Resource constrained project scheduling models and algorithms applied to underground mining

SE Terblanche

Thesis presented in partial fulfilment of the requirements for the degree of
Master of (Industrial) Engineering
in the Faculty of Engineering at Stellenbosch University

Supervisor: JH van Vuuren

December 2015
Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 1, 2015
Abstract

The resource constrained project scheduling problem (RCSP) involves the scheduling of a number of activities over time, where each activity consumes a unit of some resource per time period. For a feasible solution to exist, the total resource consumption per time period must be less than the available resources. In addition, the order in which activities may be scheduled is determined by a precedence graph. The nodes of this directed graph represent the various activities and each directed edge a precedence relationship. A multitude of variations on the RCSP exist in the literature as do various solution approaches. Some of the frequently applied objective functions include the minimisation of the makespan, the minimisation of a tardiness penalty cost, and the maximisation of net present value.

The advent of practical computer technology during the late 1950s has meant that various industrial problems can now be solved by computer algorithms. It soon became clear, however, that certain types of problems are inherently difficult and in some cases even impossible to solve. Even today scheduling problems exist which have no more than 60 tasks to be scheduled, but which cannot be solved to optimality within reasonable time using the latest algorithmic and computer technology.

In this thesis, the challenges of underground mine planning are addressed by employing RCSP models and algorithms. Underground mine planning entails the scheduling of mining activities in a manner that the most economical value is derived while satisfying constraints related to resource requirements and physical limitations due to the properties of the mine infrastructure. Of specific interest is the use of resource flow-based RCSP models to address resource requirements related to the movement of mining crews and equipment. In addition to the inherent ability to track the flow of these resources when computing a solution, flow-based RCSP models also allow for the modelling of transfer delay constraints which are specifically useful in mechanised mining where the movement of large machinery from one point to another may cause significant delays in a mine production schedule.

Two new RCSP formulations are proposed in this thesis, of which one is based on a resource flow formulation and the other on a time-indexed formulation. Modifications to the resource flow formulation are proposed so as to accommodate the maximisation of net present value, while the time-indexed formulation is adapted to incorporate resource flow-related constraints. Due to the computational complexity of the underground mine scheduling problem, several reformulations of the problem are suggested. In addition, a Benders decomposition approach is described which is capable of improving the computation of feasible solutions for large problem instances.

Computational results presented in this thesis are based on both randomly generated data and data from a real South African underground mine. Based on these results it is found that the best performing model reformulation involves the use of a resource flow-based model in conjunction with a constraint aggregation approach. The Benders decomposition approach, implemented within a branch-and-cut framework, has proved to scale well for problem instances with a large
number of activities and resources. This is a significant contribution within the context of mining, especially considering the large number of resources that needs to be accommodated in solving underground mine scheduling optimisation problems.
Uittreksel

Die *hulpbronbeperkte skeduleringsprobleem* (HBSP) behels die skedulering van 'n aantal aktiwiteite oor verloop van tyd, waar elke aktiwiteit 'n eenheid van 'n sekere hulpbron per tydeenheid verbruik. Vir 'n haalbare oplossing om te bestaan, moet die totale hulpbron verbruik per tydeenheid minder wees as die beskikbare hulpbronne. Die volgorde waarin aktiwiteite geskeduleer kan word, word deur 'n voorrang-grafiek bepaal. Die nodes van hierdie gerigte grafiek verteenwoordig die verskillende aktiwiteite en elke gerigte skakel dui op 'n voorrang verhouding. Verskeie variasies van die HBSP kan in die literatuur gevind word, asook verskeidenheid oplosmetodes. Van die mees gewilde doelwitfunksies behels die minimering van die totale projekduur, die minimering van 'n traagheids-strafkoste, en die maksimering van die netto huidige waarde.

Die ontwikkeling van praktiese rekenaartegnologie in die laat 1950's het beteken dat verskeie industriële probleme nou ook deur middel van rekenaar algoritmes opgelos kan word. Dit het egter gou duidelik geword dat sekere tipes probleme inherent moeilik is en dat dit in sommige gevalle selfs onmoontlik is om hierdie probleme op te los. Tot op hede bestaan daar selfs probleme wat nie meer as 60 aktiwiteite het nie, en wat nie optimaal binne 'n redelike tydsverloop opgelos kan word nie, selfs al word daar van die nuutste rekenaartegnologie en algoritmes gebruik gemaak.

In hierdie tesis word die uitdaging van ondergrondse mynbouplanning aangespreek deur die gebruik van HBSP modelle en algoritmes te oorweeg. Ondergrondse mynbouplanning behels die skedulering van mynboubedrywighede op so 'n wyse dat die mees ekonomiese waarde daaruit geput kan word. Dit moet gedoen word deur ook in ag te neem wat die vereistes ten opsigte van hulpbronbeperkings is asook beperkings wat daar gestel word deur die fisiese infrastruktuur van die myn. Van spesifieke belang is die gebruik van vloei-gebaseerde HBSP modelle om die beweging van mynbou werkspane en toerusting in ag te neem. Sodoende kan beperkings, wat byvoorbeeld van toepassing is in gemeganiseerde mynburg, geplaas word op die toelaatbare vertragingstyd wanneer groot masjinerie van een punt na 'n ander vervoer moet word.

Twee nuwe modelle vir die HBSP word in hierdie tesis voorgestel, waarvan die een gebaseer is op 'n hulpbron vloei-gebaseerde formulering en die ander op 'n tyd-geïndeksseerde formulering. 'n Aanpassing word vir die hulpbron vloei-gebaseerde formulering voorgestel om sodanig aktiwiteite te akkommodeer. Verder word daar ook 'n voorgestel gemaak om die tyd-geïndeksseerde formulering aan te pas sodat die hulpbron vloei-verwante beperkings ge-inkorporeer kan word. Verskeie herformulerings word voorgestel as gevolg van die noemenswaardige berekeningskompleksiteit van die ondergrondse mynbou skeduleringsprobleem en 'n Benders ontbindingsbenadering word beskryf wat in staat is om die berekeningstyd van oplossings vir groot probleemgevalle te verminder.

Resultate wat in hierdie tesis gerapporteer, word is gebaseer op beide lukraak-gegeneerdata en data wat afkomstig is van 'n Suid-Afrikaanse ondergrondse myn. Die bevindings van hierdie studie is dat die beste model gebruik maak van 'n hulpbron vloei-gebaseerde formulering, in samewerking met 'n formuleringsbenadering wat sekere beperkings aanvoeg. Die resultate vir
die Bender's ontbindingsbenadering wys daarop dat die metode skaalbaar is vir probleem gevalle met 'n groot aantal aktiwiteite en hulpbronne. Dit is 'n belangrike bydrae in die konteks van mynbou, veral in ag geneem dat 'n groot aantal hulpbronne in die oplossing van ondergrondse mynbou skeduleringsprobleme beskou moet word.
Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- To Prof Jan Van Vuuren, thank you for your guidance and assistance with the manuscript.
- To MineRP, for providing me with access to the necessary planning tools and data.
- To Mariet, my lovely wife. Thank you for your unconditional love and support.
Table of Contents

Abstract iii
Uittreksel v
Acknowledgements vii
List of Acronyms xiii
List of Figures xv
List of Tables xvii

1 Introduction 1
   1.1 An economic perspective ........................................... 2
   1.2 Mine production planning ........................................... 2
   1.3 Scientific and technological advances ............................. 3
   1.4 Research methodology and contributions .......................... 4
   1.5 Mathematical preliminaries ....................................... 5
   1.6 Chapter orientation ................................................. 7

2 Scheduling algorithms and computational complexity 9
   2.1 The classical scheduling problem ................................. 9
   2.2 Computational complexity ......................................... 10
   2.3 The resource constrained project scheduling problem (RCSP) .... 13
       2.3.1 Mathematical formulations ..................................... 14
       2.3.2 Algorithmic approaches ....................................... 17
   2.4 Summary ............................................................... 18

3 The underground mine scheduling optimisation problem 21
   3.1 Technical aspects of underground mine scheduling ............... 21
# Table of Contents

3.2 Solving the mine scheduling optimisation problem as an RCSP .................................. 25  
3.3 Practical considerations in mine scheduling optimisation ............................................. 28  
3.4 Modelling requirements beyond the RCSP framework .................................................. 29  
3.5 Related work .................................................................................................................... 30  
3.6 Summary .......................................................................................................................... 31  

4 Mathematical Models ......................................................................................................... 33  
4.1 Formulations of the RCSP when considering resource transfer requirements ................. 33  
4.2 Mathematical Models ....................................................................................................... 34  
4.2.1 A resource flow RCSP formulation ........................................................................... 35  
4.2.2 The resource flow mine scheduling problem (RMSP) .............................................. 37  
4.2.3 The time-indexed mine scheduling problem (TMSP) .............................................. 38  
4.3 Time-based costs and revenue ........................................................................................... 40  
4.4 Crew requirement constraints and transfer delays ............................................................ 41  
4.5 Summary .......................................................................................................................... 43  

5 Algorithmic improvements ................................................................................................. 45  
5.1 Preprocessing .................................................................................................................... 46  
5.2 Problem reformulations RMSP-AC and TMSP-AC ....................................................... 46  
5.3 Graph reduction ................................................................................................................ 47  
5.3.1 RMSP with graph reduction (RMSP-GR) .................................................................. 47  
5.3.2 TMSP with graph reduction (TMSP-GR) .................................................................. 48  
5.4 Problem reformulations RMSP-AC-GR and TMSP-AC-GR .......................................... 49  
5.5 An alternative TMSP formulation ..................................................................................... 50  
5.6 A branch-and-cut framework ........................................................................................... 50  
5.6.1 Branch-and-bound basics ......................................................................................... 51  
5.6.2 Cut generation in the branch-and-bound .................................................................. 52  
5.6.3 Benders decomposition of the RCSP ....................................................................... 53  
5.6.4 The resource flow separation problem (RFSEP) ...................................................... 54  
5.6.5 The RFSEP with aggregated constraints (RFSEP-AC) ............................................. 55  
5.6.6 An incremental implementation of the RFSEP (RFSEP-INC) ................................. 56  
5.6.7 Master problem alternatives ...................................................................................... 56  
5.6.8 The resource flow rounding heuristic (RFRH) ........................................................ 57  
5.6.9 The RFRH with guaranteed feasibility (RFRH-GF) ................................................ 57  
5.6.10 An incremental implementation of the RFRH (RFRH-INC) ..................................... 59  
5.6.11 Augmenting the master problem with valid inequalities ......................................... 59
# Table of Contents

5.7 Summary .................................................. 60

6 **Empirical results** .............................. 63

6.1 Implementation details ......................... 64
6.2 Data sets .............................................. 64
6.3 Result set I: Effect of directional and transitivity inequalities .......... 65
6.4 Result set II: The efficiency of the proposed reformulations .............. 66
6.5 Result set III: The effectiveness of Benders decomposition ................. 68
6.6 Result set IV: Solving real-world problem instances ......................... 73
  6.6.1 Practical consideration related to mine scheduling optimisation .... 74
  6.6.2 Computational efficiency of solving real-world problem instances .... 77
6.7 Summary .................................................. 80

7 **Summary and conclusion** .................. 81

7.1 Chapter summaries ................................. 81
7.2 Research outputs .................................. 83
7.3 Future work ......................................... 84

References ............................................... 87
List of Acronyms

AC: Aggregated constraints

ATMSP: Alternative time-indexed mine scheduling optimisation problem

CAD: Computer aided design

CVI: Cutset valid inequalities

GF: Guaranteed feasibility

GR: Graph reduction

ILP: Integer linear programming problem

INC: Incremental

LP: Linear programming problem

MILP: Mixed integer linear programming problem

NPV: Net present value

RCSP: Resource constrained scheduling problem

RFRH: Resource flow rounding heuristic

RFSEP: Resource flow separation problem

RMSP: Resource flow mine scheduling optimisation problem

TMSP: Time-indexed mine scheduling optimisation problem
# List of Figures

2.1 An example of reduction of well known $NP$-complete problems. .................. 12

3.1 A typical underground mine layout. ...................................................... 22

3.2 Top-view of cross-cuts accessing a reef. ............................................... 23

3.3 Sequential vs. pillar mining. .............................................................. 23

3.4 Sequencing rules for simulating a sequential mining method. The diagram on the left is for a time-based discretisation and the diagram on the right for a logical discretisation of the activities. .................................................. 24

3.5 A logical discretised section of a mine level. Identification numbers are provided next to each logical mining activity. ............................................. 25

3.6 Gantt chart showing results for the small example considered. ................. 28

3.7 Production profile of the “tonnes” resource aggregated monthly for the small example considered. .......................................................... 28

4.1 Possible scheduling outcomes for an activity $i$ relative to a period $t$. .......... 39

4.2 Crew requirements over time for the unconstrained RMSP. ........................ 41

4.3 Gantt chart showing results for the unconstrained RMSP. ......................... 42

4.4 Gantt chart showing results for the crew constrained RMSP. ...................... 42

4.5 Gantt chart showing results for the RMSP with transfer delays. .................. 42

5.1 Outline of the branch-and-cut framework. .............................................. 52

6.1 Unconstrained tonnes profile for the 12 half-levels of the L6-Y2 problem instance. 75

6.2 Tonnes profile for the 12 half-levels of the L6-Y2 problem instance when considering a half-level capacity constraint of 10000 per month. .................. 75

6.3 Comparing the effect on total tonnes and crew for the the L6-Y2 problem instance when considering a half-level capacity of 10000 tonnes per month. .... 76

6.4 Comparing the effect on total tonnes and crew for the the L6-Y2 problem instance when considering a half-level capacity of 10000 tonnes per month and a maximum crew requirement limit of 30 crews per day. ............................... 76
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Activity input values for the small example considered.</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>Resource input values for the small example considered.</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Activity input values for the small example considered.</td>
<td>41</td>
</tr>
<tr>
<td>5.1</td>
<td>A list of abbreviations used for distinguishing between all combinations of problem formulations.</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>The detrimental effect of directional and transitivity inequalities on computing times.</td>
<td>65</td>
</tr>
<tr>
<td>6.2</td>
<td>The computational benefit of the newly proposed reformulations for the PSPLIB data sets.</td>
<td>66</td>
</tr>
<tr>
<td>6.3</td>
<td>The computational benefit of the newly proposed reformulations for the randomly generated data sets. Results are only provided for problem formulations for which at least one feasible solution was computed.</td>
<td>67</td>
</tr>
<tr>
<td>6.4</td>
<td>The computational benefit of the newly proposed separation sub-problems within a branch-and-cut approach.</td>
<td>68</td>
</tr>
<tr>
<td>6.5</td>
<td>The computational benefit of the newly proposed separation sub-problems within a branch-and-cut approach.</td>
<td>69</td>
</tr>
<tr>
<td>6.6</td>
<td>The computational benefit of the newly proposed heuristics within a branch-and-cut approach.</td>
<td>71</td>
</tr>
<tr>
<td>6.7</td>
<td>The computational benefit of the newly proposed heuristics within a branch-and-cut approach.</td>
<td>72</td>
</tr>
<tr>
<td>6.8</td>
<td>The computational benefit of the newly proposed heuristics within a branch-and-cut approach. Results are only provided for branch-and-cut implementations for which at least one feasible solution was computed.</td>
<td>73</td>
</tr>
<tr>
<td>6.9</td>
<td>Problem instances created from real mine planning data.</td>
<td>74</td>
</tr>
<tr>
<td>6.10</td>
<td>The computational efficiency for solving real-world instances while considering a half-level capacity of 10,000 tonnes per month.</td>
<td>77</td>
</tr>
<tr>
<td>6.11</td>
<td>The computational efficiency for solving real world instances while considering a half-level capacity of 10,000 tonnes per month and a maximum crew requirement limit of 30 crews per day.</td>
<td>77</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.12</td>
<td>The computational efficiency for solving real world instances with a mixed scheduling calendar while considering a half-level capacity of 10,000 tonnes per month.</td>
<td>78</td>
</tr>
<tr>
<td>6.13</td>
<td>The computational efficiency for solving real world instances with a mixed scheduling calendar while considering a half-level capacity of 10,000 tonnes per month and a maximum crew requirement limit of 30 crews per day.</td>
<td>78</td>
</tr>
<tr>
<td>6.14</td>
<td>Relative objective function values of the ATMS-SP-based formulation compared to the RMSP-based formulation.</td>
<td>79</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

Contents

1.1 An economic perspective ............................................. 2
1.2 Mine production planning ........................................... 2
1.3 Scientific and technological advances ............................. 3
1.4 Research methodology and contributions ......................... 4
1.5 Mathematical preliminaries ......................................... 5
1.6 Chapter orientation .................................................. 7

“A change so unexpected and a development never known before were due to
the discovery in 1886 of the greatest gold mines of all history, ancient and
modern. From 1886 the story of South Africa is the story of gold.”

C.W. de Kiewiet, South Africa, 1941

Mining in South Africa has its origins in the discovery of the first diamond on the banks of the
Orange River in 1867. Soon afterwards the discovery of gold in Pilgrim’s Rest and Barberton
followed, leading to the establishment of a promising South African economy. The real water-
shed, however, was the discovery of the Witwatersrand gold deposits which precipitated the
Anglo-Boer War. During the British occupation of South Africa from 1910 to 1961, mine pro-
duction became more efficient, leading to increased industrial support and promoting economic
growth. Gold production in South Africa peaked in 1970 with a 68 per cent contribution to the
world-wide gold production.

Apart from gold, South Africa has become the leading producer of chrome, manganese, platinum,
vanadium and vermiculite. Needless to say, mining has had and will most likely continue to have
a huge impact on the South African economy. In the remainder of this chapter an economic
perspective on mining will be provided, followed by an overview of the impact that mining has
had on scientific and technological advances in South Africa. Of specific interest to this study
is the use of resource constrained project scheduling models and algorithms for improving mine
production planning, taking into account economic factors, infrastructure limitations, ore body
distribution and resource requirements. The research methodology and contributions related
to the use of mathematical optimisation will be presented below, followed by an outline of the
remainder of the thesis.
Chapter 1. Introduction

1.1 An economic perspective

Mining in South Africa currently contributes about 8% directly to the Gross Domestic Product (GDP) [17]. Accounting for the indirect impact due to mining suppliers or downstream consumers, however, this percentage is more likely to be 17% of South Africa’s GDP [19]. The total annual mineral sales for South Africa in 2013 was recorded as R279.5 billion, of which gold, platinum group metals, iron ore and coal accounted for 81%. The resulting contribution to the national fiscus in the form of taxes and royalties amounted to an average of R23 billion per annum for the period 2011 to 2014. Mining and mineral investments on the Johannesburg Stock Exchange (JSE) accounted for approximately 24% of the total market capitalisation at the end of 2012. The total workforce employed by the mining industry was approximately five hundred thousand, with total employee earnings of R93 billion. These figures exclude the number of jobs created by the secondary mining industries. According to the Chamber of Mines of South Africa [18], for every mining job that is created, two further jobs are created in other sectors of the economy.

A cost breakdown of the total mining industry reflects the challenges faced by the mining companies to remain profitable. Almost 50% of mining costs, based on total industry figures, are due to operational and labour costs. This can most likely be attributed to the complex and dangerous environment of the typical South African mine, specifically underground gold mines. It is, therefore, imperative for mining companies to excel in their planning and operational efforts in order to be more cost efficient.

1.2 Mine production planning

The initial setup of a typical South African underground mine requires detailed mine layout planning based on exploration data, detailing the characteristics of the mineral deposits. Once in operation, continuous planning of the mine layout is required, since further information about the mineral deposits is unlocked during the underground excavation process. From an operational point of view, planning is done to manage the resources required for all mining activities. In its simplest form, operational mine planning entails the scheduling of mining activities such that the most economical value is derived while coping with physical limitations due to the properties of the mine infrastructure, as well as resource requirements. Different time horizons are used in planning for different purposes. For instance, short term mine planning is required for detailed activity planning and resource allocation, whereas for longer term planning, production output is profiled for the purpose of budgeting and corporate management.

In summary, the two primary planning activities encountered in mine production planning are mine layout planning and mine activity scheduling. Although the scope of this study is limited to the latter, the mathematical models employed in this thesis are dependent on a mine layout plan with accompanying data on the distribution of the mineral deposits.
1.3 Scientific and technological advances

Apart from the economic spin-off from mining, important advances have been made in both science and technology over the years. The well-known geostatistical method called Kriging is based on the empirical work conducted by the South African engineer, Danie Krige [44]. The use of Kriging is not limited to mining – its application can be found in all spheres of spatial analysis. Proof of further scientific contributions, specifically within a South African context, can be found by considering the journal publications hosted by the South African Institute of Mining and Metallurgy since 1969.

Planning production of an underground mine requires the use of specialised three-dimensional (3D) draughting software. Historically, mine layout planning was done by means of engineering drawings, projected onto a two-dimensional plane. With advances in computer technology, the use of 3D computer aided design (CAD) software became an indispensable tool for engineering design. The development of CAD software customised for mine planning gained momentum during the early 1990’s and today’s CAD technology encompasses a whole suite of tools ranging from the modelling of mineral deposits as 3D solids to simulation of underground activities visible as coloured polygons within a 3D space. Enabling the 3D CAD environment are computer algorithms responsible for finding solutions to various mathematical models, e.g. rotation of objects in 3D, projecting solids onto two dimensional planes, finding feasible scheduling solutions, etc.

It is evident that the successful completion of this study was dependent on the use of a 3D CAD system. Specifically, the software system Mine2-4D [55], which was made available by MineRP, was used for generating a mine layout plan and capturing the mineral deposit data.

The use of technology in mine planning has greatly improved the efficiency of mine production all over the world. Apart from improving resource utilisation, the use of 3D CAD systems also promotes flexibility in the mine planning process such that layout plans and schedules can be changed on short notice to adapt to changing economic or environmental conditions. This, of course, puts more pressure on the developers of scheduling systems to constantly improve the computational efficiency of their products. Furthermore, as computing abilities improve, the demand for more realistic mathematical models increases. For instance, initial mine scheduling systems were restricted to calculating the starting times of activities based on their durations and sequencing rules that dictate the order in which the activities may be executed. It soon became clear that a solution to such a scheduling problem could render infeasible results due to capacity restrictions imposed by the physical mine infrastructure. Furthermore, these early scheduling systems were not designed to incorporate financial and economic features, resulting in iterative approaches to improve economic viability of mine planning schedules.

More sophisticated mine scheduling systems exist today that are geared towards generating mining schedules that optimise some pre-specified objective function, e.g. the maximisation of net present value (NPV), while adhering to mine-specific requirements such as sequencing rules, production capacities and other resource limitations. The primary factor determining the ability of a scheduling system to handle mining complexities is the computational efficiency of the scheduling algorithms employed. In practice, more sophisticated mathematical models, capturing more complex mining environments, generally require more computing effort. Therefore, in order for a mine planning system to promote flexibility and agility in the mine planning process, its underlying scheduling algorithms should be efficient in finding good solutions for complex mathematical models, in a reasonable amount of time.
1.4 Research methodology and contributions

The theory of scheduling algorithms dates back to the formalisation of the critical-path scheduling problem by [39]. Informally, however, the use of horizontal bar charts, where each bar maps to a construction activity, can be traced back to the early 1900s [87]. Although the advent of practical computer technology during the late 1950s meant that various industrial problems could be solved by computer algorithms, it soon became clear that certain types of problems are inherently difficult, and in some cases even impossible to solve. Even today scheduling problems exist having no more than 60 tasks to be scheduled, that cannot be solved to optimality within reasonable time using the latest algorithmic and computer technology [67].

In order to make any contribution with respect to the underground mine scheduling problem, an understanding of the foundation of scheduling theory and the ability to build upon previous successes are required. In later chapters of this thesis an overview of the classification of general scheduling problems will be provided, to show specifically how the underground mine scheduling problem relates to the well-defined resource constrained project scheduling problem (RCSP). Furthermore, it will also be shown that most of the typical underground mining capacity constraints can be accommodated within the RCSP framework. There are, however, problem requirements for which new modelling constructs are needed. More specific, with the increased use of mechanisation within mining, the tracking of resources and and the ability to take transfer delays into account require some modifications to existing RCSP models.

The proposed methodology of this study is to make use of RCSP models and algorithms to solve the mine scheduling optimisation problem. The following is a summary of the anticipated contributions of this study.

- A thorough literature study is provided to explore i), the latest mathematical advances towards efficient solution approaches in the context of the RCSP, and ii), the technical issues pertaining to underground mine scheduling specifically within a South African context.

- It is shown that the underground mine scheduling optimisation problem is a special case of the well-defined RCSP and that most of the practical issues related to resource management in underground mine scheduling can be accommodated in standard RCSP models.

- Two mathematical formulations, based on existing RCSP formulations, are provided for solving the underground mine scheduling optimisation problem. The first model is based on a resource flow formulation and naturally extends to accommodate the modelling of delays when transferring resources. A new modelling construct is, however, proposed to allow the maximisation of net present value. The second model is based on a time-indexed formulation which is augmented with resource flow-related constraints in order to cater for delays in the transfer of resources.

- A strengthened formulation of the time-indexed RCSP is provided that proves to significantly reduce computing times.

- The underground mine scheduling optimisation problem is solved within an exact framework using CPLEX [36] as the branch-and-bound solver. The performance of the two proposed mathematical models, in terms of computing times, can easily be compared when using the same underlying solver. Furthermore, the exact framework allows for the implementation of generic side constraints without having to change existing algorithms or implement new ones.
1.5 Mathematical preliminaries

- Algorithmic improvements, in terms of computing times for the underground mine scheduling optimisation problem, are achieved by modelling considerations for the standard RCSP. That is, a simple modification to the RCSP formulation is shown to significantly improve computing times for both RCSP benchmark instances and real world mine scheduling instances. Furthermore, modifications to both the resource flow and time-indexed formulations are suggested for improving computing times.

- Additional speed-up is achieved through a preprocessing algorithm with the aim of reducing the number of variables in the RCSP model. The earliest and latest possible start times are determined for each of the activities, resulting in a reduction in variables for the time-indexed formulation and a stronger formulation of the resource flow problem.

- A branch-and-cut approach is suggested to improve tractability of large scale RCSP instances having a large number of resources. This is especially useful within the mine scheduling optimisation context where problem instances comprise many resources that need to be tracked and for which transfer delays are applicable. A decomposition approach is suggested which incorporates all resource flow-related variables in the form of Benders feasibility cuts. A separation problem is suggested for generating the feasibility cuts and a primal heuristic is proposed for the generation of feasible solutions within the branch-and-cut framework.

- A thorough computational study was performed to investigate the various modelling suggestions and also the proposed algorithmic improvements. Two sets of randomly generated data and a data set from a real underground mine were used.

1.5 Mathematical preliminaries

The solution approach adopted in this thesis is set within an exact mathematical programming framework. It suffices, therefore, to provide an overview of well known mathematical concepts and terminology in the fields of linear algebra, polyhedral theory, linear and integer programming. This section is not intended as an introduction, but rather as a reference to assist with notation.

Let the sets of real and integer numbers be denoted by \( \mathbb{R} \) and \( \mathbb{Z} \), respectively. For arbitrary index sets \( I = \{1, 2, \ldots, m\} \) and \( J = \{1, 2, \ldots, n\} \), the set \( \mathbb{R}^n \) or equivalently \( \mathbb{R}^{|J|} \), denotes all the vectors of size \( n \) that have components in \( \mathbb{R} \), and the set \( \mathbb{R}^{m \times n} \) or equivalently \( \mathbb{R}^{|I| \times |J|} \), denotes a matrix space of size \( m \times n \) that has components in \( \mathbb{R} \). Let \( j \in J \) be an index for the vector \( \mathbf{x} \) such that \( \mathbf{x} = (x_j)_{j \in J} \). In the remainder of this thesis all vectors are treated as column vectors. The transposed of the vector \( \mathbf{x} \in \mathbb{R}^n \) is denoted by \( \mathbf{x}^T \in \mathbb{R}^n \).

A vector \( \mathbf{x} \in \mathbb{R}^n \) can be expressed as a **linear combination** of the vectors \( \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in \mathbb{R}^n \) if there exist some \( \lambda \in \mathbb{R}^k \) such that \( \mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}_i \). If in addition \( \lambda \geq 0 \) and \( \sum_{i=1}^{k} \lambda_i = 1 \), \( \mathbf{x} \) is called a **convex combination** of the vectors \( \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in \mathbb{R}^n \).

The set \( H = \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} = a_0 \} \) with \( a_0 \in \mathbb{R} \) denotes a **hyperplane** with gradient \( \mathbf{a} \in \mathbb{R}^n \) and the set \( \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \leq a_0 \} \) denotes a **halfspace**. The intersection of a finite set of halfspaces defined by the set \( \{ \mathbf{x} \in \mathbb{R}^n : A \mathbf{x} \leq \mathbf{b} \} \), with \( A \in \mathbb{R}^{m \times n} \) and \( \mathbf{b} \in \mathbb{R}^m \), is called a **polyhedron**. An inequality of the form \( \mathbf{a}^T \mathbf{x} \leq a_0 \) with \( a_0 \in \mathbb{R} \) and \( \mathbf{a} \in \mathbb{R}^n \) is called a **valid inequality** for a polyhedron \( P \), if \( P \subseteq \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} \leq a_0 \} \).

A **linear programming problem** (LP) entails finding a vector \( \mathbf{x}^* \in P = \{ \mathbf{x} \in \mathbb{R}^n : A \mathbf{x} \leq \mathbf{b} \} \) that optimises the objective function \( \mathbf{c}^T \mathbf{x} \). The vector \( \mathbf{x}^* \) is called a **feasible solution** if \( \mathbf{x}^* \in P \) and
is called an optimal solution if \( c^T x^* \geq c^T x \) for all \( x \in P \) in the case of a maximisation problem, or if \( c^T x^* \leq c^T x \) for all \( x \in P \) in the case of a minimisation problem.

The standard form for representing an LP is:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\] (1.1)

Alternatively, the short notation \( \min \{ c^T x : Ax \leq b, x \in \mathbb{R}^n_+ \} \) is used instead. It should be noted that an LP with a minmisation objective function can be transformed into an LP with a maximization objective function (and vice versa), an LP with unbounded variables can be transformed to an LP with non-negative bounded variables, and an LP with inequality constraints can be transformed into an LP with equality constraints.

An optimal solution of an LP over a polyhedron \( P \) will always be at a vertex of \( P \), or a convex combination of some of the vertices of \( P \), provided \( P \) is bounded. Otherwise the LP may be unbounded with \( c^T x = \pm \infty \), depending on the choice of the objective function \( c^T x \).

Associated with each primal LP

\[
\min \{ c^T x : Ax \leq b, x \in \mathbb{R}^n_+ \}
\]
is the dual problem

\[
\max \{ w^T b : w^T A \geq c, w \in \mathbb{R}^m_+ \}.
\]

The relationship between the primal and dual problems is described by the following well-known theorems:

**Theorem 1.1** (Duality theorem). With regard to the primal and dual problems, exactly one of the following statements is true:

1. Both problems have optimal solutions \( x^* \in \mathbb{R}^n_+ \) and \( w^* \in \mathbb{R}^m_+ \) with \( c^T x^* = b^T w^* \).
2. One problem is unbounded, in which case the other must be infeasible.
3. Both problems are infeasible.

**Theorem 1.2** (Complementary slackness theorem). Let \( x^* \in \mathbb{R}^n_+ \) and \( w^* \in \mathbb{R}^m_+ \) be any feasible solutions to the primal and dual problems, respectively. These solutions are optimal if and only if:

\[
(c_j - a_j^T w^*) x_j^* = 0 \quad j = 1, 2, \ldots, n
\]

and

\[
w_i^* (a_i^T x^* - b_i)^T = 0 \quad j = 1, 2, \ldots, n
\]

where \( a_j \) denotes the \( j \)-th column and \( a_i^* \) denotes the \( i \)-th row of \( A \), respectively.

The objective of solving an integer linear programming problem (ILP) is to find an integer vector \( x^* \in \mathbb{Z}^n \cap P \) with \( P = \{ x \in \mathbb{R}^n_+ : Ax \leq b \} \), that optimises the objective function \( c^T x \).

The problem obtained by omitting the integrality restrictions on ILP is again an LP, called the LP relaxation of the ILP. The objective of solving a mixed integer linear programming problem
1.6. Chapter orientation

(MILP) is to find $x^* \in \mathbb{Z}^n \cap P$ and $y^* \in \mathbb{R}^n \cap P$ with $P = \{(x, y) \in \mathbb{R}_{+}^{n \times n} : Ax + Dy \leq b\}$, so that the objective function $c^T x + d^T y$ is optimised.

Consider a minimisation problem where $x^U \in \mathbb{Z}^n \cap P$ denotes a feasible solution to an ILP with associated objective function value $z^U = c^T x^U$, and let $x^L \in P$ be an optimal solution of the LP relaxation with associated objective function value $z^L = c^T x^L$. For a minimisation problem the quantities $z^U$ and $z^L$ are referred to as the upper bound and lower bound, respectively. For a maximisation problem it is the converse. The quantity $(z^U - z^L)/z^L$ is called the integrality gap.

1.6 Chapter orientation

In Chapter 2, the RCSP is formally introduced within the context of classical scheduling theory. A literature review of the computational complexity involved in solving the RCSP is presented and references are provided of the main streams of mathematical formulations of the RCSP. A section on recent algorithmic advances is provided with the main focus on exact approaches.

An introduction to underground mine scheduling is provided in Chapter 3. In order to gain an understanding of the problem to be solved, an overview of the technical aspects and terminology in underground mining is provided. A small example is used to demonstrate that the underground mine scheduling optimisation problem is a special case of the RCSP. There are, however, some practical considerations in underground mining that do not fit into the RCSP framework and have to be addressed by generic mathematical programming formulations.

Two mathematical models for solving the underground mine scheduling optimisation problem are presented in Chapter 4. Both these models are based on existing RCSP formulations, but were augmented in order to address, i) the modelling of transfer delays with respect to the time-indexed RCSP model, and ii) the linearisation of the objective function of the RCSP flow-based model in the case of maximising net present value.

Algorithmic efforts in speeding up computing times of the underground mine scheduling optimisation problem are presented in Chapter 5. The primary contributions of this chapter are the introduction of problem reformulations for both the flow-based and the time-indexed RCSP, as well as a Benders decomposition of the RCSP implemented within a branch-and-cut framework.

Computational results are provided in Chapter 6. The primary objective of this chapter is to show comparative results for the newly proposed problem formulations and the application of Benders decomposition to the underground mine scheduling optimisation problem. Empirical results are based on well known RCSP benchmark instances as well as real instances from the mining industry.

A final summary and conclusion are presented in Chapter 7. Concluding remarks are provided and suggestions for future work are discussed.
Recent advances in computer technology and the progress made in developing new algorithmic ideas, made it possible to solve some industrial-sized problems. Certain types of problems are, however, inherently difficult, and in some cases even impossible to solve to optimality in reasonable time. The basic concepts of computational efficiency will be presented in this chapter in an attempt to demonstrate the challenges faced in solving underground mine scheduling problems. Furthermore, the most recent algorithmic advances in solving related problems will be presented in the form of a literature overview.

2.1 The classical scheduling problem

The outcome of a classical scheduling problem is the processing of jobs in one or more time intervals, resulting in a schedule. Each job may, however, comprise one or more operations, which may be required to be processed by one or more machines. To be precise, let \( \mathcal{J} \) denote the set of all jobs involved in the current scheduling problem, \( \mathcal{O} \) the set of all operations and \( \mathcal{M} \) the set of all machines. Assuming that different jobs may comprise different operations, \( \mathcal{O}(j) \subseteq \mathcal{O} \) is defined as the set of all operations involved in processing job \( j \in \mathcal{J} \). Similarly, let \( \mathcal{M}(o) \) be the set of machines required to complete operation \( o \in \mathcal{O} \). If, for all operations \( o \in \mathcal{O} \) it holds that \( |\mathcal{M}(o)| = 1 \), then the problem involves dedicated machines. Otherwise, if for all operations \( o \in \mathcal{O} \) it holds that \( \mathcal{M}(o) = \mathcal{M} \), the problem involves parallel machines. In the case where the same machine \( m \in \mathcal{M} \) is also used in two different operations (not necessarily from the same job), i.e. \( m \in \mathcal{M}(o_1) \) and \( m \in \mathcal{M}(o_2) \), with \( o_1 \neq o_2 \), then multi-purpose machines are involved and each machine is equipped with the appropriate tools to handle different operations. Scheduling...
problems involving multi-processor tasks entail the simultaneous use of machines $\mathcal{M}(o)$, with $|\mathcal{M}(o)| > 1$, by an operation $o \in \mathcal{O}$.

Due to the vast number of different scheduling problem types, a classification scheme was developed over time [31]. The classification proceeds according to a three-field parameter list $\alpha|\beta|\gamma$ and is especially useful in discussions related to the computational complexity of different types of scheduling problems, see e.g. [12]. The first parameter in the three-field classification, $\alpha = \alpha_1\alpha_2$, is used to distinguish between scheduling problems having different machine environments. For instance if $\alpha_1 = \circ$, it is an indication that each job must be processed on a dedicated machine, i.e. $|\mathcal{M}(o)| = 1$ for a given operation $o \in \mathcal{O}$. Otherwise, if $\alpha_1$ is assigned a value in the set $\{P,Q,R\}$, it is an indication that parallel machines are involved. More specifically, $P$ represents the use of identical parallel machines (i.e. processing different operations at different speeds), and $R$ represents the use of unrelated parallel machines (i.e. processing different operations of different jobs at different speeds). Assigning the values $P_{MPM}$ or $Q_{MPM}$ to $\alpha_1$ distinguishes between multi-purpose machines having identical and uniform speeds, respectively. For $\alpha_1 \in \{\circ, P, Q, R, P_{MPM}, Q_{MPM}\}$ it is assumed that each job $j \in \mathcal{J}$ comprises only one operation, i.e. $|\mathcal{O}(j)| = 1$. The parameter $\alpha_2$ is used to specify the number of machines involved. For $\alpha_2 = \circ$ an arbitrary number of machines is assumed.

The second parameter in the parameter list, $\beta = \beta_1\beta_2\beta_3\beta_4\beta_5\beta_6$, is used to classify problems according to the job characteristics. That is, $\beta_1$ indicates whether or not preemption (job splitting) is allowed. The parameter $\beta_2$ is used to describe the precedence relationship between the different jobs. Furthermore, $\beta_3$ is used to specify whether release dates exist for the different jobs and $\beta_4$ indicates the required processing times of the different operations. The parameter $\beta_5$ specifies any deadlines on the different jobs and $\beta_6$ indicates whether or not the problem at hand involves the processing of jobs in batches. For a thorough explanation and accompanying examples on using these different parameters for classification, the reader is referred to [12].

The third parameter $\gamma$ relates to the specific objective function used for solving the scheduling problem. For instance, by letting $c_j$ be the completion time of a job $j \in \mathcal{J}$, the objective function $\gamma = \sum C_j = \min \sum_{j \in \mathcal{J}} c_j$ represents the minimisation of the total processing time of the scheduling problem. Or alternatively, the minimisation of the makespan of the schedule is specified by $\gamma = C_{\text{max}} = \min \{\max \{c_j | j \in \mathcal{J}\}\}$.

### 2.2 Computational complexity

Despite the significant progress made in solving difficult computational problems using advanced computer technology, the solution of certain problems of practical size remains a challenge. The term computational complexity may, therefore, loosely be described as meaning the effort required in solving computational problems, measured in computing time. The question is, however, whether the problem is inherently difficult to solve or whether, perhaps, it is being solved using an inefficient algorithm. A theoretical framework, based on the hypothetical Turing Machine [83], is used as a standard to facilitate the analysis of algorithms in general. The framework has allowed researchers over many decades to distinguish mathematically between “easy” and “hard” problems. It should be noted that proving computational complexity is an ongoing endeavor and that there are still many computational problems for which no verdict has been reached.

For the purpose of subsequent discussions it suffices to distinguish between a problem description, a problem instance and a problem formulation. A problem description provides details
about the problem to be solved. For example, the introductory section of this chapter can be considered as a problem description of the generic scheduling problem. It describes the typical input data, the relationship between the data elements, and the criteria for what constitutes a solution. In fact, the classification scheme outlined above serves as a very compact problem description. A problem instance, on the other hand, is the actual data to be used for solving a scheduling problem. It should be noted that for a specific problem description there may be several problem instances. Within the context of underground mine scheduling, the problem description of finding optimal schedules remains the same, even if a scheduling algorithm is applied for different mines, i.e. different problem instances. A problem formulation is considered to be the mathematical description of the problem. More formally, it is the mathematical model in terms of variables and constraints. In the remainder of this thesis, references to a problem formulation and a model will be used interchangeably.

In order to compare one scheduling algorithm with another in terms of computational efficiency, a hypothetical computer is assumed which performs elementary instructions in unit time. That is, all basic instructions performed by the computer program in the execution of an algorithm are assumed to require the same amount of time. Instead of measuring absolute computing time of an algorithm, a performance measure of an algorithm is obtained by relating the number of elementary instructions used in executing the algorithm, to the size of the problem instance. Consider, for example, the multiplication of 983 by 127 as a computational problem. The “high school” algorithm would require three multiplication steps involving a single digit by a number, followed by three additions involving \((2 \times 3)\)-digit numbers (the effect of shifting). The three multiplication steps require a total of \(3^2\) instructions and the three addition steps require a total of \(2(3)^2\) instructions. That is, \(3^2 + 2(3)^2\) instructions. If the multiplication is performed with two 4-digit numbers, then it would require a total of \(4^2 + 2(4)^2\) instructions. That is, in general, at most \(n^2 + 2(n)^2\) number of steps are required to perform the multiplication algorithm. The result is that the “running time” of the multiplication algorithm is now expressed as the function \(g(n) = n^2 + 2n^2\), with the number of digits \(n\) referred to as the size of the problem instance. It should be noted that since one is not interested in the absolute running time of an algorithm, but only in the increase in running time with respect to the size of the problem instance, the term in \(g(n)\) with the highest growth rate is of concern. That is, it suffices to express the worst-case running time of the multiplication algorithm as \(f(n) = n^2\). The relationship between the function \(g(n)\) and \(f(n)\) is formalised through the big \(O\) notation.

**Definition 2.1.** Let \(f(n)\) and \(g(n)\) be functions mapping positive integers \(n\) onto the positive real numbers. If there exists a constant \(c > 0\), such that \(f(n) \leq cg(n)\) for a large enough \(n\), then \(f(n) = O(g(n))\).

Using the big \(O\) notation, the complexity of the above multiplication algorithm is reported to be of \(O(n^2)\). It should be noted that other multiplication algorithms exist which exhibit different complexity properties. For instance, a very naive algorithm is to simply add the value 983 to itself 127 times. The complexity of this algorithm is of \(O(n10^{n-1})\), obviously much worse than that of the high school algorithm. In general, algorithms that have a complexity of at most polynomial order are considered tractable, i.e. practically solvable. It is customary to refer to these algorithms as polynomial-time algorithms.

Complexity theory assists in evaluating whether it is worthwhile to seek for a polynomial-time algorithm, or to accept that a problem is inherently hard and that no polynomial-time algorithm exists. The classification of a computation problem according to its complexity requires analysing the decision version of the problem. For instance, a decision version of the scheduling problem \(P|prec|C_{\text{max}}\) is to determine whether a feasible solution exist such that \(C_{\text{max}} \leq K\) for some arbitrary \(K\). That is, the answer to a decision problem is either “yes” or “no”. It should be
noted that some computational problems are already in a decision format, e.g. given the numbers $x$ and $y$, is $y$ divisible by $x$?

**Definition 2.2.** Let $\mathcal{P}$ denote the class of all decision problems that are polynomially solvable.

Working with the decision version of an optimisation problem, e.g. the one described above for the problem $P|\text{prec}|C_{\text{max}}$, turns out to be just as difficult as solving the original problem. Therefore, it can be shown that, if no polynomial-time algorithm exists for solving the decision version of an optimisation problem, then clearly such a problem cannot be in $\mathcal{P}$. In order to classify these problems, the existence of a problem instance is required for which a “yes” answer is certain. That is, for the scheduling example, a feasible solution for which $C_{\text{max}} \leq K$ is a certificate that can be tested for validity.

**Definition 2.3.** Let $\mathcal{NP}$ denote the class of all decision problems for which their certificates can be tested for validity in polynomial-time.

It should be clear that $\mathcal{P} \subseteq \mathcal{NP}$ since testing the validity of the certificate of a decision problem, which is in itself polynomial time solvable, would also require at most a polynomial number of steps. There exists another subset of $\mathcal{NP}$, which is not in $\mathcal{P}$, which is called the set of $\mathcal{NP}$-complete problems. To formalise this complexity class, the concept of polynomial reduction is introduced.

**Definition 2.4.** Let $D_1$ and $D_2$ be decision problems with $d_1$ a problem instance of $D$ that would result in a “yes” answer. Then, $D_1$ reduces to $D_2$, if there exists a polynomial-time algorithm $g$ such that $d_2 = g(d_1)$ is a problem instance of $D_2$ also resulting in a “yes” answer.

**Definition 2.5.** A decision problem $D \subseteq \mathcal{NP}$ is $\mathcal{NP}$-complete if any problem in $\mathcal{NP}$ reduces to $D$.

The significance of the $\mathcal{NP}$-complete class is that if a problem $Q \in \mathcal{NP}$-complete could be solved in polynomial time, then all problems in $\mathcal{NP}$ are solvable in polynomial time.

**Definition 2.6.** $\mathcal{NP}$-hard denotes the class of all optimisation problems for which their decision versions are $\mathcal{NP}$-complete.

In order to prove that a given optimisation problem is $\mathcal{NP}$-hard, it needs to be shown that an existing $\mathcal{NP}$-complete problem reduces, according to Definition 2.4, to the decision version of the given optimisation problem. The ground work for proving the complexity of problems was done by [21], who demonstrated that the well-known combinatorial problem, satisfiability (SAT), is
2.3. The resource constrained project scheduling problem (RCSP)

NP-complete. Subsequently, proving the complexity of many other optimisation problems was achievable by showing that SAT reduces directly, or indirectly, to these problems. Figure 2.1 is an example of the reduction relationship between SAT and other well-known problems proven to be NP-complete.

Examples 3.3 and 3.4 from [12] illustrate the use of reduction to show that the scheduling problems F3||Cmax and P2|prec; pj ∈ {1, 2}|Cmax are NP-hard, by showing that the Partitioning problem is reducible to the decision version of F3||Cmax and the Clique problem is reducible to the decision version of P2|prec; pj ∈ {1, 2}|Cmax.

2.3 The resource constrained project scheduling problem (RCSP)

In the remainder of this thesis, to be more in-line with current literature on resource constrained scheduling, reference will be made to the scheduling of activities rather than the scheduling of jobs. Let A denote the index set of all activities and let di be the duration of an activity i ∈ A. The order in which the activities have to be scheduled is specified by means of a directed acyclic graph, called a precedence graph. From this graph we can derive the index set P(i) ⊆ A as the set of all predecessor activities of i ∈ A. The execution of an activity implies that one or more resources will be consumed. For this purpose R is defined as the index set of all resources and vr as the quantity of resource r ∈ R being consumed by activity i ∈ A. The availability of resources is in most cases restricted and, therefore, Ur is defined as an upper limit on the consumption of resource r ∈ R.

Finding a solution to the RCSP entails determining starting times for each of the activities such that all precedence rules are obeyed and resource usage per time period is within the specified limits. For that purpose si ≥ 0 is defined as the starting time of an activity i ∈ A. It should be noted that, depending on the specific formulation approach (see Section 2.3.1 below), the starting time si does not necessarily have to coincide with the start times of the scheduling periods. Furthermore, to facilitate a review of the literature below, involving different ways of formulating resource constraints, the generic notation V(i, r, t) is used to express the quantity of resource r ∈ R being consumed by activity i ∈ A during time period t ∈ T. That is, the function V(i, r, t) depends on the solution obtained for the starting time si of an activity, the consumption vr of the activity and the duration di of the activity. Consider the following example with one activity i = 1 having a duration d1 = 2 (days) and a resource consumption vr1 = 1 for some resource r. Consider a scheduling horizon of three days, i.e. T = {1, 2, 3}. If the solution to this schedule is s1 = 0, that is, the activity is scheduled to start at the beginning of period t = 1, then V(1, r, 1) = vr1/d1 = 1/2 since the activity stretches over two days and for the first day it consumes half of the resource (over the total duration it consumes vr1 = 1). Consequently, the other half of the resource is consumed during period t = 2, so that V(1, r, 2) = 1/2.

A conceptual formulation of the RCSP when minimising the makespan, is to

\[
\begin{align*}
\text{minimise} & \quad \max_{i \in A} \{s_i + d_i\} \\
\text{s.t.} & \quad \sum_{i \in A} V(i, r, t) \leq U_r, \quad \forall \ r \in R, \ \forall \ t \in T, \\
& \quad s_i \geq 0.
\end{align*}
\]

If provided with cash-flow values c_i for each activity i ∈ A and a discount rate α, the objective
function of the RCSP, in the case of maximising net present value (NPV), is to

$$\text{maximise } \sum_{i \in A} e^{-\alpha s_i} c_i.$$  \h[2]

It was shown in [30] that the RCSP, with a single resource and no precedence constraints, is \( \mathcal{NP} \)-hard. The intuitive expectation that the RCSP with precedence constraints would also be hard is confirmed by [10] and by [48].

The complexity status of the RCSP necessitates the exploration of different algorithmic approaches to either improve lower bounds obtained through the relaxation of the original problem, or improve feasible solutions obtained through the use of heuristic approaches, or both. In the next section, two classical problem formulations are provided for modelling the RCSP, followed by a literature overview of recent algorithmic contributions towards solving the RCSP.

### 2.3.1 Mathematical formulations

Finding solutions to the RCSP, whether approximate or optimal, requires an explicit mathematical model. The use of mixed integer linear programming (MILP) as a modelling approach is well suited for the formulation of the RCSP due to the logical decision-making nature of the problem. It should be noted that several different mathematical formulations may exist to address the same problem. These different formulations may be equivalent in terms of representing the feasible region and the objective function of the RCSP, but they may differ in the number of variables and constraints, as well as the efficiency of the underlying algorithm in finding solutions to these models. In the literature, three main classes of RCSP formulations can be found, namely time-indexed formulations, resource flow-based formulations, and event-based formulations.

Time-indexed formulations are based on the discretisation of time. Binary variables, indexed by both an activity and a time period, are used to indicate at which time period each activity will start. An obvious drawback of this approach is that a fixed time horizon is required which may result in an exponential number of variables. Several alternative time-indexed formulations exist, of which the earliest is by [66]. Their formulation is probably one of the most simplistic, catering for arbitrary precedence and resource availability constraints. The formulation by [56] is obtained by considering only feasible sets of activities per time period. An auxiliary variable is introduced to select from these feasible sets only the one that would result in an optimal solution.

In a resource flow formulation, the resource consumption by activities is modelled as a network flow problem. That is, continuous variables are defined that represent the flow of a resource from one activity to the next [5]. In [4], the start time of an activity is modelled as a continuous variable, while binary decision variables are required to fix the ordering of the activities. The formulations in both the aforementioned papers were driven by an algorithmic approach rather than by an application requirement. In [45] a resource flow formulation is adopted to model the transfer of resources among multiple projects, while in [68] a resource flow formulation is used to model delays for when resources are transferred from one activity to another.

Event-based formulations of the RCSP rely on the fact that the end time of an activity, or a set of activities, coincides with the start time of another activity, or set of activities. This point in time is called an event. In [2] and [42], continuous variables are introduced to keep track of the time of each event and binary variables are used to indicate at which event each activity will start and end, respectively. The total number of events is therefore bounded by the number of
activities plus one. An earlier version of the event-based formulation involving more variables is given by [89].

Due to the large number of practical applications of the RCSP, the literature available on the topic is vast. Many variants of the RCSP have been developed over the years as a result of having to solve scheduling problems with application-specific requirements. The survey by [40] is a good source to help navigate through the many variants and extensions to the basic RCSP. Specific attention is given to objective functions of the RCSP that involve makespan minimisation and NPV maximisation, and the close relationship between their models and approaches. In the survey by [33] the distinction between some variations of the classical RCSP is made according to the following criteria:

1. **Activity related attributes.** Scheduling problems are described to either have splittable or non-splittable activities (that is, preemptive vs. non-preemptive scheduling). The latter implies that once an activity has started, it may not be interrupted until it has been completed. Different forms of preemptive scheduling exist, e.g. continuous or preemption at discrete points in time.

   The conceptual RCSP formulations above all assume a constant resource consumption rate \( v_{ri} \) per time period of a resource \( r \) by an activity \( i \). The alternative is to allow for a varying resource consumption rate per period. This can simply be achieved by introducing a time index to the resource consumption parameter, i.e. \( v_{rit} \).

   Associated with certain types of activities are setup times. These are of specific concern in production-related problems where machines need to be configured prior to being able to execute an activity. The most basic case is to simply alter the duration of an activity in order to take setup times into account.

   For the classical RCSP a constant resource efficiency is assumed through the parameter \( v_{ri} \), such that an activity will be completed within the specified duration \( d_i \). In multi-mode formulations, each mode of an activity will dictate a different duration and a different resource consumption. Part of the optimisation problem then becomes the selection of an optimal mode for each activity.

2. **Resource-related attributes.** Constraint (2.2) above allows a specific resource \( r \in R \), called a cumulative resource, to process more than one activity at a time. A disjunctive resource can only process one activity at any given time, and this can be achieved by simply requiring that \( v_{ri} = U_r \) for all \( i \in A \).

   Resources can also be classified as renewable and non-renewable. The resource constraint (2.2) where \( U_r \) is the total resource available during any given time period \( t \in T \) is an implementation of a renewable resource. That is, on consumption of the resources at the end of period \( t \), the resource is renewed at the beginning of time period \( t + 1 \). In the case of a non-renewable resource, the total available resource is cumulatively consumed until depletion.

   The resource constraint, as stated in (2.2) above, implies the same resource availability for each time period. The alternative is to allow varying resource availability per time period. This can simply be achieved by introducing a time index to the resource parameter, i.e. defining \( U_{rt} \) as the total amount of resource \( r \) available at a given time \( t \).

3. **Scheduling-related attributes.** The precedence relationship in the classical RCSP does not allow for any lags between the completion of one activity and the start of its successor. Minimal time lag constraints can be introduced to force a successor activity to be delayed. This is, of course, another way to cater for machine setup times. Maximal time lag
constraints, on the other hand, introduce flexibility to the schedule by allowing a certain amount of delay, if required by the final solution.

The earliest time at which an activity is allowed to start may be implemented by a release date. That is, even if such an activity’s predecessor has been completed, the activity may only start after the release date. Implementing a deadline for an activity would force it to be completed before the deadline date, provided its predecessor was completed in time.

The concept of time-switching constraints can be introduced to cater for periods of work and rest within a schedule.

Constraints can be implemented to enforce activity-specific policies. For instance, it may be required that two activities may not start/finish at the same time, or conversely, that the two activities should always start or always finish simultaneously. An extension to this is the modelling of logical operators related to, e.g. multiple predecessors. For instance, for a logical “AND” operator, all predecessors to an activity must be completed before the activity itself is allowed to start.

Constraints related to resource transferring may be implemented to capture the delay effects due to the transfer of resources from one location to another. As an extension to this approach, the transfer itself may be represented by another activity, which may in turn consume additional resources.

4. **Objective function alternatives.** The minimisation of the makespan is an example of a time-based objective function. Other variations on this theme include the minimisation of lateness or the maximisation of earliness, or possibly even a combination of several time-based measures.

Objective functions with the aim of improving the robustness of a scheduling solution are referred to as robustness-based objective functions. One way of deriving a robust schedule is the maximisation of the minimum free slack between the activities. An alternative form is proactive scheduling, which entails the scheduling of backup activities in the event of unplanned delays, e.g. machinery replacement in the case of breakdowns.

In contrast to proactive scheduling, rescheduling objective functions are used in cases where the project is already under way, but due to unforeseen circumstances, rescheduling of the remaining activities is required with minimal deviation from the original plan.

Renewable and non-renewable resource-based objective functions are driven by (non-)renewable resource-oriented decisions. For example, by associating a unit cost for each resource, a resource investment problem can be formulated that would minimise total (non-)renewable resource cost, provided that a deadline or due date is achieved by the resulting schedule. Resource leveling or smoothing is another important criterion in calculating schedules in an attempt at maintaining a certain level of resource usage.

The presence of cash flows over time, negative or positive, warrants the use of net present value-based objective functions. This is of particular interest in scheduling problems for which both costs and revenue generating activities are scheduled. The maximum net present value for a schedule would dictate the acceleration of revenue generating activities and the delay of cost generating activities.

Finally, using a single type of objective function might be considered impractical, motivating the use of multiple objective functions.
2.3.2 Algorithmic approaches

Despite the fact that the complexity status of the RCSP has been established as $\mathcal{NP}$-hard, continued efforts in developing new algorithms have facilitated the progressive solution of larger and more difficult problem instances, as evidenced by the success in solving benchmark instances over time. The project scheduling problem library [67] was created as a repository of RCSP problem instances and has been referenced extensively over the years to benchmark newly developed models and algorithms. In addition to the problem instances themselves, information is recorded and maintained on optimal solutions or best bounds found thus far. Currently there are four data sets called J30, J60, J90 and J120 comprising RCSP instances with respectively 30, 60, 90 and 120 activities. Each data set contains 480 different problem instances, except for the J120 data set which contains 600 problem instances. Details on how these problem instances were created can be found in [41]. Although significant progress has been made in solving many of the instances in PSPLIB to optimality, including instances from the J120 data set, at this point in time there are still no optimal solutions for 49 of the J60 problem instances. All of the J30 problem instances have been solved to optimality.

Algorithmic approaches towards solving the RCSP are in general either exact or heuristic. An exact approach has a finite running time and will terminate with an optimal solution, provided the problem being solved has at least one feasible solution (see Section 1.5 for an overview of optimality conditions and solution bounds). The philosophy adopted in this thesis is that, although proven optimality may be seen as a luxury in practice, the continuous efforts in devising exact algorithms are important in terms of understanding problem-specific properties that may lead to the development of improved approaches, whether exact or heuristic. Furthermore, from a practical point of view, the solvers used for solving general MILP problems provide solution bounds when prematurely terminated, which serve as a quality certificate for the incumbent feasible solution. In a mine scheduling optimisation context, the ability to have a quality guarantee on a solution allows decision makers to be confident in corporate decisions involving billions of Rands. Therefore, in this literature overview, the focus will be on work related to exact approaches.

Most of the exact methods reported for the RCSP are based on either the general branch-and-bound method, proposed by [47], for solving binary linear programming problems, or on implicit enumeration methods incorporating customised branch-and-bound schemes. In the latter approach, implicit enumeration involves the exploration of a branch-and-bound tree, with each node in the tree corresponding to a partial scheduling solution. Pruning of the branch-and-bound nodes is done based on dominance rules, see for example [37] and [73]. Various extensions to the basic enumeration scheme have since been proposed, such as using a breadth-first approach [80] or a depth-first approach [20] for selecting a new node to be explored. Other state of the art customised branch-and-bound schemes include, for example, the approaches by [14, 25, 46, 56]. Customised branch-and-bound algorithms for specifically maximising NPV were suggested by, amongst others, [27], [86] and [88].

The early popularity of customised branch-and-bound schemes over the more general binary programming branch-and-bound method, was a result of the inefficiency of the latter approach [59]. Technological advances and the growing maturity of mathematical programming methodologies has since changed the situation. Computational results of [42] show improved computing times of the binary programming branch-and-bound method over the MCS exact approach by [46]. They also show that for problem instances with a short time horizon, the time-indexed formulation of the RCSP outperforms both the continuous-time and event-based formulations. In the case of longer time horizons, either the MCS or event-based formulations appear to be
superior, depending on the specific problem instance. A clear advantage over customised branch-and-bound schemes is an ability to solve the RCSP as a binary linear programming problem, while taking other “business-related” side constraints into account. For instance, binary programming formulations have been proposed for maximising NPV while taking constraints on capital expenditure [26] and material use [79] into account.

More recent advances within an exact framework include the use of constraint programming approaches. Significant progress in solving many of the open PSPLIB instances is due to [34]. A satisfiability solver was employed for solving the RCSP problem and it managed to solve 80 of the open J60 instances and 44 open J90 instances to optimality. In [76] it was reported that by using a lazy clause generation approach, they were able to close a total of 60 open instances, of which 20 are from the J120 data set. A similar approach was employed by [75], who managed to solve 631 open instances of the PSPLIB, for the RCSP with generalised precedence constraints. A custom branch-and-bound scheme based on constraint propagation was proposed by [49] which makes use of the resource flow formulation of [4].

An approach is generally considered to be a heuristic if it does not terminate with a proven optimal solution. There are, however, approaches that do terminate with upper and lower bounds that give a quantification of the quality of feasible solutions. The Lagrangian relaxation-based heuristic by [58] falls within this category. In their approach, the resource constraints for the RCSP are relaxed and the resulting Lagrangian subproblems are solved as minimum cut problems. Solution information from these subproblems are then used by list scheduling algorithms to generate primal solutions. In [7], a Lagrangian relaxation approach is adopted for computing lower bounds of the resource-constrained modulo scheduling problem. Although the formulation by [56] may be hampered by an exponential number of feasible sets, their work on the derivation of lower bounding procedures have paved the way for very successful approaches by others, see e.g. [8, 13]. By stronger formulation of the problem, or the application of valid inequalities to the LP relaxation, improved lower bound solutions can be obtained. For instance, it was shown by [84] that the disaggregated version of the precedence constraints by [20] improved the lower bound by 75% for a specific problem instance. In [24] a constraint propagation approach is used to derive valid inequalities for both sequence-based and time-indexed formulations within a cutting plane framework. Instead of only strengthening the initial LP relaxation, a branch-and-cut approach can be followed whereby the separation of valid inequalities is performed during the processing of the branch-and-bound nodes [65]. In [90] a branch-and-cut algorithm is used to solve the multimode RCSP while [77] use it to solve a process scheduling problem.

### 2.4 Summary

The development of scheduling algorithms has been an ongoing endeavour since the formalisation of the critical-path scheduling problem during the late 1950s [39]. Despite the significant progress made over the years in solving bigger scheduling problem instances, the ability to obtain solutions for industrial-sized problems remain a challenge. In this chapter an overview of the theory of computational complexity was provided, which provides a useful framework to distinguish mathematically between “easy” and “hard” problems.

The majority of problems encountered in industry that involve the scheduling of activities, are most likely special cases of the RCSP, since the consumption of resources is typically involved and the order in which activities are scheduled to start is determined by a precedence graph. In this chapter, an overview of the different options in formulating the RCSP was provided as part
of a literature study.

The use of heuristics may be considered as an attractive approach for solving the RCSP due to its classification as being \( \mathcal{NP} \)-hard [10]. The emphasis in this study is, however, on exact solution approaches due to the benefit of being able to add generic constraints to the RCSP formulation and being able to obtain a certificate on solutions calculated as part of the branch-and-bound approach. Furthermore, algorithmic enhancements proposed in later chapters show that it is possible to obtain good solutions for the RCSP in reasonable time, despite its complexity status.

The application of the RCSP within a underground mining context is discussed in more detail in the next chapter.
CHAPTER 3

The underground mine scheduling optimisation problem

Contents

3.1 Technical aspects of underground mine scheduling . . . . . . . . . . . . . . . . . 21
3.2 Solving the mine scheduling optimisation problem as an RCSP . . . . . . . . . 25
3.3 Practical considerations in mine scheduling optimisation . . . . . . . . . . . . . 28
3.4 Modelling requirements beyond the RCSP framework . . . . . . . . . . . . . . . 29
3.5 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

Excavation activities in mining are very costly and in order to maintain profitability, optimal use of resources is imperative. This may be achieved firstly, by extracting high-grade minerals through proper mine layout designs, and secondly, by the optimal scheduling of mining activities that would maximise net present value (NPV), while taking mine-specific constraints into account. These two tasks are typically treated as disjoint from each other and the mine layout is used as input to the scheduling process. Several commercial software packages are available for integrating mine layout design and scheduling within a 3D CAD environment [55].

In this chapter, a brief overview of the technical aspects of underground mine scheduling is provided. A comprehensive discussion of different mining methods and geological modelling is, however, beyond the scope of this thesis. With a basic understanding of an underground mining operation, various modelling assumptions will be presented that will facilitate the formulation of mathematical models in later chapters.

3.1 Technical aspects of underground mine scheduling

A concentration of underground mineral deposits is commonly referred to as an ore body. Depending on the characteristics of an ore body, the extraction of minerals is done according to specific mining methods. The primary distinction to be made is open pit mining and underground mining. In the case of open pit mining, the ore body is usually very localised and close to the surface whereas with underground mining, access to the ore body is gained through underground shafts and tunnels. In typical South African gold and platinum mines, the ore body forms a reef that may in some cases be only a few centimeters thick, and depending on its inclination with the surface, could reach depths of up to several kilometers.
Chapter 3. The underground mine scheduling optimisation problem

Figure 3.1: A typical underground mine layout.

The type of underground mining operation to be considered in this study is for a typical South African gold or platinum reef, as depicted in Figure 3.1. Main access to the reef is obtained by a vertical tunnel called a \textit{shaft}. Horizontal tunnels, called \textit{haulages}, are excavated from the shaft in order to provide access to the reef. Haulages running along the plane of the reef at different depths are referred to as \textit{levels} and on each level several smaller tunnels, called \textit{cross-cuts}, provide access to the reef. A \textit{step-over} connects a cross-cut to a tunnel running along the plane of the reef, called a \textit{raise line}. This configuration is illustrated in Figure 3.2 which is a top view where the reef is transparent in the diagram.

Blocks of ore-bearing rock, alongside raise lines and between consecutive levels, are called \textit{stoping blocks} and are demarcated into smaller pieces to form \textit{stoping panels}. The size of the individual stoping panels is determined by the specific mining technique employed as well as by geological constraints. The activity of excavating stoping panels is referred to as \textit{stoping} whereas the excavation of tunnels giving access to the ore body is referred to as \textit{development}. A distinction is made between \textit{off-reef development} and \textit{on-reef development}, where the latter refers to the excavation of tunnels (\textit{e.g.} raise lines) within the ore body, with the result that some minerals are also mined out, but with a high dilution factor. Off-reef development, on the other hand, refers to the excavation of tunnels through waste rock to give access to the ore body.

Different mining methods can be applied for the excavation of stoping panels. Two of the most commonly used mining methods in shallow dip reef mining are \textit{sequential} and \textit{pillar} mining. For a more complete reference on different mining methods, see [35]. Figure 3.3 illustrates the differences between sequential and pillar mining. In the latter, parts of the stoping block are left behind as pillars for safety purposes, whereas sequential mining involves clearing out the entire stoping block.
Rock that has been excavated from both development and stoping panels needs to be transported all the way up the shaft. Only the ore from the stoping panels will be sent to a processing plant where the valuable minerals are extracted. The routing of rock underground, both waste and ore, is called \textit{tramming}. The different tramming routes will be dictated by the mine layout and the location of the shaft.

In reality, the rather ideal layout of a reef, as shown in Figure 3.1, is very unlikely, especially during the initial exploration phases of the ore body. As excavation efforts intensify, more information about the ore body becomes available. The main statistical tool used for estimating the grade distribution of the ore body is called \textit{Krigging} \cite{44}. The output from a Krigging model is a map of the ore body discretised into \textit{evaluation blocks}. A grade estimate is associated with each evaluation block and is calculated through a process of interpolation. The mine planning process commences with an initial mine layout that overlays the evaluation blocks as best possible in order to extract as much as possible of the valuable minerals while leaving behind most of the waste.

The next step in the mine planning process is the discretisation of the mine layout into \textit{mining activities}. Traditionally, time-based discretisation was performed to simplify the scheduling of the activities. For instance, if it takes one month to excavate ten metres of an underground tunnel, then the entire tunnel in the three-dimensional design would have been broken up into
Chapter 3. The underground mine scheduling optimisation problem

Figure 3.4: Sequencing rules for simulating a sequential mining method. The diagram on the left is for a time-based discretisation and the diagram on the right for a logical discretisation of the activities.

ten one-metre activities. Alternatively, a functional discretisation would entail the creation of logical mining segments that would need to be completed in an uninterrupted fashion. For instance, a logical mining segment may be an entire raise line, or the piece of haulage between two adjacent raise lines, etc.

Mining activities are scheduled according to strict precedence rules, commonly referred to as sequencing rules within the mining environment. The physical creation of sequencing rules typically takes place within the same 3D CAD environment in which the mine layout drawing is done [55]. The application of the sequencing rules is dictated by the mining method employed. For instance, Figure 3.4 shows the placement of sequencing rules to simulate a sequential mining method, for both time-based discretisation and logical discretisation of the activities.

With sequencing rules in place, generating a schedule entails finding starting times for each of the activities such that all precedence rules are obeyed. In order to generate schedules that are feasible with respect to mine-specific constraints and that are optimal with respect to some financially derived objective function, certain properties need to be associated with each of the activities that can be aggregated per scheduling time period. For instance, if the final schedule should be feasible with respect to the capacity of a shaft, specified in terms of tonnes per month, each of the activities should have tonnes associated with it. Depending on the starting time of each activity, the tonnes from each activity are accumulated per period and compared to the shaft capacity to test for feasibility. The final step prior to scheduling is, therefore, the assignment of the necessary properties to each of the activities.

The following physical properties are required in order to solve a capacitated mine scheduling problem:

- **Physical dimension.** This is the physical area \( (m^2) \) specified for each activity as part of the mine layout design. In the case of development, the length \( (m) \) of the activity is required and in the case of stoping, the stope-width \( (m) \) is required.

- **Rock density.** This is typically constant for an entire mine and is specified as \( t/m^3 \), with \( t \) measured in metric tonnes.

- **Tonnes.** This measure refers to metric tonnes and is derived from the physical dimension
3.2 Solving the mine scheduling optimisation problem as an RCSP

The mathematical formulation of the underground mine scheduling optimisation problem can be accommodated by the conceptual formulation of the RCSP (2.1)–(2.3) from Section 2.3. From the discussion above, it is clear that all of the data elements are available to construct the necessary parameter sets for the RCSP formulation. For instance, the set of activities $A$ is obtained through the discretisation of the mine design layout (either time-based or logical). The set of predecessors $P(i)$ for an activity $i \in A$ can be derived from the mine sequencing rules. The creation of the resources $R$ depends on the type of resource constraints that will be considered. For instance, if a shaft capacity has to be imposed in terms of allowable tonnes per month, the resource set $R$ should include “tonnes” as an element, in which case an upper limit $U_r$ should be defined for this resource. The parameter $v_{ir}$ in the RCSP formulation can now be interpreted as a quantity of resource $r \in R$ being either consumed or produced. For instance, the scheduling of each activity $i \in A$ will result in the production of the “tonnes” resource according to the quantity $v_{ir}$. That is, the value of $v_{ir}$ for an activity $i$ corresponds to the physical tonnes property from the section above, as estimated from the physical dimensions of the activity.

In order to use NPV as an objective function criterion, a cash flow per activity needs to be estimated. Within a mining environment, costs are typically allocated based on activity types. For instance, the cost of excavating a haulage is expressed as Rand per metre ($R/m$) and for stoping it is Rand per square metre ($R/m^2$). A simple way to deal with activity-based costing within the RCSP problem is to allocate cost via a resource assignment. For instance,
by extending the set of resources $\mathcal{R}$ to include “metres” as an element, a cost parameter $c_r$ is introduced to denote the unit cost of excavating one metre of a haulage. The modified objective function of the RCSP, which includes activity-based costing, is to

$$\text{maximise} \sum_{i \in \mathcal{A}} e^{-\alpha s_i} \sum_{r \in \mathcal{R}} c_r v_{ir}.$$  

The same mechanism can also be applied to account for the revenue estimates per activity. For instance, the set of resources $\mathcal{R}$ may be extended to include “mineral” as a resource, representing the kilograms ($\text{kg}$) of the specific minerals extracted from the mine. The quantity $v_{ir}$ may be used to indicate how many kilograms of the mineral are produced by activity $i \in \mathcal{A}$ based on the grade, which is one of the physical properties as described above. The mineral price is then captured as the parameter $c_r$ in the objective function (3.1).

To make these concepts more concrete, a small example is considered. Figure 3.5 shows the layout of a section of a mine level consisting of a haulage from which two cross-cuts originate. For each cross-cut, a step-over is followed by a raise line which provides access to four stoping panels.

<table>
<thead>
<tr>
<th>$i$</th>
<th>Type</th>
<th>$d_i$</th>
<th>$P(i)$</th>
<th>Off-reef$(m)$</th>
<th>On-reef$(m)$</th>
<th>Stoping$(m^2)$</th>
<th>Tonnes$(t)$</th>
<th>Mineral$(kg)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Haulage</td>
<td>200</td>
<td>-</td>
<td>180</td>
<td>0</td>
<td>0</td>
<td>4500</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Cross-cut</td>
<td>120</td>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2800</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Step-over</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Raise line</td>
<td>180</td>
<td>3</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>3000</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Stope panel</td>
<td>250</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Stope panel</td>
<td>250</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>Stope panel</td>
<td>250</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Stope panel</td>
<td>250</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Haulage</td>
<td>200</td>
<td>1</td>
<td>180</td>
<td>0</td>
<td>0</td>
<td>4500</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Cross-cut</td>
<td>120</td>
<td>9</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2800</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>Step-over</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>Raise line</td>
<td>180</td>
<td>11</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>3000</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Stope panel</td>
<td>250</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1500</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>Stope panel</td>
<td>250</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>Stope panel</td>
<td>250</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>Stope panel</td>
<td>250</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>1500</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 3.1: Activity input values for the small example considered.

By following a logical discretisation approach of the layout depicted in Figure 3.5, a total of 16 activities are obtained. A capacity restriction of 750 tonnes per month is assumed for this small section of the mine layout and the objective is to find a schedule that will maximise NPV. For this purpose it is assumed that the cost of doing off-reef development is R10 000 per metre, the cost of on-reef development is R5 000 per metre, the cost of stoping is R3 000 per square metre and the current mineral price is assumed to be R500 000 per kilogram. From this information it is clear that several resources will have to be defined, while the production values $v_{ir}$ will need to be specified for each activity-resource combination. Table 3.1 lists the sixteen activities and the first four columns of the table correspond to the activity identification $i$, the type of the activity, the duration $d_i$ and the predecessor $P(i)$ of the activity, respectively. For this example only a single predecessor is assumed, whereas in reality there could be an arbitrary number of predecessors. The remaining columns correspond to resources and each cell entry in the table corresponds to the production $v_{ir}$ of a resource $r \in \mathcal{R}$ by an activity $i \in \mathcal{A}$. It is important to
3.2. Solving the mine scheduling optimisation problem as an RCSP

note that not all activities produce the same resources. For example, the first row in the table corresponds to the first haulage segment in Figure 3.5. Since this haulage segment is classified as an off-reef development type, there is an entry for it under the resource “Off-reef(m)”, but no entries (zeroes) under the resources “On-reef(m)”, “Stoping(m^2)” and “Mineral(kg)”. The only exception is for the resource “Tonnes(t)” since there is a capacity specified for it and the tonnes produced by all of the activities have to be aggregated and tested for feasibility. The only activities contributing to the production of the “Mineral(kg)” resource are the stoping panels and the raise lines.

In order to capture the production limits on each of the resources, as well as the financial parameters necessary for an NPV calculation, resource data are introduced in Table 3.2. The first column lists the resources \( r \in R \) as they appear in Table 3.1 as the column headings. The unit costs or revenues per resource \( c_r \) are provided in the second column. The negative entries denote costs whereas the positive entries denote revenue. The entries in the last column \( U_r \) provide the upper limits for each of the resources to be used in the resource capacity constraints.

<table>
<thead>
<tr>
<th>( r )</th>
<th>( c_r )</th>
<th>( U_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off-reef Dev(m)</td>
<td>-10 000</td>
<td>infinity</td>
</tr>
<tr>
<td>On-reef Dev(m)</td>
<td>-5 000</td>
<td>infinity</td>
</tr>
<tr>
<td>Stoping(m^2)</td>
<td>-3 000</td>
<td>infinity</td>
</tr>
<tr>
<td>Tonnes(t)</td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>Mineral(kg)</td>
<td>500 000</td>
<td>infinity</td>
</tr>
</tbody>
</table>

Table 3.2: Resource input values for the small example considered.

The underground mine scheduling optimisation problem, as captured by the information in Tables 3.1 and 3.2, can now be formulated conceptually as

\[
\text{maximising } \sum_{i \in A} e^{-\alpha s_i} \sum_{r \in R} c_r v_{ir} \quad (3.2)
\]

s.t. \[ \sum_{i \in A} V(i, r, t) \leq U_r, \quad \forall \ r \in R, \quad \forall \ t \in T, \quad (3.3) \]

\[ s_i \geq 0. \quad (3.4) \]

The explicit formulation of the above problem will be given in the next chapter. For now it suffices to know that this problem can be solved to optimality using standard mathematical programming technology. The most important result expected by a mine planner from an automated scheduling approach is the graphical display of the scheduling solution and production profiles. These are used by the mine planner for validating the solution in terms of feasibility. If the mine planner is satisfied that the solution constitutes a feasible schedule, the next consideration would be the value of the solution in terms of NPV. A solution to the above optimisation problem comprises starting times \( s_i \) for each activity \( i \in A \). Using these starting times together with the duration of each activity provided as input, a Gantt chart can be created, as illustrated in Figure 3.6.

The graphical nature of a Gantt chart makes it a valuable tool for accessing the feasibility of scheduling solutions. For example, Figure 3.6 shows that all predecessor rules are obeyed, since for this particular section, no parallel activities can occur from a cross-cut (activities 2 and 9) onwards, but the two cross-cuts and their branches may be executed simultaneously. Also interesting to observe is that in order to maximise NPV, it is suggested that the stoping panels that are accessed via cross-cut 9 should be excavated sooner than the stoping panels that are
accessed via cross-cut 2. This is in line with the input data which shows that there is less mineral content in the latter. That is, the optimisation model prefers to incur revenue as soon as possible and delay costs for as long as possible, which is exactly what one would expect from a function that maximises NPV.

In addition to validating the solution against the given predecessor rules, the aggregated resources produced per time period by all of the activities should be checked against the resource upper limits. Specifically, for our example, the constraint of 750 tonnes per month for the “tonnes” resource, as specified in Table 3.2, may not be exceeded. The graph depicted in Figure 3.7 shows the “tonnes” resource aggregated per month for all of the activities. The graph shows that the capacity constraint of 750 tonnes per month is satisfied by the optimal solution for the small example.

3.3 Practical considerations in mine scheduling optimisation

For the purpose of operational and strategic planning by mine management, a production report is created from a scheduling solution that is expressed in terms of the starting times $s_i$ for each activity $i \in \mathcal{A}$. Depending on the planning horizon, production profiles such as the one in
Figure 3.7 can be generated for different time period sizes, i.e. for different scheduling calendars. For strategic decision making, reporting typically occurs according to either annual or monthly scheduling calendars, whereas for operational purposes, reporting usually occurs according to a higher time period resolution such as weekly or even daily calendars. In terms of the conceptual underground mine scheduling model (3.2)–(3.3), the choice of scheduling calendar should only affect reporting and not the solutions that are obtained for the starting times \( s_i \). This is, however, not the case for certain explicit mathematical formulations of the RCSP. If a time-indexed formulation is used (see the literature overview in Section 2.3.1), the starting times \( s_i \) are forced to align with the starting times of the calendar time periods.

When using a scheduling calendar with shorter time periods for reporting, it is not required that the discretisation of the mine layout be performed according to a higher resolution. If an activity is scheduled, which overlaps one or more periods, the contribution of the activity to each smaller time period may be calculated proportionately by considering the duration of the activity and its starting time relative to the said time period. There is, however, benefit to a finer grained discretisation, specifically in cases where high variability in the grade distribution is observed. In such cases the optimisation model can pick out parts of a stoping panel with high mineral content first and delay the excavation of the less valuable parts until later. This will, of course, have a positive effect on the NPV calculation. The other benefit of a finer grained discretisation is the ability to interrupt activities. For example, by breaking up a haulage into smaller pieces, its excavation can be delayed midway if insufficient capacity is available and the NPV for that period can be boosted by rather excavating a stoping panel.

In the example presented above, only a single revenue generating resource was considered, namely the “mineral” resource in Table 3.2. In reality, more than one mineral is typically processed at a mine, with each mineral having its own grade and price. This can easily be accommodated within the RCSP framework by simply adding a resource for each mineral. Furthermore, although no mention was made of plant processing costs in the above example, it is possible to accommodate such costs by introducing a resource dedicated to mineral bearing ore sent to the plant. Only mineral-producing activities like stoping and on-reef development will contribute to this resource. In addition to processing costs \( (R/t) \), plant capacity \( (t/\text{month}) \) can be specified as the resource upper limit.

Several operational constraints need to be considered in scheduling mine production activities. For instance, there is a tramming capacity per level which limits the volume of rock that can be transported through the haulages. This can simply be achieved by adding a “level” resource such that all activities belonging to this level contribute to the volume of rock for this resource. Since each level will have its own resource, a capacity \( (t/\text{month}) \) can be specified as the resource upper limit.

3.4 Modelling requirements beyond the RCSP framework

It is evident that the RCSP framework is able to incorporate several operational requirements through the use of generic resources. There are, however, some modelling requirements that do not fit nicely into this framework and which need to be addressed through the use of standard modelling approaches. An incomplete list of some of these requirements are provided below.

- **Fixed vs. variable cost splitting.** The RCSP resource framework, as illustrated by the small example above, facilitates unit costs per resource. That is, only a variable cost component is considered. Fixed costs depend on production volumes and are associated with
infrastructure expenses. For instance, if total production in a mine exceeds some threshold, additional ventilation infrastructure is required that will require additional capital injection and will cause ongoing operational expenses to be incurred.

- **Budget constraints.** The planning of production is typically subject to available capital and cash flow requirements. For this reason the allowable operating cost per period would be a very practical constraint to consider.

- **Ore flow networks and blending.** Mining operations that have multiple shafts, multiple processing plants or which operate on multiple reef types, are candidates for scheduling optimisation with a component of ore flow routing. Furthermore, in the context where multiple minerals are being mined from different reef types, a blending problem is encountered where processing plants are configured to accept a certain mix of ore from the different reef types.

- **Equipment and crew logistics.** For every planning horizon considered, a certain level of detail is applicable. The movement of crews or equipment is relevant for short term planning and may have a significant effect on the feasibility of the final schedule. Although work crews and equipment may be viewed as resources with certain upper limits, the key requirement here is to be able to track the movement of crews and equipment and plan for the delays that will be introduced when moving from one location to another.

### 3.5 Related work

Various approaches toward solving the underground mine scheduling problem have been proposed in the literature. The use of MILP approaches have been found by many to be well suited for formulating the mine scheduling optimisation problem. Compared to open pit applications [6, 11, 15, 29], the work done on underground mine scheduling problems is limited. Earlier references to the use of MILP formulations for solving underground mine scheduling problems can be found in [16], [69] and [78], although no model formulations were provided in these papers. MILP formulations for underground mine scheduling problems are presented in [71] and [74], but without any algorithmic contributions toward improving computing times.

In the paper by [72], a MILP formulation is proposed for solving a long-term production scheduling problem in which the allocation of mining units to mine sections are considered. A Benders decomposition approach is followed to improve the solution time of the MILP for small to medium-sized randomly generated problem instances. In [82], two preprocessing algorithms are presented for reducing the number of binary decision variables in the MILP formulation. Also in an attempt to improve computing times, the approach followed by both [52] and [60] is to aggregate production activities that follow a natural continuous sequence, with the result that the number of variables is reduced. Similarly, the approach followed by [61] is to heuristically generate feasible solutions by aggregating time periods. By doing this, improvements on computing times are achieved, but to the detriment of optimality. In a follow-up study involving the same mine [54], optimisation-based decomposition heuristics are proposed for solving the corresponding MILP. A heuristic approach is presented by [28] which involves the iterative solving of the linear programming relaxation and the fixing of the binary decision variables.

An improved formulation of the underground mine scheduling optimisation problem is proposed by [81]. The MILP formulation is based on a low resolution time discretisation of the mine layout, while maintaining detailed mineral information through the use of a mining-method-dependent
grade tonnage curve. Optimisation-based heuristics are proposed by [64] which appear to be successful in generating good approximate solutions within a short time. A review of underground mine scheduling optimisation can be found in [62].

3.6 Summary

Profitability of mining operations is highly dependent on proper planning and effective resource utilisation. The optimal scheduling of mining activities, with the objective to maximise NPV, plays an important role in achieving this. However, prior to embarking on any kind of optimisation exercise, it is imperative to have domain knowledge of mining and to understand the complexities thereof. In this chapter, a brief overview of the technical aspects of underground mine scheduling was provided. Although a comprehensive discussion of different mining methods and geological modelling is beyond the scope of this thesis, some aspects regarding typical underground mine layout and mining operations were described.

A small example was used to illustrate how the underground mine scheduling problem may be accommodated within the RCSP framework. An important consideration, however, is the requirement that certain strategic decisions be incorporated into the mine scheduling optimisation problem in the form of generic constraints. In the next chapter, it will be shown that the exact framework provided by MILP solvers makes it possible to incorporate these constraints over and above the constraints needed to capture the underlying resource scheduling problem.
CHAPTER 4

Mathematical Models

Contents
4.1 Formulations of the RCSP when considering resource transfer requirements . . 33
4.2 Mathematical Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
   4.2.1 A resource flow RCSP formulation . . . . . . . . . . . . . . . . . . . . . 35
   4.2.2 The resource flow mine scheduling problem (RMSP) . . . . . . . . . . 37
   4.2.3 The time-indexed mine scheduling problem (TMSP) . . . . . . . . . . 38
4.3 Time-based costs and revenue . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.4 Crew requirement constraints and transfer delays . . . . . . . . . . . . . . . . . 41
4.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

Underground mine scheduling optimisation, in its simplest form, entails the scheduling of mining activities to be executed in future in order to maximise net present value (NPV), while taking resource and mining-specific constraints into account. It is therefore considered to be a special case of the RCSP. The sequencing order of the activities is determined by a precedence graph, which is constructed as part of the mine layout. Typical underground capacity constraints can be accommodated within the RCSP framework as upper limits on generic resources. There are, however, modelling requirements that do not fit into the RCSP framework, as shown in the previous chapter. The aim of this chapter is to present modifications to existing RCSP models that will address some of these modelling requirements. Specifically, the emphasis will be on how to incorporate the movement of crews and equipment into the RCSP framework for the purpose of tracking these resources and imposing transfer delays. This is a very relevant topic in current mining conversations due to the increased use of mechanisation [57]. Large machines are used for excavation and these machines need to be scheduled effectively in order to minimise the time it takes to transfer them from one location to the other.

4.1 Formulations of the RCSP when considering resource transfer requirements

Recall from Chapter 2 that RCSP formulations can be classified into three main streams, namely, time-indexed formulations, resource flow formulations, and event-based formulations. Intuitively, it makes sense to explore the use of resource flow formulations to address the requirements above, and more specifically the work by [45] and [68], in which transfer delays are explicitly considered. The resource flow formulations of the RCSP have, however, been stated in the literature
with the objective to minimise makespan. In contrast to the time-indexed formulation that lends itself naturally to be formulated as a linear objective function, the resource flow formulations require optimisation of a non-linear function in order to maximise NPV (see equation (3.1) in Section 3.2). Therefore, a modification to the resource flow formulation is required if the objective is to maximise NPV. The research question posed here is whether a time-indexed formulation, augmented with resource flow features, might be more efficient in terms of computing times than the resource flow formulation adapted to maximise NPV. In the subsequent sections of this chapter, adapted versions of the time-indexed and resource flow formulations are provided.

4.2 Mathematical Models

The formulation of the underground mine scheduling problem is based on the RCSP framework. That is, associated with each activity is a set of resources, where each resource may either be produced or consumed. In order to facilitate the formulation of subsequent models, the notation from Section 2.3 is extended below.

- Let $R$ denote the set of resources. A potential set of resources may, for example, include the total on-reef and off-reef tonnes of production, the volume of minerals produced, or the amount of explosives consumed for blasting.

- Let $F \subseteq R$ denote the subset of resources for which a transfer delay applies. Examples include, work force crews or heavy machinery that have to be transported from one work place to another.

- Let $A$ denote the index set of all activities. An activity $i \in A$ may, for example, relate to the excavation of part of an underground tunnel, or the placement of machinery that will enable the excavation.

- Let $d_i$ be the duration of activity $i \in A$, measured in days.

- Let $E^A_i$ be the earliest start time and let $L^A_i$ be the latest start time of an activity $i \in A$. The earliest and latest start times of an activity are functions of the precedence graph and the planning horizon. The calculation of $E^A_i$ and $L^A_i$ is considered part of preprocessing, which will be presented in Chapter 5.

- Let $\mathcal{P}(i) \subseteq A$ denote the set of immediate predecessor activities of activity $i \in A$. That is, all incident predecessor activities according to the precedence graph.

- Let $\mathcal{S}(i) \subseteq A$ denote the set of immediate successor activities of activity $i \in A$. That is, all incident successor activities according to the precedence graph.

- Let $\delta_{rij}$ be the delay in transferring resource $r \in R$ from activity $i \in A$ to activity $j \in A$.

- Let $v_{ri}$ be a numerical value for the quantity of resource $r \in R$ being produced/consumed by activity $i \in A$, over its entire duration.

- Let $\mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$ denote the time period indices. The number of periods $|\mathcal{T}|$ depends on the choice of the scheduling calendar. The standard calendar options within the mining environment are annual, quarterly, monthly and weekly calendars.
4.2. Mathematical Models

- Let $S^T_t$ be the start time of period $t \in T$. The planning horizon is measured in terms of days. Therefore, the start time of a period is calculated as the number of days relative to time zero. The end time of a period $t$ is equal to $S^T_{t+1}$. The start and end times of periods are dictated by the choice of scheduling calendar and are necessary to accommodate the composition of mixed calendars, e.g. first monthly, then quarterly and finally annually. The application of a mixed calendar is presented in Chapter 6.

- Let $p_t$ be the duration in days of the $t$-th time period, calculated as $p_t = S^T_{t+1} - S^T_t$.

- Let $T(i) \subseteq T$ be the set of eligible start time periods for activity $i \in A$ based on the earliest and latest start times $E_i^A$ and $L_i^A$, respectively. Specifically, for each activity $i \in A$ the inequality $E_i^A \leq S^T_t \leq L_i^A$ must hold for all $t \in T(i)$.

- Let $A(t) \subseteq A$ be the set of activities eligible to start at time period $t \in T$ based on the earliest and latest start times. That is, $E_i^A \leq S^T_t \leq L_i^A$, for all $i \in A(t)$.

- Let $c_r$ be the cost per unit of consuming/producing a resource $r \in R$ during time period $t \in T$. Note that the coefficient $c_r$ may either be negative or positive, depending on the type of resource. For instance, in the case of a maximisation problem, the coefficient associated with a resource that denotes the tonnes of rock from stoping production would be negative, since the production of the resource would incur costs. The coefficient associated with a resource that denotes, for example, the volume of minerals would be treated as positive since revenue is incurred by the production of this resource.

- Let $U_r$ be the upper limit on the quantity of resource $r \in R$ that may be consumed/produced per day. Generalisation of $U_r$ to other time period resolutions may be achieved by multiplying it by the period duration $p_t$, measured in days.

- Let $\alpha_t$ be the effective NPV rate per period at which future cash flows will be discounted. Its dependency on a time period index $t$ ensures that the discounting is performed according to the correct period durations when schedules are generated according to different time period resolutions.

In order to facilitate the formulation of resource flow constraints below, two artificial activities are introduced, both with a duration of zero. A source activity $i^+$ is introduced with $P(i^+) = \emptyset$ and $S(i^+) = \{i \in A : P(i) = \emptyset\}$, and a sink activity $i^-$ is introduced with $S(i^-) = \emptyset$ and $P(i^-) = \{i \in A : S(i) = \emptyset\}$. Furthermore, for the source and sink activities, $v_{r_i^+} = U_r$ and $v_{r_i^-} = U_r$, respectively, for all $r \in F$.

4.2.1 A resource flow RCSP formulation

The earliest resource flow MILP formulation of the RCSP is by [4], in which a polynomial insertion algorithm is proposed for solving the RCSP. This formulation is, however, driven by an algorithmic approach and is not formulated for the purpose of solving it with a MILP solver. The work by [42] is the first to provide numerical results for a resource flow RCSP formulation solved using an off-the-shelf commercial MILP solver. The flow-based formulation is compared to both a time-indexed and an event-based formulation purely on the basis of computational efficiency. It is shown that for problem instances having activities with longer durations, and consequently longer scheduling horizons, the flow-based and event-based formulations are superior compared to the time-indexed formulation. The alternative resource flow formulations in this chapter are based on the model by [42], which is reproduced below for the sake of completeness.
The primary decision variables are the starting times \( s_i \geq 0 \) for each of the activities \( i \in \mathcal{A} \). In order to formulate the resource transfer constraints, the resource flow variables \( f_{rij} \geq 0 \) are introduced to denote the flow of a resource \( r \in \mathcal{F} \) from activity \( i \in \mathcal{A} \) to \( j \in \mathcal{A} \). The variables \( z_{ij} \in \{0, 1\} \), called the *linear ordering variables*, are used to indicate the ordering of activities. That is, if \( z_{ij} = 1 \) it indicates that activity \( j \) is scheduled to start only after completion of activity \( i \). Consequently, the linear ordering variables also indicate whether the transfer of a resource is permitted from activity \( i \in \mathcal{A} \) to \( j \in \mathcal{A} \).

The objective of the resource flow RCSP is to

\[
\text{minimise } s_{i^-} \tag{4.1}
\]

subject to the constraints

\[
z_{ij} = 1, \quad j \in \mathcal{A}, i \in \mathcal{P}(j), \tag{4.2}
\]

\[
z_{ij} + z_{ji} \leq 1, \quad (i, j) \in \mathcal{A}^2, i < j, \tag{4.3}
\]

\[
z_{ij} + z_{jk} - z_{ik} \leq 1, \quad (i, j, k) \in \mathcal{A}^3, i \neq j, i \neq k, j \neq k, \tag{4.4}
\]

\[
s_j - s_i - (d_i + M)z_{ij} \geq -M, \quad (i, j) \in \mathcal{A}^2, i \neq j, r \in \mathcal{F}, \tag{4.5}
\]

\[
\sum_{i \in \mathcal{A} \setminus \{j\}} f_{rij} = v_{rj}/d_j, \quad j \in \mathcal{A}, r \in \mathcal{F}, \tag{4.6}
\]

\[
\sum_{j \in \mathcal{A} \setminus \{i\}} f_{rij} = v_{ri}/d_i, \quad i \in \mathcal{A}, r \in \mathcal{F}, \tag{4.7}
\]

\[
f_{rij} - \min\{v_{ri}, v_{rj}\}/d_iz_{ij} \leq 0, \quad (i, j) \in \mathcal{A}^2, i \neq j, r \in \mathcal{F}. \tag{4.8}
\]

The objective function (4.1) minimises the makespan by minimising the starting time of the sink activity \( i^- \) while constraint set (4.2) is required to ensure feasibility in terms of activity precedence. According to [42], constraint set (4.3) ensures that resource flow is either in one direction or the other, or that activities \( i \) and \( j \) are being processed in parallel, i.e. \( z_{ij} = 0 \) and \( z_{ji} = 0 \). Constraints (4.4) are called *transitivity constraints* and according to [3] they are responsible for ensuring that there are no cycles in the permutations.

Constraint set (4.5) is collectively called the *linear ordering constraints* and they determine the linear ordering variables \( z_{ij} \) based on the starting time \( s_j \) of activity \( j \) and the completion time of its predecessor \( i \), given by \( s_i + d_i \). A reasonable choice of the large number \( M \) in (4.5) would be the latest possible finishing time of the schedule according to the calendar, i.e. \( M = S_{\max} \).

The resource flow requirements are imposed by constraint sets (4.6) and (4.7), stating that all the flow of resources into an activity (4.6) and all the flow of resources out of an activity (4.7) should match the daily resource requirement \( v_{ri}/d_i \) of an activity \( i \), for any resource \( r \). According to constraint set (4.8), the flow of resources from activity \( i \) to \( j \) is permitted only if activity \( j \) is scheduled to start after the completion of activity \( i \), that is when \( z_{ij} = 1 \).

It should be noted that, according to [43], constraint sets (4.3) and (4.4) are redundant. They mention that these are valid inequalities, but no formal proofs or evidence of improvement in computing times are provided. In fact, all computational results related to the flow-based RCSP by both [42] and [2] are generated with the constraints sets (4.3) and (4.4) included in the formulation. No results are provided of the flow-based RCSP without these valid inequalities. For the sake of completeness, it is shown below that these constraints are indeed redundant. Furthermore, computational results presented later in this thesis show that constraint sets (4.3) and (4.4) have a detrimental effect on computing times when included in the formulation of resource flow-related models.
4.2. Mathematical Models

Proposition 4.1. Constraint set (4.3) is redundant and the same effect is obtained by the linear ordering constraints (4.5).

Proof. Consider activities $i$ and $j$ with $d_i > 0$ the duration of $i$. If $z_{ij} = 1$, then according to the linear ordering constraints (4.5), $s_i < s_j$. Assume now that $z_{ji} = 1$, then $s_j < s_i$ which is a contradiction, implying that $z_{ji}$ has to be zero.

Proposition 4.2. Constraint set (4.4) is redundant and the same effect is obtained by the linear ordering constraints (4.5).

Proof. Consider activities $i$, $j$ and $k$ with respective durations $d_i > 0$, $d_j > 0$ and $d_k > 0$. If $z_{ij} = 1$ and $z_{jk} = 1$, then according to the linear ordering constraints (4.5), $s_i < s_j$ and $s_j < s_k$. Assume now that $z_{ki} = 1$, then $s_k < s_i$ which is a contradiction, implying that $z_{ki}$ has to be zero.

The redundant constraints (4.3) and (4.4) are omitted in subsequent resource flow formulations of the mine scheduling optimisation problem in this thesis.

4.2.2 The resource flow mine scheduling problem (RMSP)

The RMSP is an adaptation of the resource flow RCSP formulation. Additional variables and constraints are introduced to facilitate the formulation of an objective function that maximises NPV. Recall that the objective function (3.2), which maximises NPV, is non-linear in the variables $s_i$ and can be written as

$$\max \sum_{i \in A} f_i(s_i),$$

with

$$f_i(s_i) = e^{-\alpha s_i} \sum_{r \in R} c_r v_{ri}.$$ 

A piece-wise approximation of the objective function is suggested to deal with each non-linear function $f_i(s_i)$ according to the approaches by [23] and [53]. Let the points $(s_{iv}, f_{iv})$, $v \in V = \{0, 1, 2, \ldots, N - 1\}$ be the vertices for the piece-wise linear approximation of the function $f_i(s_i)$. The decision variable $y_i \in \mathbb{R}$ is introduced to capture the approximate value of $f_i(s_i)$ according to the piece-wise linear approximation. For that purpose, auxiliary variables $\lambda_v \geq 0$, with $v \in V$ and $l_v \in \{0, 1\}$, with $v \in V \setminus \{0\}$ are defined. The latter serves the purpose of selecting the most appropriate line segment for local approximation with respect to the objective function, whereas the former are needed to express the decision variables $s_i$ and $y_i$ as convex combinations of the knots $(s_{iv}, y_{iv})$, $v \in V$. Note that since all resource constraints in the RMSP formulation are treated as resource flow constraints, it follows that $\mathcal{F} = \mathcal{R}$. The objective in the RMSP is to

$$\text{maximise} \sum_{i \in A} y_i,$$
subject to the constraints

\begin{align}
  s_j - s_i - (d_i + \delta_{rij} + M)z_{ij} & \geq -M, \quad (i, j) \in \mathcal{A}, i \neq j, r \in \mathcal{F} \quad (4.12) \\
  \sum_{i \in \mathcal{A}\setminus\{j\}} f_{rij} &= v_{rj}/d_j, \quad j \in \mathcal{A}, r \in \mathcal{F} \quad (4.13) \\
  \sum_{j \in \mathcal{A}\setminus\{i\}} f_{rij} &= v_{ri}/d_i, \quad i \in \mathcal{A}, r \in \mathcal{F} \quad (4.14) \\
  f_{rij} - \min\{v_{ri}, v_{rj}\}/d_z z_{ij} & \leq 0, \quad (i, j) \in \mathcal{A}, i \neq j, r \in \mathcal{F} \quad (4.15) \\
  z_{ij} &= 1, \quad j \in \mathcal{A}, i \in \mathcal{P}(j), \quad (4.16) \\
  s_i - \sum_{v \in \mathcal{V}} \lambda_{iv}s_{iv} &= 0, \quad i \in \mathcal{A}, \quad (4.17) \\
  y_i - \sum_{v \in \mathcal{V}} \lambda_{iv}y_{iv} &= 0, \quad i \in \mathcal{A}, \quad (4.18) \\
  \sum_{v \in \mathcal{V}} \lambda_{iv} &= 1, \quad i \in \mathcal{A}, \quad (4.19) \\
  \lambda_{i0} - l_{i1} & \leq 0, \quad i \in \mathcal{A}, \quad (4.20) \\
  \lambda_{iv} - l_{iv} - l_{i(v+1)} & \leq 0, \quad i \in \mathcal{A}, v \in \mathcal{V}\setminus\{0, N-1\}, \quad (4.21) \\
  \lambda_{i(N-1)} - l_{i(N-1)} & \leq 0, \quad i \in \mathcal{A}. \quad (4.22)
\end{align}

The objective function (4.11) expresses the NPV as the sum of the linear piece-wise approximations \( y_i \approx f_i(s_i) \), for all activities \( i \in \mathcal{A} \). Constraint sets (4.12)–(4.16) are the same resource flow constraints as in the resource flow RCSP above, except for the linear ordering constraints (4.12) which now take the delay \( \delta_{rij} \) in transferring resource \( r \) from activity \( i \) to \( j \) into account. Constraint sets (4.17) and (4.18) express \( s_i \) and \( y_i \) as convex combinations of the piece-wise linearisation knots of \( f_i(s_i) \), for all \( i \in \mathcal{A} \). Convexity conditions are maintained by (4.19), while constraint sets (4.20), (4.21) and (4.22) are responsible for enabling the convexity variable \( \lambda_{iv} \) to take on an appropriate value based on the selection of a specific line segment \( l_{iv} \).

### 4.2.3 The time-indexed mine scheduling problem (TMSP)

The formulation of the TMSP is based on the model by [66], augmented with the necessary variables and constraints to facilitate resource flows as formulated above. In the TMSP formulation, the variable \( x_{it} \in \{0, 1\} \) is used to indicate the start time of an activity. That is, \( x_{it} = 1 \) indicates that activity \( i \in \mathcal{A} \) is scheduled to start at the beginning of time period \( t \in \mathcal{T} \). The start time of an activity is therefore aligned with the start time of a calendar period. More specifically, \( s_i = \sum_{t \in \mathcal{T}} S_i^T x_{it} \). Keeping with the definition above, the variable \( z_{ij} \in \{0, 1\} \) is used to indicate whether the transfer of resources is permitted from activity \( i \in \mathcal{A} \) to \( j \in \mathcal{A} \). If a transfer of a resource \( r \in \mathcal{F} \) is allowed, the flow of the resource from activity \( i \in \mathcal{A} \) to \( j \in \mathcal{A} \) is denoted by the continuous variable \( f_{rij} \geq 0 \).

The duration \( d_i \) of an activity \( i \in \mathcal{I} \) is assumed to be continuous and, within a mining context, may extend over several time periods. Furthermore, in contrast to the time-indexed formulations in the literature [42], \( d_i \) is typically not divisible by multiples of the time period durations. Therefore, the formulation of the objective and the resource constraints below are treated differently. The possible ways in which an activity may extend over time periods are illustrated in Figure 4.1, depending on its starting time and its duration. From the top of Figure 4.1, the cases that are of interest are the ones that overlap time period \( t \). Consider the second case where activity \( i \) is scheduled to start at time \( t - 1 \), that is \( s_i = S_{t-1}^T \), such that its completion extends into
4.2. Mathematical Models

In order to determine the feasibility of a resource constraint for period $t$, it is necessary to calculate the proportion of resource contribution by activity $i$ to period $t$. For the second case in Figure 4.1 this proportion is calculated as $(s_i + d_i - S^T_k)/p_t$ with $p_t = S^T_{t+1} - S^T_t$. Note, however, that the fourth case may also be accommodated using the same formula. Similarly, the third and fifth cases are catered for by means of the formula $(S^T_{t+1} - S^T_t)/d_i$.

The following function,

$$
\phi_{itk} = \begin{cases} 
S^T_t + d_i - S^T_{t+1}, & \text{if } S^T_k \leq S^T_t \leq S^T_k + d_i, \\
\frac{p_t}{d_i}, & \text{if } S^T_k \leq S^T_t \text{ and } S^T_k + d_i \geq S^T_{t+1}, \\
0, & \text{otherwise,}
\end{cases}
$$

captures the various cases depicted in Figure 4.1 and returns a resource multiplier which represents the proportion of the resource consumed by an activity $i$ during time period $t$, based on the duration of the activity and its start period $k$. Note that the start time $s_i$ of activity $i$ has been substituted with $S^T_k$ in the expression $(s_i + d_i - S^T_t)/p_t$ for the above function.

The objective in the TMSP augmented with resource flow constraints is to,

$$
\text{maximise } \sum_{i \in A} \sum_{r \in R} \sum_{k \in T} \left( \sum_{t=k}^{S^T_{t+1}-S^T_k} c_r \phi_{itk} v_{ri}(1 + \alpha_t)^{-S^T_t} \right) x_{ik}, \quad (4.23)
$$

subject to the constraints

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure41.png}
\caption{Possible scheduling outcomes for an activity $i$ relative to a period $t$.}
\end{figure}
\[ \sum_{t \in T(i)} x_{it} = 1, \quad i \in A, \quad (4.24) \]
\[ \sum_{i \in A(t)} \sum_{k \in T, k \leq t} \phi_{itk} v_{rk} x_{ik} \leq p_t U_r, \quad r \in R \setminus F, \quad t \in T, \quad (4.25) \]
\[ \sum_{t \in T(j)} S_t^T x_{jt} - \sum_{t \in T(i)} S_t^T x_{it} - (d_i + \delta_{rij} + M)z_{ij} \geq -M, \quad (i, j) \in A^2, \quad r \in F, \quad (4.26) \]
\[ \sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj} / d_j, \quad j \in A, \quad r \in F, \quad (4.27) \]
\[ \sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri} / d_i, \quad i \in A, \quad r \in F, \quad (4.28) \]
\[ f_{rij} - \min\{v_{ri}, v_{rj}\} / d_i z_{ij} \leq 0, \quad (i, j) \in A^2, \quad i \neq j, \quad r \in F, \quad (4.29) \]
\[ z_{ij} = 1, \quad j \in A, \quad i \in P(j). \quad (4.30) \]

The objective function (4.23) maximises NPV at a discount rate of \( \alpha^{(p)} \). The inner summation involving the index \( t \) is used to aggregate the NPV values for the time periods over which activity \( i \) extends, in the case where activity \( i \) is scheduled to start at time period \( k \). The function \( \phi_{itk} \) is used for this purpose to proportionately adjusts the revenue or cost for each activity, based on the start time and duration of the activity. The sign of the cost coefficients \( c_r \) are defined according to the meaning of each resource \( r \in R \). For instance, for a mineral resource representing the kilograms of minerals produced, the corresponding coefficient \( c_r \) should reflect the mineral prices per kilogram as positive. Similarly, if a resource incurs costs, the corresponding coefficient \( c_r \) should be negative.

Constraint set (4.24) forces each activity to be scheduled in one of the time periods \( t \in T \) and the constraints (4.25) enforce an upper limit on the consumption/production of resources. The remaining constraints correspond to the resource flow constraints as formulated above. Note, however, that the linear ordering constraints (4.26) are now based on the starting times expressed in terms of the binary start-time variables, i.e. \( s_i = \sum_{t \in T(i)} S_t^T x_{it} \).

### 4.3 Time-based costs and revenue

In the above formulations, the parameter \( c_r \) is used to capture the unit cost or unit revenue for a resource \( r \in R \). In practice, however, costs are typically escalated into the future according to the consumer price index (CPI). The effect of applying the CPI to resource costs is that a different unit cost per time period is essentially obtained. This can easily be accommodated within the TMSP formulation by adding a time index \( t \) to the cost parameter \( c_r \). In the case of revenue drivers, such as mineral prices, escalation may also be considered to facilitate future mineral price scenarios. This can again be achieved by adding a time index to the specific revenue generating resource parameter \( c_r \).

In this thesis, however, the escalation of unit costs and revenues is not considered, although the mathematical formulations can easily be adapted to cater for it. The reason for this exclusion is due to the diverse interpretation of cost and revenue escalations in industry which make it difficult to incorporate in empirical studies. Furthermore, some interpolation approach would have to be performed on the input data in order to incorporate cost escalation into the objective function of the RMSP problem formulation.
4.4 Crew requirement constraints and transfer delays

The primary motivation for adopting a resource flow-based formulation of the underground mine scheduling problem is to incorporate the movement of crews and equipment into the RCSP framework for purposes of tracking these resources and imposing transfer delays. In order to illustrate this requirement more concretely, consider the small mine scheduling example from Section 3.2, for which the mine layout is depicted in Figure 3.5. In order to incorporate crew scheduling the input data are augmented with crew requirements as depicted in Table 4.1.

<table>
<thead>
<tr>
<th>i</th>
<th>Type</th>
<th>$d_i$</th>
<th>$P(i)$</th>
<th>$v_{ri}/d_i$</th>
<th>Development crew(#)</th>
<th>Stoping crew(#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Haulage</td>
<td>200</td>
<td>-</td>
<td>200/200 = 1</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Cross-cut</td>
<td>120</td>
<td>1</td>
<td>120</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Step-over</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Raise line</td>
<td>180</td>
<td>3</td>
<td>180</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Stope panel</td>
<td>250</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>6</td>
<td>Stope panel</td>
<td>250</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>7</td>
<td>Stope panel</td>
<td>250</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>8</td>
<td>Stope panel</td>
<td>250</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>9</td>
<td>Haulage</td>
<td>200</td>
<td>1</td>
<td>200/200 = 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Cross-cut</td>
<td>120</td>
<td>9</td>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>Step-over</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>Raise line</td>
<td>180</td>
<td>11</td>
<td>180</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>Stope panel</td>
<td>250</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>14</td>
<td>Stope panel</td>
<td>250</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>15</td>
<td>Stope panel</td>
<td>250</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>16</td>
<td>Stope panel</td>
<td>250</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
</tbody>
</table>

Table 4.1: Activity input values for the small example considered.

4.4 Crew requirement constraints and transfer delays

The primary motivation for adopting a resource flow-based formulation of the underground mine scheduling problem is to incorporate the movement of crews and equipment into the RCSP framework for purposes of tracking these resources and imposing transfer delays. In order to illustrate this requirement more concretely, consider the small mine scheduling example from Section 3.2, for which the mine layout is depicted in Figure 3.5. In order to incorporate crew scheduling the input data are augmented with crew requirements as depicted in Table 4.1.

The column which is labeled “Development crew(#)” specifies the crew requirement for each development activity. Consider the development crew requirement for activity $i = 1$ which is $200$. Since the duration of this activity is 200 days, the development requirement for activity $i$ is, therefore, one crew per day. This is exactly in line with the resource flow requirement constraints (4.13)–(4.14), in the case of the RMSP and (4.27)–(4.28), in the case of the TMSP, since the right-hand side values are $v_{ri}/d_i = 200/200 = 1$. The column which is labeled “Stoping crew(#)” specifies the crew requirement for each stoping activity and the same logic in terms of the resource requirement constraints is applied for this resource.

Figure 4.2: Crew requirements over time for the unconstrained RMSP.
Figure 4.3: Gantt chart showing results for the unconstrained RMSP.

Figure 4.4: Gantt chart showing results for the crew constrained RMSP.

Figure 4.5: Gantt chart showing results for the RMSP with transfer delays.
For the purpose of illustrating the crew scheduling feature, an unconstrained version of the RMSP was solved using the data provided in Table 4.1. That is, the original capacity constraint as described in Section 3.2 has been omitted. The crew requirements as a result of solving the unconstrained version of the RMSP is provided in Figure 4.2. For both development and stoping, the crew requirements vary mostly between one and two crews per day. There is an exception where, for a specific day, a total of three development crews are required. The same applies to stoping where, for a specific day, a total of three stoping crews are required. These results are confirmed by the Gantt chart in Figure 4.3 showing how activities overlap with the result that more than one crew may be required per day.

For the sake of illustration, assume that there are one development and one stoping crew available per day for this synthetic mining operation. In terms of the RMSP problem formulation, the upper bound $U_r$ is set to one for both the development crew and stoping crew resources. The Gantt chart in Figure 4.4 shows the new schedule obtained by re-optimising the RMSP with the newly added resource constraints. In order to maintain the constraint of only one development crew per day, the development activity $i = 9$ has been scheduled to start later and commences only after completion of the development activity $i = 4$. Furthermore, stoping activities $i = 7$ and $i = 8$ have also been scheduled to start later so that they do not coincide with the other stoping activities. Recall from the above formulations that the variable $f_{rij}$ denote the flow of resource from activity $i$ to $j$. A direct use of the solution values for the flow variables $f_{rij}$ is the ability to track the movement of crews. For instance, the solution values for the optimal schedule depicted in Figure 4.4 suggest that the “route” of the development crew is to traverse the activities 1, 2, 3, 4, 9, 10, 11 and 12. Similarly, the route of the stoping crew is to traverse the activities 5, 6, 7, 8, 13, 14, 15 and 16.

In order to illustrate the effect of transfer delays on the solution of the RMSP, consider a situation in which a delay of 100 days is expected for the development crew to move from the first raise line, depicted in Figure 3.5, to the next. That is, on completion of activity $i = 4$, there will be a delay of 100 days before the development crew may start excavating activity $i = 9$. The financial impact of the delay can be clearly seen from the Gantt chart depicted in Figure 4.5. According to the optimal schedule it is more profitable to carry on with the excavation of the second raise line and finish stoping panels 13–16, before returning the the first raise line to complete the remaining stoping panels.

### 4.5 Summary

The underground mine scheduling problem, as a special case of the RCSP, entails determining the start times of mining activities in order to maximise NPV, while taking resource and mining-specific constraints into account. In this chapter, a time-based formulation (TMSP) and a resource flow formulation (RMSP) were presented to capture mining-specific requirements as part of the mathematical formulation. More specifically, the delays introduced as a result of crew and equipment movement were incorporated into the TMSP formulation by augmenting it with resource flow constraints. Although the RMSP formulation is already based on a classical resource flow model, it was necessary to follow a linearisation approach in order to approximate the non-linear objective function that maximises NPV.

It should be noted that although the two proposed formulations are different in terms of decision variables and constraints, the same feasible region is described by the two formulations when considering the same problem instance. Although there might be a slight difference in objective function values due to the linearisation of the objective function in the RMSP formulation, the
two formulations should provide the same optimal solutions.

The ability of the two proposed formulations to cater for mining specific requirements was demonstrated in this chapter. For this purpose, the example from Chapter 3 was extended to include crew requirement constraints and transfer delays.

The emphasis in this study is the use of an exact solution framework. The implication is that the method for generating solutions is separated from the problem formulation. For instance, the solution for the small example above was generated by applying a commercial solver using the RMSP problem formulation. The benefit of this is that generic constraints may be added to the RMSP and TMSP formulations in order to cater for other mining specific requirements that may not have been addressed in this study.

Due to the inherent difficulty in solving the mine scheduling optimisation problem, several algorithmic enhancements are proposed in the next chapter to improve computational efficiency.
Algorithmic improvements

Contents

5.1 Preprocessing .............................................. 46
5.2 Problem reformulations RMSP-AC and TMSP-AC ................. 46
5.3 Graph reduction ............................................. 47
   5.3.1 RMSP with graph reduction (RMSP-GR) .................. 47
   5.3.2 TMSP with graph reduction (TMSP-GR) .................. 48
5.4 Problem reformulations RMSP-AC-GR and TMSP-AC-GR .............. 49
5.5 An alternative TMSP formulation ................................ 50
5.6 A branch-and-cut framework .................................. 50
   5.6.1 Branch-and-bound basics ................................. 51
   5.6.2 Cut generation in the branch-and-bound ................ 52
   5.6.3 Benders decomposition of the RCSP ..................... 53
   5.6.4 The resource flow separation problem (RFSEP) ............ 54
   5.6.5 The RFSEP with aggregated constraints (RFSEP-AC) ....... 55
   5.6.6 An incremental implementation of the RFSEP (RFSEP-INC) .... 56
   5.6.7 Master problem alternatives .............................. 56
   5.6.8 The resource flow rounding heuristic (RFRH) ............. 57
   5.6.9 The RFRH with guaranteed feasibility (RFRH-GF) .......... 57
   5.6.10 An incremental implementation of the RFRH (RFRH-INC) .... 59
   5.6.11 Augmenting the master problem with valid inequalities ..... 59
5.7 Summary ..................................................... 60

The computational complexity of the RCSP has been established as being \( \text{NP} \)-hard \([10]\). It is, therefore, unlikely that an algorithm can be developed to solve the underground mine scheduling optimisation problem to optimality in reasonable time for realistically sized problem instances. A typical course of action is, for instance, either to improve lower bounds through stronger problem formulations and the application of valid inequalities, or to improve the generation of primal solutions through the use of heuristics. Depending on the problem formulation adopted, an alternative is to make use of a decomposition approach in an attempt to speed up computations.

A preprocessing procedure is described below, which may result in the RCSP having a stronger problem formulation. Furthermore, the resource flow-related formulations of the RMSP and the TMSP are reviewed and a graph reduction approach is described. The resulting reformulations may possibly be weaker than the original, but they may have fewer variables and constraints
describing the same feasible region, with the result that there is less of a burden on the LP solver when solving the sub-problems of the branch-and-bound method. The benefit of these new formulations, in terms of computing times, are presented in the next chapter.

The final contribution of this chapter is a Benders decomposition approach which allows the mine scheduling optimisation problem to be solved within a branch-and-cut framework. The motivation for exploring the use of decomposition is to improve scalability, specifically when problem instances with a large number of resources are considered. A separation procedure and a primal heuristic are also proposed as part of the branch-and-cut approach.

5.1 Preprocessing

Recall from Chapter 4 that $E_i^A$ and $L_i^A$ are defined as the earliest and latest start times, respectively, of an activity $i \in A$. Conceptually, the approach for determining $E_i^A$ involves solving an optimisation problem for which the objective is to minimise the start time $s_i$ of an activity $i$, subject to the precedence constraints of the RCSP. Similarly, an optimisation problem that maximises the start time $s_i$ of an activity $i$ is solved to determine $L_i^A$. It should be noted, however, that some upper bound, say $T$, is required on $s_i$ in order to prevent an unbounded solution in the case of solving the maximisation problem. A conservative estimation of $T$ is to assume that all the activities will be scheduled in serial, that is, $T = \sum_{i \in A} d_i$, where $d_i$ is the duration of activity $i$.

The preprocessing problem involves

\[
\text{minimising / maximising } s_i \quad (5.1)
\]

subject to the constraints

\[
s_j - s_i \leq d_i, \quad (i,j) \in A^2, \quad (5.2)
\]
\[
s_i \leq T, \quad i \in A, \quad (5.3)
\]

for each activity $i \in A$.

5.2 Problem reformulations RMSP-AC and TMSP-AC

The resource flow related constraints of both the RMSP and the TMSP are based on the formulation by [42]. A reformulation is suggested here that results in a reduction of constraints, and consequently, a possible reduction in computing times. Consider the constraint sets (4.15) and (4.29),

\[
f_{rij} - \min\{v_{ri}, v_{rj}\}/d_i z_{ij} \leq 0, \quad (i,j) \in A^2, \quad i \neq j, \quad r \in F, \quad (5.4)
\]

in the RMSP and TMSP formulations, respectively. These constraints are responsible for allowing the flow of resources whenever the linear ordering variable $z_{ij}$ has been set to one. They may also be written in the aggregated form

\[
\sum_{r \in F} f_{rij} - \left( \sum_{r \in F} U_r \right) z_{ij} \leq 0, \quad (i,j) \in A^2, \quad i \neq j. \quad (5.5)
\]

Both constraint sets (5.4) and (5.5) are based on a big-M formulation, which is considered to be an undesirable property. However, the latter constraint set clearly comprises fewer constraints since the individual resource flows $f_{rij}$ are aggregated for all resources $r \in F$. Although the
aggregated version (5.5) is considered to be theoretically weaker than (5.4), some practical benefit is expected in terms of computing times, especially in the case where a large number of different resources is considered.

The resource flow mine scheduling problem with aggregated constraints (RMSP-AC) is obtained if constraint set (4.15) in the RMSP is replaced by (5.5). Similarly, the time-indexed mine scheduling problem with aggregated constraints (TMSP-AC) is obtained if constraint set (4.29) in the TMSP is replaced by (5.5).

5.3 Graph reduction

As an additional measure to reduce the number of variables and constraints in the RMSP and TMSP formulations, a graph reduction approach is followed. The formulation of the resource flow-related constraints in both the RMSP and the TMSP is in terms of a resource flow graph \( G(A, E) \), with the set of activities \( A \) corresponding to the nodes of the graph and the set \( E \) representing its arcs. The graph \( G(A, E) \) is a complete graph with bi-directional arcs. This is evident from the qualifiers \((i, j) \in A^2\) used in the formulation of the constraints that involve the linear ordering variable \( z_{ij} \), for both the RMSP and the TMSP.

The graph reduction approach proposed in this section is inspired by the observation that resource flows are consistent with precedence relationships. More specifically, it is evident from the linear ordering constraints that, for an activity \( i \) which is a predecessor of \( j \), i.e. \( s_i + d_i \leq s_j \), resources are not allowed to flow from \( j \) to \( i \). Applying this recursively, no flows are allowed from \( j \) to any of the predecessors of \( i \). Based on this observation, some of the variables and constraints in the original RMSP and TMSP formulations may be ignored by constructing a subgraph \( G'(A, E') \), with \( E' \subset E \), such that there are no arcs from a node \( i \) to any of its predecessors. In order to incorporate the reduced resource flow graph into both the RMSP and the TMSP, new notation is required. Recall that the set of predecessor activities of activity \( i \in A \) is defined as \( P(i) \subseteq A \). These predecessor activities are the immediate predecessors of activity \( i \in A \). For each of the predecessors \( j \in P(i) \), a predecessor list \( P(j) \) exists. Continuing in a recursive manner, all of the activities along each possible path from node \( i \to \) a root node of the precedence graph can be obtained. This set of activities is denoted by \( P^+(i) \subseteq A \). Similarly, the set of successor activities along each possible path from node \( i \to \) a leaf node of the precedence graph is denoted by \( S^+(i) \subseteq A \).

Instead of creating an explicit subgraph \( G'(A, E') \), such a graph is implied by the use of the predecessor and successor sets \( P^+(i) \) and \( S^+(i) \) in the formulation of resource flow-related constraints. The basic concept is that, for a resource flow-related constraint, a resource may only be transferred from activity \( i \) to activity \( j \) if \( j \) is not an immediate predecessor, or a predecessor through a recursive relationship, that is if \( j \notin P^+(i) \). This requirement also implies that \( i \) may not be an immediate successor, or a successor through a recursive relationship to \( j \), that is \( i \notin S^+(j) \).

For the sake of completeness, both the resource flow and the time-indexed mine scheduling problems are reformulated below in terms of the reduced resource flow graph.

5.3.1 RMSP with graph reduction (RMSP-GR)

The RMSP-GR formulation, which is based on the reduced resource flow graph \( G'(A, E') \), does not require any new variables. The objective function and all the constraints related to the
Chapter 5. Algorithmic improvements

linearisation of the objective function remain the same. The major difference, compared to the original RMSP formulation, is that the constraints are now defined in terms of the reduced resource flow graph $G'(A, E')$.

As a safety measure, the precedence constraints (4.16), from the original RMSP formulation, are replaced by the explicit precedence constraints (5.7) below. This is required since the linear ordering variables $z_{ij}$ in (4.16) depend on the underlying resource flow graph and a reduction of this graph may compromise feasibility in terms of the precedence requirements. For example, in addition to the graph reduction approach described above, directed arcs between an activity $i$ and $j$ may be removed, provided that activities $i$ and $j$ do not have any resources in common. In this case the linear ordering variables will be absent from constraints (4.16).

The objective of the new RMSP-GR formulation is to

$$\text{maximise } \sum_{i \in A} y_i,$$

subject to the constraints

$$s_j - s_i \geq d_i, \quad i \in A, \quad i \in S(i),$$

$$s_j - s_i - (d_i + \delta_{ij} + M) z_{ij} \geq -M, \quad i \in A, \quad j \in A \setminus \{P^+(i) \cup i\}, \quad r \in F,$$

$$\sum_{i \in A \setminus \{S^+(j) \cup j\}} f_{rij} = v_{rj}/d_j, \quad j \in A, \quad r \in F,$$

$$\sum_{j \in A \setminus \{P^+(i) \cup i\}} f_{rij} = v_{ri}/d_i, \quad i \in A, \quad r \in F,$$

$$f_{rij} - \min\{v_{ri}, v_{rj}\}/d_iz_{ij} \leq 0, \quad i \in A, \quad j \in A \setminus \{P^+(i) \cup i\},$$

$$s_i - \sum_{v \in V} \lambda_{iv}s_{iv} = 0, \quad i \in A,$$

$$y_i - \sum_{v \in V} \lambda_{iv}y_{iv} = 0, \quad i \in A,$$

$$\sum_{v \in V} \lambda_{iv} = 1, \quad i \in A,$$

$$\lambda_{i0} - l_{i1} \leq 0, \quad i \in A,$$

$$\lambda_{iv} - l_{iv} - l_{i(v+1)} \leq 0, \quad i \in A, \quad v \in V \setminus \{0, N - 1\},$$

$$\lambda_{i(N-1)} - l_{i(N-1)} \leq 0, \quad i \in A.$$

The constraints that are directly affected by the graph reduction are the linear ordering constraints (5.8), the resource flow requirement constraints (5.9)–(5.10) and constraints (5.11), which are responsible for allowing the flow variables to take on values based on the linear ordering variables $z_{ij}$. Note that constraints (5.11) are presented in a disaggregated form in order to facilitate measuring the effect of graph reduction without constraint aggregation.

5.3.2 TMSP with graph reduction (TMSP-GR)

Similar results are obtained when formulating the time-indexed mine scheduling problem in terms of a reduced resource flow graph $G'(A, E')$. Explicit precedence constraints are added as a safety measure to ensure precedence feasibility. The objective of the TMSP-GR is to,
maximise \[ \sum_{i \in A} \sum_{r \in R} \left( \sum_{t=k}^{C_{ij}^{T} + d_i} c_{r \delta_{it} k} v_{ri} (1 + \alpha_t)^{-S_T^T} \right) x_{ik}, \] 
subject to the constraints

\[ \sum_{t \in T(i)} S_T^T x_{jt} - \sum_{t \in T(i)} S_T^T x_{it} \geq d_i, \quad i \in A, \quad i \in S(i), \] (5.19)

\[ \sum_{t \in T(i)} x_{it} = 1, \quad i \in A, \] (5.20)

\[ \sum_{i \in A(t)} \sum_{k \in T} \phi_{itk} v_{ri} x_{ik} \leq p_t U_r, \quad r \in R \setminus F, \quad t \in T, \] (5.21)

Constraint set (5.19) encapsulates the new explicit precedence requirements. The constraints that are affected by the graph reduction are the linear ordering constraints (5.22), the resource flow requirement constraints (5.23)–(5.24) and constraints (5.25). Constraint set (5.25) is again presented in a disaggregated form in order to facilitate measuring the effect of graph reduction without constraint aggregation.

5.4 Problem reformulations RMSP-AC-GR and TMSP-AC-GR

Combining the RMSP-AC and the RMSP-GR results in a formulation of the resource flow mine scheduling problem with aggregated constraints and graph reduction (RMSP-AC-GR). The formulation of the RMSP-AC-GR is exactly the same as that of the RMSP-GR, except for the constraints (5.11) that are replaced by the aggregated version

\[ \sum_{r \in F} f_{rij} - \left( \sum_{r \in F} U_r \right) z_{ij} \leq 0 \quad i \in A, \quad j \in A \setminus \{P^+(i) \cup i\}. \] (5.26)

A similar result is obtained for the TMSP variants. A formulation of the time-indexed mine scheduling problem with aggregated constraints and graph reduction (TMSP-AC-GR) is obtained by combining the TMSP-AC and the TMSP-GR. This formulation is essentially the TMSP-GR, except for the constraints (5.25) that are replaced by the aggregated version (5.26) above.
5.5 An alternative TMSP formulation

The primary decision variables in the TMSP formulation are the binary variables $x_{it}$. An activity $i$ is scheduled to start at the beginning of time period $t$ if $x_{it} = 1$. The relative start time of an activity is expressed as $s_i = \sum_{t \in T} S^T_t x_{it}$. The formulation of the newly proposed alternative TMSP (ATMSP) involves substituting the starting time of activity $i$ in the TMSP, expressed in terms of the binary decision variable $x_{it}$, by the continuous start time variable $s_i$. In addition to the constraint $s_i = \sum_{t \in T} S^T_t x_{it}$, which is added to the new formulation, explicit precedence constraints are added which are expressed in terms of the continuous start time variable $s_i$.

The objective of the ATMSP is to

$$\text{maximise} \sum_{i \in A} \sum_{r \in R} \sum_{k \in T} \left( S^T_i + d_i \right) c_{ri} \phi_{itk} v_{ri} (1 + \alpha_t) x_{ik},$$

subject to the constraints

$$\sum_{t \in T(i)} x_{it} = 1 \quad i \in A,$$  (5.28)

$$\sum_{i \in A(t)} \sum_{k \in T} \phi_{itk} v_{ri} x_{ik} \leq p_t U_r, \quad r \in R \setminus F, \quad t \in T,$$  (5.29)

$$s_i - \sum_{t \in T} S^T_t x_{it} = 0, \quad i \in A,$$  (5.30)

$$s_j - s_i \geq d_i, \quad (i, j) \in A^2,$$  (5.31)

$$s_j - s_i - (d_i + \delta_{rij} + M) z_{ij} \geq -M, \quad (i, j) \in A^2, \quad r \in F,$$  (5.32)

$$\sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj} / d_i \quad j \in A, \quad r \in F,$$  (5.33)

$$\sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri} / d_i \quad i \in A, \quad r \in F,$$  (5.34)

$$f_{rij} - \min\{v_{ri}, v_{rj}\} / d_i z_{ij} \leq 0, \quad (i, j) \in A^2, \quad i \neq j, \quad r \in F.$$  (5.35)

The same variations previously proposed for the TMSP formulation are also applicable to the ATMSP. The ATMSP-AC is obtained by replacing constraint set (5.35) with the aggregated version (5.5) and the ATMSP-GR is obtained by adapting constraints (5.32)–(5.35) to incorporate the reduced graph $G'(A, E')$. This is achieved in a manner similar to the approach followed in formulating TMSP-GR. Furthermore, the combined effect of both the aggregated constraints and graph reduction can be achieved by means of the formulation of ATMSP-AC-GR, which is similar to the TMSP-AC-GR formulation, except that all starting times are expressed in terms of the continuous variable $s_i$ and the additional constraints (5.30) and (5.31) are included.

5.6 A branch-and-cut framework

The branch-and-cut approach is an adaptation of the branch-and-bound method which has been applied successfully in the solution of many large-scale optimisation problems [65]. The approach basically entails relaxing some of the constraints in the original problem and only incorporating them back into the branch-and-bound if needed. In order to illustrate the concept more
concretely, an overview of the branch-and-bound method is provided in the next section. A
discussion then follows on the implementation of the branch-and-cut method within the con-
text of the branch-and-bound approach, providing details on how to solve the mine scheduling
optimisation problem by means of this method.

5.6.1 Branch-and-bound basics

The branch-and-bound method is based on the notion of a divide-and-conquer approach[47] and
involves the iterative solution of LP sub-problems. That is, the feasible region of the MILP to
be solved is systematically partitioned into smaller regions based on the solutions obtained by
solving the LP sub-problems. To illustrate the process, let \( x \) denote the vector of integer decision
variables in the formulation of a MILP. Let \( LP_0 \) denote the first LP sub-problem obtained by
relaxing the integer requirements on \( x \). If \( LP_0 \) is infeasible, the original MILP is also infeasible.
If, on the other hand, the solution \( x^0 \) to \( LP_0 \) satisfies the integer requirements for all of the
variables in \( x \), an optimal solution has been found. Otherwise, at least one of the components
of \( x \) is non-integral and, assuming that the MILP is a minimisation problem, the current lower
bound \( Z^* \) is assigned the objective function value \( Z_0 \) obtained by solving the sub-problem
\( LP_0 \). The initial upper bound, again assuming that the MILP is a minimisation problem, is taken
as \( Z^* = \infty \). Partitioning of the feasible region of the MILP is now achieved by creating two
sub-problems \( LP_1 \) and \( LP_2 \). If \( x_k \) is one of the components in \( x \) that does not have an integer
solution when solving \( LP_0 \), then \( LP_1 \) is obtained by adding the constraint \( x_k \leq \lfloor x^0_k \rfloor \) to the
parent problem \( LP_0 \), with \( x^0_k \) the solution of the \( k \)-th component of \( x \). Similarly, the sub-
problem \( LP_2 \) is obtained by adding the constraint \( x_k \geq \lfloor x^0_k \rfloor \) to the parent problem \( LP_0 \). This
step in the branch-and-bound process is called branching. It should be noted that in the case
where more than one component of \( x \) is non-integral, some rule has to be applied in order to
select the variable on which branching will be performed. Commercial MILP solvers typically
provide several branching strategies from which to choose. The computational study by [51]
showed, however, that no branching strategy dominates and that the effect of a particular
strategy depends on the specific problem instance being solved.

If, during the branch-and-bound process, a solution to a sub-problem, say \( p \), is found to be
integral, it becomes the incumbent integer solution, provided that the objective function value
\( Z_p \) of the corresponding sub-problem is less than the current upper bound \( Z^* \). If this is the case,
the upper bound is updated by setting \( Z^* = Z_p \). Otherwise, if the sub-problem is not integral
and its objective function is still less than the current upper bound \( Z^* \), it is added to an open
list of sub-problems. A sub-problem may be pruned, and consequently not be added to the open
list of sub-problems, whenever it is either infeasible or its objective function value exceeds the
current upper bound \( Z^* \).

It is convenient to represent the branch-and-bound process by means of a binary tree, with the
nodes of the tree corresponding to each of the sub-problems. The sub-problems in the open
list are referred to as dangling nodes and the process of selecting the next sub-problem for
branching is called node selection. Similar to different branching strategies, different criteria
exist for selecting a node from the open list. Nodes are removed from the open list if they
are selected for branching or if their objective function values exceed that of a newly found
incumbent integer solution.
5.6.2 Cut generation in the branch-and-bound

The branch-and-cut approach provides a generic framework in which a relaxed version of the MILP problem is solved using the branch-and-bound method. Feasibility with respect to the original problem is achieved through the dynamic generation of constraints, which are called cuts. Consider a sub-problem $p$ being solved as part of the branch-and-bound process. The current solution $x^p$ is tested for feasibility against the constraints of the original MILP problem. If $x^p$ is found to be infeasible, a cut is derived and added to the current sub-problem. The sub-problem is repeatedly solved until no more violated cuts can be found. The simplest implementation of such a process is to treat the relaxed constraints as lazy constraints. That is, each lazy constraint is explicitly tested for satisfaction at each of the sub-problems of the branch-and-bound approach. The benefit in terms of computing times may, however, be negated when a large number of lazy constraints are present.

An alternative to using lazy constraints is to implicitly generate cuts through the use of sepa-
5.6. A branch-and-cut framework

That is, given an existing solution $Z_p$ at a node of the branch-and-bound, a separation algorithm is applied to generate a cut using information from the current subproblem solution. The cut is then appended to the relaxed MILP formulation to cut off the current infeasibility. Figure 5.1 provides a flow diagram of the branch-and-cut process in which the application of a separation algorithm for cut generation as part of the branch-and-bound approach is illustrated.

The choice of constraints to relax is driven in most cases by the model complexity. For instance, the resource flow-based formulation of the RCSP will grow exponentially in the number of variables with an increase in the number of activities. Intuitively, by relaxing all the resource flow-related constraints, the computational burden should be less when solving the branch-and-bound sub-problems. The challenge, however, is to find a suitable separation algorithm which will be able to find violated cuts for the remaining MILP problem.

5.6.3 Benders decomposition of the RCSP

The suggested approach for solving the resource flow RCSP within a branch-and-cut framework is to make use of Benders decomposition [9]. The resource flow variables and their associated constraints are projected out of the original MILP formulation and a separation algorithm is presented below which is responsible for generating cuts so as to ensure feasibility.

Consider the resource flow RCSP formulation of Section 4.2.1. The variables in the RCSP formulation are the start time variables $s \in \mathbb{R}_{+}^{|A|}$, the linear ordering variables $z \in \{0, 1\}^{|A|^2}$ and the resource flow variables $f \in \mathbb{R}_{+}^{|R| \times |A|^2}$. For illustration purposes, let $x \in \mathbb{R}_{+}^{|A|} \times \{0, 1\}^{|A|^2}$ denote the adjoint vectors $(s, z)$ and let $X$ denote the polyhedron comprising constraint sets (4.2)–(4.5) in the resource flow RCSP formulation. Let $A$ and $B$ be appropriate coefficient matrices associated with $x$ and $f$, respectively, which correspond to the coefficients in the RCSP constraints (4.6)–(4.8). Furthermore, let $b$ correspond to the right-hand side of the same constraints and let $c$ and $d$ be the cost vectors associated with $x$ and $f$, respectively. It is evident from the RCSP objective function (4.1) that all entries of $c$ will be zero except for the component corresponding to the sink activity $s_i$. Consequently, all entries of the cost vector $d$ will also be zero since there are no costs associated with the flow variables $f$ in the objective function.

The objective of the resource flow RCSP, in matrix form, is to

\[
\begin{align*}
\min & \quad c^T x + d^T f, \\
\text{s.t.} & \quad Ax + B f \leq b, \\
& \quad x \in X.
\end{align*}
\]  

Since no flow variables are present in the polyhedron $X$, the above problem can be decomposed into a master problem

\[
\begin{align*}
\min & \quad c^T x, \\
& \quad x \in X,
\end{align*}
\]  

and a sub-problem

\[
\begin{align*}
\min & \quad d^T f, \\
\text{s.t.} & \quad B f \leq b - Ax^*.
\end{align*}
\]
Chapter 5. Algorithmic improvements

where \(x^*\) is the solution obtained by solving the master problem.

The master problem (5.39)–(5.40) is a MILP problem and is solved using the branch-and-bound method. The sub-problem (5.41)–(5.42) is an LP problem given a fixed \(x^*\). The outcome of solving the LP sub-problem is that either an optimal solution \(f^*\) is obtained, or the problem is infeasible as a result of \(x^*\). In the case where the sub-problem is infeasible, a cut \((b - Ax^*)^T w \geq 0\) can be derived where \(w\) is an extreme direction [9]. By adding the feasibility cut to the master problem, the infeasible point \(x^*\) is “separated” from the feasible region of the master problem.

5.6.4 The resource flow separation problem (RFSEP)

A separation problem is suggested based on sub-problem (5.41)–(5.42), which will produce a Benders cut \((b - Ax^*)^T w \geq 0\) if infeasibility in the sub-problem is detected. A slack variable \(\alpha \geq 0\) is introduced below for this purpose.

The objective of the resource flow separation problem (RFSEP) is to minimise \(\alpha\),

subject to the constraints

\[
\sum_{i \in A \setminus \{j\}} f_{rij} = \frac{v_{rj}}{d_j}, \quad j \in A, \ r \in F, \tag{5.44}
\]

\[
\sum_{j \in A \setminus \{i\}} f_{rij} = \frac{v_{ri}}{d_i}, \quad i \in A, \ r \in F, \tag{5.45}
\]

\[
f_{rij} + \alpha = \min\{v_{ri}, v_{rj}\}/d_i z_{ij}^*, \quad (i,j) \in A^2, \ i \neq j, \ r \in F. \tag{5.46}
\]

The vector \(z^* \in \{0, 1\}^{|A^2|}\) is the solution obtained for the linear ordering variables \(z_{ij}\) by solving the master problem. It is clear that the value of \(\alpha\) will be an indicator of feasibility for the sub-problem. If \(\alpha > 0\), the sub-problem is infeasible, otherwise it is feasible.

**Theorem 5.1.** For a given vector \(z^* \in \{0, 1\}^{|A^2|}\) in the RFSEP, the cut

\[
\sum_{(i,j) \in A^2} \sum_{i \neq j} \min\{v_{ri}, v_{rj}\}/d_i z_{ij}^* \mu_{ijr} \leq -\sum_{r \in F} \sum_{i \in A} (v_{ri}/d_i \pi^1_{ir} + v_{ri}/d_i \pi^2_{ir}),
\]

where \(\pi^1 \in \mathbb{R}^{|A| \times |F|}\), \(\pi^2 \in \mathbb{R}^{|A| \times |F|}\) and \(\mu \in \mathbb{R}^{|(|A^2| - |A|) \times |F|}|\) are the dual vectors associated with (5.44), (5.45) and (5.46) respectively, is a feasibility cut and will separate the infeasible point \(z^*\) if \(\alpha > 0\).

**Proof.** The dual objective function of the RFSEP is to

\[
\max \left\{ \sum_{i \in A} \sum_{r \in F} (v_{ri}/d_i) \pi^1_{ir} + \sum_{i \in A} \sum_{r \in F} (v_{ri}/d_i) \pi^2_{ir} + \sum_{(i,j) \in A^2} \sum_{r \in F} \min\{v_{ri}, v_{rj}\}/d_i z_{ij}^* \mu_{ijr} \right\}.
\]

In the case of infeasibility, \(\alpha > 0\) is an optimal solution to the RFSEP and, therefore,

\[
\sum_{i \in A} \sum_{r \in F} (v_{ri}/d_i) \pi^1_{ir} + \sum_{i \in A} \sum_{r \in F} (v_{ri}/d_i) \pi^2_{ir} + \sum_{(i,j) \in A^2} \sum_{r \in F} \min\{v_{ri}, v_{rj}\}/d_i z_{ij}^* \mu_{ijr} > 0. \tag{5.49}
\]
5.6. A branch-and-cut framework

In order to separate the infeasible point $z^*$, the violated cut

$$
\sum_{(i,j) \in A^2} \sum_{r \in F} \min\{v_{ri}, v_{rj}\} / d_i z_{ij} \mu_{ijr} \leq -\sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{1r}^i + v_{ri} / d_i \pi_{2r}^i) \tag{5.50}
$$

is added to the master problem.

5.6.5 The RFSEP with aggregated constraints (RFSEP-AC)

Recall that the constraints (5.4), which allow the flow variables $f_{rij}$ to take on values if the corresponding linear ordering variable $z_{ij}$ has been set, can also be written in an aggregated form. For this purpose a separation problem, called the RFSEP with aggregated constraints (RFSEP-AC), is introduced.

The objective of the RFSEP-AC is to

$$
\text{minimise } \alpha, \tag{5.51}
$$

subject to the constraints

$$
\sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj} / d_j, \quad j \in A, \ r \in F, \tag{5.52}
$$

$$
\sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri} / d_i, \quad i \in A, \ r \in F, \tag{5.53}
$$

$$
\sum_{r \in F} f_{rij} + \alpha = \left( \sum_{r \in F} U_r \right) z^*_{ij}, \quad (i, j) \in A^2, \ i \neq j. \tag{5.54}
$$

The vector $z^* \in \{0, 1\}^{A^2}$ is the solution obtained for the linear ordering variables $z_{ij}$ by solving the master problem. If $\alpha > 0$, the sub-problem is infeasible, otherwise it is feasible.

**Theorem 5.2.** For a given vector $z^* \in \{0, 1\}^{A^2}$ in the RFSEP-AC, the cut

$$
\sum_{(i,j) \in A^2} \left( \sum_{r \in F} U_r \right) z^*_{ij} \mu_{ij} \leq -\sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{1r}^i + v_{ri} / d_i \pi_{2r}^i), \tag{5.55}
$$

where $\pi^1 \in \mathbb{R}^{\{|A|\} \times |F|}$, $\pi^2 \in \mathbb{R}^{\{|A|\} \times |F|}$ and $\mu \in \mathbb{R}^{(|A|^2 - |A|)}$ are the dual vectors associated with (5.52), (5.53) and (5.54) respectively, is a feasibility cut and will separate the infeasible point $z^*$ if $\alpha > 0$.

**Proof.** The dual objective function of RFSEP-AC is to,

$$
\text{maximise } \left\{ \sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{1r}^i) + \sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{2r}^i) + \sum_{(i,j) \in A^2} \left( \sum_{r \in F} U_r \right) z^*_{ij} \mu_{ij} \right\}. \tag{5.56}
$$

In the case of infeasibility, $\alpha > 0$ is an optimal solution to the RFSEP-AC and, therefore,

$$
\sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{1r}^i) + \sum_{i \in A} \sum_{r \in F} (v_{ri} / d_i \pi_{2r}^i) + \sum_{(i,j) \in A^2} \left( \sum_{r \in F} U_r \right) z^*_{ij} \mu_{ij} > 0. \tag{5.57}
$$
In order to separate the infeasible point $z^*$, the violated cut
\[
\sum_{(i,j) \in A^2} \left( \sum_{r \in F} U_r \right) z_{ij} \mu_{ij} \leq -\sum_{i \in A} \sum_{r \in F} (v_{ri}/d_i \pi^1_{ir} + v_{ri}/d_i \pi^2_{ir})
\] (5.58)
is added to the master problem.

5.6.6 An incremental implementation of the RFSEP (RFSEP-INC)

The argument for following a decomposition approach is to improve scalability, specifically when problem instances are considered with a large number of resources. Although a speed-up can be expected as a result of adopting a branch-and-cut approach, computing efficiency may be hampered by the large number of flow variables $f_{rij}$ present in the RFSEP.

The proposed incremental implementation of the RFSEP (RFSEP-INC) is to solve the RFSEP in a sequential manner, involving only a partial optimisation problem. Consider the constraints of the RFSEP, given by (5.44)–(5.46). A sequential approach is achieved by reformulating the resource flow-related constraints for a specific resource $r \in F$ as
\[
\sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj}/d_j, \quad j \in A, \tag{5.59}
\]
\[
\sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri}/d_i, \quad i \in A, \tag{5.60}
\]
\[
f_{rij} - \min\{v_{ri}, v_{rj}\}/d_iz_{ij} \leq 0, \quad (i,j) \in A^2, \ i \neq j. \tag{5.61}
\]
The notation RFSEP($r$) is used to denote the RFSEP being solved with (5.59)–(5.61) for a specific resource $r \in F$. The feasibility cuts obtained from the RFSEP remain valid, since they are parameterised according to the resource set $F$, and solving RFSEP($r$) implies that $|F| = 1$ for the cut formulations (5.47) and (5.55). The RFSEP-INC, therefore, entails solving RFSEP($r$) for each $r \in F$ and adding the resulting feasibility cuts to the master problem.

5.6.7 Master problem alternatives

The exposition of the Benders decomposition in Section 5.6.3 above, is in terms of the RCSP. The master problem for the decomposed problem corresponds to the original RCSP formulation in Section 4.2.1, but with the resource flow-related constraints removed. The primary requirement for the application of the separation problem RFSEP in a branch-and-cut paradigm, is for the master problem to provide it with solutions $z^*$ to the linear ordering variables. This implies that, irrespective of other possible auxiliary variables, the only links between the master and the sub-problem are the linear ordering variables $z_{ij}$.

The RMSP, TMSP and ATMSP problem formulations may easily be incorporated into the branch-and-cut framework, since the only requirement for them is to provide the separation problem RFSEP with a solution vector $z^*$. Therefore, by removing the resource flow-related constraints (4.13)–(4.15), the reduced RMSP formulation is fit to be a master problem. Similarly, by removing the resource flow-related constraints (4.27)–(4.29), the reduced TMSP formulation may be used as a master problem. The same logic may be applied to the ATMSP formulation. The variations on the RMSP, TMSP and ATMSP defined above, may also be incorporated into a branch-and-cut framework. For instance, recall that the RMSP-AC and the RMSP-GR
are formulations of the RMSP with aggregated constraints and a reduced resource flow graph, respectively. Each of these variations may serve as a master problem within a branch-and-cut approach, provided the flow-related constraints are removed and a solution vector $z^*$ can be provided to the separation routine.

5.6.8 The resource flow rounding heuristic (RFRH)

The branch-and-cut approach, outlined in Figure 5.1, suggests the use of a heuristic for generating primal solutions on completion of the separation process. Consider the master problem (5.39)–(5.40). Recall that the vector $x$ is the adjoined vector $(s, z)$, where $s$ is the vector of starting variables and $z$ the vector of linear ordering variables. The latter is also the only link required between the master and the sub-problem within the branch-and-cut.

The motivation for the rounding heuristic suggested in this section is that the LP solutions obtained for each of the nodes in the branch-and-bound tree may give an indication of which linear ordering variables $z_{ij}$ are likely to take on a value of one in an optimal solution of the overall MILP problem. The argument is that the variables $z_{ij}$ act as capacities for the corresponding resource flow variables $f_{rij}$. That is, the resource flow values dictate what the entries of $z$ will be in the final solution.

The resource flow rounding heuristic (RFRH) is simply a copy of the MILP problem being solved by the branch-and-cut method (comprising both the master and sub-problems), together with the requirement that the variables $z_{ij}$ are fixed to one if $z_{ij}^* > 0$, where $z_{ij}^*$ is the LP solution of the corresponding variable at a branch-and-bound node. It is evident that the performance of the RFRH is highly dependent on the number of variables that are fixed to one.

The exact formulation of the RFRH is implied by the MILP problem being solved according to the branch-and-cut method. For instance, if the MILP to be solved by the branch-and-cut method is the RMSP-AC-GR, then this will also be the formulation of the RFRH.

5.6.9 The RFRH with guaranteed feasibility (RFRH-GF)

The RFRH as described above entails setting some of the linear ordering variables $z_{ij}$ to one, based on solution values obtained from the LP relaxation of a branch-and-bound node. This is, however, a greedy approach which is not guaranteed to produce a feasible solution. A two-step approach, called the RFRH with guaranteed feasibility (RFRH-GF), is suggested to address this shortcoming and will always output a feasible solution.

Consider that the problem being solved is the RMSP and let $z^F$ be the fractional solution vector of a node in the branch-and-bound tree. The first step in the RFRH-GF involves a sub-problem, referred to as RFRH-GF(1), which attempts to set the appropriate linear ordering variables $z_{ij}$ to one by

$$\text{maximising } \sum_{(i, j) \in A^2} \left\lceil z^F_{ij} \right\rceil z_{ij}$$

subject to the constraints
s_j - s_i \geq d_i, \quad i \in A, \; i \in S(i), \quad (5.63)

s_j - s_i - (d_i + M)z_{ij} \geq -M, \quad (i, j) \in A^2, \; i \neq j, \; r \in F, \quad (5.64)

\sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj}/d_j, \quad j \in A, \; r \in F, \quad (5.65)

\sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri}/d_i, \quad i \in A, \; r \in F, \quad (5.66)

f_{rij} - \min\{v_{ri}, v_{rj}\}/d_i z_{ij} \leq 0, \quad (i, j) \in A^2, \; i \neq j, \; r \in F. \quad (5.67)

The effect of the objective function (5.62) is that, for an optimal solution, the linear ordering variables $z_{ij}$ will be set to one based on the LP relaxation solution $z^F_{ij}$, provided the resource flow-related constraints (5.64)–(5.67) are satisfied. At worst, it may be that the suggested pattern for the linear ordering variables as prescribed by the solution vector $z^F$ is unattainable, in which case an arbitrary feasible solution is produced.

Let $z^I$ and $s^I$ be the integral solution vectors obtained by solving the RFRH-GF(1). Recall from the previous section that the RFRH is a copy of the MILP problem being solved which comprises both the master problem and the sub-problem, as defined for the branch-and-cut framework. Therefore, the next step in the RFRH-GF approach is to determine the values of the remaining variables in the RMSP, given the integral solution vectors $z^I$ and $s^I$. It is anticipated that the solution of the RMSP model would be made easier, if solutions to the “difficult” variables already exist. It should be noted, however, that solving the RMSP problem when provided with values to the linear ordering variables $z_{ij}$ and the start time variables $s_i$, does not require all of the original variables and constraints in the formulation. More specifically, the linear ordering variables $z_{ij}$, as well as the constraints that involve these variables, may be omitted since the objective function of the RMSP may be calculated as a function of the remaining variables.

The objective of the second sub-problem, referred to as RFRH-GF(2), is to

$$\text{maximise } \sum_{i \in A} y_i, \quad (5.68)$$

subject to the constraints

$$s^I_i - \sum_{v \in V} \lambda_{iv}s_{iv} = 0, \quad i \in A, \quad (5.69)$$

$$y_i - \sum_{v \in V} \lambda_{iv}y_{iv} = 0, \quad i \in A, \quad (5.70)$$

$$\sum_{v \in V} \lambda_{iv} = 1, \quad i \in A, \quad (5.71)$$

$$\lambda_{i0} - l_{i1} \leq 0, \quad i \in A, \quad (5.72)$$

$$\lambda_{iv} - l_{iv} - l_{i(v+1)} \leq 0, \quad i \in A, \; v \in V \setminus \{0, N - 1\}, \quad (5.73)$$

$$\lambda_{i(N-1)} - l_{i(N-1)} \leq 0, \quad i \in A. \quad (5.74)$$

Constraint set (5.69) is provided with the solution vector $s^I$. The remaining variables are required in order to calculate the objective function value.

In the case where the MILP problem being solved is the TMSP, the objective of the sub-problem RFRH-GF(1) is to

$$\text{maximise } \sum_{(i, j) \in A^2, \; i \neq j} \lfloor z^F_{ij} \rfloor z_{ij} \quad (5.75)$$
subject to the TMSP constraints (4.24)–(4.30). Similarly, in the case where the MILP problem being solved is the ATMSP, the same objective function is maximised subject to the ATMSP constraints (5.28)–(5.35).

The primary output of the RFRH-GF(1), for both the TMSP and the ATMSP formulations, is the integral solution vector \( x^I \). Note that \( x^I \) is already feasible with respect to the precedence and resource constraints. Therefore, the second subproblem RFRH-GF(2) for both the TMSP and the ATMSP only has to calculate the objective function value

\[
Z = \sum_{i \in A} \sum_{r \in R} \sum_{k \in T} \left( \sum_{t=k}^{S^T_i} \sum_{r=k}^{S^T_r} c_{r \max} v_{ri} (1 + \alpha_t) - S^T_t \right) x_{ik}.
\]

(5.76)

### 5.6.10 An incremental implementation of the RFRH (RFRH-INC)

As with the RFSEP-INC, the motivation for an incremental implementation of the RHRH (RFRH-INC), is to improve scalability of the branch-and-cut approach, specifically when problem instances with a large number of resources are considered. The suggested approach entails solving the RFRH in a sequential manner, involving only one resource at a time. Consider the resource flow-related constraints of both the RMSP and the TMSP, reformulated for a specific resource \( r \in F \) as

\[
\sum_{i \in A \setminus \{j\}} f_{rij} = v_{rj}/d_j, \quad j \in A, \tag{5.77}
\]

\[
\sum_{j \in A \setminus \{i\}} f_{rij} = v_{ri}/d_i, \quad i \in A, \tag{5.78}
\]

\[
f_{rij} - \min\{v_{ri}, v_{rj}\}/d_i z_{ij} \leq 0, \quad (i, j) \in A^2, \ i \neq j. \tag{5.79}
\]

The notation \( \text{RFRH}(r) \) is used to denote the RFRH being solved subject to (5.77)–(5.79) for a specific resource \( r \in F \). It should be noted, however, that the sequence of solutions \( z^r \) obtained by solving \( \text{RFRH}(r) \) for each \( r \in F \) may be inconsistent. In this case a greedy approach is suggested whereby the final solution is calculated as \( z = z^1 \otimes z^2 \otimes \cdots \otimes z^{|F|} \), where \( \otimes \) denotes the binary OR operator.

### 5.6.11 Augmenting the master problem with valid inequalities

Valid inequalities have been applied successfully to many large-scale MILP problems for improving computing times [22]. In the context of a branch-and-cut approach, they may either be added explicitly to the master problem, or implicitly through the use of separation routines.

Let \( \mathcal{V} \subseteq A \) be a subset of activities and let \( \bar{\mathcal{V}} = A \setminus \mathcal{V} \) be its complement. The set of arcs \( \mathcal{E}(\mathcal{V}, \bar{\mathcal{V}}) \), where each arc is incident from an activity in \( \mathcal{V} \) and incident to an activity \( \bar{\mathcal{V}} \), is called a cut. For each arc \( (i, j) \in A \), these exist a linear ordering variable \( z_{ij} \). Recall from the various formulations above that one of the uses of the variable \( z_{ij} \) is to allow the corresponding flow variables \( f_{rij} \) to take on values, if \( z_{ij} = 1 \). Conceptually, the variable \( z_{ij} \) may be interpreted as a capacity variable within a network flow context. More specifically, the capacity of an arc \( (i, j) \) is defined as \( \min\{v_{ri}, v_{rj}\}/d_i z_{ij} \) according to the disaggregated constraint formulations (4.8), (4.15) and (4.29) in the RCSP, RMSP and TMSP, respectively.

Recall from Section 4.2 that a source activity \( i^+ \) and a sink activity \( i^- \) are added to the set \( A \) in order to facilitate the formulation of the resource flow-related constraints. Furthermore, the
resource requirements for the source and sink activities are \( v_{ri^+} = U_r \) and \( v_{ri^-} = U_r \), respectively, for each \( r \in F \). This implies that, for a specific resource \( r \in F \), a total of \( U_r \) resource units need to flow from the sink to the source. Since there is at least one direct arc from the sink to the source, which is the case for both the complete graph \( G(A,E) \) and the reduced graph \( G'(A,E') \), the resource requirement that \( U_r \) resource units must flow from the sink to the source will always be satisfied. This implies, for any cut \( E(V,\bar{V}) \), that there will be at least one variable \( z_{ij} \) set to one in any feasible solution, provided that \( i^+ \in V \) and \( i^- \in \bar{V} \). If it is assumed that for all activities \( i \in A, v_{ir} > 0 \) for at least one \( r \in F \), the latter requirement of having \( i^+ \in V \) and \( i^- \in \bar{V} \) may be relaxed.

For any subset \( V \) the valid inequality

\[
\sum_{(i,j) \in E(V,\bar{V})} z_{ij} \geq 1, \tag{5.80}
\]

called a cutset valid inequality (CVI), must hold in order for any given solution \( z^* \) to be feasible. For the computational studies that involve the application of cutset valid inequalities, which are presented in the next chapter, cuts are generated up front and added explicitly to the branch-and-cut master problem. Furthermore, cutsets were generated only for combinations with \( |V| = 2 \) and \( |V| = 3 \), in order to limit the number of constraints added to the master problem.

The notation used above for distinguishing between the various problem formulations is augmented with the abbreviation CVI to include the application of cutset valid inequalities. For instance, to indicate the application of valid inequalities in the case of the master problem TMSP, the notation TMSP-CVI is used.

### 5.7 Summary

The primary objective of this chapter was to propose algorithmic enhancements for solving the mine scheduling optimisation problem. Problem reformulations were presented in an attempt to reduce the number of variables and constraints in both the RMSP and TMSP formulations. A branch-and-cut approach was suggested that relies on the decomposition of both the RMSP and the TMSP. Since both problems have the same underlying resource flow formulation, a single separation problem was proposed that is able to separate an infeasible point from the feasible region of the master problem. A heuristic was also proposed which constructs primal solutions based on the LP relaxation solutions obtained by solving the sub-problems of the branch-and-bound approach. An incremental implementation of both the separation problem and the primal heuristic was further suggested to improve the scalability of the branch-and-cut approach, specifically for problem instances having a large number of resources. The final contribution of this chapter was the proposed use of cutset inequalities, which are explicitly added to the master problem of the branch-and-cut approach.

The final section of this chapter has demonstrated that several variations are available when formulating the resource flow-based mine scheduling optimisation problem. To assist the reader in navigating through all these combinations, a list of problem formulations and their abbreviations is provided in Table 5.1.
### Abbreviation Description Formulation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCSP</td>
<td>Flow-based resource constrained scheduling problem</td>
<td>Section 4.2.1</td>
</tr>
<tr>
<td>RMSP</td>
<td>Resource flow mine scheduling optimisation problem</td>
<td>Section 4.2.3</td>
</tr>
<tr>
<td>RMSP-AC</td>
<td>Resource flow mine scheduling optimisation problem with aggregated constraints</td>
<td>Section 5.2</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>Resource flow mine scheduling optimisation problem based on graph reduction</td>
<td>Section 5.3.1</td>
</tr>
<tr>
<td>RMSP-AC-GR</td>
<td>Resource flow mine scheduling optimisation problem based on both aggregated constraints and graph reduction</td>
<td>Section 5.4</td>
</tr>
<tr>
<td>TMSP</td>
<td>Time-indexed mine scheduling optimisation problem</td>
<td>Section 4.2.3</td>
</tr>
<tr>
<td>TMSP-AC</td>
<td>Time-indexed mine scheduling optimisation problem with aggregated constraints</td>
<td>Section 5.2</td>
</tr>
<tr>
<td>TMSP-GR</td>
<td>Time-indexed mine scheduling optimisation problem based on graph reduction</td>
<td>Section 5.3.2</td>
</tr>
<tr>
<td>TMSP-AC-GR</td>
<td>Time-indexed mine scheduling optimisation problem based on both aggregated constraints and graph reduction</td>
<td>Section 5.4</td>
</tr>
<tr>
<td>ATMSP</td>
<td>An alternative time-indexed mine scheduling optimisation problem</td>
<td>Section 5.5</td>
</tr>
<tr>
<td>ATMSP-AC</td>
<td>An alternative time-indexed mine scheduling optimisation problem with aggregated constraints</td>
<td>Section 5.5</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>An alternative time-indexed mine scheduling optimisation problem based on graph reduction</td>
<td>Section 5.5</td>
</tr>
<tr>
<td>ATMSP-AC-GR</td>
<td>An alternative time-indexed mine scheduling optimisation problem based on both aggregated constraints and graph reduction</td>
<td>Section 5.5</td>
</tr>
<tr>
<td>X-CVI</td>
<td>Master problem used in the branch-and-cut augmented with cutset valid inequalities where X may be substituted with any of the following: RMSP, RMSP-AC, RMSP-GR, RMSP-AC-GR, TMSP, TMSP-AC, TMSP-GR, TMSP-AC-GR, ATMSP, ATMSP-AC, ATMSP-GR and ATMSP-AC-GR</td>
<td>Section 5.6.11</td>
</tr>
<tr>
<td>RFSEP</td>
<td>Resource flow separation problem used as a sub-problem in the branch-and-cut</td>
<td>Section 5.6.4</td>
</tr>
<tr>
<td>RFSEP-AC</td>
<td>Resource flow separation problem with aggregated constraints</td>
<td>Section 5.6.5</td>
</tr>
<tr>
<td>RFSEP-INC</td>
<td>Resource flow separation problem based on an incremental implementation</td>
<td>Section 5.6.6</td>
</tr>
<tr>
<td>RFRH</td>
<td>Resource flow rounding heuristic</td>
<td>Section 5.6.8</td>
</tr>
<tr>
<td>RFRH-GF</td>
<td>Resource flow rounding heuristic with guaranteed feasibility</td>
<td>Section 5.6.9</td>
</tr>
<tr>
<td>RFRH-INC</td>
<td>Resource flow rounding heuristic based on an incremental implementation</td>
<td>Section 5.6.10</td>
</tr>
</tbody>
</table>

**Table 5.1:** A list of abbreviations used for distinguishing between all combinations of problem formulations
In earlier chapters of this thesis it was shown that the underground mine scheduling optimisation problem may be formulated as an RCSP. Due to the inherent hardness of the problem, alternative problem formulations and a decomposition approach were suggested for the purpose of improving computing times. The main objective of this chapter is to report on the efficiency of the proposed model reformulations and the resulting branch-and-cut implementation that is based on the proposed Benders decomposition.

The solution methodology adopted in this thesis for solving the underground mine scheduling optimisation problem is set within an exact framework. The implication is that problem instances are either solved to optimality, or if terminated prematurely with a feasible solution, a quantification of the degree to which optimality has been achieved is obtained. The primary measure of efficiency applied in this chapter is, therefore, the number of problem instances that could be solved to optimality within a specified time limit. The secondary success indicators are the average time required to find an optimal solution and the average integrality gap in the case where problems could not be solved to optimality but for which at least one feasible solution could be computed.

In the following section, implementation details of the suggested approaches and the software and hardware used as part of the computing environment are described. For the purpose of evaluating the proposed reformulations and algorithmic contributions, three data sets were used of which the details are provided in the section to follow. Finally, the last section in this chapter is devoted to computational results demonstrating the success of the contributions made in this study.
Chapter 6. Empirical results

6.1 Implementation details

All of the empirical tests reported in this chapter were performed on an HP Compaq Elite 8300, with eight cores and 32GB of RAM. SuSE Linux was used as operating system and all coding was done in the programming language C++. The IBM product, CPLEX v12.6 [36], was used as MILP solver. The Concert Technology library was used as the interface for specifying the mathematical model and the callback functionality in CPLEX was utilised for implementing the branch-and-cut approach.

6.2 Data sets

Several data sets for the RCSP and its variants are available in the research community for the purpose of testing algorithmic ideas. For instance, the project scheduling problem library (PSPLIB) [67] is a repository of RCSP problem instances which has been referenced extensively over the years. The PSPLIB comprises the data sets J30, J60, J90 and J120, which are sets of RCSP instances with respectively 30, 60, 90 and 120 activities. Each data set has 480 different problem instances, except for the J120 data set which has 600 problem instances. Details on how these problem instances were created can be found in [41].

The data sets from the PSPLIB used in this study are limited to J30 and J60 due to the large number of different proposed problem formulations that need to be evaluated. More specifically, according to Table 5.1 in the previous chapter, there are 12 different problem formulations and 144 master-, separation- and heuristic-problem combinations possible for the branch-and-cut approach. Therefore, only the first 100 out of the total of 480 problem instances were considered for both J30 and J60.

In addition to the PSPLIB, further data sets were generated randomly using the software RanGen2 [70]. Details on the design of RanGen2 can be found in [85]. The major benefit of using RanGen2 is the ability to specify several input parameters which influence the properties of the randomly generated problem instances. For instance, one of the parameters in RanGen2, which is called $I_2$, determines the degree of serialisation with respect to the precedence graph. More specifically, if the value $I_2 = 1$ is specified by the user, a random problem instance is created for which all the activities are serial according to the precedence graph. On the other hand, if $I_2 = 0$, all the activities are in parallel. Another attractive feature of RanGen2 is its ability to generate an arbitrary sized network with an unlimited number of activities and resources. This is especially useful for this study since one of the objectives is to test the scalability of the newly proposed formulations and decomposition approach. A range of problem instances having 100 and 150 activities, respectively, were generated using RanGen2. For the purpose of testing the scalability of the newly proposed formulations and algorithms, these instances were generated with 30 resources each. In the remainder of this chapter, the problem instances with 100 activities are collectively referred to as the RG100 data set, while the instances with 150 activities are collectively referred to as the RG150 data set. The RG100 and RG150 data sets both comprise 50 problem instances. These instances were generated to be progressively more “difficult” by incrementing the parameter $I_2$ with 10% after the generation of every 10th problem instance. That is, the first group of 10 instances were generated with $I_2 = 0.1$ and the last group of 10 instances were generated with $I_2 = 0.5$.

For the purpose of conducting a computational study, a time limit was imposed on the solution of each problem instance. For example, a time limit of 150 seconds was imposed on the solution times for the J30 data set, 300 seconds for the J60 data set, 900 seconds for the RG100 data and 1 800 seconds for the RG150 data set.
6.3. Result set I: Effect of directional and transitivity inequalities

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RCSP</td>
<td>70</td>
<td>13.2</td>
<td>30</td>
</tr>
<tr>
<td>RCSP without (4.3)–(4.4)</td>
<td>74</td>
<td>5.1</td>
<td>26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RCSP</td>
<td>17</td>
<td>89.2</td>
<td>72</td>
</tr>
<tr>
<td>RCSP without (4.3)–(4.4)</td>
<td>60</td>
<td>33.9</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 6.1: The detrimental effect of directional and transitivity inequalities on computing times.

Details of a data set obtained from a South African mining company are provided in Section 6.6 of this chapter. Results pertaining to the practical use of mathematical models for solving real-world underground mine scheduling problems are provided and computational results are discussed which highlight the efficiency of the proposed mathematical formulations and algorithmic enhancements.

The following section provides details of the computational results obtained for the newly proposed problem formulations and algorithmic approaches applied to the PSPLIB benchmark instances and the problem instances generated randomly using RanGen2.

6.3 Result set I: Effect of directional and transitivity inequalities

The resource flow-based RCSP formulation (4.1)–(4.8) introduced in Chapter 4, has been applied in previous studies for the purpose of comparing it to other formulations, see e.g. [2, 3]. In Chapter 4 it was shown that the directional constraints (4.3) and the transitivity constraints (4.4) are redundant. The reason for their inclusion in RCSP formulations is not clearly stated in either [2] or [3], other than that they are valid inequalities. It is assumed, in general, that valid inequalities may in some cases improve computing times, depending on the problem formulation and the problem instance.

Computational results showing the effect on computing times as a result of including the redundant constraints (4.3)–(4.4), are provided in Table 6.1. The computational tests were performed on the J30 and J60 data sets. The first column provides information on the instances solved to optimality. The second column provides information on the instances that could not be solved to optimality, but for which at least one feasible solution could be obtained. The last column reveals the number of problem instances for which no solution could be found in the allowed solution time. For the RCSP formulation, 70 instances of the J30 data set were solved to optimality with an average running time of 13.2 seconds and for 30 instances not solved to optimality an average integrality gap of 27.9% was obtained. However, solving the RCSP without the redundant constraints (4.3)–(4.4) significantly improved the situation. A total of 74 instances of the J30 data set were solved to optimality with an average running time of 5.1 seconds. The average integrality gap was 22.1% for the remaining 26 instances not solved to optimality.

The effect of excluding the redundant constraints (4.3)–(4.4) is even more significant for the J60 data set. Compared to the RCSP formulation for which only 17 instances could be solved to optimality, 60 problem instances were solved to optimality when excluding the redundant constraint, with an average running time of 33.9 seconds. Furthermore, an average integrality
Chapter 6. Empirical results

Results for the J30 data set

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>86</td>
<td>6.17</td>
<td>14</td>
</tr>
<tr>
<td>RMSP</td>
<td>86</td>
<td>6.22</td>
<td>14</td>
</tr>
<tr>
<td>RMSP-AC-G</td>
<td>86</td>
<td>6.50</td>
<td>14</td>
</tr>
<tr>
<td>RMSP-AC</td>
<td>86</td>
<td>7.03</td>
<td>14</td>
</tr>
<tr>
<td>ATMS-P-GR</td>
<td>84</td>
<td>9.24</td>
<td>16</td>
</tr>
<tr>
<td>ATMS-P</td>
<td>84</td>
<td>9.31</td>
<td>16</td>
</tr>
<tr>
<td>ATMS-P-AC</td>
<td>83</td>
<td>7.56</td>
<td>17</td>
</tr>
<tr>
<td>ATMS-P-AC-G</td>
<td>83</td>
<td>8.57</td>
<td>17</td>
</tr>
<tr>
<td>TMSP-AC-G</td>
<td>72</td>
<td>7.20</td>
<td>11</td>
</tr>
<tr>
<td>TMSP-AC</td>
<td>71</td>
<td>11.62</td>
<td>14</td>
</tr>
<tr>
<td>TMSP-GR</td>
<td>70</td>
<td>6.13</td>
<td>12</td>
</tr>
<tr>
<td>TMSP</td>
<td>70</td>
<td>8.25</td>
<td>11</td>
</tr>
</tbody>
</table>

Results for the J60 data set

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RMSP</td>
<td>69</td>
<td>21.77</td>
<td>31</td>
</tr>
<tr>
<td>RMSP-AC</td>
<td>68</td>
<td>20.01</td>
<td>32</td>
</tr>
<tr>
<td>RMSP-AC-G</td>
<td>67</td>
<td>15.17</td>
<td>33</td>
</tr>
<tr>
<td>RMSP-G</td>
<td>67</td>
<td>15.71</td>
<td>33</td>
</tr>
<tr>
<td>ATMS-P-G</td>
<td>59</td>
<td>25.60</td>
<td>38</td>
</tr>
<tr>
<td>ATMS-P-AC</td>
<td>59</td>
<td>30.18</td>
<td>38</td>
</tr>
<tr>
<td>ATMS-P-AC-G</td>
<td>58</td>
<td>19.12</td>
<td>40</td>
</tr>
<tr>
<td>ATMS-P</td>
<td>58</td>
<td>26.06</td>
<td>40</td>
</tr>
<tr>
<td>TMSP-AC-G</td>
<td>39</td>
<td>41.89</td>
<td>13</td>
</tr>
<tr>
<td>TMSP</td>
<td>38</td>
<td>44.30</td>
<td>9</td>
</tr>
<tr>
<td>TMSP-GR</td>
<td>37</td>
<td>36.35</td>
<td>11</td>
</tr>
<tr>
<td>TMSP-AC</td>
<td>36</td>
<td>59.28</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 6.2: The computational benefit of the newly proposed reformulations for the PSPLIB data sets.

A gap of 24% was obtained for the remaining problem instances when excluding the redundant constraints. The inclusion of these redundant constraints in the RCSP formulation resulted in 11 problem instances in the J60 data set for which no solution could be computed within the allowed solution time.

It is evident from the above results that the inclusion of the redundant constraints (4.3)–(4.4) has a detrimental effect on computing times when solving the RCSP. For this reason these constraints are excluded from the RMSP and the TMSP formulations in all of the experimental work presented in the remainder of this chapter.

6.4 Result set II: The efficiency of the proposed reformulations

Several adaptations of the RMSP and TMSP were proposed in the previous chapter. The first of these adaptations involves the aggregation of the constraints (5.4) to produce the constraints (5.5). The aggregation approach applies to both the RMSP and the TMSP, and the abbreviations RMSP-AC and TMSP-AC, are used for this purpose. A graph reduction approach was proposed for both the RMSP and the TMSP, and the abbreviations RMSP-GR and TMSP-GR are used to refer to the resulting reformulations. The use of a combined constraint aggregation and graph reduction formulation is indicated by the postfix AC-GR. Finally, the ATMS-P is an alternative
6.4. Result set II: The efficiency of the proposed reformulations

### Results for the RG100 data set

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RMSP-AC-GR</td>
<td>11</td>
<td>296.02</td>
<td>34</td>
</tr>
<tr>
<td>RMSP-AC</td>
<td>10</td>
<td>303.67</td>
<td>33</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>9</td>
<td>423.12</td>
<td>21</td>
</tr>
<tr>
<td>RMSP</td>
<td>7</td>
<td>370.29</td>
<td>21</td>
</tr>
<tr>
<td>ATMSP-AC</td>
<td>2</td>
<td>618.88</td>
<td>7</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>1</td>
<td>490.41</td>
<td>12</td>
</tr>
</tbody>
</table>

### Results for the RG150 data set

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Solved to optimality</th>
<th>Time limit reached</th>
<th>No solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Instances</td>
<td>Time (s)</td>
<td># Instances</td>
</tr>
<tr>
<td>RMSP-AC-GR</td>
<td>0</td>
<td>—</td>
<td>32</td>
</tr>
<tr>
<td>RMSP-AC</td>
<td>0</td>
<td>—</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 6.3: The computational benefit of the newly proposed reformulations for the randomly generated data sets. Results are only provided for problem formulations for which at least one feasible solution was computed.

The computational results for the newly proposed formulations are provided in Tables 6.2 and 6.3. Consider the results for the J30 data set in Table 6.2. The application of the RMSP formulation and all of its variants produced optimal solutions for 86 of the problem instances. The best computing time was recorded for the RMSP-GR with an average processing time of 6.17 seconds. For the problem instances that could not be solved to optimality, the best integrality gap was due to the RMSP-AC with an average of 1.87%. The TMSP formulation and its variants performed much worse and only produced optimal solutions for a maximum of 72 problem instances. Furthermore, they were the only formulations for which no solutions could be found for up to 19 of the problem instances. The performance of the ATMSP formulation and its variants is noticeably better and produced optimal solutions for up to 84 problem instances.

A similar pattern in computational results may be observed for the J60 data set. The best performing problem formulation is the RMSP. All of the proposed variants of the RMSP improved on the average computing times. The best improvement is due to the RMSP-AC-GR with an average computing time of 15.17 seconds. Once again the TMSP formulation and its variants were the worst performers and no solutions could be found for more than half of the J60 problem instances. The ATMSP formulations showed considerable improvement over the TMSP formulations. The average computing time of the best ATMSP performer, namely ATMSP-AC-GR, was 19.12 seconds, which is lower than the average computing time of the RMSP.

Computational results for the randomly generated data sets RG100 and RG150 are provided in Table 6.3. Out of the 50 instances of the RG100 data set, a maximum of eleven instances could be solved to optimality by applying the RMSP-AC-GR formulation. Feasible solutions were computed for up to 34 problem instances, with an average gap of 9.26%. In contrast to the RMSP formulations, only two problem instances could be solved to optimality according to the ATMSP-AC problem formulation and only one instance according to the ATMSP-GR formulation. Applying the ATMSP-AC and ATMSP-GR formulations, feasible solutions were obtained within the specified time limit for 7 and 12 problem instances, respectively. No feasible solution could be found by using the TMSP formulation or any of its variants.

The outcome for the RG150 data set is even worse and no optimal solutions could be computed.
Results for the J30 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th># Instances</th>
<th>Time (s)</th>
<th># Instances</th>
<th>Gap (%)</th>
<th># Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>74</td>
<td>13.00</td>
<td>13</td>
<td>3.39</td>
<td>13</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>74</td>
<td>13.08</td>
<td>13</td>
<td>3.44</td>
<td>13</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-INCR</td>
<td>74</td>
<td>16.44</td>
<td>12</td>
<td>6.50</td>
<td>14</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-INCR</td>
<td>74</td>
<td>16.53</td>
<td>12</td>
<td>5.14</td>
<td>14</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-INCR</td>
<td>74</td>
<td>16.82</td>
<td>14</td>
<td>12.99</td>
<td>12</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>73</td>
<td>16.37</td>
<td>25</td>
<td>21.04</td>
<td>2</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>73</td>
<td>16.38</td>
<td>25</td>
<td>19.80</td>
<td>2</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INCR</td>
<td>73</td>
<td>16.55</td>
<td>25</td>
<td>19.70</td>
<td>2</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INCR</td>
<td>73</td>
<td>20.47</td>
<td>25</td>
<td>23.04</td>
<td>2</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INCR</td>
<td>73</td>
<td>20.78</td>
<td>25</td>
<td>22.36</td>
<td>2</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INCR</td>
<td>73</td>
<td>22.13</td>
<td>25</td>
<td>27.60</td>
<td>2</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>72</td>
<td>11.36</td>
<td>13</td>
<td>7.04</td>
<td>15</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP-INCR</td>
<td>71</td>
<td>27.89</td>
<td>28</td>
<td>39.73</td>
<td>1</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>69</td>
<td>21.10</td>
<td>31</td>
<td>30.93</td>
<td>0</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>68</td>
<td>15.58</td>
<td>14</td>
<td>17.48</td>
<td>18</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INCR</td>
<td>66</td>
<td>15.08</td>
<td>14</td>
<td>17.06</td>
<td>20</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-AC</td>
<td>66</td>
<td>19.08</td>
<td>9</td>
<td>9.85</td>
<td>25</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-AC</td>
<td>66</td>
<td>19.17</td>
<td>9</td>
<td>9.80</td>
<td>25</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-AC</td>
<td>66</td>
<td>25.89</td>
<td>32</td>
<td>37.34</td>
<td>2</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-AC</td>
<td>66</td>
<td>25.98</td>
<td>32</td>
<td>39.31</td>
<td>2</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-AC</td>
<td>64</td>
<td>16.77</td>
<td>10</td>
<td>14.82</td>
<td>26</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-AC</td>
<td>63</td>
<td>25.20</td>
<td>35</td>
<td>34.68</td>
<td>2</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP-AC</td>
<td>59</td>
<td>27.82</td>
<td>41</td>
<td>46.39</td>
<td>0</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-AC</td>
<td>56</td>
<td>13.58</td>
<td>11</td>
<td>26.42</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 6.4: The computational benefit of the newly proposed separation sub-problems within a branch-and-cut approach.

by applying any of the problem formulations. Feasible solutions could be computed for 32 of the problem instances using the RMSP-AC-GR formulation and 31 instances using the RMSP-AC formulation. The success of the RMSP-AC-GR and the RMSP-AC formulations was an expected result since the problem instances for the RG100 and the RG150 data sets have, in addition to a larger number of activities, also a larger number of resources. The aggregation of resources achieved through the RMSP-AC formulation results in a reduction of constraints which, in turn, places less of a burden on the MILP solver.

### 6.5 Result set III: The effectiveness of Benders decomposition

The proposed Benders decomposition approach, as outlined in the previous chapter, entails the formulation of a master problem and an accompanying sub-problem. The sub-problem, which is derived from the resource flow-based RCSP, involves only the resource flow-related variables and constraints. The anticipated advantage is that scalability may be improved due to the exponential number of flow variables that are being projected out of the master problem. Their effect, in terms of defining the feasible region of the resource flow-based RCSP, is incorporated back into the master problem in the form of Benders feasibility cuts. The separation problem RFSEP introduced in the previous chapter, which is essentially the resource flow-based sub-problem within the Benders decomposition framework, is responsible for generating the feasibility cuts. The RFSEP-AC and the RFSEP-INC are two variations on the RFSEP that were introduced in the previous chapter.
6.5. Result set III: The effectiveness of Benders decomposition

Results for the J60 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th># Instances</th>
<th>Time (s)</th>
<th># Instances</th>
<th>Gap (%)</th>
<th># Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>42</td>
<td>110.58</td>
<td>8</td>
<td>28.94</td>
<td>50</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>41</td>
<td>95.67</td>
<td>4</td>
<td>19.54</td>
<td>55</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>40</td>
<td>90.33</td>
<td>5</td>
<td>15.88</td>
<td>55</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>39</td>
<td>77.36</td>
<td>25</td>
<td>47.28</td>
<td>36</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>38</td>
<td>73.82</td>
<td>22</td>
<td>45.21</td>
<td>40</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>38</td>
<td>74.46</td>
<td>22</td>
<td>44.69</td>
<td>40</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>37</td>
<td>142.88</td>
<td>9</td>
<td>26.95</td>
<td>54</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>37</td>
<td>143.78</td>
<td>9</td>
<td>27.37</td>
<td>54</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-INC</td>
<td>33</td>
<td>97.49</td>
<td>22</td>
<td>41.36</td>
<td>45</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-INC</td>
<td>33</td>
<td>97.98</td>
<td>22</td>
<td>39.12</td>
<td>45</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>33</td>
<td>136.89</td>
<td>13</td>
<td>56.46</td>
<td>54</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-INC</td>
<td>32</td>
<td>98.16</td>
<td>24</td>
<td>45.56</td>
<td>44</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INC</td>
<td>32</td>
<td>141.92</td>
<td>11</td>
<td>40.45</td>
<td>57</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-AC</td>
<td>30</td>
<td>114.36</td>
<td>21</td>
<td>58.82</td>
<td>49</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-AC</td>
<td>30</td>
<td>114.96</td>
<td>21</td>
<td>29.65</td>
<td>49</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-AC</td>
<td>27</td>
<td>73.11</td>
<td>25</td>
<td>45.54</td>
<td>48</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-AC</td>
<td>25</td>
<td>103.70</td>
<td>25</td>
<td>49.82</td>
<td>50</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-AC</td>
<td>25</td>
<td>128.22</td>
<td>5</td>
<td>52.04</td>
<td>70</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-AC</td>
<td>25</td>
<td>129.16</td>
<td>5</td>
<td>52.04</td>
<td>70</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>23</td>
<td>118.93</td>
<td>24</td>
<td>41.66</td>
<td>53</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-AC</td>
<td>22</td>
<td>116.30</td>
<td>9</td>
<td>30.49</td>
<td>69</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>21</td>
<td>152.31</td>
<td>20</td>
<td>43.41</td>
<td>59</td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP-AC</td>
<td>19</td>
<td>127.92</td>
<td>18</td>
<td>53.56</td>
<td>63</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-AC</td>
<td>17</td>
<td>81.50</td>
<td>28</td>
<td>49.64</td>
<td>55</td>
</tr>
</tbody>
</table>

Table 6.5: The computational benefit of the newly proposed separation sub-problems within a branch-and-cut approach.

Recall from the previous chapter that a Benders decomposition approach may be applied to any of the three formulations RMSP, TMSP and ATMSP, and their different variants. In subsequent tables reporting on the computational efficiency of the proposed branch-and-cut approach, references to master problems which involve the application of graph reduction, for instance RMSP-GR, implies that graph reduction is also relevant to the corresponding separation sub-problem and the heuristic. However, for the sake of presentation, the acronym GR will only be appended to the name of the master problem formulation and not to the separation sub-problem and the heuristic. It should be noted that, due to the poor performance of the TMSP formulation reported in the previous section, it has been omitted from any further empirical tests.

Heuristic algorithms are incorporated as part of the branch-and-cut framework for the purpose of computing primal solutions. The three heuristic approaches, RFRH, RFRH-GF and RFRH-INCR, that were introduced in the previous chapter, utilise solution information from the nodes of the branch-and-bound tree to construct new primal solutions.

Computational results obtained when solving the resource flow-based RCSP with Benders decomposition are provided below in two parts. The first set of results demonstrates the effect on computing time when applying the three different separation sub-problems. The second set of results reports on the computing times achieved through the joint application of the separation sub-problems and the three heuristic approaches.

Table 6.4 provides information on the computational results for the J30 data set. The problem instances were solved by applying the three different separation sub-problems in conjunction with
the various RMSP and ATMSP formulations that were used as branch-and-cut master problems. A clear pattern is discernible from the results, showing that irrespective of the master problem formulation, the RFSEP and the RFSEP-INC dominated in terms of the number of problem instances solved to optimality. Furthermore, by examining the last column in Table 6.4, it is clear that the RMSP formulations were much more successful in the computation of feasible solutions compared to the ATMSP formulations. The verdict on the use of CVIs is uncertain considering that formulations augmented with CVI are present in both the top ranking and bottom ranking performers.

Similar results were obtained for the J60 data set, as reported in Table 6.5. The RFSEP and the RFSEP-INC dominated in terms of the number of problem instances solved to optimality, irrespective of the master problem formulation. Although it appears as if there is no clear pattern showing which of the master problem formulations are computationally more efficient, a ranking of the data according to the last column of Table 6.5 indicates otherwise. It suggests that the ATMSP formulations were more successful in the computation of feasible solutions than the RMSP formulations. Due to the poor performance of employing RFSEP-AC as a separation sub-problem within the branch-and-cut approach, it has been omitted from any further empirical tests.

It should be noted that for both the J30 and the J60 data sets, the average computing times for solving the problem instances with a branch-and-cut approach, as reported in Tables 6.4 and 6.5, are considerably higher than the computing times provided in Table 6.2. Furthermore, no feasible solutions could be computed for the RG100 and RG150 data sets by employing a branch-and-cut approach. Although it may seem that no computational benefit is realised by the implementation of a branch-and-cut approach, subsequent results based on the application of the proposed heuristics prove otherwise.

Computational results for incorporating the newly proposed heuristics into a branch-and-cut framework are provided in Tables 6.6–6.8. For the J30 data set, the top performing branch-and-cut approach, which managed to solve 82 problem instances to optimality, is using ATMSP-CVI for the master problem, RFSEP-INC for the separation sub-problem and RFRH-INC for the heuristic. The top four master formulations are ATMSP and the top seven separation sub-problems are RFSEP-INC. The worst performing heuristic for the J30 data set appears to be the RFRH-GF.

Compared to the J30 data set, the results for the J60 data set, as provided in Table 6.7, reveal a different picture. The two top performing branch-and-cut approaches were able to solve 48 problem instances to optimality. The first approach employed the ATMSP-CVI as a master problem, the RFSEP as a separation sub-problem and RFRH-INC as a heuristic. The only difference in the second approach is that the cutset valid inequalities have been omitted from the master problem. A very distinctive pattern for the J60 data set is that the RFSEP separation routine clearly outperformed the RFSEP-INC. For both the J30 and J60 data sets, the verdict on the use of CVIs is uncertain considering that the results are mixed and formulations augmented with CVI are present in both the top ranking and bottom ranking performers. The same applies to the use of graph reduction-related formulations.

The computational results for the RG100 and RG150 data sets are provided in Table 6.8. None of the RG100 and the RG150 problem instances could be solved to optimality. Furthermore, the branch-and-cut approach with ATMSP and its variants as master problems did not manage to compute any feasible solutions within the allowed time limit. Feasible solutions could only be obtained for the bare RMSP formulation as master problem. The inability of the RMSP formulation augmented with CVI to produce any feasible solutions may be attributed to the large number of valid inequalities being generated as a result of the larger number of activity
## Results for the J30 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th>Heuristic</th>
<th># Inst.</th>
<th>Optimal Time (s)</th>
<th>Time limit</th>
<th>No solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>82</td>
<td>11.09</td>
<td>18</td>
<td>3.40</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>81</td>
<td>7.26</td>
<td>19</td>
<td>4.98</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>81</td>
<td>8.10</td>
<td>19</td>
<td>4.50</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>81</td>
<td>8.49</td>
<td>19</td>
<td>3.09</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>78</td>
<td>18.68</td>
<td>22</td>
<td>2.97</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>78</td>
<td>19.11</td>
<td>22</td>
<td>3.07</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>77</td>
<td>18.17</td>
<td>23</td>
<td>3.37</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>76</td>
<td>16.03</td>
<td>24</td>
<td>4.56</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>76</td>
<td>20.06</td>
<td>24</td>
<td>4.86</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>76</td>
<td>17.67</td>
<td>24</td>
<td>3.12</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>75</td>
<td>15.33</td>
<td>25</td>
<td>4.36</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>74</td>
<td>16.29</td>
<td>26</td>
<td>4.40</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>74</td>
<td>14.01</td>
<td>26</td>
<td>3.93</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>73</td>
<td>10.42</td>
<td>27</td>
<td>3.10</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>73</td>
<td>10.90</td>
<td>27</td>
<td>3.33</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>73</td>
<td>13.68</td>
<td>27</td>
<td>7.88</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>73</td>
<td>19.35</td>
<td>27</td>
<td>27.10</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>73</td>
<td>19.38</td>
<td>27</td>
<td>26.77</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>72</td>
<td>12.30</td>
<td>27</td>
<td>3.31</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>72</td>
<td>12.06</td>
<td>27</td>
<td>4.55</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>13.10</td>
<td>27</td>
<td>3.73</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>72</td>
<td>10.67</td>
<td>25</td>
<td>3.07</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>72</td>
<td>11.35</td>
<td>25</td>
<td>3.08</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>11.45</td>
<td>26</td>
<td>3.80</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>73</td>
<td>12.91</td>
<td>25</td>
<td>3.11</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>73</td>
<td>11.01</td>
<td>27</td>
<td>3.76</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>13.68</td>
<td>28</td>
<td>5.23</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>26.92</td>
<td>28</td>
<td>30.14</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>26.91</td>
<td>28</td>
<td>29.46</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>15.05</td>
<td>28</td>
<td>25.62</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>72</td>
<td>11.04</td>
<td>28</td>
<td>3.50</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>72</td>
<td>9.72</td>
<td>26</td>
<td>2.97</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>72</td>
<td>9.49</td>
<td>28</td>
<td>3.83</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>72</td>
<td>12.92</td>
<td>27</td>
<td>3.68</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>10.10</td>
<td>28</td>
<td>4.49</td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>72</td>
<td>9.55</td>
<td>28</td>
<td>3.90</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>71</td>
<td>11.09</td>
<td>29</td>
<td>25.56</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>71</td>
<td>15.53</td>
<td>29</td>
<td>29.12</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>71</td>
<td>16.41</td>
<td>29</td>
<td>28.98</td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>71</td>
<td>14.47</td>
<td>29</td>
<td>29.66</td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>71</td>
<td>14.53</td>
<td>29</td>
<td>28.79</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>71</td>
<td>13.20</td>
<td>29</td>
<td>4.54</td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>70</td>
<td>14.40</td>
<td>30</td>
<td>19.49</td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>68</td>
<td>10.77</td>
<td>32</td>
<td>17.62</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>68</td>
<td>15.14</td>
<td>22</td>
<td>6.59</td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>68</td>
<td>18.01</td>
<td>21</td>
<td>9.15</td>
</tr>
</tbody>
</table>

Table 6.6: The computational benefit of the newly proposed heuristics within a branch-and-cut approach.
### Chapter 6. Empirical results

#### Results for the J60 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th>Heuristic</th>
<th>Optimal # Inst.</th>
<th>Time (s)</th>
<th>Time limit # Inst.</th>
<th>Gap (%)</th>
<th>No solution # Inst.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>48 58.40</td>
<td>49 5.02</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>48 58.50</td>
<td>50 5.04</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>46 39.24</td>
<td>54 4.09</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>46 39.56</td>
<td>54 4.33</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>46 45.58</td>
<td>54 2.44</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>46 54.00</td>
<td>53 2.31</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>46 66.94</td>
<td>53 4.09</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>45 43.45</td>
<td>55 2.11</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>45 43.78</td>
<td>55 2.11</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>45 44.46</td>
<td>55 3.53</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>45 47.56</td>
<td>54 3.94</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>44 45.88</td>
<td>56 57.13</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>44 47.74</td>
<td>56 57.16</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>44 61.57</td>
<td>56 3.22</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>44 62.49</td>
<td>55 2.08</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>43 41.23</td>
<td>57 4.54</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>43 63.04</td>
<td>54 5.80</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>43 65.56</td>
<td>57 55.25</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>43 101.24</td>
<td>53 4.17</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>42 35.80</td>
<td>58 4.77</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>42 54.89</td>
<td>56 7.34</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>42 95.04</td>
<td>55 4.91</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>41 70.55</td>
<td>59 63.55</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>40 69.57</td>
<td>59 3.73</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>39 71.05</td>
<td>60 3.57</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>39 74.22</td>
<td>54 2.53</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>39 81.17</td>
<td>56 2.60</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>39 89.12</td>
<td>61 60.20</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>38 63.36</td>
<td>62 44.06</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>38 68.45</td>
<td>62 43.16</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>38 85.97</td>
<td>62 45.94</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>38 112.77</td>
<td>62 3.24</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>38 117.90</td>
<td>62 3.39</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>38 136.63</td>
<td>62 2.14</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>37 69.08</td>
<td>63 46.36</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>37 69.92</td>
<td>53 2.77</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>36 83.40</td>
<td>52 3.07</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>36 122.00</td>
<td>64 2.10</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>34 87.93</td>
<td>58 4.88</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>33 41.40</td>
<td>62 4.83</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>31 95.95</td>
<td>27 14.43</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>31 161.29</td>
<td>47 32.24</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-GR-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>28 98.92</td>
<td>25 5.25</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>27 143.85</td>
<td>73 24.24</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>27 148.45</td>
<td>73 25.33</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSP-CVI</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>25 157.33</td>
<td>51 25.72</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.7: The computational benefit of the newly proposed heuristics within a branch-and-cut approach.
6.6. Result set IV: Solving real-world problem instances

## Results for the RG100 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th>Heuristic</th>
<th># Inst.</th>
<th>Time (s)</th>
<th># Inst.</th>
<th>Gap (%)</th>
<th># Inst.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-GF</td>
<td>0</td>
<td>—</td>
<td>50</td>
<td>19.52</td>
<td>0</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-GF</td>
<td>0</td>
<td>—</td>
<td>50</td>
<td>19.49</td>
<td>0</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>0</td>
<td>—</td>
<td>48</td>
<td>1.91</td>
<td>2</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>0</td>
<td>—</td>
<td>46</td>
<td>1.83</td>
<td>4</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH</td>
<td>0</td>
<td>—</td>
<td>29</td>
<td>0.82</td>
<td>21</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH</td>
<td>0</td>
<td>—</td>
<td>29</td>
<td>0.79</td>
<td>21</td>
</tr>
</tbody>
</table>

## Results for the RG150 data set

<table>
<thead>
<tr>
<th>Master problem</th>
<th>Separator</th>
<th>Heuristic</th>
<th># Inst.</th>
<th>Time (s)</th>
<th># Inst.</th>
<th>Gap (%)</th>
<th># Inst.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-GF</td>
<td>0</td>
<td>—</td>
<td>48</td>
<td>20.61</td>
<td>2</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-GF</td>
<td>0</td>
<td>—</td>
<td>42</td>
<td>20.00</td>
<td>8</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP-INC</td>
<td>RFRH-INC</td>
<td>0</td>
<td>—</td>
<td>20</td>
<td>5.03</td>
<td>30</td>
</tr>
<tr>
<td>RMSP</td>
<td>RFSEP</td>
<td>RFRH-INC</td>
<td>0</td>
<td>—</td>
<td>17</td>
<td>5.73</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 6.8: The computational benefit of the newly proposed heuristics within a branch-and-cut approach. Results are only provided for branch-and-cut implementations for which at least one feasible solution was computed.

In order to put the success of the branch-and-cut implementation into context, consider Table 6.3, which shows the computational results of applying the different RCSP reformulations to the problem instances of the RG100 and RG150 data sets. In contrast to the branch-and-cut approach, which could not solve any problem instance to optimality, the RMSP-AC-GR was responsible for computing optimal solutions for eleven RG100 problem instances. The benefit of employing the branch-and-cut, however, is that feasible solutions could be computed for all 50 of the RG100 problem instances. The performance of the branch-and-cut approach becomes even more significant when considering the RG150 data set, for which the best performing formulation, the RMSP-AC-GR, only managed to generate 32 feasible solutions. By applying the branch-and-cut approach feasible solutions were computed for a total of 48 out of the 50 problem instances, with an average gap of 20.61%.

It is interesting to note that the best performing RCSP reformulation, without branch-and-cut, involves the use of constraint aggregation. More specifically, the best performing reformulations for both the RG100 and RG150 problem instances were identified in Table 6.3 as RMSP-AC-GR and RMSP-AC. However, the use of constraint aggregation in the branch-and-cut approach, in form of the separation sub-problem RFSEP-AC, did not produce satisfactory results (see Tables 6.4 and 6.5).

### 6.6 Result set IV: Solving real-world problem instances

Mine planning data for a South African underground mine were used as a reference for constructing 12 real world problem instances. The name of the mining company and financial-specific information may, however, not be disclosed due to the sensitive nature of the data. Recall from Chapter 3 that an underground mine is demarcated vertically into several levels. The number of levels are typically an indication of the size of a mining operation. One of the attributes associated with a mining activity is the level to which it belongs. Therefore, problem instances
of varying sizes may be created by filtering the database of activities according to mine level identifiers. For the purpose of this study, the mine planning data at hand were partitioned into three smaller sets comprising 6, 10 and 14 levels, respectively. By doing this, real-world problem instances corresponding to a small operation with 6 levels, a medium operation with 10 levels and a large operation with 14 levels, are considered.

In addition to filtering activities based on mine level identifiers, a planning horizon criterion was applied in the final construction of the different problem instances. By making use of earliest start times of the activities (computed as part of the preprocessing procedure in Section 5.1), problem instances corresponding to specific planning horizons were created. More specifically, problem instances were created for a 2-year, 5-year, 10-year and 15-year planning horizon, respectively.

Table 6.9 provides information on the problem instances created for varying mining operation sizes and different planning horizons. The last column shows the number of activities included in each of the problem instances.

Computational results are provided in the remainder of this chapter to showcase the efficiency of the proposed problem formulations and algorithmic approaches when applied to the problem instances listed in Table 6.9. It is, however, informative to consider the practical aspects in underground mine scheduling and how they are accommodated in the mathematical models considered in this thesis.

### 6.6.1 Practical consideration related to mine scheduling optimisation

Recall the small mine scheduling example presented in Section 3.2. As indicated by the input data for this example in Table 3.1, resource consumption/production values $v_{ir}$ for each resource $r$ are associated with each activity $i$. The same approach applies to the problem instances considered in this section. The main resources of concern in this section are related to infrastructure capacities and crew requirements.

Each level of a typical underground mine is demarcated into two sections, with each section referred to as a half-level. Capacity constraints related to the volume of rock that may be trammed within a given level are specified per half-level. A typical half-level capacity is a value of between 10,000 to 20,000 tonnes per month, depending on the layout and the physical infrastructure of the mine.
6.6. Result set IV: Solving real-world problem instances

Figure 6.1: Unconstrained tonnes profile for the 12 half-levels of the L6-Y2 problem instance.

Figure 6.2: Tonnes profile for the 12 half-levels of the L6-Y2 problem instance when considering a half-level capacity constraint of 10,000 per month.

For illustrative purposes, the L6-Y2 problem instance is used and the resources that are reported on include tonnes per half-level and the number of crews required for stoping activities. In practice there may be several other resources required to capture all of the costing parameters and resource limitations. The motivation for the development of algorithmic approaches to improve computing times of RCSP problems with a large number of resources now becomes apparent, considering that for the L6-Y2 problem instance, a minimum of 12 resources are required, one for each half-level.

As a first course of action typically followed in practice, an unconstrained version of the L6-Y2 problem instance is solved. The tonnes profile for the 12 half-levels obtained by solving either the RMSP or the TMSP without any resource restrictions, is depicted in Figure 6.1. Note that the resulting solution is a schedule based on a daily calendar. Therefore, the tonnes profile in Figure 6.1 is per day.

An unconstrained scheduling solution is typically required to validate the precedence requirements of the activities, specified as sequencing rules. By considering a tonnes profile, an experienced mine planner should be able to access quickly whether erroneous input data were used in the construction of the scheduling solution. Furthermore, the starting solutions of the activities may be imported into a 3D mine planning system and the resulting graphical display of the scheduling solution may be used for further validation.

The next step following the validation of the unconstrained schedule is the systematic application
Chapter 6. Empirical results

Figure 6.3: Comparing the effect on total tonnes and crew for the L6-Y2 problem instance when considering a half-level capacity of 10000 tonnes per month.

Figure 6.4: Comparing the effect on total tonnes and crew for the L6-Y2 problem instance when considering a half-level capacity of 10000 tonnes per month and a maximum crew requirement limit of 30 crews per day.

of the remaining constraints. For instance, a half-level capacity of 10000 tonnes per month is assumed for all of the remaining empirical tests that involve the L6, L10 and L14 problem instances. The tonnes profile obtained when re-solving the L6-Y2 problem instance while taking the half-level capacity constraint into account is given in Figure 6.2. Note that since the resulting tonnes profile is displayed per day, the maximum allowable tonnes per day is given by $10000 \times \frac{12}{260} \approx 462$, if the number of production days in a year is assumed to be 260.

The tonnes profiles depicted in Figure 6.3(a) are for the unconstrained case and for the case where the half-level capacity constraint is imposed. By applying the capacity constraint, a reduction in total tonnes can clearly be observed. The effect of the constraint is also noticeable when considering the profiles of other resources. For instance, the crew requirement profile in Figure 6.3(b) clearly shows a reduction in the number of crews required over time, as a result of activities being scheduled to start at a later date in order to satisfy the capacity constraint.

As a further illustration of the effect that constraints have on resource profiles, consider the case where a limit is imposed on the number of crews available per day. The crew profile in Figure 6.3(b), for the case where the half-level capacity constraint is applied, suggests that a maximum of up to 38 crews are required to realise the corresponding tonnes profile. In the remainder of this chapter, however, a maximum limit of 30 crews per day is considered for the L6 problem instances, 40 crews per day for the L10 problem instances and 50 crews per day for the L14 problem instances.

The resulting profiles for the L6-Y2 problem instance when considering a crew limit of 30 per day are shown in Figure 6.4. The combined effect of the crew requirement and the half-level capacity constraints is clearly visible in the tonnes profile depicted in Figure 6.4(a). Interesting to note, however, is that although the volume of tonnes decreased during the early periods due to the crew requirement constraint, the tonnes profile is higher for subsequent days as a result of activities being scheduled to start later. This effect is also visible for the crew requirement profile shown in Figure 6.4(b).
6.6. Result set IV: Solving real-world problem instances

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>RMSP-AC Time(s)</th>
<th>Gap(%)</th>
<th>Branch-and-cut (RMSP, RFSEP, RFRH-GF) Time(s)</th>
<th>Gap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L6-Y2</td>
<td>3</td>
<td>0.00</td>
<td>233</td>
<td>0.00</td>
</tr>
<tr>
<td>L10-Y2</td>
<td>900</td>
<td>1.02</td>
<td>900</td>
<td>3.01</td>
</tr>
<tr>
<td>L14-Y2</td>
<td>73</td>
<td>0.00</td>
<td>900</td>
<td>1.75</td>
</tr>
<tr>
<td>L6-Y5</td>
<td>1800</td>
<td>1.31</td>
<td>1800</td>
<td>3.60</td>
</tr>
<tr>
<td>L10-Y5</td>
<td>1800</td>
<td>1.04</td>
<td>1800</td>
<td>2.84</td>
</tr>
<tr>
<td>L14-Y5</td>
<td>419</td>
<td>0.00</td>
<td>1800</td>
<td>1.79</td>
</tr>
<tr>
<td>L6-Y10</td>
<td>3600</td>
<td>1.66</td>
<td>3600</td>
<td>3.94</td>
</tr>
<tr>
<td>L10-Y10</td>
<td>—</td>
<td>—</td>
<td>3600</td>
<td>3.01</td>
</tr>
<tr>
<td>L14-Y10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L6-Y15</td>
<td>—</td>
<td>—</td>
<td>7200</td>
<td>3.82</td>
</tr>
<tr>
<td>L10-Y15</td>
<td>—</td>
<td>—</td>
<td>7200</td>
<td>2.73</td>
</tr>
<tr>
<td>L14-Y15</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 6.10: The computational efficiency for solving real-world instances while considering a half-level capacity of 10,000 tonnes per month.

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>RMSP-AC Time(s)</th>
<th>Gap(%)</th>
<th>Branch-and-cut (RMSP, RFSEP, RFRH-GF) Time(s)</th>
<th>Gap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L6-Y2</td>
<td>101</td>
<td>0.00</td>
<td>900</td>
<td>1.56</td>
</tr>
<tr>
<td>L10-Y2</td>
<td>900</td>
<td>1.63</td>
<td>900</td>
<td>5.50</td>
</tr>
<tr>
<td>L14-Y2</td>
<td>900</td>
<td>2.07</td>
<td>900</td>
<td>30.88</td>
</tr>
<tr>
<td>L6-Y5</td>
<td>1800</td>
<td>1.80</td>
<td>1800</td>
<td>10.61</td>
</tr>
<tr>
<td>L10-Y5</td>
<td>1800</td>
<td>4.90</td>
<td>1800</td>
<td>35.18</td>
</tr>
<tr>
<td>L14-Y5</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L6-Y10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L10-Y10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L14-Y10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L6-Y15</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L10-Y15</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L14-Y15</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 6.11: The computational efficiency for solving real-world instances while considering a half-level capacity of 10,000 tonnes per month and a maximum crew requirement limit of 30 crews per day.

6.6.2 Computational efficiency of solving real-world problem instances

Initial results in this section were generated by considering only the RMSP-AC problem formulation and a branch-and-cut approach that comprises the RMSP as a master problem, RFSEP-INC as the separation sub-problem and RFGF as the primal heuristic. The motivation for this selection stems from the computational results that were presented above for the RG100 and RG150 data sets. It is shown later in this section that the application of the ATMSP-based problem formulation in conjunction with a branch-and-cut approach may be applied successfully in generating feasible solutions to difficult real world problem instances.

The computational results to follow are for all of the problem instances listed in Table 6.9. A time limit of 900 seconds was imposed on solution times for Y2 problem instances, a limit of 1800 seconds on the Y5 problem instances, 3600 seconds on the Y10 problem instances and 7200 seconds on the Y15 problem instances.

The results obtained when solving all of the problem instances by taking the half-level capacity constraint into account, are provided in Table 6.10. Positive results are reported for problem instances relating to the two-year and five-year planning horizons. Optimal solutions
Chapter 6. Empirical results

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>ATMSP-AC</th>
<th>Branch-and-cut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time(s)</td>
<td>Time(s)</td>
</tr>
<tr>
<td></td>
<td>Gap(%)</td>
<td>Gap(%)</td>
</tr>
<tr>
<td>L6-Y10</td>
<td>129</td>
<td>1533</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>L10-Y10</td>
<td>3600</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>1.04</td>
<td>1.09</td>
</tr>
<tr>
<td>L14-Y10</td>
<td>3600</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>1.72</td>
<td>1.56</td>
</tr>
<tr>
<td>L6-Y15</td>
<td>3332</td>
<td>7200</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>1.12</td>
</tr>
<tr>
<td>L10-Y15</td>
<td>—</td>
<td>7200</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>2.75</td>
</tr>
<tr>
<td>L14-Y15</td>
<td>—</td>
<td>7200</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>2.17</td>
</tr>
</tbody>
</table>

Table 6.12: The computational efficiency for solving real world instances with a mixed scheduling calendar while considering a half-level capacity of 10,000 tonnes per month.

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>ATMSP-AC</th>
<th>Branch-and-cut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time(s)</td>
<td>Time(s)</td>
</tr>
<tr>
<td></td>
<td>Gap(%)</td>
<td>Gap(%)</td>
</tr>
<tr>
<td>L6-Y10</td>
<td>3600</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>68.00</td>
<td>79.7</td>
</tr>
<tr>
<td>L10-Y10</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L14-Y10</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L6-Y15</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L10-Y15</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>L14-Y15</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 6.13: The computational efficiency for solving real world instances with a mixed scheduling calendar while considering a half-level capacity of 10,000 tonnes per month and a maximum crew requirement limit of 30 crews per day.

were computed for L6-Y2, L14-Y2 and L14-Y5 by employing the RMSP-AC formulation. The corresponding solution times are 3 seconds, 73 seconds and 419 seconds. The L6-Y2 problem instance is the only instance for which the branch-and-cut approach was able to compute an optimal solution. Very low integrality gaps were measured for all the other instances not solved to optimality, but for which at least one feasible solution could be computed. The scalability of the branch-and-cut approach is demonstrated through its ability to compute feasible solutions for some of the Y10 and Y15 problem instances.

Table 6.11 shows the computational results obtained when solving all of the problem instances by taking both the half-level capacity constraint and the crew requirement constraint into account. Feasible solutions could only be computed for all of the Y2 problem instances and two of the Y5 problem instances. No solution could be computed for the Y10 and Y15 problem instances within the allowed time limit.

The inability of the RMSP-based formulation and branch-and-cut approach to generate feasible solutions to some of the problem instances, as indicated by Tables 6.10 and 6.11, does not signify a dead end. An approximation approach is suggested which involves the time indexed formulation ATMSP. Although it may seem counterintuitive to use an ATMSP-based approach due to its poor performance as reported above, the ability to make use of a mixed scheduling calendar in the formulation of ATMSP allows for the generation of approximate solutions. More specifically, recall from Section 4.2 that the start time of a period $S^T_t$ and the end time of a period $S^T_{t+1}$, are dictated by the choice of a scheduling calendar. For instance, if a daily calendar is considered, then $S^T_{t+1} - S^T_t = 1$, while if a weekly calendar is considered, then $S^T_{t+1} - S^T_t = 5$, etc. This allows for the creation of a mixed scheduling calendar comprising, for instance, monthly periods followed by quarterly periods and finally, annual periods. The argument for such a calendar configuration is that more detailed scheduling information is required during the earlier periods, while the aggregation of information during later periods is acceptable.
The effect on the formulation of the ATMSP is that fewer time-indexed variables are required to represent the underground mine scheduling problem. An added benefit of employing the ATMSP-based formulation is that resources which are not subjected to transfer delays may be treated as ordinary resources within the upper bound constraints (5.29). The result is that less flow related variables are required in the problem formulation. The resources of the problem instances considered here, which are related to the half-level capacity constraints, falls within this category.

Table 6.12 shows the results obtained for solving the problem instances using the ATMSP-AC formulation and the branch-and-cut approach which includes ATMSP as the master problem, RFSEP as the separator and RFRH-GF as the primal heuristic. Use of the ATMSP-based formulation shows an improvement over the RMSP-based formulation, since solutions could be computed for problem instances that were previously unsolvable. For instance, the ATMSP-AC and the associated branch-and-cut approach managed to compute feasible solutions for the L14-Y10 and L14-Y15 problem instances within the allowed time limit and with relatively small integrality gaps. Furthermore, the ATMSP-based formulation was also successful in computing feasible solutions for the L6-Y10 problem instance while considering both half-level and crew requirement constraints, as indicated by Table 6.13. Unfortunately, no improvements were possible for any of the other problem instances in Table 6.13.

As mentioned above, the use of ATMSP-based formulations in conjunction with mixed scheduling calendars may only provide approximate solutions. Although relatively small integrality gaps were reported in Tables 6.12 and 6.13, the corresponding solutions may be far from optimal. In order to quantify how far from optimal the solutions are, a comparison of relative objective function values is performed, since some solutions were obtained for the same problem instances by both the RMSP and ATMSP-based formulations.

Table 6.14 provides a comparison of relative objective function values showing the effect of employing an ATMSP-based formulation with a mixed scheduling calendar. Since all objective function values of the problem instances solved according to the RMSP-based formulation were greater than those of the ATMSP-based formulations, the entries in the first column of Table 6.14 are all at 100%. It is clear from the second column in the table that the approximate solutions obtained from ATMSP-based formulations have objective function values that are slightly more than half of what the objective function values are for the RMSP-based formulations. Although this may appear to be discouraging, it should be remembered that the ATMSP-based formulations with a mixed scheduling calendar may at least compute feasible solutions to previously unsolvable problem instances. This suggests a possible new approach in which the feasible solutions from an ATMSP-based formulation may be used as good starting solutions for the RMSP-based formulation.

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>RMSP-based</th>
<th>ATMSP-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>L6-Y10</td>
<td>100</td>
<td>57</td>
</tr>
<tr>
<td>L10-Y10</td>
<td>100</td>
<td>53</td>
</tr>
<tr>
<td>L6-Y15</td>
<td>100</td>
<td>55</td>
</tr>
<tr>
<td>L10-Y15</td>
<td>100</td>
<td>47</td>
</tr>
</tbody>
</table>

Table 6.14: Relative objective function values of the ATMSP-based formulation compared to the RMSP-based formulation.
6.7 Summary

The computational results presented in this chapter have demonstrated the success with which the proposed algorithmic enhancements have improved computing times for both randomly generated and real-world problem instances. The first set of results showed that by excluding redundant valid inequalities from the problem formulations that were used several times in other studies, significant improvements in computing times are possible.

Extensive tests were performed on both the PSPLIB benchmark instances and the problem instances that were generated by using Rangen2, in order to compare the newly proposed reformulations and Benders decomposition approach. The top performing reformulation was identified to be the aggregated constraint formulation RMSP-AC. In some test cases benefit was realised by considering graph reduction as well, that is, applying the formulation RMSP-AC-GR. Although the ATMSP-based formulations did show promising results for the J30 and J60 data sets, the RMSP-based formulations were responsible for solving most of the problem instances in the larger RG100 and RG150 data sets.

Less favourable results were obtained for the branch-and-cut implementation which only incorporated the separation sub-problems RFSEP, RFSEP-AC and RFSEP-INCl. Inclusion of primal heuristics, however, resulted in significant improvements in computing times. In fact, the application of the RFRG-GF heuristic within a branch-and-cut approach meant that solutions could be computed for problem instances for which no solutions could otherwise be obtained.

The final set of results is based on problem instances that were created for real mine planning data. Twelve problem instances were created by considering different combinations of the number of mining levels and planning horizons. Feasible solutions could be computed with RMSP-based formulations for the instances with 6 to 10 mine levels that corresponded to a two-year or five-year planning horizon. Less favourable results were obtained for the larger instances for which an additional crew requirement constraint was imposed. As a last resort, an approximation approach which uses an ATMSP-based formulation with a mixed scheduling calendar was considered. Although the quality of the solutions obtained by this approach may be considered to be poor, it nevertheless demonstrated that it is possible to compute feasible solutions to some of the harder problem instances.
CHAPTER 7

Summary and conclusion

Contents

7.1 Chapter summaries ................................................................. 81
7.2 Research outputs ................................................................. 83
7.3 Future work ............................................................................ 84

Global economic downturn in recent years and the increasing competitiveness of corporations as a result of globalisation are only some of the challenges faced by mining companies. In order to remain profitable, proper planning and sound decision making are imperative for managing the complexity of mining operations.

A key aspect in the production planning life cycle at an underground mine is the use of highly sophisticated mine planning tools to assist with decision making. The latest 3D CAD planning systems encapsulate both mine layout design functionality as well as computer algorithms for generating production schedules. Although significant progress has been made over the last couple of decades in developing improved mine planning systems, the ability of these systems in terms of catering for all of the mining complexities is still very limited. Furthermore, as computing abilities improve, the demand for more functionality in these systems increases.

The use of mathematical models and algorithms play an important role in the development of mine planning systems. More specifically, it was shown in this thesis that the underground mine scheduling problem is a special case of the well-known resource constrained scheduling problem. The approach followed in this study has therefore been to obtain an in-depth understanding of the theory underlying the resource constrained scheduling problem, and to explore algorithmic approaches that show potential in terms of improving computing efficiency when solving an underground mine scheduling problem.

A summary of the content of this thesis is provided in the first section of this thesis. The findings of this study and explicit reference to contributions made in the thesis are summarised in the following section. The chapter closes with a brief overview of possible future follow-up work and a discussion on potentially open research questions.

7.1 Chapter summaries

The introductory chapter of this thesis provided context for the motivation to embark on this study. The use of appropriate mathematical models and algorithms has become a necessity for mining companies to ensure that short and long-term production plans culminate in sustainable
Chapter 7. Summary and conclusion

The primary goal of this study was to improve on existing mathematical models in order to address as much of the typical underground mine planning requirements as possible. Furthermore, since the solution of these models is inherently hard, the search for algorithmic enhancements was identified as the secondary objective of this study.

An overview of the theoretical foundations of scheduling algorithms was provided in Chapter 2. The emphasis in this chapter was on the formulation of resource constrained scheduling problems and algorithmic approaches for providing solutions to these problems. An introduction to complexity theory provided a framework for discussing the hardness of problem instances and ways of measuring the efficiency of algorithms. A literature review was also presented of the most recent contributions in the field of resource constrained scheduling. The scope of the literature study was, however, limited to work within an exact solution framework.

Chapter 3 contained an overview of the technical aspects pertaining to underground mining operations and the process of mine planning and scheduling. This chapter provided a glimpse into the complexities involved in planning the execution of mining activities while taking infrastructure and resource limitations into account. An important contribution of this chapter has been the use of a small example to demonstrate that the underground mine scheduling optimisation problem is a special case of the well-known resource constrained scheduling problem. Furthermore, it was also shown that the standard framework for handling resource constrained scheduling problems does not necessarily accommodate all of the operational requirements encountered in underground mining. It is for this reason that an exact modelling framework approach was adopted in this thesis, in order to address some of these requirements.

A number of newly proposed mathematical models were presented in Chapter 4. The first model presented is an existing resource flow model, adapted to incorporate an objective function which maximises net present value. The second model, which is a time-indexed model, was augmented with flow-related variables and constraints. The reason for this, and for considering resource flow models in general, is to address the operational requirement in mining where the movement of crews and equipment is subjected to transfer delays. An added advantage of using a resource flow-based formulation is the ability to perform crew tracking by reporting on the movement of crews as part of the scheduling solution. The chapter closed with a small example demonstrating the implementation of crew requirement constraints and the effect of considering transfer delays on the scheduling solution.

The heart of this thesis is captured in the contents of Chapter 5. Various reformulations were proposed that have the potential to improve computing times associated with the underground mine scheduling problem. As a first suggestion, a constraint aggregation approach was formulated which resulted in a reduced number of resource flow-related constraints. A graph reduction approach was also suggested to reduce the number of variables in both the resource flow-based and time-indexed formulations, by taking logical movement of resources into account with respect to the underlying precedence graph. The distinctive feature of a typical underground mine scheduling problem instance of having many resources was the primary motivation for considering a decomposition approach. Details of applying a Benders decomposition of the underground mine scheduling problem were provided, and two theorems were established in support of the derivation of three separation sub-problems. The implementation of these separation routines as part of a branch-and-cut approach was described and three heuristic approaches were suggested for the purpose of generating primal solutions. The chapter closed with an innovative suggestion of applying valid inequalities that resemble the well-known cutset inequalities found in solution approaches for solving network design problems.

The penultimate chapter of this thesis showcased the success of the proposed model reformulations and algorithmic enhancements. An extensive report was provided detailing computational
results on both randomly generated data as well as data emanating from a real mine planning problem. The main conclusion drawn from the empirical work performed is that most of the proposed enhancements improve computing times when compared to the application of existing formulations found in the literature. The top performing model reformulation involves the use of the resource flow-based model in conjunction with a constraint aggregation approach. The Benders decomposition approach, implemented within a branch-and-cut framework, has shown to scale well for problem instances with a large number of activities and resources. This is a significant contribution within the context of mining, especially considering the large number of resources that need to be accommodated in solving underground mine scheduling optimisation problems.

### 7.2 Research outputs

Tangible outputs in the form of three journal publications are anticipated from this thesis. The following is an overview of the expected publications and the content in the thesis from which it emanates.

- **Resource constrained scheduling with future cash-flows and transfer delays.** The primary contribution of this first publication will be the introduction of the RMSP and TMS model formulations as alternatives for solving resource constrained scheduling problems with future cash-flows and transfer delays. The models will be presented under different names in order to fit into a general resource constraint scheduling context. This will be the first publication to introduce a linearisation approach for approximating the NPV of resource flow-based formulations. An additional contribution will be to demonstrate the computational efficiency when applying the constraint aggregation and graph reduction approaches for problem instances with a large number of resources. The content of this paper will be extracted from Chapters 4 and 5, and will be submitted for publication in Annals of Operations Research.

- **A Benders decomposition approach for solving the resource constrained scheduling problem.** The main contribution of the second publication will be the model formulations derived for the master and sub-problem by applying Benders decomposition. Details of the sub-problem implementation based on the derivation of the Benders feasibility cuts will be provided, as well as an overview of the proposed heuristics for generating primal solutions. A thorough empirical study will be presented to showcase the computational efficiency of the various sub-problem and heuristic variants. The content of this paper will be based on the material of Chapters 5 and 6, and will be submitted for publication in Computers and Operational Research.

- **The application of resource constrained scheduling models and algorithms for solving the underground mine scheduling problem.** There is a need in the mining community to achieve a better understanding of the models and algorithms applicable to underground mine planning problems. The objective of this paper would be to make use of simplified examples in order to illustrate how the models proposed in this thesis would address the requirements of typical underground mine scheduling problems. Specific attention will be afforded to crew requirement constraints and the use of transfer delays in the case where mechanised mining is of concern. The real-world problem instances from Chapter 6 will be used to demonstrate the practicality of using the proposed solution methodology within the context of short and long-term planning. The journal of choice to which this paper would be submitted is the Journal of the South African Institute for Minerals and Mining.
7.3 Future work

Computational results obtained for the branch-and-cut approach demonstrated the success of applying a decomposition approach. The inability to compute feasible solutions for some of the real-world instances is, however, enough reason to further efforts in developing more efficient approaches. The establishment of the branch-and-cut framework for solving the resource constraint scheduling problem also opened up the opportunity to address other modelling requirements, which were previously unattainable due to the inefficiency of existing solution approaches. Details of possible further algorithmic work and modelling extensions are provided below.

1. **Separation of cutset inequalities.** The application of cutset inequalities was proposed in order to improve computing times. These inequalities were enumerated up front as part of preprocessing and added explicitly to the model formulation. The empirical results provided in this thesis showed benefit when applying cutset inequalities in the solution of small to medium sized problem instances. The benefits were, however, negated by the large number of constraints that had to be added explicitly when considering larger problem instances. It is proposed that cutset inequalities be added implicitly through a separation mechanism [1]. This will be similar to the separation implementation of the Benders feasibility cuts.

2. **Precedence-related valid inequalities.** The order in which activities may be scheduled is determined by the precedence relationships which are captured in the precedence graph. A possible extension to the work in this thesis is the exploration of information that is locked up in the precedence graph in order to allow for the derivation of valid inequalities. Once again these inequalities may be added explicitly as part of the problem formulation or be generated during the branch-and-bound process by means of a separation routine. The work by Hardin et al. [32] is suggested as a basis for further work on the topic.

3. **Column generation for the resource flow variables.** The Benders decomposition approach proposed in this thesis involves a separation sub-problem that contains variables related to the flow of resources. The number of resource flow variables grows exponentially with an increase in the number of activities. The inability to compute solutions to large-scale problems may, therefore, be attributed to insufficient memory as a result of the enormous growth in variables. The use of column generation approaches within a branch-and-cut framework have been applied successfully in network design applications for improving memory usage [63]. It should, therefore, be possible to incorporate a column generation approach within the branch-and-cut framework suggested in this thesis, for generating the required resource flow-related variables during the branch-and-bound process.

4. **Constraint programming as primal heuristic.** The application of constraint programming has demonstrated to be very efficient in generating feasible solutions, and in several cases optimal solutions, to the resource constraint scheduling problem [50]. It may, therefore, be reasonable to expect that the application of constraint programming within a branch-and-cut approach may improve computing times when applied as a primal heuristic.

5. **Resource constraint scheduling with uncertain resource requirements.** The models considered in this thesis are based on the assumption that the resource consumption/production values, provided as part of the input data, are deterministic. This may be an over-simplifying assumption considering the uncertainty in real-world applications.
This is especially true for underground mine scheduling where many parameters are, in reality, stochastic and should be treated as such in corresponding mathematical models. For this purpose, a stochastic programming framework is suggested to cater for stochastic resources such as mineral content. The branch-and-cut solution approach established in this thesis lends itself naturally to a stochastic programming approach due to the Benders decomposition of the problem. Extending the branch-and-cut implementation only requires changes to the separation sub-problem.

6. **A robust approach to resource constraint scheduling.** The application of a robust formulation of the resource constraint scheduling problem is another way to deal with uncertainty in resource requirements. Compared to stochastic programming where uncertainty is treated by means of stochastic scenarios, incorporating uncertainty within a robust framework involves the use of uncertainty sets. Once again, extending the branch-and-cut implementation for a robust approach will only require changes to the separation sub-problem.

Various outputs in the form of journal publications are possible from the potential extensions outlined above. A possible first publication would be on the enhancements of the branch-and-cut implementation in which the separation of the new valid inequalities and the column generation approach (for implicitly generating the resource flow-related variables during the branch-and-bound process) are considered. For a second publication, the emphasis may be on the practical considerations of uncertainty in resource constrained scheduling and how a stochastic programming approach, as well as a robust optimisation approach, could be incorporated into the existing branch-and-cut framework.
References


REFERENCES


 REFERENCES


