-and-row generation algorithm, and (ii) a network representation of the problem and a corresponding network flow formulation. In Section 4.1, we formulate the problem and introduce the column-and-row generation algorithm; details of the algorithm, an initial solution procedure for RMP, the proposed solution method for the pricing subproblem and the flow of the column-and-row generation algorithm is discussed. We discuss the network flow problem and the corresponding mathematical model in Section 4.2; we give details of the network representation and the corresponding integer programming formulation. Finally, in Section 4.3, we show the results of our computational study that compares the two solution methods devised for the RCCP-C.

4.1 A Simultaneous Column-and-Row Generation Algorithm for RCCP-C

To formulate this version of the problem, the set covering problem (3.6)-(3.8) is enriched with an additional parameter, a new decision variable and constraints that represent the connectivity relationship among the schedules.

Let the new parameter denoted by $l_{jj'}$ indicate the connectivity relationship between schedules as follows:

$$l_{jj'} = \begin{cases} 1, & \text{if schedule } j \text{ can be connected to schedule } j'; \\ 0, & \text{otherwise.} \end{cases}$$

Decision variable $y_{jj'}$ is defined as:

$$y_{jj'} = \begin{cases} 1, & \text{if schedule } j \text{ is connected to schedule } j'; \\ 0, & \text{otherwise.} \end{cases}$$

Then, the mathematical programming formulation of the tactical crew capacity planning problem with connectivity considerations in pairings (RCCP-C) becomes:

$$\begin{array}{llll} [\text{RCCP-C}]_{sc} & \text{minimize} & (3.6) & \\ & \text{subject to} & (3.7) - (3.8), & \\ & & \sum_{j' \in J} l_{jj'}\, y_{jj'} - x_j = 0, \qquad j \in J, & (4.1) \\ & & \sum_{j' \in J} l_{j'j}\, y_{j'j} - x_j = 0, \qquad j \in J, & (4.2) \\ & & y_{jj'} \in \{0, 1\}, \qquad j, j' \in J. & (4.3) \end{array}$$

Constraints (4.1) guarantee that each selected schedule follows (is connected to) another selected schedule in the solution. Likewise, constraints (4.2) guarantee that a selected schedule is being followed by (connects to) another selected schedule in the solution. We may call these two set of constraints as linking constraints in the general context of set-covering type mathematical programming formulations, and as connectivity constraints in the case of RCCP. Constraints (4.3) represent the binary nature of the new decision variables.

Solving [RCCP-C]$_{sc}$ directly is not practical due to the size of set J, which contains all possible schedules. However the formulation [RCCP-C]$_{sc}$ belongs to the set of column-dependent-rows (CDR) problems [17, 18]. In this respect, a simultaneous column-and-row generation algorithm may be employed to solve the LP relaxation.

In problem [RCCP-C]$_{sc}$, existence of connectivity constraints (4.1) and (4.2) (i.e. rows of the formulation) depends on the existence of columns (represented by x_j) in the problem; addition of a new column induces new linking constraints to be added to RMP. However, when this new variable is to be generated by PSP, the dual information associated with the constraints that are linking the new decision variable(s) to the existing variables is missing from the solution of the most recent RMP. Therefore, the reduced cost of the new column may not be accurately computed with PSP. In order to correctly compute the reduced cost of a column (with PSP), we need to know the associated dual variables of these linking constraints (rows) a priori. In this respect, a traditional column generation algorithm would not suffice to solve this problem due to this missing dual information associated with the missing rows that do not yet exist. To overcome this challenge, we develop a column-and-row generation algorithm that simultaneously generates feasible crew schedules (i.e. decision variables/columns) and associated connectivity constraints (i.e. linking constraints/rows).

As in the column generation algorithm, the simultaneous column-and-row generation (CRG) algorithm starts with an RMP that includes only a selected subset of columns (schedules), and then iteratively adds new columns to RMP to improve its objective function value. At each iteration, RMP is solved to optimality and the optimal dual values from RMP are used to solve PSP which allows generating new variables and their associated linking constraints simultaneously by appropriately estimating the dual values of the missing linking constraints.

In order to mathematically describe PSP that generates a new schedule for RMP of problem [RCCP-C]$_{sc}$, let J_c be the set of existing schedules in RMP, and the set of remaining feasible schedules be $\overline{J_c}$. Let u_i, v_j and w_j be the dual variables corresponding to, respectively, constraints $i \in I$ in (3.7) and constraints (4.1) and (4.2) for only $j \in J_c$.

The dual of RMP can then be formulated as

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i\in I} u_i && && (4.4)\\
\text{subject to} \quad & \sum_{i\in I} a_{ij}u_i - v_j - w_j \leq 1, && j \in J_c, && (4.5)\\
& l_{jj'}(v_j + w_{j'}) \leq 0, && j, j' \in J_c, && (4.6)\\
& u_i \geq 0, && i \in I, && (4.7)\\
& v_j, w_j \text{ u.i.s} && j \in J_c. && (4.8)
\end{aligned}
$$

PSP is to find a schedule $j \in \overline{J_c}$ that has a negative reduced cost (i.e. violates the corresponding dual constraint in (4.5)). If we search for the schedule that is expected to make the most improvement in the objective function value (3.6) of RMP, the pricing problem becomes the following two-stage problem to find

$$
j^* = \arg\min_{j\in\overline{J_c}} \left\{ 1 - \sum_{i\in I} a_{ij}u_i + v_j + w_j \right\} \tag{4.9}
$$

$$
\begin{aligned}
\text{minimize} \quad & v_j && (4.10)\\
\text{subject to} \quad & l_{jj'}(v_j + w_{j'}) \leq 0, \qquad j \in \overline{J_c}, j' \in J_c, && (4.11)\\
& \qquad\qquad \text{at least one constraint in (4.11) is tight;} && (4.12)
\end{aligned}
$$

$$
\begin{aligned}
\text{minimize} \quad & w_j && (4.13)\\
\text{subject to} \quad & l_{j'j}(v_{j'} + w_j) \leq 0, \qquad j' \in J_c, j \in \overline{J_c}, && (4.14)\\
& \qquad\qquad \text{at least one constraint in (4.14) is tight.} && (4.15)
\end{aligned}
$$

where problems (4.10)-(4.12) and (4.13)-(4.15) impose the connectivity relations between the new schedule $j \in \overline{J_c}$ and the existing schedules in J_c.

Constraints (4.11) and (4.14) impose that the dual constraints (4.6) are not violated as we try to find the schedule with the largest violation in dual constraint (4.5). The new schedule, represented by x_{j^*}, is expected to enter the optimal basis of RMP (i.e. the new schedule is included in the optimal solution) at the next iteration of the CRG algorithm. We know that when x_{j^*} enters the basis, the connectivity constraints in (4.1) and (4.2) are to be honored with two new basic linking variables $y_{j^*j'} = 1, j' \in J_c$, and $y_{j'j^*} = 1, j' \in J_c$. Optimal solution of (4.10)-(4.12) indicates which $y_{jj'}$ variable is to enter the basis (takes a value of 1) in (4.1), corresponding to a tight constraint in (4.11). This is indeed imposed by the complementary slackness condition associated with constraint (4.12). Similarly, the optimal solution of (4.13)-(4.15) indicates which $y_{j'j}$ variable is to enter the basis (takes a value of 1) in (4.2), corresponding to a tight constraint in (4.14) with respect to the complementary slackness condition associated with constraint (4.15).

We may exploit the solution of PSP in (4.9)-(4.15) as follows:

- For each schedule $j \in \overline{J_c}$ (new candidate column), if the schedule can be connected to the existing schedule $j' \in J_c$, (i.e. $l_{jj'} = 1, j \in \overline{J_c}, j' \in J_c$) the corresponding constraint (4.11) appears as $v_j \leq -w_{j'}$. Therefore, for each $j \in \overline{J_c}$ (new candidate schedule), the solution of (4.10)-(4.12) becomes

$$v_j = \min_{j' \in J_c, l_{jj'}=1} (-w_{j'}) = \max_{j' \in J_c, l_{jj'}=1} (w_{j'}). \tag{4.16}$$

- Likewise, for all the existing schedules $j' \in J_c$ that can connect to the new schedule j (i.e. $l_{j'j} = 1, j' \in J_c, j \in \overline{J_c},$) the corresponding constraint (4.14) appears as $w_j \leq -v_{j'}$ and for each $j \in \overline{J_c}$ (new candidate schedule), the solution of (4.13)-(4.15) becomes

$$w_j = \min_{j' \in J_c, l_{j'j}=1} (-v_{j'}) = \max_{j' \in J_c, l_{j'j}=1} (v_{j'}). \tag{4.17}$$

Then, the two stage problem (4.9)-(4.15) can be reformulated as

$$j^* = \arg\min_{j \in \overline{J_c}} \left\{ 1 - \sum_{i \in I} a_{ij} u_i + \max_{j' \in J_c, l_{jj'}=1} (w_{j'}) + \max_{j' \in J_c, l_{j'j}=1} (v_{j'}) \right\} \tag{4.18}$$

As in the column generation algorithm, the CRG algorithm achieves optimality when the objective function value (4.18) is non-negative (i.e. no column exists with negative reduced cost). Consequently, if

$$1 - \sum_{i \in I} a_{ij^*} u_i + v_{j^*} + w_{j^*} < 0,$$

x_{j^*} is added to RMP and RMP is augmented along with one new linking constraint of type (4.1) and one new linking constraint of type (4.2). Then, the correct termination condition for the CRG algorithm that optimally solves the LP relaxation of [RCCP-C]$_{sc}$ may be formalized with the following theorem.

Theorem 2 *Let $u_i, i \in I, v_j, j \in J_c$, and $w_j, j \in J_c$ be the optimal dual solution corresponding to the optimal basis $\mathcal{B}$ of the current RMP. The primal solution associated with $\mathcal{B}$ is optimal for the LP relaxation of [RCCP-C]$_{sc}$ if*

$$1 - \sum_{i \in I} a_{ij} u_i + \max_{j' \in J_c, l_{jj'}=1} (w_{j'}) + \max_{j' \in J_c, l_{j'j}=1} (v_{j'}) \geq 0 \tag{4.19}$$

for every $j \in \overline{J_c}$.

The proof of this theorem follows from the analysis of the CRG algorithm developed in Muter [17] and Muter et al. [18].

The LP relaxation of [RCCP-C]$_{sc}$ can be solved to optimality with a CRG algorithm when the termination condition in Theorem 2 is reached. From a methodological and computational point of view there are potential obstacles in the CRG algorithm. First, to initialize the CRG algorithm, an initial feasible solution to RMP ([RCCP-C]$_{sc}$) should be known. Second, the two stage PSP (4.18) is solved at each iteration. In order to find a feasible solution to the problem, one could develop a constructive algorithm in order to produce a set of schedules which together cover all the duties in the schedule. A trivial procedure would be to produce schedules each of which cover a single train duty. However, such schedules would be infeasible with respect to rest period restrictions. To find a new schedule that is expected to improve the solution to RMP, one could calculate the reduced costs of all remaining feasible schedules ($\forall j \in J_c$). Yet, this would first of all require knowing all feasible schedules, and it would also be computationally cumbersome to calculate the reduced costs for all of them.

We develop a tailored CRG algorithm to solve the LP relaxation of [RCCP-C]$_{sc}$ where both the initial solution procedure and a computationally efficient solution method for PSP are developed based on the network representation of the original problem discussed in Section 3.1.

4.1.1 Initial Solution Procedure

To start the simultaneous column-and-row generation algorithm, an initial feasible solution to [RCCP-C]$_{sc}$ is needed to construct the initial RMP. A trivial idea is to generate dummy schedules that cover a specific duty which can connect to every other schedule and vice versa. Such schedules are clearly infeasible; thus they would need extra care. Another idea that constructs an initial RMP is an enhanced version of the inital solution procedure given in Section 3.2.1 that uses again the network representation of RCCP. A feasible solution to [RCCP-C]$_{sc}$ requires that all duties are covered (i.e. corresponding duty arcs in the network are included in selected paths) and each schedule (i.e. $s-t$ path) is connected to another schedule (i.e. $s-t$ path), and connected by a schedule (i.e. $s-t$ path). And, the overall objective is to attain the minimum number of schedules (i.e. $s-t$ paths). In order to find such a set of schedules, we develop a greedy search algorithm that is based on the similar idea of finding a path at every iteration.

The algorithm iteratively covers all duty arcs by finding a new $s-t$ path in each iteration as in the initial solution procedure described for the column generation algorithm for RCCP in Section 3.2.1. In order to maintain connectivity of schedules throughout the iterations, we use an ordered list of connected schedules as follows:

- The first schedule on the list does not necessarily follow any other schedule found so far while the new path to be found is expected to connect to the last schedule on the list. In order to achieve this, we modify the length of the source arcs at the beginning of the iteration in such a way that the first duty of the new path to be found is connected to the last duty of the path found in the previous iteration.

- At the end of the iteration, we check if the new path may connect to the first schedule on the list. If this is the case, then the current list of schedules makes a closed circuit of connected schedules, we initiate a new list at the next iteration. Otherwise, we again modify the source arcs at the beginning of the next iteration.

We terminate the algorithm when all duty arcs are covered in some path. Flowchart of the algorithm is given in Figure 4.1. At any iteration, the new path may not cover any previously uncovered duty arc. In this case, we modify this path such that it can connect to the first path in the active list of connected schedules, thus forming a closed circuit of connected schedules.

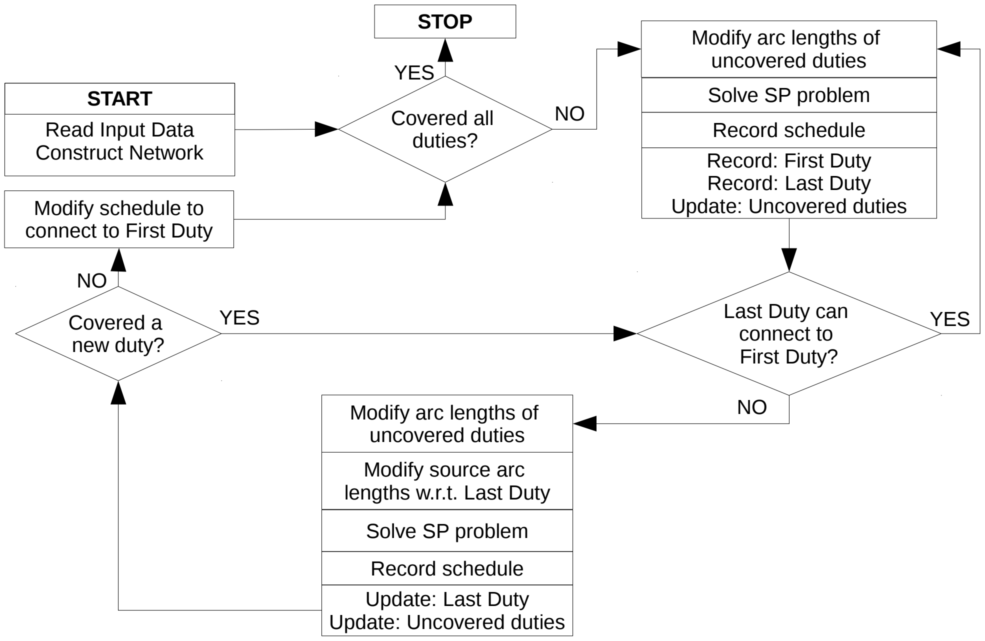

Figure 4.1: The flow of the initial solution procedure

4.1.2 Solution Method for PSP

PSP as stated in (4.18) is seemingly a bi-level optimization problem. However, it can be solved by using the network representation in Section 3.1. PSP can be formulated to find a path on the network representation of the problem where the arc costs are arranged in such a way that the total length of an $s - t$ path represents the reduced cost of the corresponding new column (i.e. crew schedule). A solution to PSP at any iteration of the CRG algorithm can be found using the dual values (u_i, v_j and w_j) obtained from the optimal solution of the most recent RMP. At each iteration of the algorithm arc lengths are arranged as follows:

- the dual price of each duty ($-u_i$) is introduced as the arc length of the corresponding duty arcs (in a multi-layer network all copies of the same duty arc are updated);
- the length of a source arc (s, i) is set to

  $$\max_{j \in J_c^{s(i)}} (v_j)$$

  where $J_c^{s(i)}$ is the set of schedules in the current RMP which can connect to a schedule that starts with duty i, and

- the length of a sink arc (i, t) is set to

$$\max_{j \in J_c^{e(i)}} (w_j)$$

  where $j \in J_c^{e(i)}$ is the set of schedules in the current RMP which can be connected by a schedule that ends with duty i.

Consequently, solving a shortest $s-t$ path problem on this network with modified arc lengths corresponds to finding a path that represents a schedule j^* with a total length of

$$c_{si_{j^*}} - \sum_{i \in I} a_{ij^*} u_i + c_{i'_{j^*}t}$$

where $c_{si_{j^*}}$ is the length of the source arc that enters the on-duty node of the first duty in schedule j^* denoted by i_{j^*} and $c_{i'_{j^*}t}$ is the length of the sink arc that emenates from the tie-up node of the last dusty in schedule j^* denoted by i'_{j^*}. If

$$1 + c_{si_{j^*}} - \sum_{i \in I} a_{ij^*} u_i + c_{i'_{j^*}t} < 0$$

(i.e. negative reduced cost), the corresponding schedule (column) is added to RMP; otherwise, the termination criteria is satisfied as no schedule with negative reduced cost is found.

4.1.3 Procedure for the Column-and-Row Generation Algorithm

CRG algorithm that solves the LP relaxation of [RCCP-C]$_{sc}$ has three main components: initial solution, RMP and PSP. The iterative mechanism of the algorithm is depicted in Figure 4.2. The main difference in the flow of the CRG algorithm when compared to the algorithm in Section 3.2.3 (depicted in Figure 3.3) is the expansion of RMP, which now includes not only the new column but also the associated new connectivity constraints along with the new column. Furthermore, the initial solution procedure uses the algorithm presented in Section 4.1.1 and PSP is solved as explained in Section 4.1.2.

The optimal solution to LP relaxation of [RCCP-C]$_{sc}$ may not be integer feasible; we use the same heuristic idea to find an integer feasible solution with the information obtained from the terminating iteration of the CRG algorithm. Alternatively, optimal solution of [RCCP-C]$_{sc}$ can be obtained by implementing a branch-and-bound algorithm.

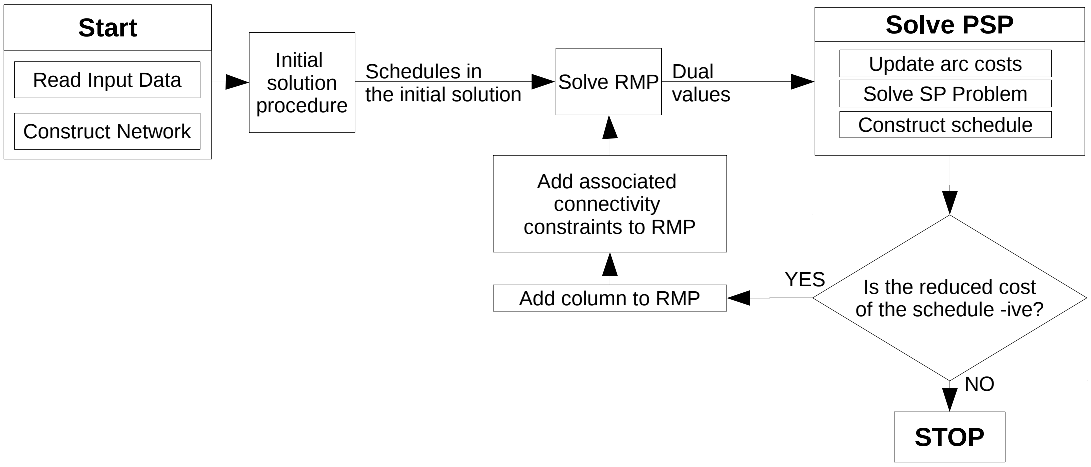
Start
Read Input Data
Construct Network
Initial solution procedure
Schedules in the initial solution
Solve RMP
Dual values
Solve PSP
Update arc costs
Solve SP Problem
Construct schedule
Add associated connectivity constraints to RMP
Add column to RMP
YES
Is the reduced cost of the schedule -ive?
NO
STOP

Figure 4.2: The flow of the column-and-row generation algorithm

Solving the LP relaxation of [RCCP-C]$_{sc}$ with CRG algorithm corresponds to solving only the root node problem of B&B. Within a branch-and-bound-and-price framework, the node problems are solved with CRG algorithm to obtain the optimal solution of problem [RCCP-C]$_{sc}$.

4.2 Network Flow Formulation for RCCP-C

RCCP-C problem requires the continuity of schedules over the recurrence of the schedule period. In order to incorporate the connectivity issue in a network flow formulation of the problem, we first modify the network representation given in Section 3.1. Then, we solve a minimum flow problem over the modified network to solve RCCP-C to optimality.

4.2.1 Network Representation

The challenge in developing a network representation of RCCP-C is to represent the connectivity of the schedules. Considering the network representation of the RCCP, this challenge corresponds to linking the $s - t$ paths (i.e. schedules) such that a flow may circulate on these paths.

To represent this circulating flow, a new arc type is defined: a *connectivity arc* emanates from a tie-up node that may mark the end of a schedule to an on-duty node at home station that may mark the beginning of a schedule. Unlike any other arc in the network, time attribute of the tail node will be later/greater than the time attribute of the head node;

a *connectivity arc* goes from the end of the planning horizon to the beginning, representing the recurrence of the planning horizon. In other words, head node of a *connectivity arc* corresponds to a duty that will be covered in the next recurrence of planning period. The flow on this arc represents the rest period of a crew member that is between her last duty (i.e. tail node) and the first duty (i.e. head node) in the next period. In order to represent the rest activity in between, time difference of the head and tail nodes have to honor resting constraints; and connectivity arcs are created between such nodes.

A unit flow on a *connectivity arc* links two connected schedules of a crew member by marking the end of one and the beginning of the latter. The head node of a *connectivity arc* defines the end of a path that corresponds to the last duty of that schedule, and the head node defines the first duty of a path that corresponds to the first duty of another schedule that is connected to the previous one. Consequently, source and sink nodes that represent the beginning and the end of the planning horizon are no longer needed in the updated network representation. Moreover, source arcs that define the beginning of a schedule and sink arcs that define end of a schedule are naturally excluded as there are no source and sink nodes.

We give an example to demonstrate the use of connectivity arcs; two feasible crew schedules and modifications of a single day-off requirement problem on a two-layer network representation is given in Figure 4.3. In part (a), two distinct $s - t$ paths (i.e. crew schedules) are marked with bold arcs. One of the schedules starts with a source arc (S,15) and ends with a sink arc (20,T), the other starts with a source arc (S,2) and ends with a sink arc (13,T). The second network illustrates the same case where connectivity arcs (13,15) and (20,2) represents the connection between the two schedules from one period to the next. Crew following these schedules continue working without violating rules and regulations through the end of the first period into the second.

4.2.2 Mathematical Model

The objective of RCCP-C is to minimize the number of required crew members to cover all the duties while maintaining the continuity of schedules over the recurrence of the planning period. In the modified network, total flow on connectivity arcs corresponds to number of schedules linked with each other (i.e. number of crew members), thus corresponding mathematical programming formulation aims to minimize the total flow

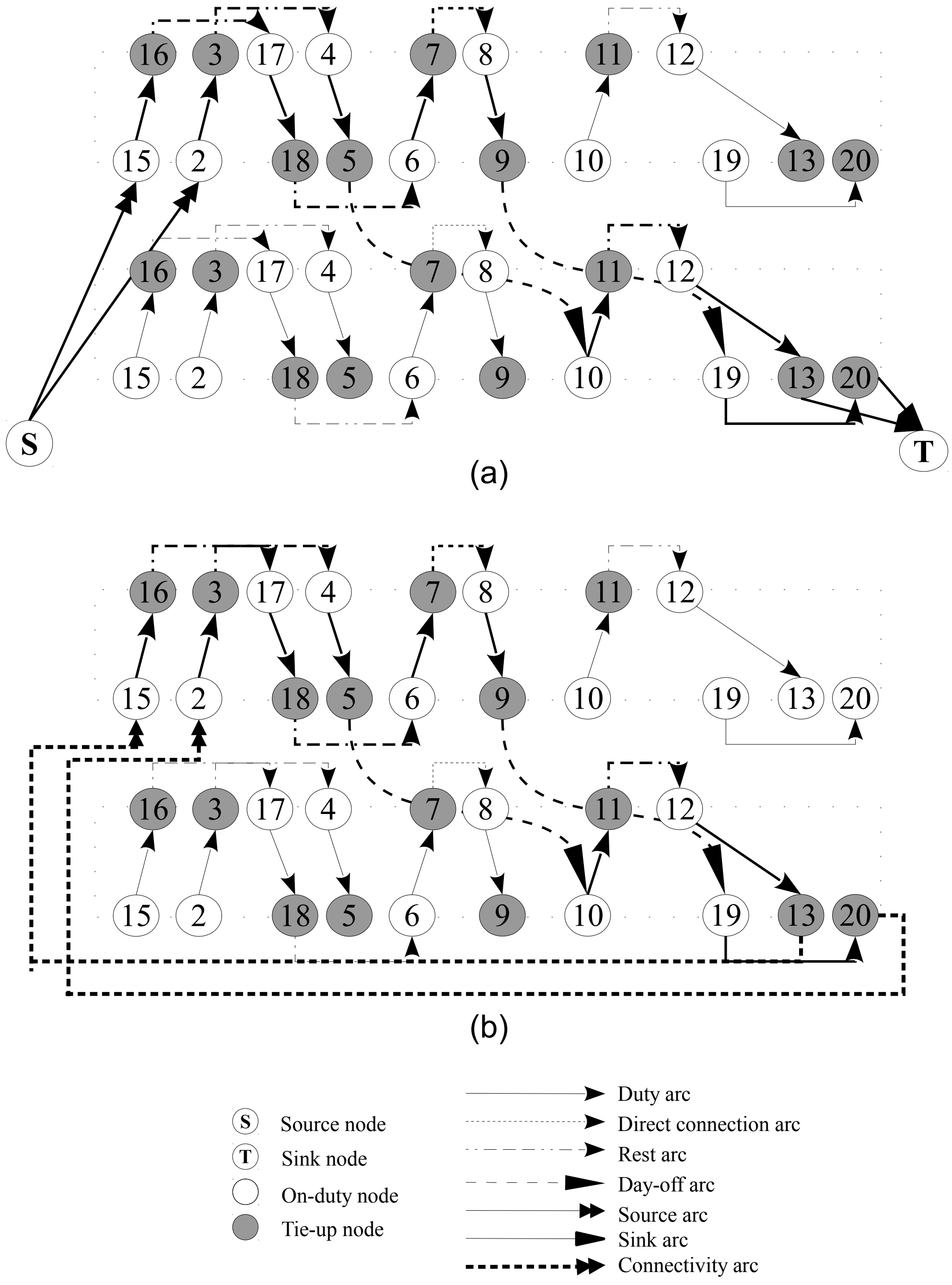

Figure 4.3: An illustration of two sample schedules

on connectivity arcs.

In addition to the notation described in Table 3.2, A_c represents the set of connectivity arcs. The integer programming formulation of the corresponding network flow problem is as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{a \in A_c} \bar{x}_a && && (4.20)\\
\text{subject to} \quad & \sum_{a \in A_{n+}} \bar{x}_a = \sum_{a \in A_{n-}} \bar{x}_a, && \forall n \in N, && (4.21)\\
& \sum_{l \in L} \bar{x}_a^l \geq c_a, && \forall a \in A_d, && (4.22)\\
& \bar{x}_a \in \mathbb{Z}_+, && \forall a \in A. && (4.23)
\end{aligned}
$$

The objective function (4.20) minimizes the total amount of flow on connectivity arcs, which corresponds to minimizing the number of crew members required to operate the given duties in the region. Constraint (4.21) is the flow-balance constraint. The coverage constraint (4.22) guarantees that the total amount of flow on all copies of a duty arc is at least as much as the required amount, c_a, to ensure that duties are covered by the required number of crew members. Constraints (4.23) represent the domain of the decision variable.

4.3 Computational Study

We perform a computational study in order to empirically observe both the efficiency and effectiveness of our proposed solution methods. The computational study is performed on the same data set as in Chapter 3. We implement the CRG algorithm and the network flow formulation in C++ using ILOG Optimization Studio 12.2 as the LP and IP solver on a PC with Intel Core i7 @3.20 GHz CPU and 24GB RAM.

With the computational study, we want to first understand the quality of the solutions obtained with our CRG algorithm and the heuristic idea to obtain integer feasible solutions. For this purpose, we present results for three TCDD crew regions where we create problem instances with a planning horizon of one week and two weeks and different day-off requirements.

In Table 4.1, we present results for two versions of the CRG algorithm based on how the initial RMP is constructed. In CRG-DC, initial RMP is constructed by dummy columns that cover only a single duty and can connect to every other column and vice versa. On the other hand, in CRG-AC initial RMP is constructed by the algorithm given in Section 4.1.1. CRG-DC-IP and CRG-AC-IP corresponds to the heuristic procedure for generating integer feasible solutions. In Table 4.1, LP-LB corresponds to the objective function value of the optimal solution of the LP relaxation of $[\text{RCCP-C}]_{sc}$ obtained with our CRG algorithm. ”Itr” shows the number of iterations/schedules generated until termination. IP-UB indicates the number of schedules generated by the initial solution procedure. With the existing columns in the terminating RMP, we solve the problem as an IP to optimality. IP-FS shows the objective function value of the IP problem solution where Time indicates the time (in seconds) required by CPLEX solver to solve the IP problem (bounded by one hour). We obtain the following results:

- Neither versions of the algorithm are capable of solving the LP relaxation to optimality within reasonable computational time (3600 seconds). We have faced both convergence and stalling problems with the CRG algorithm which we believe that is due to the unary column cost and high degeneracy in our data sets.

- When DC version of the algorithm is used, CPLEX is not able to solve the IP problem. It shows that a good initial solution significantly improves both solution quality and computational time required to solve the problem.

			CRG-DC			CRG-DC-IP			CRG-AC			CRG-AC-IP		
Region	Weeks	Days-Off	Itr	LP-LB	Time	IP-UB	IP-FS	Time	Itr	LP-LB	Time	IP-UB	IP-FS	Time
Ankara	1	0	569	63.33	3600	322	67	3600	1577	54.52	3600	57	57	2783
	1	1	541	74.73	3600	322	76	2380	1621	64.01	3600	67	65	53
	1	2	506	85.16	3600	322	87	3600	1567	78.14	3600	83	80	82
	2	0	235	66.85	3600	644	89	3600	1315	55.86	3600	56	56	1
	2	1	248	81.58	3600	644	105	3600	1329	60.66	3600	65	65	3600
	2	2	243	86.17	3600	644	109	3600	1444	66.17	3600	69	69	122
Haydarpasa	1	0	390	47.63	3600	329	54	3600	1307	40.46	3600	41	41	1
	1	1	400	55.02	3600	329	64	3600	1395	47.66	3600	49	49	35
	1	2	419	62.09	3600	329	79	3600	1441	58.50	3600	59	59	1
	2	0	194	49.49	3600	658	72	3600	1110	41.00	3600	41	41	1
	2	1	194	57.17	3600	658	82	3600	1228	46.20	3600	47	47	1
	2	2	193	60.75	3600	658	85	3600	1087	47.99	3600	50	50	35
Eskisehir	1	0	404	80.60	3600	420	84	3600	1455	69.67	3600	73	72	2034
	1	1	421	91.88	3600	420	106	3600	1533	76.20	3600	78	78	49
	1	2	429	107.50	3600	420	110	3600	1432	93.18	3600	96	96	3600
	2	0	210	91.65	3600	840	100	3600	1286	71.56	3600	72	72	1
	2	1	214	102.33	3600	840	129	3600	1275	71.77	3600	73	73	21
	2	2	217	105.04	3600	840	132	3600	1288	76.05	3600	77	77	1

Table 4.1: Results of CRG algorithm for RCCP-C

In order to overcome convergence and stalling problems, we modified the CRG algorithm to add multiple columns instead of a single column in each iteration. In each iteration, after we solve PSP with the solution method given in Section 4.1.2, we modify the arc costs, simply by increasing the length of the arcs on the current optimal path, and resolve the shortest path problem to generate an alternative path. The first path represents the column with the most negative reduced cost; the second is the second best column, and so on. We may continue generating alternative paths until a path with nonnegative reduced cost is found. In essence, any of these paths may enter the basis in the next iteration of the CRG algorithm. By generating multiple decision variables at each iteration, we hope to overcome the stalling problem that occurs when generating a single decision variable in each iteration. In Table 4.2, we present two more versions of the CRG algorithm namely DC-MC and AC-MC where MC corresponds to generating multiple columns in one iteration. In CRG-DC-MC, initial RMP is constructed by dummy columns that only covers a single duty and can connect to every other column and vice versa. On the other hand, in CRG-AC-MC initial RMP is constructed by the algorithm given in Section 4.1.1. Both algorithms expand RMP with multiple columns in each iteration, and CRG-DC-MC-IP and CRG-AC-MC-IP corresponds to the heuristic procedure for generating integer feasible solutions using CPLEX. We obtain the following result:

- In some cases, multi column version of the CRG algorithms solves the LP relaxation of [RCCP-C]$_{sc}$ to optimality within reasonable computational time.

- When we compare the IP-FS columns of the dummy column start (CRG-DC-MC-IP) and advanced column start (CRG-AC-MC-IP) versions of the algorithm, we observe significant difference between the best integer feasible solutions. Overall, we see that good integer feasible solutions are obtained by our heuristic procedure when we construct initial RMP with the algorithm given in Section 4.1.1. Contrary to our expectation, results obtained from the single column version (in Table 4.1) is better compared to the multi column version.

			CRG-DC-MC			CRG-DC-MC-IP			CRG-AC-MC			CRG-AC-MC-IP		
Region	Weeks	Days-Off	Itr	LP-LB	Time	IP-UB	IP-FS	Time	Itr	LP-LB	Time	IP-UB	IP-FS	Time
Ankara	1	0	31	50.00	245	322	68	3600	59	50.00	314	57	57	3600
	1	1	72	62.20	3600	322	75	3600	71	54.80	1780	67	67	3600
	1	2	55	67.72	3600	322	100	3600	39	65.20	537	83	83	3600
	2	0	64	59.67	3600	644	83	3600	143	52.85	3600	56	56	3600
	2	1	50	57.21	3600	644	95	3600	75	53.80	3600	65	65	3600
	2	2	39	68.46	3600	644	106	3600	71	60.95	3600	69	69	3600
Haydarpasa	1	0	70	38.00	1930	329	76	3600	50	38.00	160	41	41	3600
	1	1	27	42.66	528	329	70	3600	31	42.66	156	49	49	3600
	1	2	49	55.10	3600	329	104	3600	20	51.20	109	59	59	3600
	2	0	60	42.83	3600	658	43	333	124	38.37	3600	41	41	3600
	2	1	44	49.04	3600	658	97	3613	80	40.68	3600	47	47	3600
	2	2	36	50.94	3600	658	127	3600	62	44.36	3600	50	50	3600
Eskisehir	1	0	56	65.00	1547	420	118	3600	38	65.00	116	73	73	3600
	1	1	60	75.00	3600	420	102	3600	86	67.24	3600	78	78	3600
	1	2	53	86.13	3600	420	115	3600	67	84.70	3600	96	96	3600
	2	0	52	86.84	3600	840	106	3600	132	68.12	3600	72	72	3600
	2	1	38	80.97	3600	840	141	3600	88	68.87	3600	73	73	3600
	2	2	30	85.86	3600	840	144	3600	64	73.01	3600	77	77	3600

Table 4.2: Results of multi-column version of the CRG algorithm for RCCP-C

We also want to investigate the quality of the solutions obtained with the network flow formulation. For this purpose, we present results for three TCDD crew regions where we create the same problem instances with a planning horizon of one week and two weeks with different day-off requirements.

In Table 4.3, we present results for network flow formulations of both problems RCCP and RCCP-C to observe the effect of connectivity issue as well. In Table 4.3, LB corresponds to the objective function value of the optimal solution of the LP relaxation of (4.20)-(4.23). OPT indicates the optimal objective function value which corresponds to required number of crew members for both problems where Time shows the time required (in seconds) to solve the problem (4.20)-(4.23) to optimality. Finally, OPT* indicates optimal integer objective function value of (3.1)-(3.5), i.e. the optimal solution to RCCP. We obtain the following results:

- When we compare OPT against OPT*, we observe an increase in the number of required crew members in all scenarios. From the decision-maker's point of view, results indicate that the decisions on regional crew capacities without connectivity of the schedules might significantly differ from those where connectivity of schedules are integrated into the problem at the planning/pairing phase.

- Our numerical experiments indicate that the proposed network flow formulation is capable of solving all of the cases to optimality within reasonable computational time. Thus, our network flow formulation is a valid approach for the RCCP-C.

- When we compare LP-LB values of the cases where CRG algorithm terminates (see Table 4.2) to the corresponding LB values of network flow representation, we observe that network flow formulation gives better lower bounds to the problem, not to mention that the optimality of the LP relaxation is guaranteed.

- Both solution methods, CRG and network flow formulation, are capable of generating integer feasible solutions but network flow formulation is better off not only by solution quality but also with respect to computational time. Therefore, it is a more efficient and effective solution approach which could be used for solving RCCP-C as it is to RCCP.

				RCCP-C		
Region	Weeks	Days-Off	OPT*	LB	OPT	Time
Ankara	1	0	50	54.00	54	1
	1	1	55	59.80	60	1
	1	2	66	70.00	71	5
	2	0	50	54.00	54	1
	2	1	52	54.15	55	15
	2	2	57	58.66	59	87
Haydarpasa	1	0	38	39.00	39	1
	1	1	43	44.50	45	9
	1	2	52	53.40	54	15
	2	0	38	39.00	39	1
	2	1	41	41.00	41	45
	2	2	44	44.41	45	130
Eskisehir	1	0	65	69.00	69	1
	1	1	67	72.20	73	5
	1	2	76	86.20	87	4
	2	0	65	69.00	69	1
	2	1	65	69.00	69	20
	2	2	68	70.91	71	259

Table 4.3: Results of the network flow formulation for RCCP-C

Chapter 5

Rostering Strategies for RCCP-C

Crew rostering is an operational level problem which involves sequencing crew schedules that will be performed by a set of crew members over a specified planning horizon. A roster is also subject to rules and regulations. These rules and regulations which also constrain the feasibility of crew schedules, usually involve minimum and maximum working hour regulations which may be binded by safety standards or union agreements. In addition, the quality of a roster may be even subject to quality of life objectives based on the satisfaction level of industrial and individual preferences.

Crew rostering is usually treated as a second phase planning problem in crew scheduling literature and practice. In the first phase, crew pairing, feasible schedules are generated. Then, in the second phase, these schedules are sequenced together to make up the rosters. Rosters can be divided into two groups: cyclic and non-cyclic. In a cyclic roster, all crew members visit each schedule one by one, resulting in the same cyclic working pattern starting with a different particular schedule for each crew member. For example, with five crew schedules, say A,B,C,D and E, first crew member would execute a schedule sequence ABCDE; the second crew member would be given the sequence BCDEA; the third CDEAB; the forth DEABC, and the last EABCD. Over a planning horizon of five periods, each crew member completes the same workload but in a different order. This type of cyclic rostering allows the workload distribution to be inherently fair [10]. On the other hand a non-cyclic roster, as the name implies, does not require each crew member to visit every schedule. Constructing a non-cyclic roster is easier compared to a cyclic one, but the resulting uneven workload distribution in the short-term planing horizon is a drawback. In Figure 5.1, we provide examples of cyclic and multi-non-cyclic rosters schematically. Consider that each of the five nodes represent the first on-duty node of schedules A,B,C,D, and E, respectively while the five shaded nodes on the right are the last tie-up nodes of the schedules correspondingly. In this respect, the arcs from left to right represent the schedules while the arcs from right to left act as the connectivity arcs. In part (a), the schedules are connected to one another in a cyclic manner; note that there's only one cycle in this case. However, in part (b), there are three separate cycles as AB, C, and DE. Schedule sequence in part (a) can be executed by a group of five crew members that switch in each period; on the other hand, schedule sequence in part (b) can be executed by three group of crew members where two crew members execute A and B, two crew members execute D and E, and the last crew member executes C repeatedly.

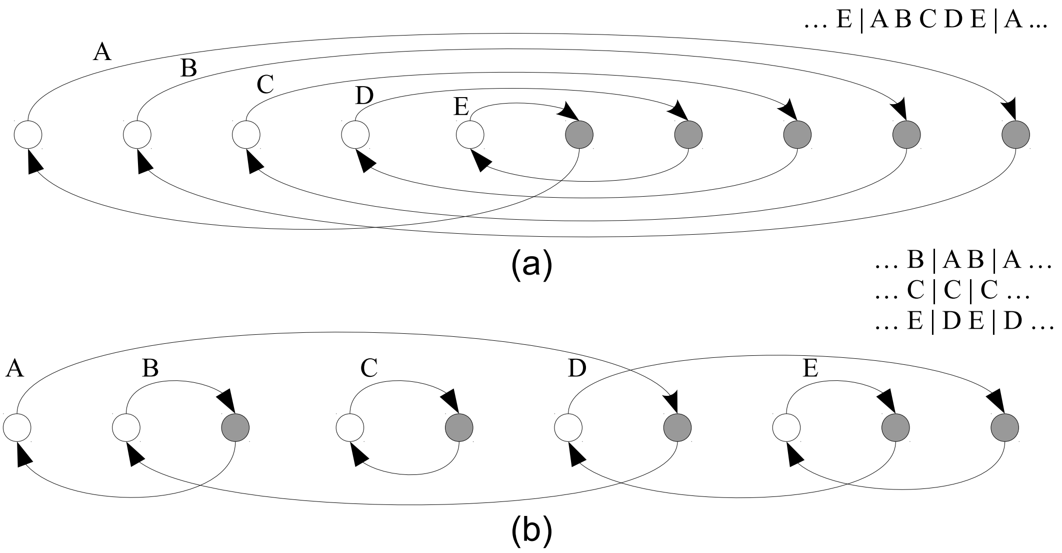

Figure 5.1: Examples of a cyclic (a) and a non-cyclic (b) rostering solutions

Ernst et al. [10] propose two schemes for generating cyclic and non-cyclic rosters in their study. They aim for feasibility and their algorithms do not contain an objective function. Although their non-cyclic rostering algorithm guarantees a feasible solution, the cyclic rostering algorithm does not. Ernst et al. [10] does not provide a valid integer programming formulation for cyclic rostering.

In our study of RCCP-C, by incorporating the connectivity issue into the RCCP problem, we explicitly consider the non-cyclic rosters and integrate this operational concern into our tactical level problem. In this regard, our focus is close to that of Ernst et al. [10]. Any feasible solution to RCCP-C contains a feasible roster, which may turn out to be a non-cyclic roster in the worst case. In the following sections, we present our preliminary work on generating cyclic rosters for our problem RCCP-C.

5.1 A Mathematical Model for the RCCP-C with Cyclic Rosters

In RCCP-C, the connectivity relationship between schedules could be used for constructing feasible rosters. In fact, any feasible solution contains a feasible rostering solution since continuity of schedules in the next recurrence of planning horizon is guaranteed. In order to find a cyclic solution from scratch, the set covering type formulation of the problem in [RCCP-C]$_{sc}$ should be extended as follows:

- An artificial schedule, *H*, that can connect to any other feasible schedule is introduced as the first schedule in the roster, and
- an artificial schedule, H', that can be connected by any other feasible schedule is introduced as the last schedule in the roster.

In Figure 5.2, we demonstrate H and H' and the corresponding sequence that starts with H and ends with H' by modifying the cyclic roster example given part (a) in Figure 5.1.

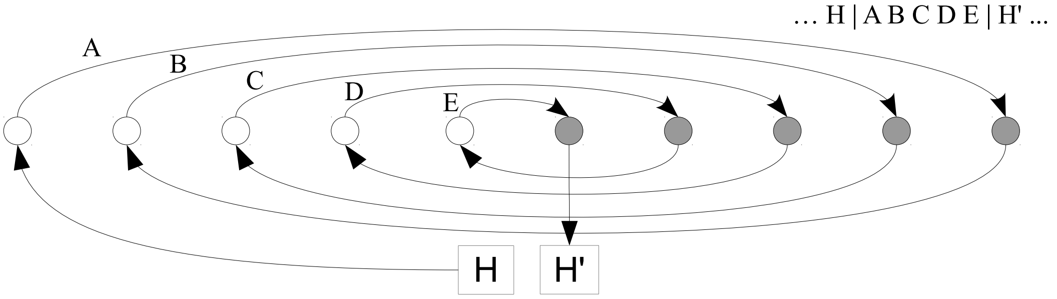

Figure 5.2: Example of a cyclic roster solution

In this setting, a cyclic roster corresponds to a sequence of feasible schedules, in which the sequence starts with schedule *H* and ends with schedule H'. On top of that, it is required that the last schedule before H' should connect to the first schedule after *H* to make a closed circuit of connected schedules. We are able to model these new restrictions with constraints that are similar to sub-tour elimination constraints in an open tour TSP.

If a new decision variable s_j denotes the sequence of schedule j is in roster ($j \neq H$), then, the mathematical programming formulation of the RCCP-C with cyclic rostering becomes:

$$
\begin{array}{lllr}
\text{minimize} & (3.6) & & \\
\text{subject to} & (3.7)-(3.8), & & \\
 & (4.1)-(4.3), & & \\
 & s_j - s_{j'} + nl_{jj'}y_{jj'} \leq n-1, & j, j' \in J - \{H, H'\}, & (5.1) \\
 & y_{Hj'} + y_{jH'} - l_{jj'} \leq 1, & j, j' \in J, & (5.2) \\
 & s_j \in \mathbb{Z}_+, & j' \in J. & (5.3)
\end{array}
$$

where n is greater than or equal to cardinality of J. In this extended model, constraint (5.1) is the subtour elimination constraint that would align the selected schedules into a sequence and constraint (5.2) guarantees that the last schedule on the sequence connects to the first one so that the schedules are connected to one another in a cyclic manner.

Computational results given in Section 4.3 reveals that CRG algorithm that we propose to solve [RCCP-C]$_{sc}$ performs poorly. Solving [RCCP-C]$_{sc}$ with additional constraints (5.1)-(5.3) would require a more complicated CRG algorithm in which PSP may turn out to be a more difficult problem. Based on this observation and our computational experience, we have not yet attempted to develop a solution algorithm for solving [RCCP-C]$_{sc}$ with additional constraints (5.1)-(5.3). On the other hand, we do not yet have an extended model for the network flow formulation of RCCP-C that would generate a cyclic roster.

5.2 A Cyclic Rostering Algorithm for RCCP-C

In the previous section, the objective is to propose a mathematical model that would find a cyclic solution. An alternative approach is to develop a scheme that would generate alternative solutions for the non-cyclic version and try to identify a solution that is in fact a cyclic one. By using the connectivity relationship between the schedules, we are able to propose such an algorithm.

Consider a feasible solution to problem [RCCP-C]$_{sc}$, let every schedule in the solution be represented by a node and the connectivity relationship between those schedules (i.e. nodes) are represented by arcs (i.e. whenever $l_{jj'} = 1$ a directed arc is placed between j and j'). In this setting, finding a cyclic roster corresponds to finding a circuit that includes every node by exactly once. This problem is known as detection of Hamiltonian circuits in a directed graph; Karmakar [12] proposes a polynomial time algorithm that detects such a circuit if there is one. It is possible to check whether a set of feasible schedules can be sequenced in a cyclic roster or not, and to identify that circuit within reasonable computational effort.

With our network flow formulation given in Section 4.2, we are able to solve RCCP-C problem to optimality within reasonable computational time. We propose an iterative algorithm to find a solution that contains a cyclic roster with the optimal number of crew schedules; the main steps of this algorithm is depicted on a flowchart in Figure 5.3. We first solve the problem (4.20)-(4.23), and then check whether this solution contains a cyclic roster. If this is the case, we have identified the optimal solution. Otherwise, we resolve the problem to find the next best solution to (4.20)-(4.23) by introducing a new constraint that would prevent the model to come up with a previously found solution. We might check whether this new solution contains a cyclic roster; if this is not the case, we might continue this procedure until we find one such solution.

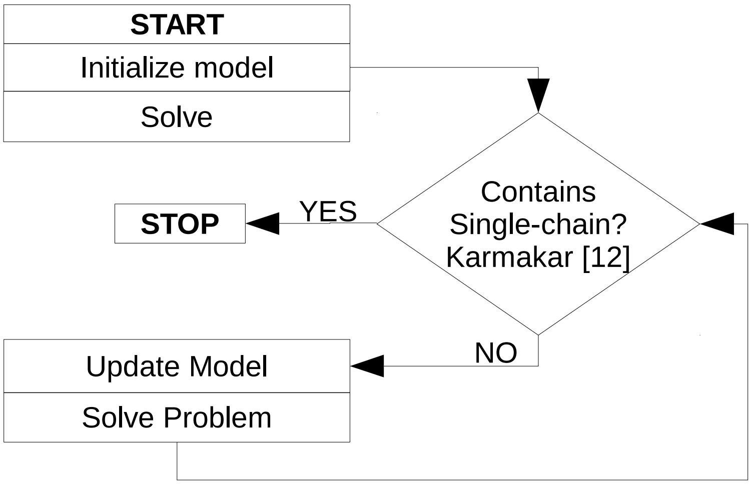

Figure 5.3: Algorithm for generating a cyclic roster

To find the next best solution of (4.20)-(4.23), we need to update the model in each iteration of the algorithm accordingly. In order to describe the updated mathematical model throughout the iterations of the algorithm, we use the approach in Acar et al. [3].

N denotes the set of previous iterations, and n denotes the index of iterations. M, a sufficiently large number, represents the penalty cost associated with the previously found solution at any iteration. To keep track of the solutions found in previous iterations, we define a parameter as:

$$T_{na} = \bar{x}^n_a, a \in A_c,$$

where $\bar{x}^n_a$ denotes the value of $\bar{x}_a$ in (4.20)-(4.23) (in candidate solution n) and a decision variable

$$Z_n = \begin{cases} 1, & \text{if the solution currently considered by the mathematical model} \\ & \text{is suggested in previous iteration } n; \\ 0, & \text{otherwise.} \end{cases}$$

that is used to incorporate the penalty cost for candidate solution at iteration n into the objective function. Then, the resulting mathematical model is

$$\begin{aligned} \text{minimize} \quad & (4.20) + \sum_n MZ_n && && (5.4) \\ \text{subject to} \quad & (4.21) - (4.23), \\ & \sum_{a \in A_c} (2T_{na}\bar{x}_a - T_{na} - \bar{x}_a) \leq Z_n - 1, && \forall n \in N, && (5.5) \\ & \sum_{a \in A_c} (2T_{na}\bar{x}_a - T_{na} - \bar{x}_a) \geq M(Z_n - 1), && \forall n \in N, && (5.6) \\ & Z_n \in \{0, 1\}, && \forall n \in N. && (5.7) \end{aligned}$$

In this formulation, constraints (5.5)-(5.6) identify a previously obtained solution through the values of flow on connectivity arcs. If such a solution is found (i.e. $\bar{x}_a = T_{na}$, $\exists n \in N$), that solution is penalized in the objective function.

We have tested our single-chain rostering algorithm for RCCP-C with TCDD data sets. For each data set, we have observed the same results: initially none of the solutions of (4.20)-(4.23) was a cyclic roster, but in the first iteration it turns out that the optimal schedules can be rostered in a cyclic fashion. This is due to the high connection possibility of the schedules in our data sets; in other words, $l_{jj'}$ of the selected schedules in the first optimal solution is a dense matrix.

Chapter 6

Conclusions and Future Research

Effective crew management is a critical planning problem in railways. We study a tactical level crew planning problem where we consider different policies and practical considerations that also include the ones as applied in TCDD. We first focus on RCCP problem which determines the minimum number of crew members required to cover all duties in a given region. We suggest improvements for the network flow formulation of the problem in an earlier study; this improvement results in a significant decrease in the number of required crew members. Moreover, we develop a set-covering type formulation for the same RCCP problem; we propose a column generation algorithm to solve the LP relaxation and a heuristic to obtain integer feasible solutions. We observe that good-quality integer feasible solutions are obtained by our heuristic procedure based on the LP relaxation solutions obtained by the column generation algorithm. Yet, the column generation performs poorly not only by solution quality but also in terms of the required computational effort for both the LP relaxation and integer feasible solutions when we compare the results with those obtained with the network flow formulation of the problem. In conclusion, our computational study indicates that the network flow formulation is a valid solution approach to the RCCP problem.

While determining the minimum required crew capacity for a region, tactical decisions should take into account the operational level considerations as much as possible. In the planning process, tactical decisions are made by considering a finite planning horizon to repeat periodically; but the recurrence of the planning period is overlooked, which may lead to crew schedules that are not implementable in practice. We extend RCCP to RCCP-C (RCCP with connectivity considerations in pairings) to consider finding a set of feasible crew schedules that can be connected to other schedules from one period to the other. For this extension of the problem, we follow the footsteps of the solution methods for the original RCCP: (i) a set-covering type formulation and a simultaneous column-and-row generation algorithm and (ii) a network representation of the problem along with a corresponding network flow formulation. We perform a computational study with data sets acquired from TCDD. Our numerical experiments show that the proposed network flow formulation is capable of solving all real-life cases to optimality within reasonable computational time. On the other hand, CRG algorithm fails to solve the LP relaxation of RCCP-C problem to optimality within reasonable computational time where we experience unexpected convergence and stalling problems with the CRG algorithm. Both solution methods, CRG and network flow formulation, are capable of generating integer feasible solutions but network flow formulation outweighs not only by solution quality but also with respect to computational time. In conclusion, our computational study indicates that the network flow formulation is a more efficient and effective solution approach which could be used for solving RCCP-C as it is to RCCP. From the decision-maker's point of view, the results clearly shows that the decisions on regional crew capacities that ignore the connectivity of the schedules might significantly differ from those where connectivity of schedules are integrated into the problem at the planning/pairing phase.

Lastly, we focus on additional crew rostering issues, which could be considered in the context of operational level planning. Crew rostering involves assigning feasible and but also preferable crew schedules to a set of crew members over a specified planning horizon. Balanced allocation of workload among crew members is a particular quality aspect in rostering which may be attained by carefully planning the schedule connectivity in the long-term. The cyclic rosters provide perfect balance over several recurrences of the planning period in the long term. We first provide a mathematical model based on the set-covering formulation of the RCCP-C and then propose an algorithm that generates a cyclic rostering solution. This preliminary study leads to ideas for future research directions that would incorporate more operational level concerns into tactical level planning.

For future research, we consider the workload balancing as a valuable venue. Cyclic rostering is a trivial management solution, and literature lacks alternative ones; yet, obtaining perfectly cyclic rosters may not be that trivial from a methodological point of view. In addition, integrating other operational concerns such as minimizing operational costs, fair allocation of different type of duties among the crew and other quality of life objectives based on the satisfaction level of industrial and individual preferences are open research questions to be studied.

Bibliography

[1] E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401, 2005. ISSN 0092-2102. doi: http://dx.doi.org/10.1287/inte.1050.0158.

[2] E. Abbink, J. V. Wout, and D. Huisman. Solving large scale crew scheduling problems by using iterative partitioning. In C. Liebchen, R. K. Ahuja, and J. A. Mesa, editors, *ATMOS 2007 - 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Dagstuhl, Germany, 2007. Internationales Begegnungs und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. ISBN 978-3-939897-04-0.

[3] Yavuz Acar, Sukran N. Kadipasaoglu, and Jamison M. Day. Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems. *Computers & Industrial Engineering*, 56 (1):106 – 112, 2009.

[4] A. Caprara, M. Fischetti, P. L. Guida, P. Toth, and D. Vigo. Solution of large-scale railway crew planning problems: the italian experience. In *N.H.M. Wilson (Ed.), Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, pages 1–18. Springer, 1997.

[5] A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P. L. Guida. Algorithms for railway crew management. *Math. Program.*, 79(1-3):125–141, 1997. ISSN 0025-5610. doi: http://dx.doi.org/10.1007/BF02614314.

[6] A. Caprara, M. Monaci, and P. Toth. A global method for crew planning in railway applications. In *S. Voß and J. R. Daduna (Eds.), Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, 505*, pages 17–36. Springer, 2001.

[7] Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. Chapter 3 passenger railway optimization. In Cynthia Barnhart and Gilbert Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 129 – 187. Elsevier, 2007. doi: 10.1016/S0927-0507(06)14003-7. URL `http://www.sciencedirect.com/science/article/pii/S0927050706140037`.

[8] G. Şahin and B. Yüceoğlu. Tactical crew planning in railways. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1221 – 1243, 2011. ISSN 1366-5545. doi: 10.1016/j.tre.2011.05.013. URL `http://www.sciencedirect.com/science/article/pii/S136655451100072X`.

[9] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, February 2004. doi: 10.1016/S0377-2217(03)00095-X.

[10] A.T. Ernst, H. Jiang, M. Krishnamoorthy, H. Nott, and D. Sier. An integrated optimization model for train crew management. *Annals of Operations Research*, 108:211–224, 2001. ISSN 0254-5330. URL `http://dx.doi.org/10.1023/A:1016019314196`. 10.1023/A:1016019314196.

[11] R. Freling, R. M. Lentink, and M. A. Odijk. Scheduling train crews: a case study for the dutch railways. In *Proceedings of CASPT 2000*, 2000.

[12] Sudhangshu B. Karmakar. Detection of hamiltonian circuits in a directed graph. *The Journal of the Australian Mathematical Society Series B. Applied Mathematics*, 24:234–242, 1982. doi: 10.1017/S0334270000003702.

[13] L. G. Kroon and M. Fischetti. Crew scheduling for netherlands railways - destination: Customer, 2000.

[14] L. G. Kroon and M. Fischetti. Crew scheduling for netherlands railways: "destination: Customer". Technical Report ERS-2000-56-LIS, Erasmus Research Institute of Management (ERIM), December 2000. URL `http://ideas.repec.org/p/dgr/eureri/200057.html`.

[15] Viswanathan P. Kumar, Michel Bierlaire, and Olivier Gallay. Generalized Algorithms for Crew Planning: Survey and Future Directions for Railways. Technical report, September 2010. URL `http://transp-or2.epfl.ch/proceedings/KumarBierGall2010.pdf`.

[16] E. M. Morgado and J. P. Martins. CREWS_NS: Scheduling train crews in the netherlands. *AI Magazine*, 19(1):25–38, 1998. ISSN 0738-4602.

[17] Ibrahim Muter. *Large-scale Linear Programs with Column-Dependent-Rows*. PhD thesis, Sabanci University, Istanbul, aug 2011.

[18] Ibrahim Muter, Ilker Birbil, and Kerem Bulbul. Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. Technical Report 2010/0004, Sabanci University, April 2010. URL `http://research.sabanciuniv.edu/14197/`.

[19] Turkish State Railways. Turkish state railways annual statistics 2000-2004, . URL `http://www.tcdd.gov.tr/tcdding/20002004ist.pdf`. retrieved on May 1, 2012.

[20] Turkish State Railways. Turkish state railways annual statistics 2004-2008, . URL `http://www.tcdd.gov.tr/tcdding/2004_2008.pdf`. retrieved on May 1, 2012.

[21] Turkish State Railways. Turkish state railways annual statistics 2006-2010, . URL `http://www.tcdd.gov.tr/tcdding/20062010yillik.pdf`. retrieved on May 1, 2012.

[22] B. Vaidyanathan, K. C. Jha, and R. K. Ahuja. Multicommodity network flow approach to the railroad crew-scheduling problem. *IBM Journal of Research and Development*, 51(3):325–344, 2007.