Approaches to Supply Chain Coordination: Decomposed and Decentralised Decision Making Models

submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

of the
Indian Institute of Technology Bombay, India
and
Monash University, Australia
by

Anu Thomas

Supervisors:

Prof. Mohan Krishnamoorthy
Department of Mechanical and Aerospace Engineering, Monash University

Prof. Jayendran Venkateswaran
Industrial Engineering and Operations Research, IIT Bombay

Dr. Gaurav Singh
CSIRO Digital Productivity Flagship, Australia

The course of study for this award was developed jointly by the Indian Institute of Technology Bombay, India and Monash University, Australia and given academic recognition by each of them. The programme was administered by The IITB-Monash Research Academy.

2014

"Dream is not that which you see while sleeping
it is something that does not let you sleep."

— `Dr. A.P.J. Abdul Kalam, Wings of Fire: An Autobiography`

Dedicated to

my parents, who inspired me to dream high
my family, who persuaded me to achieve the dream
my teachers, who guided me to conquer the dream
my friends, who were with me to transpire the dream.

Thesis Approval

The thesis entitled

Approaches to Supply Chain Coordination: Decomposed and Decentralised Decision Making Models

by

Anu Thomas

(IIT Bombay Roll Number: 09409401,

[redacted]

is approved for the degree of

Doctor of Philosophy

Examiners [redacted]

1. Prof. M. Mathirajan (IISc Bangalore)
2. Prof. Narayan Rangaraj (IIT Bombay)

Supervisors [redacted]

1. Prof. Mohan Krishnamoorthy
2. Prof. Jayendran Venkateswaran [redacted]
3. Dr. Gaurav Singh [redacted]

Chairman [redacted]

Prof. N. S. Punekar (IIT Bombay)

Date: 17-10-2014 Place: Powai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/-source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute/the Academy and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Notice 1

Under the Copyright Act 1968, this thesis must be used only under normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgment should be made for any assistance obtained from this thesis.

Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owners permission.

[redacted]

Place: Powai

Date: 17-10-2014

Anu Thomas

Abstract

Modern supply chains have shifted from hierarchical, one-dimensional supply chains and increasingly operate in a global supply network. Supply Chain Coordination (SCC) focuses on optimising operations such as supply, manufacturing and distribution in the supply network of an enterprise. The major concern in such networks is to achieve coordination without compromising the autonomy of individual units or partner organisations. This area provides ample opportunities for research.

Different modelling approaches based on operational and decision-making aspects of the supply chain can be used to address SCC issues. An *integrated model* (IM) is one which combines the constraints and objectives of different decision making units (DMUs) into a single, huge optimisation model. Integration allows the supply chain to solve all its sub-components (DMUs) simultaneously and hence, it can guarantee feasibility across DMUs compared to solving the DMUs separately. However, such models, being large and complex, are computationally difficult to solve. The solution, if any, need not be optimal for all DMUs. Often, the independent players in the system are reluctant to share all their competitive information in public. If the players are not willing to share complete information, then an attempt at an IM is extremely difficult. Therefore, the development of alternative approaches for SCC becomes increasingly necessary.

In this thesis, new approaches for SCC are proposed using decomposed and decentralised decision making models based on Lagrangian relaxation (LR) and column generation (CG) methods. This is motivated by a real-world coal mining example, which is generalised to a multi-resource constrained scheduling problem. The transition of solution approaches from integrated approaches to decomposed approaches and then to decentralised ones is presented. The industry seeks a coordination approach that can deliver quality solutions in a reasonable amount of time without compromising their autonomy and their confidential and competent information. Therefore, decentralised decision-making will be the driving force in the future for supply chain coordination.

A two-party coordinated production-planning and resource-scheduling problem involving a set of independent producers (multiple mines) and a shared resource manager (rail operator) is considered. The decisions in this SC are decomposed by relaxing the resource

sharing constraints which link the DMUs. Decomposition approaches based on LR and CG are then developed. Several strengthening methods and stabilisation techniques have been implemented to improve the LR and the CG algorithms. The decomposition approaches are compared with the integrated approach to benchmark the performance of distributed decision making.

Decentralised approaches are developed by further reducing the information-sharing and eliminating the central coordinator in the decomposed approaches. The role of information-sharing in a decentralised approach and how to quantify the usefulness of an information, are also addressed in this thesis. For the two-party case of the coal supply chain, two vital pieces of information are identified—the production capacity (at mines) and the resource capacity (at rail operator). The decentralised scheme proposed, based on LR, guarantees convergence and is found to outperform IM in terms of obtaining solutions in reasonable time. The *value of information* is also quantified using experimental results.

The study is further expanded to a three-party case by including the second resource manager (the terminal in the coal supply chain). The three-party decentralised model was solved using a modified CG algorithm. A decentralised method called *secure-sum* has been implemented in this algorithm to compute the lower bound in all iterations. This algorithm is also strengthened with column management and other techniques. The computational results highlight the impact of an additional player and the value of information-sharing. The results show that the decentralised model could achieve better or equivalent solutions compared to that from the integrated model with significantly less information and interactions.

In summary, the thesis proposes a scalable and robust, decentralised framework of decision-making for a generic multi-party supply chain, which is a better alternative to the integrated approach. It requires only minimal information sharing between the players and guarantees convergence by means of the underlying decomposition algorithm. The approach can be used even as the level of coordination (information-sharing) improves. The proof of the concept has been demonstrated using a large and complex multi-party coal supply chain.

Publications from the thesis work

1. Thomas, A., Singh, G., Krishnamoorthy, M., & Venkateswaran, J. (2013). Distributed optimisation method for multi-resource constrained scheduling in coal supply chains. *International Journal of Production Research*, 51, 2740-2759. doi: 10.1080/00207543.2012.737955.

2. Thomas, A., Venkateswaran, J., Singh, G., & Krishnamoorthy, M. (2014). A resource constrained scheduling problem with multiple independent producers and a single linking constraint: A coal supply chain example. *European Journal of Operational Research*, 236 (3), 946-956. doi: 10.1016/j.ejor.2013.10.006.

3. Thomas, A., Krishnamoorthy, M., Singh, G., & Venkateswaran, J. (2014). Coordination and the value of information in a decentralised producer-distributor supply chain. (`Submitted to the International Journal of Production Economics`).

4. Thomas, A., Krishnamoorthy, M., Venkateswaran, J., & Singh, G. (2014). Decentralised decision-making in a distributed computing environment. (`Submitted to Computers and Industrial Engineering`).

5. Thomas, A., Krishnamoorthy, M., Venkateswaran, J., & Singh, G. (2014). Decentralised decision-making in a multi-party supply chain. (`Submitted to the International Journal of Production Research`).

6. Thomas, A., Singh, G., Krishnamoorthy, M., & Venkateswaran, J. (2013). Decentralised supply chain coordination: Role of information and cloud computing, *17th Annual International Conference of SOM*, IIT Madras, Dec 21-23, 2013.

7. Thomas, A., Krishnamoorthy, M., Singh, G., & Venkateswaran, J. (2011). Collaborative services provision in production-logistics-scheduling: A coal industry example. In *2nd International Conference on Services in Emerging Markets (ICSEM-2011), IIT Bombay, India.* Service Research and Innovation Institute (SRII).

8. Thomas, A., Singh, G., Krishnamoorthy, M., & Venkateswaran, J. (2011). Collaborative resource constrained scheduling: A coal industry example, *19th Triennial Conference of the International Federation of Operational Research Societies (IFORS-2011)*, July 10-15, 2011 Melbourne, Australia.

9. Presented posters in the annual symposium of IITB-Monash Research Academy, Mumbai in 2011 (Best poster), 2012, and 2013.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

Commonly used abbreviations

CG	Column generation
CI	Confidence interval
LB	Lower bound
LP	Linear programming
LR	Lagrangian relaxation
MILP	Mixed integer linear programming
RCSP	Resource constrained scheduling problem
SC	Supply chain
SCM	Supply chain management
UB	Upper bound

Abbreviations defined in this thesis

CDM	Centralised decision-making model
CG	Iterative algorithm developed based on column generation for two-party case
CG-3	Iterative algorithm developed based on column generation for three-party case
DDM	Decentralised decision-making model
DM	Decentralised model
DMU	Decision making unit
DSCC	Decentralised supply chain coordination
IM	Two-party integrated model
IM-3	Three-party integrated model
LR	Iterative algorithm developed based on the Lagrangian relaxation
LBR	Lower bound ratio
MOM	Multi-operator model
MP	Master problem
MS	Multiple solutions method
NI	Number of iterations
RG	Relative gap
RMP	Relaxed master problem

SCC	Supply chain coordination
SOM	Single operator model
SS	Single solution method
UBM	Upper bound method
UBR	Upper bound ratio

Chapter 1

Introduction

Supply Chain Coordination (SCC) is a challenging and rapidly developing area in Supply Chain Management (SCM) due to the increasing pressure on businesses to remain competitive in the global marketplace. SCC focuses on optimising operations such as supply, manufacturing and distribution in the supply network of an enterprise. The major concern in such supply networks is achieving coordination without compromising the autonomy of individual units or partner organisations. Maximising profit, reducing costs and inventories, asset utilisation and responsiveness are other key issues of concern [65]. For more than 20 years the area has provided, and continues to do so, ample opportunities for research [97, 144]. Early SCC approaches focused on optimising operations of chemical, pharmaceutical and petroleum manufacturing facilities. However, the application of SCC methods has now been extended to many other areas such as consumer products and the services sector as well.

In a global modern enterprise, companies are focusing more on their core business, and therefore partnerships in supply chain (SC) are becoming increasingly important to the success of large supply chains. These partnerships are across several tiers, depending on how vital and crucial these are to the success of a SC. Moreover, supply chains are becoming global as raw materials are procured and parts are produced where they are best and cheapest [4]. As a result, supply chains are becoming much more distributed and supply chain partners are becoming more autonomous. The result is that a modern supply chain is global—one in which manufacturing operations are often carried out in a region that is different from the one in which sub-assemblies or full-assemblies are carried out; it is also a chain in which logistics play an important role [115]. Therefore, modern supply chains have shifted from a hierarchical, one-dimensional supply chain and increasingly operate in a global supply network [21, 40, 4]. Further, the partners may have important relationships in many parallel chains at the same time (see Singh et al. [124]).

Market forces have, therefore, dictated that well-integrated but distributed processes deliver better results in terms of higher profits than fully-controlled and owned operations [64, 83]. However, the challenge is that, while decentralisation of the process and ownership may lead to a higher profit, it is not necessary that the management of such operations will become any easier or even straightforward. At the same time, a sequential structure of processes within the control of a single supply chain has given way to a network structure with many players/actors that offer multiple products or services [88]. Thus, a manufacturer may contract multiple raw material suppliers in order to maintain a steady production; and at the same time, the raw material supplier may have supply contracts with multiple manufacturers in order to maintain their material flow. As a result, the need for *integration* and *coordination* has increased [81, 124]. A *coordination mechanism* enables and increases coordination amongst supply chain partners who may be working, in parallel, in several different—and perhaps even conflicting—supply chains [7]. Bahinipati et al. [13] describe supply chain coordination as any situation where upstream and downstream firms are engaged in some form of long-term cooperation or agreement.

Until the 1980'-s, researchers focused on optimising individual decision-making units (DMUs), independently. For example, in a manufacturing enterprise, procurement, production, and distribution systems are considered as DMUs with their decisions taken independently. Later, research moved on to studying how the different units could be integrated or coordinated in a decision-making sense. Many articles were published on different coordination schemes such as production-distribution, production-inventory, production-procurement and distribution-inventory [7, 15, 54, 58, 64]. Sometimes the DMUs are managed by different players and each player may have sub decision-making units. In this thesis, we are interested in developing coordination schemes for such independent players in a supply chain.

Different modelling approaches, based on the problem structure and interdependencies between the DMUs, can be used to address supply chain coordination problems. An *integrated model* is one which combines the constraints and objectives of different DMUs into a single, big optimisation system. Integration allows the enterprise to solve all its DMUs simultaneously and hence, it can guarantee feasibility across DMUs compared to solving them separately. However, such solutions need not be optimal for all DMUs in the supply chain. Often, the independent players in the system are reluctant to cooperate by sharing all their competitive information completely in public. Hence, an attempt at an integrated model is extremely difficult, especially if the players are not willing to share all of the information. Even if it is possible, the integration makes the model complex and large. Therefore, the integrated model is not viable in solving large problem instances within a reasonable amount of computational time. Using decomposition techniques, each DMU's decisions can be distributed and linked with common shared resources. In Mabert and Venkataramanan's [97] words, *"the inclusion of dependency linkages in supply chain decision-making, represents the cornerstone of an effective management for firms in the* 21^{st} *century, and the process of linking decisions across the stages in a coordinated manner is by no means an easy task; it is one of the greatest challenges faced by managers today and of the future"*. Therefore, the development of alternative approaches for SCC became increasingly necessary.

In a *decentralised model*, DMUs operate separately and decisions—that potentially affect other units—are shared between them. A decentralised model solves the overall problem by distributing and coordinating the decisions of each of the DMUs. The quality of the coordination depends directly on the information that is shared between the units (or the 'players' in the system) as information sharing is one of the key factors in designing a successful coordination system [40, 34]. The methods to capture the information and provide feedback to the various players, and the medium through which the information is shared are also becoming increasingly pertinent to the successful operation of these decentralised supply chains. Therefore, delays in information availability (and processing) and the information distortion need to be significantly reduced. However, the new modes of operation come with an overload of information security and privacy [155] which needs to be taken into account when developing new frameworks.

As a result of the advancements in technology and virtualisation, there is a shift in SCMs towards using decentralised models [105, 46] instead of the traditional centralised model for decision-making. Some of these decentralised models may even be easily solvable using existing technology. In a decentralised supply chain, coordination allows the partners to negotiate and work together (collaboratively) for the overall benefit of the supply chain, and this leads partners to arrive at some consensuses.

In this thesis, we propose new approaches for SCC using decomposed and decentralised decision-making models, based on the Lagrangian relaxation and column generation methods. These models can be employed appropriately based on information availability and the levels of interaction between the DMUs. Through detailed computational experiments, we show that the proposed models outperform the integrated model in almost all cases with significantly less information sharing and with no centralised controls. This implies that the decentralised approaches are advantageous and appropriate to solve large and intricate industrial problems.

1.1 Overall research problem and motivation

Our research is motivated by a scheduling problem from Australian coal supply chains. Australia is one of the leading coal exporters in the world. According to ABARES [1], Australia produced 446.17 and 471.09 million tonnes of coal in the year 2008-09 and 2009-10, respectively. Almost 60% of the coal produced in Australia is exported to other countries. In 2011, total world coal production increased by 6.6% compared to the previous year and reached at 7678 million tonnes. The average annual growth rate of world coal production since 1999 was 4.4%*.

A coal supply chain consists of many independent DMUs such as mines, ports, rail operators, terminal and track owners. In Australia, the old mining areas were close to ports, and transport facilities were well established. However, a surge in the demand for coal spurred its discovery and production in remote locations. As the demand increased, the transportation network also improved to include dedicated train tracks. Today, due to the increased volumes, coal is mainly transported through rail (a small percentage is transported by road too). Coal transport trains are among the longest in the world. Many of them have as many as six locomotives and 148 wagons, amounting to a total length of more than two kilometres and carrying about 8500 tonnes of coal†. However, small-sized trains (approx. 2800 tonnes capacity) are also used for everyday trips to meet the periodic demand of domestic customers such as power plants and heavy industries‡.

These train movements are facilitated by high-speed loading and unloading facilities and large storage capacities at the mines and ports (*terminals*). The intermediate coal storage is called as *stockpile*. The size of a storage field varies from a few thousand tonnes to many megatonnes. Around 10 train trips are required to make a stockpile. Generally, two or three such stockpiles are required to complete one shipload. Typically the trains are operated in a 'freight' dedicated track network. This network is single line for most part, with a few crossing loops for freight trains to overtake or pass. Therefore, train scheduling in such a network, on its own right, is a challenging problem. Several giant machines such as stackers and reclaimers are required to load or unload the coal at the mines and terminals. The unavailability of these machines, when it occurs, unnecessarily delays and creates inefficiencies. Many other scheduling complexities such as maintenance schedules of equipment, terminal stockpile capacities and other similar constraints are also present in the supply chain. Due to the high cost of infrastructure and investments required for capital equipment, the rail network and the storage capacity cannot be increased at will. Instead, existing facilities need to be utilised optimally. Long-term contracts also exist between various players in the supply chain, to ensure that everything is fair. However, short-term operational scheduling problems are also regularly required to be solved to ensure that long-term contracts are satisfied.

*http://www.worldcoal.org/resources/coal-statistics/ visited on 18-Jan-2014

†http://www.australiancoal.com.au/facts-a-figures.html visited on 18-Jan-2014

‡http://www.pacificnational.com.au/services/coal.asp visited on 18-Jan-2014

Formally the supply chain process can be described as follows. The coal shipping terminal receives orders from ships with an expected arrival date. A ship's order is made up of many parcels of coal. So, the terminal splits these ship orders and passes them to the mines—along with suitable due dates—in these smaller quantities (parcels). Every mine incurs an inventory cost at the mines, a stocking cost if the coal reaches the terminal before the due date, and a demurrage cost for any late deliveries which necessitate the ship's late departure. The individual mines typically plan for and request trains of a particular class at appropriate times (so as to minimise their inventory holding and other costs). A single rail operator acts as a common resource manager that links the mines to the terminal. The rail operator provides the trains to transport the coal from the various mines to the terminal. The terminal also acts as a common resource and links all the mines. A mine is generally not concerned about the specific train that is allocated to it. The rail operator does not bother about the orders received at the terminal and at the mines. Similarly, the only decision that is important to the terminal is on-time procurement of coal from the mines, and its shipping. Nevertheless, the decisions of the mines, the rail operator and the terminal are, as is quite obvious, interlinked.

It is possible to formulate and solve the integrated whole-of-supply chain problem as a single decision-making problem that includes all the sub-units of the supply chain. Most scheduling problems are *NP-hard* (see [93, 29]). Therefore, an integrated scheduling model, which includes all DMUs, would be intractable. In general, the non-availability of shared resources like trains, creates major bottlenecks in coal production-distribution logistics. Efficient use of these shared resources benefits all players. The problem considered in this thesis is a large and complex multi-resource constrained scheduling problem. Resource constrained scheduling problems (RCSP) are a special class of problems in which the players are required to share scarce resources such as trains, tracks and terminal facilities. Unlike the traditional RCSP, in our problem, the resource manager has more than one way of meeting the requirements.

1.1.1 Problem statement

As we have seen from the discussion in previous sections, integrated approaches are not always viable for large and complex supply chain coordination problems. Motivated by the multi-resource constrained scheduling problems, we study alternative approaches for SCC using decomposed and decentralised decision-making models. The important features of SCC such as information-sharing and negotiation are also analysed. The aim of this research is (*i*) to provide a better understanding of decomposed and decentralised decision-making in supply chains and (*ii*) to develop an implementable decentralised decision-making approach for a multi-party supply chain.

1.2 Challenges

In Section 1.1, we described a coal industry example and interlinks in its supply network. It is often difficult to capture a sophisticated real life industrial problem as-is into an optimisation framework. Sometimes we may make some assumptions or relax a few constraints to obtain a simpler problem. As we have seen, there are many internal and external factors and dynamics influencing the supply chain. The real challenge is in understanding *coordination* in such situations and converting them into suitable mathematical programming problems.

A competitive enterprise requires agility and flexibility in order to cope with rapid changes in both internal and external environments. To adapt to these environments, an enterprise may integrate all decision-making units into a single system or implement an efficient mechanism to share the decisions and information between the models in coordinated systems. For example, Enterprise Resource Planning (ERP) and Enterprise Application Integration (EAI) approaches were developed for the smooth and efficient integration of enterprise operations. The interrelation between different DMUs should be analysed more accurately and realistically, so that we do not lose any vital dependency. The models found in literature may not be directly adaptable for such integration as there are many direct and indirect challenges. For example, Grossmann [65] discusses some of the direct challenges to enterprise-wide optimisation. We have consolidated the following challenges from the literature and our own experiences.

1. **Modelling challenge:** Development of appropriate optimisation models for the various components of the supply chain, particularly for nonlinear operations through integration and coordination, is a rigorous problem. We have multiple modelling choices available, from Linear Programming (LP), Mixed-Integer Linear Programming (MILP), Model Predictive Control (MPC), to Real Time Optimisation (RTO). Coordination of the optimisation models for planning and scheduling over different time scales is another tedious challenge. For example, in the airline industry, the hub location decisions are made for years, fleet planning and route planning are made for months and tail assignment is made for a week/day. Similarly, in a manufacturing SC, planning strategies are made for months/years, while the effect of control decisions lasts for a short time—seconds/minutes. The length of the planning horizon also plays a crucial role in optimising a dynamic system.

   Table 1.1 shows the challenges in integrated production, scheduling and control. The complexity and the run-time are lesser when the models are relatively simpler and deterministic. The complexity and modelling choices, and the time constraints make control optimisation problems difficult.

Table 1.1: Integrated planning, scheduling and control (Inspired by Grossmann [65])

DMU	Planning	Scheduling	Control
Model	LP/MILP	MINLP	RTO, MPC
Planning horizon	Years/months	Weeks/days	Minutes/seconds
Focus	Economics	Feasibility	Dynamic performance
Complexity	$\xrightarrow{\hspace{6em} increases \hspace{6em}}$		
Examples	Investment on production units, infrastructure, long term contracts, and others	Procurement and distribution plans, machine scheduling	Decisions on communication networks

2. **Algorithmic and computational challenge:** Solutions to the various models in terms of efficient algorithms—particularly for large scale integer programming, global optimisation and stochastic programs—is a challenge for SCC. Modern computer architectures such as grid/distributed computing, advanced networking and the like, should be utilised to address this challenge. The new approaches discussed in this thesis are computationally efficient. Details are provided in the respective chapters.

3. **Information sharing:** As we observe from the industrial problems, a good amount of information is shared between the DMUs of a supply chain. However, before sharing of the information, it is important to address the following important questions: (*i*) What are the critical bits of information that can affect the final solution? (*ii*) How much of this information needs to be shared? (*iii*) How can we quantify the usefulness of this information? (*iv*) Is there a mechanism for sharing information and does this mechanism protect the privacy and security of the data? Quantifying the value of the information is also a challenge in a decentralised coordination model.

   The success of coordination depends heavily on the way in which information is shared between models. The players should have mutual faith and the responsibility for sharing their critical information. Information delay, distortion, variability and loss are also to be considered in information flow modelling.

4. **Negotiation and incentive sharing:** Different DMUs or players in a supply chain have to negotiate and decide on an incentive sharing scheme to achieve a better coordination amongst them. Each player needs a positive incentive to coordinate, with an aim to move towards global optimality, and to not make locally optimum decisions. Success of the coordination will be undermined if players do not get any

incentive. The incentives can be some percentage of the profit or an assurance of continuity of business partnerships or the like.

5. **Decentralisation:** Multiple owners, plants or distribution centres located in multiple countries, with non-identical multiple objectives are another major concern in SCC. It is very difficult to integrate all DMUs of multiple owners. The foreign exchange, tax structure and raw material availability are some of the constraints. The mathematical models become computationally unsolvable when the number of plants, retail outlets, product items, and periods are large [109]. The complexity of the integration process sometimes forces researchers to use heuristic or meta heuristics approaches to solve the optimisation problem [24, 25]. These methods converge to a sub-optimal solution faster than in an analytical procedure. The growing computational power enables us to use these evolutionary algorithms to find the solutions to large and complex problems.

6. **Multiple objectives:** There are many independent DMUs in any supply network. Therefore, it is not easy to design a single performance measure to compare different solutions. The different objectives of these DMUs can conflict or be similar or have different scales and units. Hence, it is important to consider these multiple objectives carefully. In general, a centralised approach expects to have a single supply chain objective by taking a combination of these individual objectives. In a decentralised approach, we have the additional freedom of giving importance to the individual player's objectives. Concerns about satisfaction and improvement of individual DMUs compared to that of a supply network will exist. The overall optimality can be defined with respect to the objective costs of the integrated model. However, as we have mentioned earlier, the decentralised models may have multiple objectives. In that case, we need to compare the optimality at the DMU level and the global feasibility. Appropriate negotiation and the corresponding incentive scheme are crucial in implementing a successful coordination.

Other issues such as stochastic variations in demand, price, supply and the like make a decision model large and complex. On top of this, non-linearity increases the complexity and the size of the decision-making model. We address some of these challenges in our research.

1.3 Objectives of research

The overall objective of the research is to study decentralised decision-making approaches for a multi-party supply chain. To achieve this, the following objectives are identified and addressed in this thesis.

1. Conduct extensive literature review of supply chain coordination and different modelling approaches. Based on this, propose a classification and a framework for SCC models.

2. Identify and formulate mathematical models of coordinated/integrated production-planning and resource-scheduling decisions in a two-player supply chain (anchored to a specific supply chain, namely, coal).

3. Develop decomposition approaches based on *Lagrangian relaxation* and *column generation* to solve the integrated problem. Explore various strengthening methods to improve convergence of the decomposition methods.

4. Benchmark and analyse the performance of decomposition approaches in solving large, realistic, randomly generated problem instances.

5. Identify the role of information-sharing in two-party SC and analyse the key components in decentralised decision-making. Then, propose a framework for the decentralised approach—which has limited access to the information and does not require any central coordinator. Develop a decentralised approach for the two-party case and quantify the impact of information sharing and decentralisation.

6. Identify and formulate mathematical models of production-planning and resource-scheduling decisions in a decentralised three-player supply chain (anchored to a specific supply chain, namely, coal). Develop a decentralised decision-making framework, and quantify the impact of having an additional player and information sharing.

7. Propose a generic framework and guidelines to develop and to implement decomposed and decentralised decision-making in multi-player supply chains.

1.4 Summary of the thesis

This chapter presents an overall idea and motivation for the research problem. The importance and practices of supply chain coordination are summarised in this chapter. Chapter 2 presents a few supply chain coordination examples and an extensive literature on supply chain coordination models, different mathematical approaches, decentralised decision-making and information sharing. Chapter 3 provides a classification based on the decision-making and operational models of the supply chain. The issues with traditional centralised models and the need for decentralised models are also presented in this chapter. An example from the coal supply chain has been developed to explain different modelling approaches. The classification of coordination models and different modelling approaches mentioned in brief in this chapter are bridged logically in the remaining chapters. It gives a coherent view of this thesis.

We start with a two-party case where we have a set of independent producers and a common resource manager. Chapter 4 introduces the integrated production-planning and resource-scheduling problem, the decision makers and their decisions. Decomposition approaches are explored to solve the integrated problem by splitting it into solvable sub-problems. Chapter 4 explains an iterative scheme developed, based on *Lagrangian relaxation* (LR). Mathematical models, an LR-based algorithm, improvements and results are presented in this chapter. In Chapter 5, the same integrated problem is decomposed with Dantzig-Wolfe decomposition and solved with *column generation* techniques. Some stabilisation methods and ideas for improvement are also presented. This model is later benchmarked with the LR model and the integrated model.

We have further reduced the information sharing between the DMUs and the role of a central coordinator and developed a decentralised approach of the two-party case, and presented it in Chapter 6. This chapter includes an in depth discussion on information sharing and the value of the information. Even though the CG algorithm performs better than the LR, we could not use it for the two-party decentralised case because there was no decentralised approach, such as secure-sum, to compute the value of a column without sharing the actual values. Hence, we have used an LR-based algorithm for this two-party case. Computational results comparing the impact of decentralisation and the value of two sections of information are also presented in this chapter.

The two-party decomposed and decentralised approaches were complex enough to explain the intricacy of distributed decision-making. However, to propose a work flow for the multi-party case, we extended the two-party decentralised approach for a three-party case which has an additional shared resource. Chapter 7 presents the three-party decentralised approach based on the CG framework. Based on different levels of information sharing, the solution algorithm can be customised. Chapter 7 concludes with a discussion on generalising the solution approaches to a multi-party case in which we have many more common resources. Chapter 9 concludes the thesis by listing the major contributions and proposing a few ideas for extension.

The thesis presents a transition of solution approaches from integrated to decomposed and then to decentralised ones. The industry seeks a coordination approach that can deliver quality solutions in a reasonable amount of time without compromising the autonomy and confidential information of the individual DMUs. In the coordination problem under consideration has a set of independent decision makers are linked with a few shared resources, it is intuitive to link the decomposed approaches with well known mathematical decomposition techniques (Benders, LR, CG etc). We have used LR and CG to illustrate our approaches. The proposed approach is not merely a solution technique, it enables independent decision making by the partners and a collaboration between them. Therefore, we envisage that decentralised decision-making will be a prime enabler of supply chain coordination in the future.

Chapter 2

Literature Review

Although the methods that have been developed in this thesis are validated within the context of coal supply chains, similar coordination problems are also seen in other supply chains. Some such examples are listed below to explain the sequence and complexity of the decision-making process, which motivated the research in this thesis to improve coordination in decentralised supply chains.

2.1 Supply chain coordination examples

2.1.1 Manufacturing industry

In any manufacturing industry, procurement, production, marketing and sales, distribution and finance are the key components. Each of these decision-making units (DMUs) are involved in strategic, tactical and operational level decisions (classified based on the expected duration of impact). The decisions taken by DMUs at various levels affect and are affected by the decisions of one or more of the other DMUs.

Figure 2.1 shows a manufacturing supply network, where the producer has to procure raw materials from the suppliers, process and send it to the retailers. There are logistics operators connecting these DMUs. The forward arrow shows the material flow from the suppliers to the retailers. The information flow is from the right to the left. All of these DMUs have to be coordinated to achieve overall and individual benefits.

In a typical manufacturing supply chain (SC), the marketing and sales unit comes up with the consolidated forecasted demand for various products for multiple periods over a time horizon. They also determine the price of products, promotional schemes, and their impact on future sales to determine the forecasted demand. The objective can be to maximise revenue. Using the forecast as the input, the production unit determines the production plan, considering the bill-of-materials (BoM), machine capacity, and process

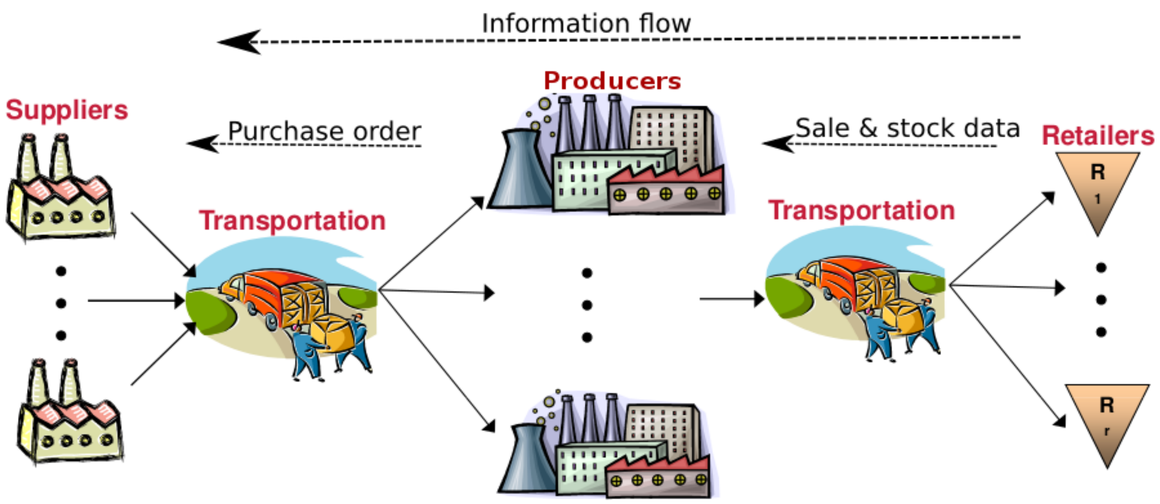


Figure 2.1: A manufacturing supply chain (Source: Venkateswaran [145])

plans. The production plan provides the quantity of products and production schedule. The objective is to minimise the inventory and production costs. Now, based on the raw material requirements plan, the procurement unit decides on the quantity of the materials to be purchased from the suppliers at different time points. The procurement unit also takes into account other costs such as the raw material holding costs, transport and purchase costs. The objective is typically to minimise the costs.

In general, multi-stages and multi-sites problems are represented as supply chain networks. Figure 2.2 shows the process of making a thin film transistor-liquid crystal display (TFT-LCD) presented in [94]. The suppliers of glass, colour filter, polariser, integrated circuit (IC), printed circuit board (PCB) and different assembly units and others, are the parts of this supply chain. The arrow shows the flow of materials. Any other super market chains, such as Wall-mart or McDonalds, can be represented in a similar way. They collect the raw materials, process, transport, and store, and then distribute to the retailer centres.

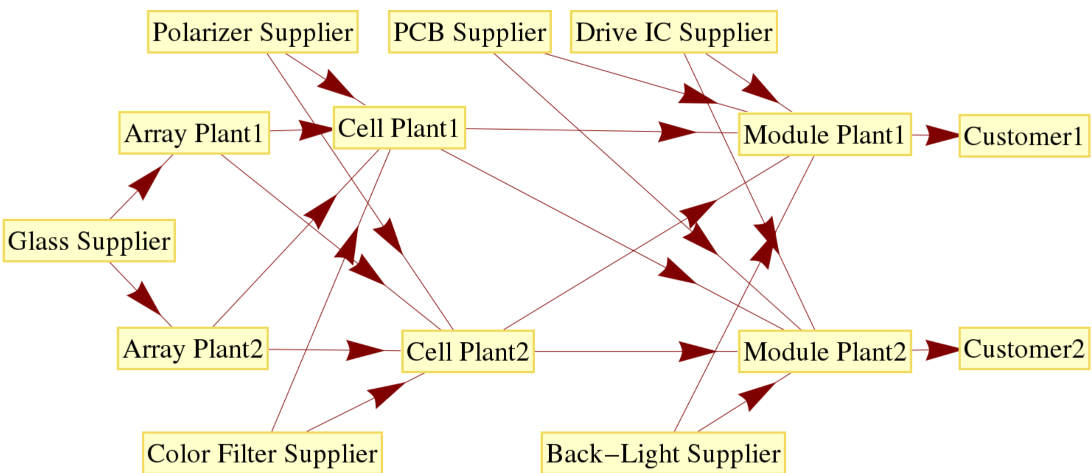


Figure 2.2: An example of multi-stages and multi-sites supply chain (Adapted from [94])

The automobile manufacturing chain can also be represented as a network of actions (see Figure 2.3). Most of the supply chain networks are not uni-directional. Sometimes, old or used products are called back due to errors or the recycling policy. Here, we can see the need for internal and external coordination between the DMUs. Sometimes, they might compete with each other, like the wholesale distributors in the Figure 2.3. The arrow shows the material flow. This is an example to show that individual DMUs (for example, plant) can have multiple sub-units. However, we focus only on coordination at supply chain level.

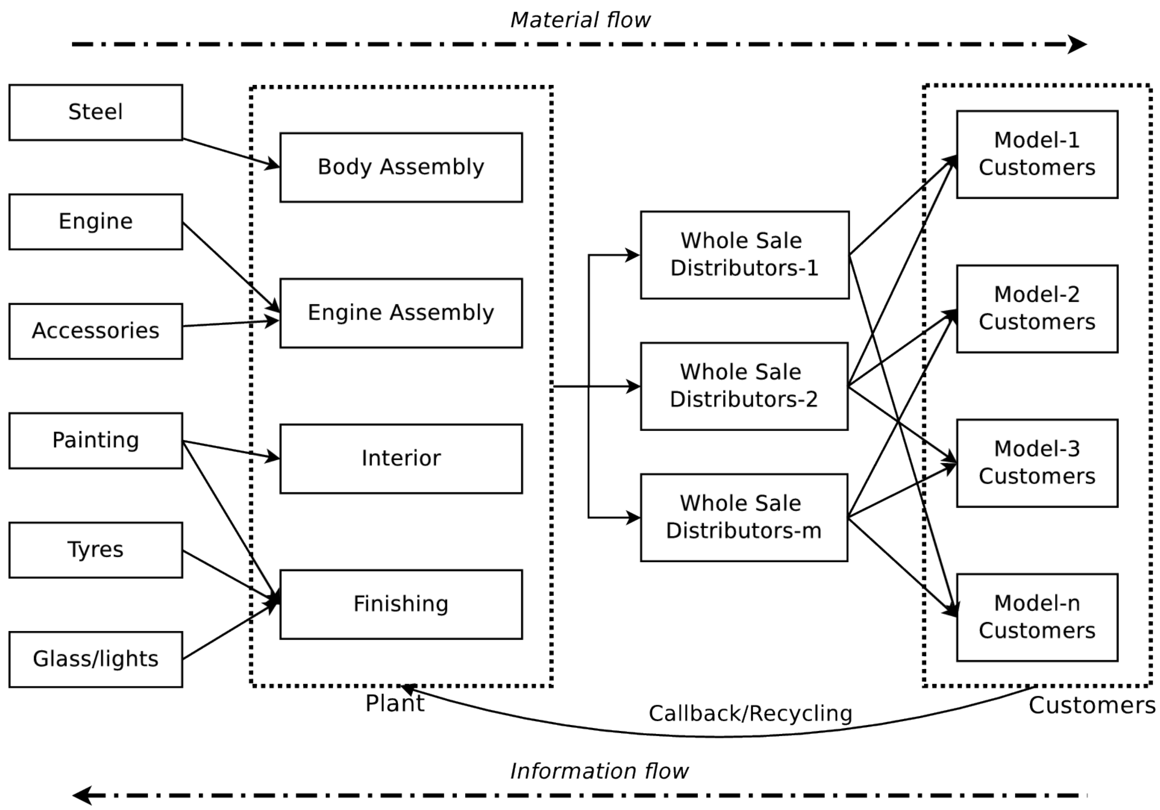


Figure 2.3: An automobile supply chain with a callback option

Steinrücke [128] also presented a similar coordination problem in an aluminium supply chain network. The stages in aluminium production involves (*i*) bauxite mining, (*ii*) preparation of aluminium oxide from bauxite, (*iii*) smelting, and (*iv*) casting. The authors modelled it as a multi-stage production-shipping and distribution-scheduling problem with a time-continuous representation.

2.1.2 Petroleum industry

The petroleum industry supply chain is large, complex, and highly dynamic. It includes a well-head, trading of crude, refinery optimisation, transfer of products, terminal operations, local distributions and the like (see Figure 2.4). In such chains, optimisation tends to have very large financial payout [65].

Figure 2.4: A supply network in the petroleum industry (Source: Grossmann [65])

Refinery optimisation Refinery operations are the core of the petroleum industry. Refinery tends to have long-term/short-term contracts with crude suppliers from many sources. The optimisation at this level is very dynamic and complex. Along with the complexity of refining, there also exists a great degree of freedom in refinery operations. Multiple products can be produced from raw crude oil, such as, naphtha, gasoline, kerosene, lubricating oils amongst others by the process of distillation. Even for a single product, for example, fuel, refineries can produce many varieties by adding sub components to boost customer satisfaction and thus the profit. This makes it a multi-objective optimisation problem. Possible objectives of the refinery are minimising crude cost, optimising the overall product mix and dispatch, minimising quality give away, optimising utilisation of the assets, optimising unit operations, maintaining the highest standards of safety, catalyst life and activity, and the like.

Scheduling decisions have a larger role in this industry. It starts from the scheduling of well-head operations and ends at the operations of distribution outlets. Scheduling decisions are very import in interlinking different sub-units. Due to multiple players and continuous operations, scheduling is not so easy in this network.

Trading of raw material and final products also contributes to the profitability of this supply chain. Raw material availability, market demand and other global issues affect trading operations.

Logistics is also an important aspect of the petroleum industry. The movement of crude from well-heads to refineries and then to distributors involves bulk cargo movement using ships, transfer through large pipes, trains and tankers. The volume and variety of products increase the complexity of the routing problem. Some of the logistics decisions impact on the long-term investment to develop required infrastructure.

In this SC, well-head and other infrastructure development decisions have a very long impact horizon. However, short-term decisions associated with the pump, such as pricing and supply, are dynamic and dependent on external factors. Multiple ownership and huge investments required in each step escalate the impact of optimisation in each of these activities. This is an example of a diverging supply chain, which requires coordination at each stage to handle the inventory efficiently and to improve profitability.

2.1.3 Coal industry

Coal supply chains also can be explained in this manner—similar to Petroleum SC. A coal supply chain has multiple players such as mines, rail operators, terminals. The terminals/ports act as 'collection centres' which receive the coal from different mines. The coal is usually transported by rail with a small percentage being transported by road. Each 'player' in the supply chain has some control over certain parts of the chain. For example, mines have control over the amount of coal that will be mined, rail operators can decide how many trains need to be sent and to which mine, the track authority provides tracks to run trains on and the port operators control operations within the port, that is, which ship needs to be loaded and when. Some of the key information such as properties of the track, coach and the like are shared between the players.

The Hunter Valley Coal Chain is one of the major coal supply chains in Australia and largest coal supply chain in the world in terms of throughput. They handle very large volumes of coal every day. It has a complex network with 35 coal mines, 31 load points, 44 trains (2200 trips per year), 9 ship berths, spread over 380km*. An elaborate and in-depth description of a Hunter Valley coal supply chain is available in [125]. Figure 2.5 shows a schematic diagram of a complex coal supply network.

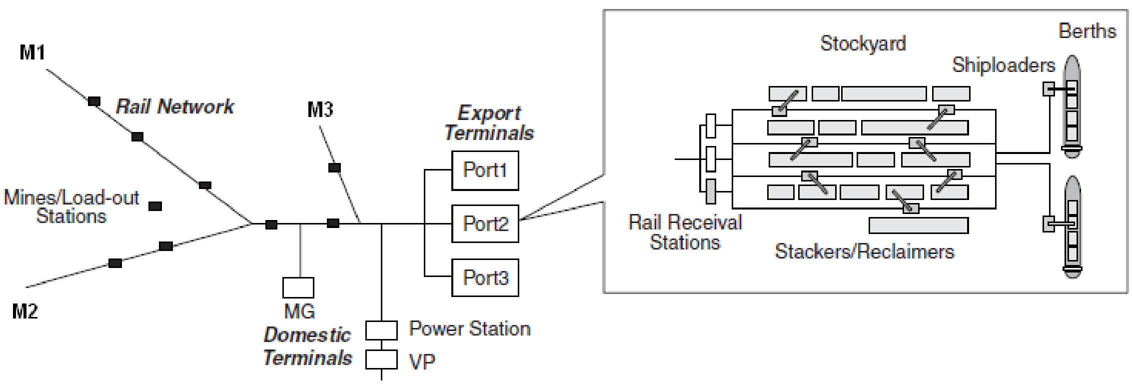


Figure 2.5: A schematic of a coal supply network (Source: Singh et al. [125])

*https://www.hvccc.com.au/AboutUs visited on 18-Jan-2014

2.1.4 Airline industry

The decisions in the manufacturing and petroleum supply chains are dependent on the flow of materials from the supplier to the customers. However, in some other supply chains, especially in the services sector, we can see the flow of services governed by many decision-making units. For example, the airline industry has a very active and dependent supply chain, including sub-components like route planning, aircrew scheduling, fleet planning, hub location and others. Figure 2.6 illustrates the hierarchical decision making and their interdependencies in an airline supply chain. The individual decisions may have different time lines and objectives. The services sector is highly dynamic and sensitive. Unlike the

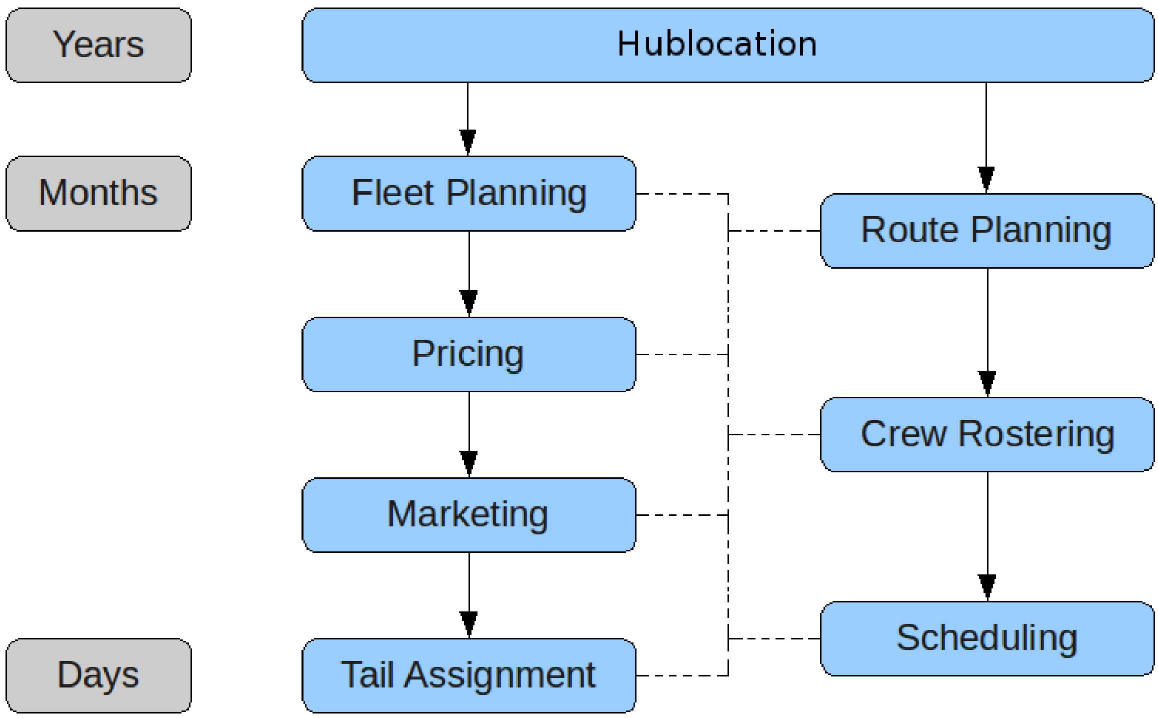


Figure 2.6: DMUs and their planning horizon in the airline industry (Adapted from [108])

manufacturing industry, a small increase in the price margin may result in huge financial returns. Some of the important sub-units are listed below:

Hub location problem is considered as a classic optimisation problem. A hub is a large airport which acts as a transfer point in between the journey. The complexity and different solution approaches make this problem very interesting to the research community. The economics of using a smaller number of high capacity aircraft against a larger number of low capacity aircraft is the core of the hub location problem. A decision concerned with hub location is a strategic decision taken for a longer planning horizon. Maximisation of revenue, minimisation of operational costs and customer satisfaction are also considered in this DMU.

Fleet planning is a critical step in airline scheduling to decide which aircraft to be operated between different locations. The general objective of this DMU is to maximise revenue with respect to the capacity, demand, and other constraints. Aircraft procurement and maintenance routing are directly dependent on this. Decisions horizon is typically in months.

Maintenance routing is necessary to schedule periodic maintenance of aircraft. The location of the maintenance centre, availability of resources and government/other agencies and safety requirements are the major constraints considered in this DMU. The most popular objective is to minimise the overall cost while improving safety measures. Here also, the decisions are taken for months.

Crew pairing is the process of assigning crew to a specific trip. The aircraft type, passenger pattern, services offered, schedule and others, are the constraints of this DMU. It is mostly planned on a monthly basis to minimise the crew cost, subject to maintaining customer service levels and other labour norms.

Pricing and marketing strategies in the aviation industry inspired to develop many optimisation models, which use dynamic pricing and competitive marketing. The airline revenue management model is very complex and sensitive. Due to market competition, service providers are forced to offer an attractive price range without heavy profit margins. The objective is to maximise the overall revenue keeping in mind the constraints on continuity in business, customer satisfaction, quality of services, and other regulations.

Integrating all airline scheduling problems into a single problem and solving, is very complex and computationally intractable [95]. Therefore, the common tendency is to update the existing schedules sequentially based on the feedback from other DMUs. Papadakos [108] highlights the origin and history of semi-integrated models used in airline scheduling. The aim of these models was to achieve better quality results, with the ultimate goal being to integrate all the stages.

2.1.5 Wine industry

The wine supply chain is complex and fragmented with more distant suppliers and diverse customers. In most of the cases, different DMUs are owned/ operated by different players, who might be involved in multiple supply chains. The major DMUs of the wine supply chain are the grape grower, the wine producer, carriers, harvester operations, and retailers. The coordination based on financial dependency is very strong in the wine supply chain. The storage and transport facilities need more investments and planning to process perishable, raw grapes. Figure 2.7 is a pictorial representation of a wine supply chain.

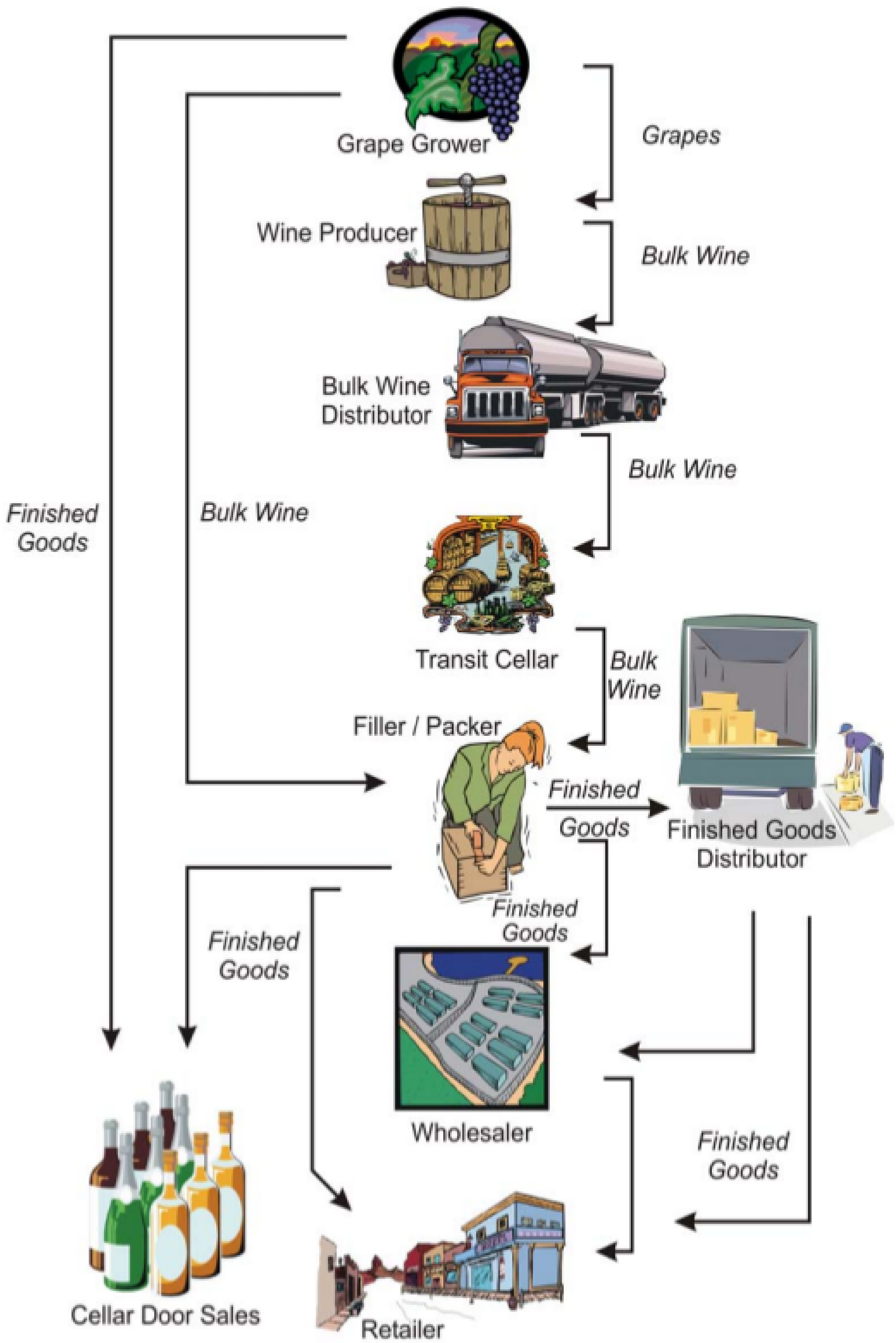


Figure 2.7: A pictorial representation of a wine supply chain (Source: GS1 [67])

Growers are responsible for the selection and production. Sometimes, vineyards are also operated under lease/partnership. Growers are involved in long term contracts between wineries and land owners. The decision-making at this level is dependent on relatively lengthy production cycles (three to five years), the complexity of final products and the risk of seasonal variability and demand.

Harvesters mainly offer their service to vineyards in harvesting, using specialised equipment and processes, till the grape is ready to be sent to wineries. Harvesters are third party partners who work for multiple vineyards. Therefore, their scheduling problem depends on route, the area, the winery schedule, and other priorities. Another important constraint is that the same type of grapes needs to be harvested in the same period, so that they can be processed together. Larger vineyards may require multiple units of harvesting operators.

Wineries The wine producer is responsible for receiving the grapes and for the production, manufacturing and/or blending of wine products. Their production problem depends on the quantity and quality of the raw grape and market demands. Wineries and Growers must be coordinated for effective production and utilisation of their products and services.

Carriers play a key role between the three players mentioned above. They also offer service to multiple and competing players. Other partners' schedule, demand-supply flow and others, influence the carriers' operations. The carriers have a tight time frame for pick-up and delivery, before deterioration of grapes occurs.

To understand the complexity, Singh et al. [124] explain an instance of the wine supply chain where the raw materials supply was: over 150000 tonnes of grapes; 35 grape varieties; over 30 distinct areas; more than 500 growers; around 100 unique wines; and over 3000 harvest units and 100 carriers and others. The 'uncertainties' in various stages of the supply chain play major roles in the undesirable behaviour of supply chain dynamics [129].

The supply network in the previous examples, the auto-mobile manufacturing, the petroleum, and the coal supply chains are somewhat deterministic because their planning horizon is longer. However, airline and wine supply chains are highly volatile as they have to make some decisions for a shorter planning horizon, repeatedly. In some sense, the DMUs of an airline supply chain are in-house, mostly under a common management. The DMUs of a coal/petroleum/wine supply chains are generally of the multi-owner type due to their huge capital investment and geographical constraints. Irrespective of the structure, the length of the planning horizon, and other factors, coordination is necessary in these supply chains to improve the performance of their overall and individual DMUs.

2.2 Background literature

In the previous section, many industrial examples are presented to understand supply chain coordination (SCC) problems. In most of the supply chain, DMUs are distributed and connected through some mechanism. Therefore, decentralised decision making (DDM) approaches are preferred over the integrated approach to address SCC problems. This section presents a summary of relevant literature to develop DDM approaches.

Integration/coordination models were first applied in manufacturing supply chains. Their successful implementation inspired many other industries to adopt these concepts into those sectors too. There are many review articles available specifically covering production distribution coordination models [146, 138, 24, 105]. Some of the relevant literature related with this thesis are grouped and given below in four sub-sections.

2.2.1 Supply chain coordination

Supply chain coordination has been studied through many mechanisms such as contracts and quick response-vendor managed inventory [31, 126, 70, 141, 113, 130]. The chapter written by Cachon [31] on supply chain contracts has been cited more than a thousand times in reputed refereed journals. It explains six types of contracts—whole sale pricing, buyback, revenue sharing, quantity flexibility, sales rebate and quantity discount contracts. Toktas-Palut and Ulengin [141] compared supply chain contracts between suppliers and manufacturer by examining subsidy, transfer pricing and cost sharing. In general, contract based coordination models are solved in parallel as functions of contract parameters. These parameters are computed in such a manner to achieve global or Pareto optimality. Rodriguez-Rodriguez et al. [112] explored a negotiation model applied on a real business situation to identify and tackle the bottlenecks in a multiple vendor-retailer supply chain. Chen and Chen [35] analysed the effect of centralisation versus decentralisation in a two-echelon supply chain using joint replenishment. The authors conclude that in terms of cost reduction, a centralized, coordinated replenishment policy is preferred over decentralized replenishment policy. [11] considered a similar replenishment problem in the pharmaceutical supply chain. Arshinder et al. [7] review the perspectives empirical studies and research directions of supply chain coordination (SCC). Since there is no unique globally accepted definition for SCC, it can be interpreted as a collaborative working framework for joint planning, development, information and revenue sharing, or a win/win arrangement that is likely to provide improved business success to all parties. Stadtler [127] provides a framework and an overview of the state-of-the-art techniques for collaborative planning. The authors have reviewed ten papers published between 1995 and

2007, which mainly use exact solutions, decomposition methods, economic order quantity (EOQ) concepts, contracts and multi-agent models to enable collaboration in the supply chain. Some of the main characteristics of collaborative models were identified and discussed in this chapter.

Bahinipati et al. [13] summarise the literature base and development of supply chain management through various coordination mechanisms and contracts. The review is mostly focused on buyer-supplier coordination models under different scenarios. The role and problems of supply chain coordination are summarised as follows.

- Objectives of coordination
  - Aligning all supply chain decisions to satisfy global system objectives and to the advantage of every supply chain player
  - Improving economic efficiency (profit or utility utilisation) in a decentralised setting and to bring them closer to those of a centralised setting
  - Unification of different decisions of the enterprises and multiple players
  - To make sure that the individual player optimises its decisions or appropriate incentive is provided to accept a sub-optimal solution.
  - Reduce the modelling and computational complexity in solving large supply chain problems
- Effects of lack of coordination
  - Double marginalisation results in a bullwhip effect
  - Ineffective utilisation of resources which will leads to lower revenue and profit
  - Local optimisation produces overall sub-optimal solutions. Sometimes, local optimisation might end up in a conflicting or gaming situations.
  - Information inaccuracy or a wrong forecast of input parameters will result in a incorrect decision making.
- Possible reasons of coordination problems
  - Decentralised decision-making due to diversity in the supply network and distributed DMUs
  - Interdependency among players can only be addressed through coordination.
  - Multiple objectives, some of them might conflict or be similar.
  - Competition between supply networks force the DMUs to coordinate to improve their performance and as well as the overall performance
  - Partial information-sharing leads to some gaps in the estimate of input parameters and system behaviour.

Hugos [78] describes some important factors affecting coordination in the context of managing the *bullwhip effect*. The author has provided an overview on the necessity of global data synchronisation and an evaluation method, 'collaborative planning, forecasting, and replenishment' (CPFR). The rapid development of information technology and the global market turns collaboration into one of the most critical factors for global companies. Hence, the collaboration between different functional units in a supply chain is decisive in (a) responding to rapid changes in customer needs and (b) increasing the efficiency of the whole supply chain [81]. The author has used an agent-based collaboration approach and discussed incomplete information-sharing. In this model, only the cost parameters are private information, all constraints are public. The global objective of the model is to meet the customer-demand with minimum cost. Shen et al. [118] explain the applications of Agent technology in SC integration, SC collaboration and other fields. The author has provided a comprehensive review of these applications. Also discussed are some key issues in implementing agent-based manufacturing systems such as agent encapsulation, agent organisation agent coordination and negotiation, system dynamics, learning, optimisation security and privacy, tools and standards and future directions. Monostori et al. [103] define a computational agent as an autonomous, intelligent, interactive and adaptive tool designed to meet a particular purpose. Multi-agent systems are very useful in a decentralised or a distributed supply chain, where incomplete sources of information are only available. The scope of agents and the multi-agent system in 3C's—Coordination, Collaboration and Cooperation—are emphasised here. Chiu and Lin [38] presented the concepts of collaborative agents and artificial neural networks in collaborative SC planning. The author makes use of the SC network for their analysis. Forget et al. [57] highlight the necessity of a coordinated supply chain and the role of agents and challenges the conventional straight forward method in highly dynamical situations. Simulation is used to compare the performances of multi-behaviour planning agents, explained with a case study. A special issue on *'Operations Research and Accounting in Supply Chain Coordination'* of OR Spectrum [9] includes articles on the principal-agent problems, risk-incentive trade-offs and dynamic contracting problem in continuous time.

2.2.2 Mathematical models

Integration/coordination models were first applied in manufacturing supply chains. Their successful implementation inspired many other industries to adopt these concepts into those sectors too. Many review articles are available, specifically covering coordinated production-distribution models (see, for example, [146, 138, 24, 105]). An invited review on mathematical programming models for supply chain production and transport planning [105] classifies models based on performance and novelty measures. The authors pointed out some of the gaps that exist in the literature, such as simultaneously optimising transport and production planning, supply chain collaboration, comparison of centralised and decentralised planning and the necessity of recent solution approaches such as multi-objective programming, fuzzy programming, heuristics and meta heuristics hybrid models. Maravelias and Sung [99] review some of the modelling approaches and solution strategies used in integrated production planning and scheduling. This paper highlights the latest trends and advances in this area. The authors discussed three solution approaches—hierarchical, iterative and full-space—to solve the integrated production scheduling problems. Steadily increasing computational power and improving decomposition methods bring many more opportunities. However, the authors assumed that the models are built when complete information is available or when all the models are controlled by a single decision-maker. Manne [98] used linear programming to solve a multi-item scheduling problem in the petroleum industry, approximately. The author has used many simplifying assumptions to create an integrated model. Lasdon and Terjung [89] proposed an alternative approach using the column generation approach and solved it effectively. Gnoni et al. [62] considered a multi-site, multi-product and multi-period lot sizing and scheduling problem from automotive industry and solved it using a hybrid modelling which combines MILP model and a simulation model.

Dempe [50] introduced the concept of bi-level programming. In this approach, decisions are made hierarchically in two levels, without any direct dependency. The multi-level programming concepts are getting popular in solving supply chain problems [131]. Anandalingam [6] proposes a mathematical programming model of decentralized multi-level systems. In this system, decisions at various levels are taken as Leader-Follower game. Abdelaziz and Mejri [2] discuss a decentralised bi-level model for shared inventory management. Bracken and McGill [27] introduces hierarchical optimisation as a generalisation of mathematical programming, where a series of optimisation problems are solved in a predetermined sequence. In this, independent decision-makers are looked at separately and in a hierarchical manner. It helps to reduce the size of the problem and the complexity of the integrated model. The application of decomposed hierarchical optimisation techniques extends to job-shop scheduling problems [28, 156], resource constrained scheduling problems [72, 85], and manufacturing problems [48, 53, 139, 151]. Pochet and Wolsey [111] discuss different production planning models and MIP formulations. The authors suggest many practical reformulation techniques and polyhedral results to strengthen the model.

Table 2.1 lists some of the coordination models from the literature and summarise their properties. It is not an exhaustive list. However, it is provided to show the wide coverage and different decision-making models for the supply chain coordination.

Table 2.1: A list of different approaches from literature

Author	**Description**	**Objective**	**Classification**	**Techniques**
Abdelaziz and Mejri [2]	Shared inventory management	Min *inventory cost and labour overtime*	Decentralised	NLP, Bi-Level model
Bard and Nananukul [15]	Integration of production, inventory, distribution and routing decisions	Min *total cost*	Integrated	MILP, Tabu search
Boudia et al. [26]	Optimisation of production and distribution of several products	Min *total cost*	Integrated, iterative	ILP, Meta-Heuristics
Chandra and Fisher [33]	Comparison of a centralised approach and a decoupled approach for coordinating production and distribution planning	Min *total cost*	Integrated, Sequential	ILP, Heuristics
Chen and Simchi-Levi [36]	Coordinated inventory control and pricing with random demand	Max *profit*	Integrated	Dynamic programming
Chien [37]	Production and transportation of a unique product directly from an origin to a destination where demands are stochastic	Max *profit*	Iterative	Monte-Carlo Simulation
Choi and Tcha [39]	A vehicle routing problem with a heterogeneous fleet of vehicles	Min *cost*	Iterative	Column generation, Dynamic programming
Cordeau et al. [44]	Simultaneous aircraft routing and crew scheduling	Min *cost*	Iterative	CG and Bender's decomposition
Dhaenens-Flipo [52]	A hierarchical scheme for multi-facility production and distribution	Min *total cost*	Hierarchical	Spatial decomposition
Guyon et al. [69]	integrated employee timetabling and production scheduling	Min *total cost*	Integrated	ILP, Benders decomposition
Haq et al. [73]	Integration of production, inventory, and distribution decisions	Min *total cost*	Integrated	MIP

Author	Description	Objective	Classification	Techniques
Kaminsky and Simchi-Levi [82]	Production and distribution lot sizing in two stages	Min *total cost*	Sequential	Analytical
Kanyalkar and Adil [83]	Aggregate production-planning, integrating procurement and distribution plans	Min *storage, cover, cost*	Integrated	Goal Programming, heuristic
Kazemi et al. [84]	A multi-agent system to solve the production-distribution planning problem	Min *total cost*	Integrated, hierarchical	Genetic algorithm
Lei et al. [92]	Integration of production, inventory, distribution and routing decisions	Min *total cost*	Two Phase	MILP, Heuristics
Papadakos [108]	Airline scheduling, and coordinated routing	Min *total cost*	Integrated	Benders decomposition, CG
Park [109]	Integrated approach for production and distribution planning	Max *profit*	MIP and heuristics	
Shah et al. [117]	Proposed a decomposition strategy for solving large scale refinery scheduling problems	Min *makespan*	Centralised and Decentralised	Structural decomposition
Singh and Weiskircher [122]	A multi-agent system for decentralised fractional shared resource constraint scheduling	Min *tardiness*	Decentralised	ILP, LR
Steinrücke [128]	Modelled an integrated production- transportation planning and scheduling in an aluminium supply chain	Min *total cost*	Integrated	MILP
Taghipour and Frayret [130]	Proposed a decentralised coordination mechanism based on a negotiation-like mutual adjustment of planning decision	Max *total profit*	Decentralised coordination	Distributed heuristic search

where, ILP: Integer linear programming; MILP: Mixed-integer programming; MIP: Mixed-integer programming; NLP: Non-linear programming; CG: Column generation; LR: Lagrangian relaxation

2.2.3 Decomposition methods

Integrated models are often solved using decomposition techniques such as Lagrangian relaxation, column generation and Benders decomposition. The books by Bertsekas [20], Wolsey and Nemhauser [150] and Conejo et al. [43] can be referred to for a quick theoretical foundation of different decomposition techniques. Most of these decomposition algorithms make use of the primal-dual relationship in computing the multipliers. The vast literature available on this topic highlights its advantages and practical applications.

Bender's decomposition [19] and Dantzig-Wolfe decomposition [45] were proposed in the early 1960's. Both these methods have similar iterative structure and cut generation format. Over iterations, Benders' decomposition adds new constraints while the Dantzig-Wolfe decomposition adds columns. A popularly used technique, Lagrangian relaxation, is developed based on the decomposition of dual prices. LR and its applications drew the attention of researchers in the 1970's and 80's (see [60, 61, 55]) itself. At present, there are many variants, generalisation and improvements developed on these decomposition methods [132, 43, 51, 79, 96, 90]. Bassett et al. [16] studied large scale scheduling problems from the chemical-processing industry. The author has proposed a time-, resource- and task-unit based decomposition strategies to solve the problem efficiently. Osman and Demirli [107] considered a safety stock placement problem of multi-stage supply chains. The authors found that a centralised policy solved with Bender's decomposition has better performance over a decentralised policy. We have used some of these concepts in later chapters. Relevant details and corresponding literature are provided, wherever it is necessary, along with the implementation.

Decomposition approaches could handle the complexity of planning and scheduling problems (see [147]). Decomposed optimisation techniques have been extended to job-shop scheduling problems (see [28], [156]), resource constrained scheduling problems (RCSP) (see [72], [85]) and manufacturing problems (see [151], [53], [139]). Column generation (CG) methods have been incorporated along with the DWD to successfully solve many problems such as the cutting-stock problem, vehicle routing problem, crew scheduling problem, p-median problem, and graph colouring problem (see [23], [96], [143]). Choi and Tcha [39] proposed a CG approach to address a heterogeneous-fleet vehicle routing problem. Capone et al. [32] compare two variants of CG schemes developed for a resource allocation problem in wireless mesh networks. Given the inherent complexity of distributed decision-making problems, decomposition methods can be employed to solve these problems.

2.2.4 Decentralised decision-making

There are many articles comparing the decentralised and centralised modelling approaches [114, 116, 141, 117, 3, 107]. It is intuitive that a centralised model produces better results than those obtained from decentralised approaches. Centralised models are built on many assumptions [130]. Most of the articles comparing these two modelling approaches ignore the impact of the underlying assumptions and their practical difficulties, when it comes to implementing either of the approaches. From the literature and our own experience, we conclude that the major assumptions of centralised modelling are: (*i*) complete information sharing is available between the DMUs, (*ii*) it is possible to develop a single mathematical programming model and that the model is solvable, (*iii*) the complexity and size of the model is manageable/tractable, (*iv*) multiple objectives of different DMUs can be combined to make a global objective, and (*v*) the loss borne by a single DMU (so as to improve the overall supply chain's objective will be acceptable to all other DMUs. In reality, each of these assumptions are practically impossible to enforce. Therefore, it is necessary to explore the decentralised optimisation of SCs. Decentralised approaches may not provide globally optimal solutions, but will offer solutions that are close to the optimal which can be generated in a reasonable amount of time. Moreover, these solutions that are obtained through a decentralised approach may be more acceptable to each of the DMUs.

Belavina and Girotra [18] discuss the benefits of decentralised decision-making. In this article, coordination is achieved through price contracts. The discussion is mainly about the price and the quantity of the material transferred in an N-tier supply chain. Arshinder et al. [8] provide an extensive review of supply chain coordination. The authors discuss different coordination mechanisms, conflicts and classifications. Their view on popular features are summarised in Table 2.2.

Table 2.2: A summary of different coordination mechanisms and their features

coordination mechanisms	coordination problems	performance measures
• information-sharing	• mismatch in goals	• inventory level/cost
• contracts	• mismatch in timing	• overall costs/profits
• joint decision-making	• independent cost calculation	• service level
• joint cost consideration	• conflict in batch size	• operational performance
• integrating role	• increase of cost and inventory	• cycle/lead time, lateness/earliness

Buyback contracts and other quantity discounts are very useful in producer-retailer supply chains. However, in a producer-distributor supply chain, the producer needs to use some of the resources managed by the distributor/resource manager; it is irreversible. Another major difference is that the core decision to be made in production is regarding the quantity, and in distribution it is regarding resource utilisation. Both of these are measured and costed differently. Hence, coordination in producer-distributor SC is more challenging than that in a producer-retailer supply chain.

2.2.5 Information sharing

The technology has changed over the years to consider everything related with the SC as an information (for example, decisions, parameters, constrains). Due to the growing awareness and concerns on information access, supply chains have to consider information as a building block in designing better coordination [153, 47, 59]. Ganesh et al. [59] states *"Information sharing is viewed as one of the key elements for successful supply chain management and coordination"*. The authors discussed the value of information-sharing in a multi-level, multi-product supply chain. A regression model, and theoretical insights were presented to highlight the impact of product substitution on the value of information-sharing.

We find a number of articles from the literature discussing different aspects of information-sharing for different applications (see [70, 154, 34, 41, 113]). For example, Ganesh et al. [59] consider the value of information-sharing in a multi-level, multi-product supply chain. Ha and Tong [70] analyse contracts and their influence on the value of information-sharing in a manufacturer-retailer supply chain. The authors modelled the problem as a two-stage information-sharing game, where investment decisions are taken in the first stage and the order quantity is decided through contracts in the second stage. The authors have discussed equilibrium concepts and a dominant strategy of the game. Zhou and Benton-Jr. [154] investigated the role of information-sharing and the supply chain practice using data from 125 North American manufacturing firms. The authors state that both features have a significant impact on a good supply chain.

Chu and Lee [41] and Chen et al. [34] discuss the role of information-sharing in supply chains. Chu and Lee [41] considered a retailer-vendor supply chain and modelled the coordination using a Bayesian game. Ryu et al. [113] studied a strategic supply chain coordination scheme under a cloud computing environment. The authors developed methods to bargain for contracts at an operational level and modified them to control some decisions at the tactical level too. Taghipour and Frayret [130] discuss different coordination techniques to achieve coordination in a decentralised SC. The authors have considered a distributed heuristic search, supported by a negotiation and an incentive structure to ensure mutual benefits and adjustments in production planning. A comparison of the centralised approach and the proposed approach, based on the computational experiment, shows that the negotiation-based model can produce close-to-optimal solutions.

Kovács et al. [87] developed four different computational approaches, namely (i) decomposition, (ii) integrated, (iii) coordinated and (iv) bilevel, to analyse a two-stage lot sizing problem. The authors considered the impact of autonomy, information asymmetry and conflicting objectives in the decision making. One of their key observation is that the classical integrated approach is limited to a single owner/full information sharing supply chain and coordination approach may bring comparable savings for decentralised multi-party supply chain.

Huang et al. [77] reviewed more than a hundred publications on the information-sharing and its impact on supply chain dynamics. The author states that the information-sharing is significant in reducing the bullwhip effect and the supply chain costs. Few guidelines and classification on information and its influence are also provided. The article closes by highlighting the necessity to carry forward the investigations at different levels and pointing a few future directions. Hall and Saygin [71] analysed the effect of information-sharing in a supply chain using simulation. They considered the effect of three factors capacity tightness, resource reliability, and information-sharing on on-time delivery rate and total costs. The authors concludes that the *"it may be more feasible to introduce additional flexibility instead of solely focusing on information-sharing capability as a co-ordination tool"*.

Quantifying the impact of information in a supply chain is non-trivial [91]. Some information may only have short term impact while some other piece of information may have direct or indirect long term impacts [86]. The same information could yield different benefits in different supply chains. Davis et al. [47] quantified the value of information as the relative reduction in a specified performance measure between the no-information-sharing and information-sharing cases. A similar relative measure is used in Yu et al. [153], Ganesh et al. [59] also. Game theoretic concepts like Shapley Value and core can also be explored further to quantify the value of information-sharing [106].

The medium through which information is shared between players has also changed significantly over the years. Traditionally much of the information that was shared was through the medium of paper (with appropriate signatures to ensure delegated authorities share the information that was required). More recently, information is shared through the medium of fax, email, internet programs and telephone. Sometimes satellite communication devices have also been used for information-sharing [119]. The growth in the use of the internet has also allowed the supply chain to share information among players.

Table 2.3 highlights the relevance of the articles cited in this thesis with respect to the key areas discussed above.

Table 2.3: Relevance of the literature to the key areas

Article	SC Coordination	Math. models	Decomposition	Info. Sharing	Decentralised
Abdelaziz and Mejri [2]	–	✓	✓	–	✓
Abdul-Jalbar et al. [3]	✓	–	–	✓	–
Amor et al. [5]	–	✓	✓	–	–
Anandalingam [6]	–	✓	–	–	✓
Arshinder et al. [7]	✓	–	–	–	–
Arya et al. [9]	✓	–	–	–	–
Atamtürk [10]	–	✓	–	–	–
Baboli et al. [11]		✓	–	–	✓
Bahiense et al. [12]	–	✓	✓	–	–
Bahinipati et al. [13]	✓	–	–	–	–
Barahona and Anbil [14]	–	✓	✓	–	–
Bard and Nananukul [15]	–	✓	–	–	–
Bassett et al. [16]	–	✓	✓	–	–
Bastert et al. [17]	–	✓	✓	–	–
Belavina and Girotra [18]	–	–	–	–	✓
Benders [19]	–	✓	–	–	–
Borndorfer et al. [23]	–	✓	✓	–	–
Boudia [24]	✓	✓	–	–	–
Boudia and Prins [25]	–	✓	–	–	–
Bracken and McGill [27]	–	✓	–	–	–
Brandimarte and Calderini [28]	–	✓	✓	–	–
Brucker et al. [30]	–	✓	–	–	–
Capone et al. [32]	–	✓	–	–	–
Chandra and Fisher [33]	✓	✓	–	–	–
Chen et al. [34]	–	✓	–	–	–
Chen and Simchi-Levi [36]	–	✓	–	–	–
Chen and Chen [35]	✓			✓	–
Chien [37]	–	✓	–	–	–
Chiu and Lin [38]	✓	✓	–	–	–
Choi and Tcha [39]	–	✓	✓	–	–
Chu and Lee [41]	✓	–	–	✓	–
Clifton et al. [42]	–	–	–	✓	–
Cordeau et al. [44]	–	✓	✓	–	–
Dantzig and Wolfe [45]	–	✓	✓	–	–
Datta and Christopher [46]	✓	–	–	✓	–
Davis et al. [47]	✓	✓	–	✓	–
Dehayem Nodem et al. [48]	–	✓	✓	–	–
Delavar et al. [49]	✓	✓	–	–	–
Dhaenens-Flipo [52]	–	✓	✓	–	–
Ebadian et al. [53]	–	✓	✓	–	–
Erenguc et al. [54]	–	✓	–	–	–

Article	SC Coordination	Math. models	Decomposition	Info. Sharing	Decentralised
Fisher [55]	–	✓	✓	–	–
Fisher [56]	–	✓	✓	–	–
Forget et al. [57]	✓	–	–	–	
Fumero and Vercellis [58]	✓	✓	–	–	–
Ganesh et al. [59]	–	–	–	✓	–
Geoffrion and Graves [61]	–	✓	✓	–	–
Gnoni et al. [62]	✓	✓	–	–	–
Goyal and Deshmukh [64]	–	✓	–	–	–
Grossmann [65]	✓	–	–	–	–
Grossmann and Biegler [66]	✓	–	–	–	–
Guignard [68]	–	✓	✓	–	–
Guyon et al. [69]	–	✓	–	–	–
Ha and Tong [70]	–	–	–	✓	–
Hall and Saygin [71]	–	–	–	✓	–
Hans et al. [72]	–	✓	✓	–	–
Haq et al. [73]	–	✓	–	–	–
Hartmann and Briskorn [74]	–	✓	–	–	–
Heydenreich et al. [75]	–	✓		✓	–
Hohn [76]	✓	–	–	–	–
Huang et al. [77]	–	–	–	✓	–
Jung and Jeong [81]	–	–	–		✓
Kaminsky and Simchi-Levi [82]	–	–	–		✓
Kanyalkar and Adil [83]	–	✓	–	–	–
Kazemi et al. [84]	–	✓	–	–	–
Kelly and Zyngier [85]	–	✓	✓		✓
Kim [86]	✓	–	–	–	–
Kovács et al. [87]	✓	✓	✓	✓	–
Lambert and Cooper [88]	✓	–	–	–	–
Lasdon and Terjung [89]	✓	✓	✓	–	–
Lee and Park [90]	–	✓	✓	–	–
Lei et al. [92]	–	✓	–	–	–
Lohatepanont and Barnhart [95]	–	✓	–	–	–
Lübbecke and Desrosiers [96]	–	✓	✓	–	–
Mabert and Venkataramanan [97]	✓	–	–	–	–
Manne [98]	✓	✓	–	–	–
Maravelias and Sung [99]	✓	–	–	–	–
Marler and Arora [100]	–	✓	–	–	–
Marston et al. [101]	–	–	–	✓	–
Mason [102]	–	✓	✓	–	–
Monostori et al. [103]	–	✓	✓	–	–
Moon et al. [104]	–	✓	–	–	–
Mula et al. [105]	–	✓	–	–	–
Osman and Demirli [107]	–	✓	✓		✓
Papadakos [108]	–	✓	–	–	–
Park [109]	–	✓	–	–	–
Pessoa et al. [110]	–	✓	–	–	–
Rodriguez-Rodriguez et al. [112]	✓	✓	–	–	–
Ryu et al. [113]	✓	–	–	✓	

Article	SC Coordination	Math. models	Decomposition	Info. Sharing	Decentralised
Sarmiento and Nagi [115]	✓	✓	–	–	–
Selim et al. [116]	✓	✓	–	–	–
Shah et al. [117]	–	✓	–	–	✓
Shen et al. [118]	✓	✓	–	–	–
Simatupang and Sridharan [119]	✓	–	–	–	–
Singh and Ernst [120]	–	✓	–	–	–
Singh and O'Keefe [121]	–	✓	–	–	✓
Singh and Weiskircher [122]	–	✓	–	–	–
Singh and Weiskircher [123]	–	✓	–	–	✓
Singh et al. [125]	–	✓	–	–	–
Stadtler [127]	✓	–	–	–	–
Steinrücke [128]	–	✓	–	–	–
Swaminathan et al. [129]	–	✓	–	–	–
Taghipour and Frayret [130]	✓	✓	–	–	–
Thomas et al. [134]	–	✓	–	–	✓
Thomas et al. [137]	–	✓	–	–	–
Thomas et al. [135]	–	–	–	✓	✓
Thomas et al. [136]	–	–	–		✓
Thomas and Griffin [138]	✓	–	–	–	–
Timm and Blecken [139]	–	✓	✓	–	–
Toktas-Palut and Ulengin [141]	✓	–	–	–	✓
Varma et al. [144]	✓	–	–	–	–
Vidal and Goetschalckx [146]	✓	–	–	–	–
Wang et al. [147]	–	✓	–	–	–
Wedelin [148]	–	✓	–	–	–
Wentges [149]	–	✓	✓	–	–
Wu and Ierapetritou [151]	–	✓	✓	–	–
Wu et al. [152]	–	–	–	✓	–
Yu et al. [153]	–	–	–	✓	–
Zhou and Benton-Jr. [154]	–	–	–	✓	–
Zissis and Lekkas [155]	–	–	–	✓	–
Zribi et al. [156]	–	✓	✓	–	–

As seen in Table 2.3, only Kovács et al. [87] covers all key areas, except decentralisation. The author has explored coordination in inventory control problems using decomposition techniques. The authors observe that full information sharing and a well developed coordination approach can achieve significant benefits in a decentralised supply chain. Nevertheless, the supply chain coordination problem considered in this thesis is very large and complex with multiple players and has very little similarity with the two-stage lot-sizing problem discussed in [87].

In our literature survey, we found only a small number of articles which discuss the decentralisation, information sharing and supply chain coordination being considered together in arriving at decisions in complex, multi-partner supply chains. This demonstrates a significant gap in the literature. We ascertain, therefore, that there is a need to analyse the role and the impact that information sharing and coordination have in a decentralised, multi-party supply chain.

2.2.6 Main gaps in the literature

Even though there are separate attempts to address decentralised decision making, supply chain coordination and information sharing; to our best knowledge, the literature does not cover all these aspects together. Application of mathematical decomposition for decentralised decision making is also an under studied research area. The growth in the use of the Internet and advancements in technology allowed us to use distributed computing facilities. However, there are security and privacy concerns with the use of distributed computing environments for information sharing and coordination. We could not find any significant industrial case study comparing different modelling approaches or any coordination mechanism which uses the value of information sharing. Considering all these gaps, we developed a decentralised framework of decision-making for a multi-party supply chain and demonstrated it using a coal supply chain.

2.3 Summary

This chapter demonstrates some examples of supply chain coordination. The decision making units in each of these examples are interconnected. Based the properties of supply network and its distributed DMUs, new decomposed and decentralised decision making approaches are proposed to solve SCC problems. Background literature related the supply chain coordination, modelling approaches, decomposition and information-sharing are also summarised. In the next chapter, we propose a classification of coordination models and methodology based on the available literature.

Chapter 3

Supply Chain Coordination Models

In this chapter, we propose a new classification of supply chain coordination (SCC) models for general supply chains. This thesis considers a multi-resource constrained scheduling problem which involves a set of producers and a group of resource managers. We suggest different modelling approaches to solve such kind of problems. A coal supply chain is taken as example to illustrate the coordination problems with a single shared resource and two shared resources and their solution approaches. The coordination problems which have a similar structure are abstracted into a generic problem, and is presented with its mathematical representation.

3.1 Classification of supply chain coordination models

It is sometimes necessary to adopt integrated approaches to arrive at coordinated operational decisions in supply chains. However, as supply chains become globally distributed and where there is an external resource provider, who also provides resources to several other competitors, coordinated operational decisions become harder to achieve through a single decision model. The decision model of supply chains where the decisions are taken by a single decision maker is referred to as *centralised* or *integrated* [76] decision model. In such models, we have a single enterprise with many smaller decision-making units (DMUs). The result is that it is possible to achieve integrated (or centralised) decision-making reasonably easily. Note that integration is possible across multiple levels (or, layers) of a supply chain. Here we only consider operational decision-making levels. However, integrated (operational) decision-making is more difficult in today's global enterprise models, which are more complex in their structure than traditional, single-enterprise, organisational models.

Some parts of this chapter were presented in ICSEM-2011 [133]

In today's large distributed SCs, there are typically two main models of operations:

MOM *multi-operator model*: There are many *independent decision-making units* (DMUs), each owned and operated by a separate player.

SOM *single-operator model*: The large supply chain that contains a plethora of independent—yet wholly owned—decision-making units.

In both MOM and SOM, we face the need for *centralised* and/or *integrated* decision-making. Moreover, in MOM, the players/partners offer services to other competitors too. Further, each of the players will need to (and want to) optimise their own processes to maximise their own profits, subject to, naturally, satisfying their service level agreements with each of the players that they offer a service to. Not all of the information that the player has in its possession can (or will) be shared with all other supply chains that the player provides a service to. Similarly, not all information that is available (to all the competitors that the player provides services to), will be available to all players in such supply networks.

Hence, under both MOM and SOM, there is a substantial need for newer optimisation models that are able to provide good/satisfactory operational decisions and solutions to all the players in the supply network.

Initial attempts at supply chain coordination in the literature mainly focused on single operator models (SOMs). However, these models are becoming increasingly irrelevant in today's MOM operations. Thus, newer decision modelling approaches are needed for the modern supply chain.

An example from the manufacturing industry may be considered, where there are units (either under MOM or SOM) that undertake production, distribution, procurement and inventory/warehousing operations. Such supply chains may consist of many *levels*, where each level contains a decision-making unit (under MOM or SOM). There are a plethora of research articles on integrated models, especially, in such manufacturing supply chains (see, for example, [138, 146, 115, 13, 105]), where two/three-level coordination is often used.

However, the studies mentioned above include many assumptions and practical difficulties that render them inapplicable in MOMs. For example, players operate independently of the many supply chains that that may provide services to. Players/partners may operate in vastly different geographies, and serve other independent entities as well. Further, independent DMUs often deal with confidential competitive information that may not be shared with the supply chains that they provide services to. Also, each decision-making unit attempts to optimise its own operation and hence, may not be able to share fully, all the information with the many partner organisations.

The above complications are not necessarily present in SOMs where the parent entity owns each of the smaller DMUs—whether they operate a two/three-level coordination model or not. Hence, new studies are required to handle decision-making in integrated models where players operate through service contracts to many large parent entities.

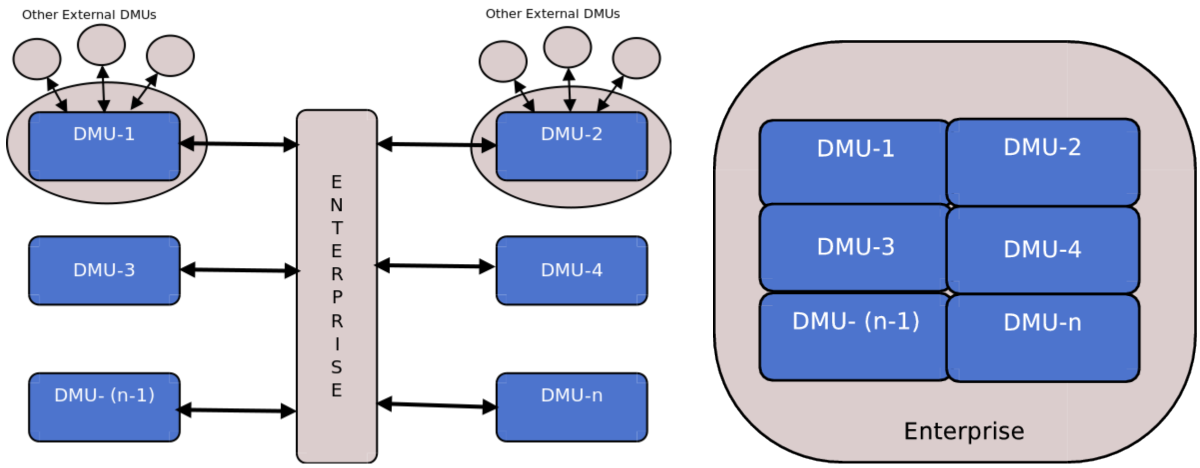
Even in SOMs where all competitive information is completely known and available, it is not often practical or possible to form a single/large centralised model to solve the variety of decision problems that arise. The scale of such problems is just too large for optimal solutions to be derived easily. This problem is, of course, further exacerbated in integrated models where, apart from the scale complexity we also face the issue of decentralisation of data.

Therefore there is a need for *decentralised decision models*, particularly in MOMs (but also in SOMs), in which there is a strong emphasis on coordination and collaboration. We formally define two ways of integrated decision-making,

CDM *centralised decision-making* where decisions are made by a single model.

DDM *decentralised decision-making* where decisions are made by different models. It can be termed as distributed decision-making also.

Figure 3.1 provides a representation of the decision-making models and the supply chain-relationship environment that they may be used in.



(a) Decentralised (b) Centralised

Figure 3.1: Decision making models

In decentralised supply chains, there are multiple operators with their own decision-making units. These DMUs need not be identical or share similar objectives. Information-sharing and coordination are necessary in such supply chains to improve the overall performance and to avoid unnecessary risks and buffers. Decentralised models have the following key characteristics:

(i) many stakeholders and players who may be geographically distributed;

(ii) players with conflicting objectives;

(iii) the presence of information asymmetry amongst players in the modelling and operational framework;

(iv) a negotiation mechanism that allows players to arrive at a decentralised solution that is, in some sense, optimal for all players [7, 116, 114, 112, 130].

Table 3.1 is a representation of how the above decision models operate within the operational models (SOM or MOM) that supply chains may adopt. The table does not imply that the availability of models corresponding to any combination is directly proportional to scale. For example, MOM-DDM combination is frequently observed in modern supply chains. However, the mathematical models to handle this particular combination are very less compared to the number of models available for the SOM-CDM combination. The ✓mark in Table 3.1 represents the chances of observing such combinations in reality. MOM-DDM problems are very commonly seen in almost all industries. If there are multiple independent operators, then the centralised decision making is not possible and vice versa (hence this is indicated by a '×' in Table 3.1). Therefore, in this thesis, we propose solution approaches for all combinations, except for MOM-CDM combination.

Table 3.1: Combinations of operational and decision models

	Decision Making	
Operational	Centralised (CDM)	Decentralised (DDM)
Single-operator (SOM)	✓	✓
Multi-operator (MOM)	×	✓✓

While decentralised decision-making models (DDMs) are applicable in both SOM as well as MOM operational models, CDMs are usually impractical in multi-operator operational models. This is because of issues of ownership, control and lack of perfect/full information. It has been pointed out in [105] that a comparison of the performance and use of both decision models is an under-explored area. The focus of our work, therefore, is strongly on the DDMs and the factors which influence them.

A round-the-clock supply chain might prefer a close-to-optimal solution in a reasonable time rather than an optimal solution after long hours. In most of the scheduling problems, the operator has to make the decision just before the planning horizon starts. In such situations, it might not be feasible to run a bigger and complex model. In the next section, we propose and develop an iterative-feedback solution approach to solve the integrated problem, which uses partial information-sharing and exhibits manageable computational complexity.

3.2 Modelling approaches

As we see in the literature, there are attempts to solve similar problems with many modelling approaches. Based on the classification, we restrict our study in the following three classes.

An integrated model is a single decision-making model which includes all the constraints and objectives of all DMUs. We assume here that these are known, although this is often not the case in complex, multi-party supply chains. This can be modelled using an optimisation program (say) and solved using commercial solvers. Due to their size and complexity, integrated models are more difficult to handle when compared to the smaller sub-problems that could be formed and considered for each of the DMUs. Moreover, such a model requires complete information-sharing.

Decomposed models allow us to split the integrated model into easily-solvable sub-problems. There are many popular decomposition techniques such as the Lagrangian relaxation (LR) and the Column generation (CG). The traditional implementation of these algorithms involves a central player/coordinator to update the bounds and the multipliers.

Decentralised models are inspired from the decomposed models which do not have any centralised player. The bounds are updated using decentralised methods. Compared to the previous approaches, the decentralised model requires only minimal information-sharing. The quality of solutions may not be good as compared to the solutions from the decomposed models.

Solutions from all these methods should not be compared based on the cost. The level of information needed for each model is not the same. The problem size and complexity are also different. A global supply chain consists of many independent decision-making units (DMUs) in different geographical locations and are 'naturally' decentralised in nature. The integration of decisions, other than being unrealistic, also makes the model too complex and large to solve efficiently. Therefore, it is important to study different modelling approaches and solution methods. The concern and awareness regarding information privacy and integrity are also growing. Hence, the challenges in SCC such as information-sharing, data security, real time synchronisation and the like, also should be addressed. Kovács et al. [87] recommend the use of decentralised decision making and a suitable coordination mechanism in a multi-party supply chain, with due consideration to information sharing, conflicting objectives and autonomy of the partners.

In this thesis, the integrated model (IM) is used as a hypothetical best alternative to benchmark the decomposed and decentralised approaches. Through the decomposed and decentralised methods we aim to achieve reasonably better solutions compared to the solutions from the "ideal" IM with minimal information sharing, and we compare their solutions.

The combinations of models represented in Table 3.1 are solved using different modelling techniques and solution approaches. This means that based on information availability and the SC structure, we need to customise our approaches. We suggest the following approaches to tackle different combinations of models in Table 3.1.

Table 3.2: Different modelling approaches for the models shown in Table 3.1

	Decision Making	
Operational	Centralised (CDM)	Decentralised (DDM)
Single-operator (SOM)	Integrated approach	Decomposed approaches - LR (Chapter 4) and CG (Chapter 5)
Multi-operator (MOM)	Not applicable	Decentralised approaches - LR (Chapter 6) and CG (Chapter 7)

Figure 3.2 shows the flow of the thesis, which address the supply chain coordination using different modelling approaches. It also shows the transition from the integrated approach to the decentralised approaches using decomposition techniques. We have developed two decomposition approaches (see Chapter 4 and Chapter 5) and two decentralised approaches (see Chapter 6 and Chapter 7) in this thesis.

3.3 Coordinated scheduling in coal supply chains

The modern coal supply chain is a prime example of an application area where coordination models are quite effective. From a situation in the past where a single operator would normally own the entire coal supply chain, through a process of divesting in a non-core business, today, different supply chains own different parts of the coal supply chain.

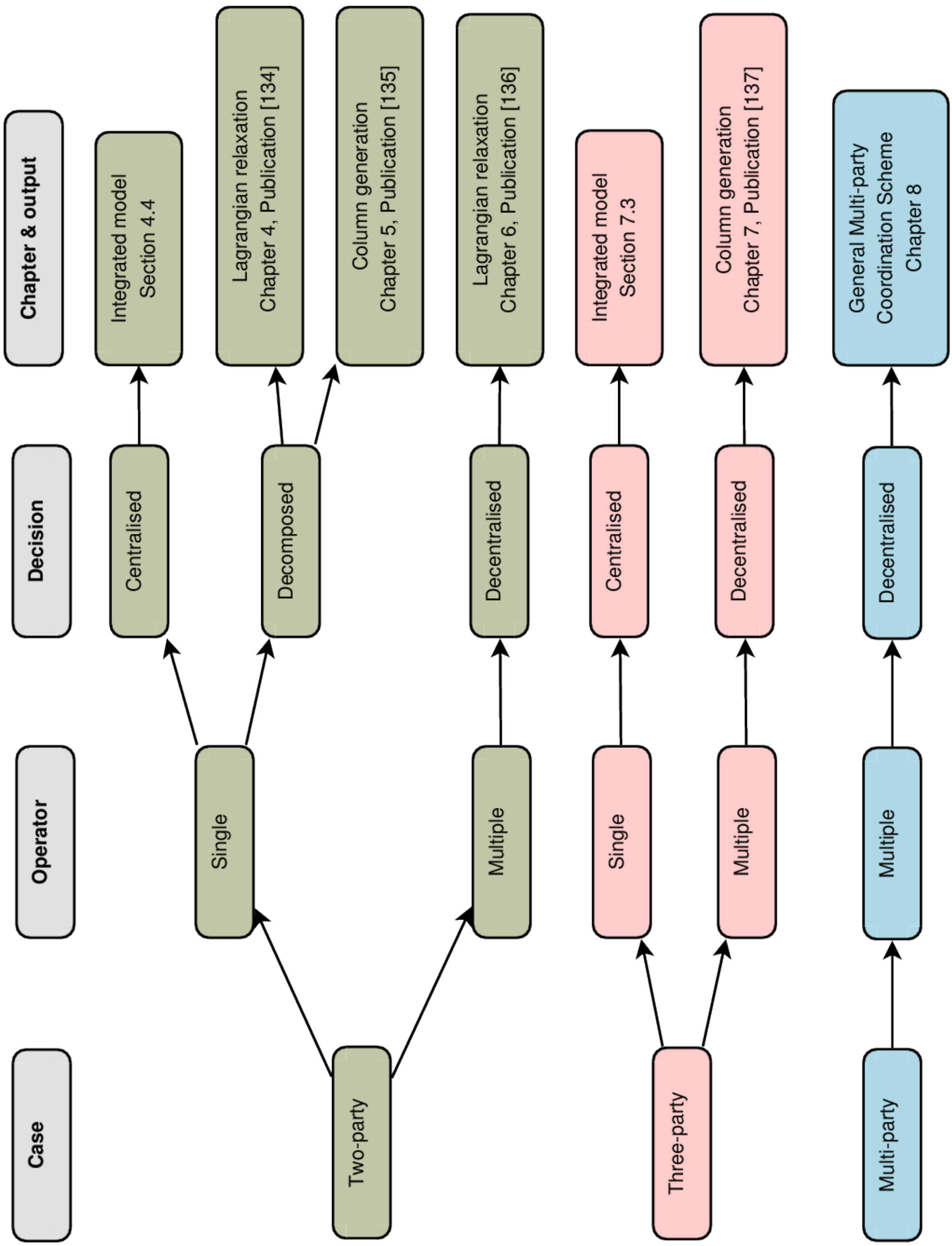


Figure 3.2: Schematic view of the thesis

The main players in this supply chain are *mines* and *mining units*, *ports*, *rail operators*, *track owners* and *terminal.* Coal normally ends up at the *terminals* or *ports* where it is loaded onto ships that set sail to their destination markets. Coal, which is mined at the various mines located at hinterlands, is usually transported by rail, (a small percentage is transported by road too), subject to track and train availability.

Each *player* in the supply chain has some control over (only) certain parts of the supply chain. For example, mines have control over their production rates and production capacities, which then determine the amount of coal that will be mined. Rail operators decide how many trains will be sent to each mine. The track authority provides and maintains the tracks that the trains use. The port operators control all operations within the port. The terminal decides which ship to load—when and from which coal stockpile. While some of the information (capacities, rates)—are known to all players in this supply network, not all information is completely shared. The result is a complex network of mines and port(s) connected by a network of track in which coal-movement logistics is facilitated by a rail operator.

This results in a need for coordinated production planning and logistics scheduling for coal SCs. Initially, we examine a coordination problem of multiple production units (*mines*) and a common resource manager (a *rail operator*) motivated by the coal supply chain of Australia. Later this is extended with one more resource manager (a terminal). The rail operator and the terminal are common resources shared amongst all the mines. Note that, the mines belong to different mining companies and operate independently of each other.

This problem is partially extended from the resource constraint scheduling problem proposed in [120] and [123]. The authors looked at a special case of resource constrained scheduling where there are multiple processors who want to execute jobs that each need a certain amount of an additional resource. A limited amount of this resource is provided by the central resource manager. Resource constrained scheduling problems have been the subject of research for more than 40 years. One of the first publications that used the term was the PhD thesis of Johnson [80]. A comprehensive survey of models and solution methods can be found in [30]. By considering the rail operator and the terminal as common resources, our coordination problem can be viewed as a multi-resource constrained scheduling problem.

Here, we consider only one terminal (port), one common rail operator and multiple mines, noting that a subset of these mines *may* be owned by a single mining unit. Usually, each mine (or mining unit) has long-term contracts with external buyers of coal. Based on these contracts and based on an expected shipping schedule, the mines are able to enter into shorter term contracts with the port/terminal based on the ships that are expected to arrive at the port within a *planning period*. The terminal computes the demand for each mine, based on coal type, ship arrival dates and priorities and passes these on to mines as a pair of *due-dates* and *quantities*. Therefore, within a planning period, based on the number of shipments expected and based on the total *parcels* that are needed, the mines may receive more than one set of requests (due-date and quantity) from the terminal. Note here, that the terminal is an independent operator and supplies such requests to all the mines that it services. Moreover, the mines do not know the set of requests that other mines receive.

The rail operator is another independent service provider. They too have contractual obligations with all the mines, to rail a certain amount of cargo within each planning horizon. Based on the amount of cargo moved, each rail operator makes a certain *profit*. The rail operator has a pool of trains which are categorised into different classes depending on train capacity. The rail operator, who links all the mines, often acts a coordinator to schedule the mines' production.

The terminal knows the order quantity from the ship and the expected date of arrival. On the basis of ship orders and past experience, the ship orders are split into smaller mine-orders. The terminal is a resource manager like the rail operator which connects all the mines. On the basis of actual coal delivery, the terminal is expected to prepare a loading schedule for the ship.

Figure 3.3 depicts a typical coal supply chain. M1, M2, . . . represent independent mining units, which consist of multiple mines. For simplicity, hereafter, we use the word 'mine' to refer the corresponding mining unit. Each mine has no information about other mines

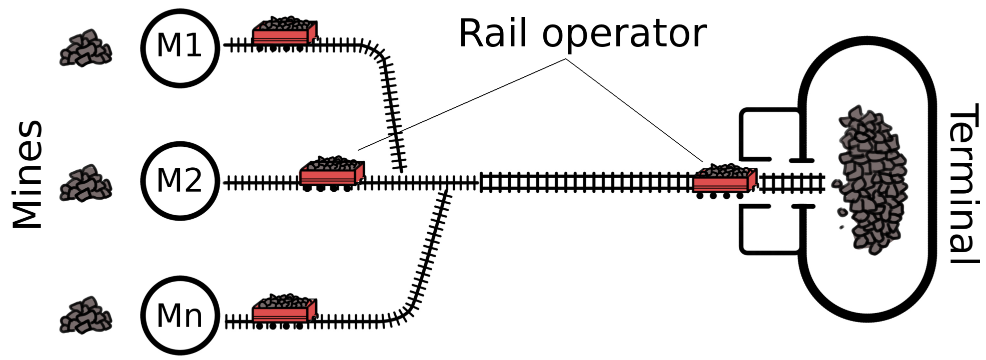


Figure 3.3: Schematic diagram of a coal supply network

and has partial information about the rail operator. At the same time, the rail operator does not know the details of each mine's costs or production capacity.

When each mine carries out production planning at the beginning of the planning period, it does not care about the specific trains that will arrive to take the mined coal away to the terminal—simply because such information cannot be determined at that time. However, at the production planning stage, each mine considers a train class (or, capacity) that it requires at various points in time in the planning horizon—this would be typically based on the 'requests' received from the port operator. At this stage, the rail operator merely announces/posts the different classes of trains that it will provide and the properties of these classes (mainly, speed and haulage capacity). At this stage, there is no information available on the actual number of trains of each class or the number of mines with whom the rail operator has contracts with—this information is not shared. Similarly, each mine's production capacity, long-term shipping contracts, port-requests and production costs are only known to the individual mines.

Each train (class) has specific properties such as journey time, loading time and load/capacity. The travelling time depends on the location of the mine and the train class. Each train trip has four stages: (*a*) empty travel from garage to mine, (*b*) loading at the mine, (*c*) transit from mine to terminal with the full train-load of coal, and (*d*) unloading at the terminal and return journey to the garage. Since this is a planning model, we do not, in our approach, allow for waiting time or queuing time between stages or during any stage. If a train is idle and un-utilised, it can only be because there is no *job* that requires the train. If this is the case, we assume that the train stays idle at the garage. Initially, we assume that the terminal has sufficient parking space and inventory holding space and so, the focus will be on the mines and the rail operator.

Some of important assumptions taken in thesis are listed below.

1. It is assumed that all trains are fully loaded at the mines. Partial loading or mixing loads from different mines are not permitted due to operational reasons. For example, in Australia, mine sites can be almost 400kms* away from the ports. Distances between mines and terminals are vast, coal trains are two to four kilometres long, and the transport of material requires excessive labour and fuel costs. Therefore, in practical instances, 'partially-loaded' trains are extremely rare.

2. Track and rail operator are one and the same. We assume that the track layout is sufficient to allow overtaking and passing. Track maintenance is also not considered.

3. There is no break allowed in between the train journey. Once it starts from the garage, it will go to a mine, loads, travel back to the terminal, unload and comes to the garage.

4. Every order received at the mines has to be satisfied before the due-date of next order. In other words, at most only one order can be delayed for any mine at any

*https://www.hvccc.com.au/AboutUs visited on 18-Jan-2014.

time since the ships cannot be delayed too much.

5. The mines do not interact with each other.

6. The terminal receives a set of orders from the ship. Based on these ship-orders, the terminal computes and conveys the production orders to the mines. Once these orders are placed, alterations or cancellations are not allowed.

7. No stochasticity in the supply chain.

8. No fixed cost for any of the operators. The variable costs are considered in the objectives.

9. The basic time unit is an hour and the planning horizon is typically a week. This assumption is taken only for the random instances used for the computational experiment.

10. We assume that the DMUs in the supply network are truthful. Whatever the information they share is true. In a decentralised environment, DMUs might not be share the complete information. Even in that case, they will not broadcast any false information.

We refer to a model involving a set of independent mines and a common rail operator as a *two-party model*. If the terminal is also considered along with the mines and the rail operator then it is referred as a *three-party model*. Equivalently, in a two-party model, a set of independent producers are linked with a shared resource and in the three-party case, these producers are linked with two shared resources.

We have tried to explore similar coordination problems in India. However, Indian Railway is the only rail service operates in India. They do not have a dedicated network for freight movement. They operate it along with their passenger segment. Hence the scenarios are completely different in India and Australia. However, the proposed approaches can be tested on a multiple producer-distributor supply chain in India or abroad. Due to time limitation, we have not included in this thesis.

3.3.1 Two-party coal supply chains

Table 3.3 summarises the information-sharing between the mines and the rail operator in a two-party model. There is no explicit information-sharing between the mines; however, they can infer it through the common rail operator.

Table 3.3: Information availability in a two-party coal supply chain

Information	**Individual Mine**	**Rail operator**
Production capacity	Yes	*No*
Holding and Demurrage costs	Yes	*No*
Demand at terminal	Yes	*No*
Train (class) capacity	Yes	Yes
Journey time	Yes	Yes
Number of trains	*No*	Yes
Number of mines	*No*	Yes
Running cost	*No*	Yes
Tardiness/Idle time costs	*No*	Yes

The production scheduling model for each mine will include constraints such as inventory balancing, production capacity and demand request satisfaction. Backlogs are not permitted at the mines. In other words, we assume that the full train quantity has to be present and available at the mine prior to the loading of a train. It is assumed, therefore, that a request for a train is made only when there is enough coal ready for railing at the mine. Partial train-loads and incomplete orders are not allowed.

It is possible for the mines to over-produce. Such over-production is penalised through inventory holding costs per unit of coal applied at the mine. Similarly, it is possible for the mines to over-rail coal into the terminal in order to meet requests ahead of time. Such over-railing is penalised through the application of inventory holding costs per unit of coal at the terminal too. Finally, and most importantly perhaps, late delivery and late satisfaction of a request at the terminal is penalised. Such late deliveries imply that a ship is waiting in the terminal/port without sufficient coal being available. This attracts a *demurrage cost*. Note that demurrage cost is a fixed cost that is applied for the whole order, irrespective of the amount of coal that a shipment is short by. However, all other holding costs are computed based on the quantity of coal.

It is conceivable that a centralised model may be applicable in such an environment. Such a model would have multiple objectives such as minimising total costs for mines and/or minimising the total tardiness (or idle time) of trains. Such models are invariably complex and intractable because of the sheer scale of problems—in terms of variables and constraints. The complexity of the integration process, sometimes forces researchers to use heuristic or meta-heuristic approaches to solve the resulting centralised optimisation problem (see [24, 25, 49, 74]). However, as explained before, given the variety and complexity of the operational scenario, we have to apply a coordinated modelling approach in this case.

In such a coordinated modelling approach, each mine optimises its individual production plan and creates a schedule of train classes at different times during the planning horizon. These are referred to as *jobs*. Each job has a time at which it is ready for collection or satisfaction by a train. This is known as its *ready time*. A mine may have several such jobs over the duration of the planning horizon. The rail operator combines all of these jobs from all the mines and then assigns specific trains to satisfy each job, on or after its ready time.

This is a clearly coordinated process. Each mine carries out its production scheduling on the basis of certain assumptions regarding the availability of train classes and without regard to the job requests created by other mines with which it is in competition for the same resource. The train operator combines these job requests and arrives at a railing schedule in order to maximise its rolling stock utilisation, without regard to the due-date for satisfaction at the port. The resulting job-schedule may be sub-optimal for one or more mines because of unwanted holding or demurrage costs.

Through an iterative procedure, however, it is possible for us to get to a solution whereby all players converge to a point where no further improvement is possible. While the final solution is unlikely to be globally optimal, it might represent a Pareto optimal solution; this is the goal of this decentralised and iterative procedure. Figure 3.4 depicts this iterative process.

Our problem has two major components: (a) production scheduling at each of the mines and, (b) a resource (train) scheduling for the rail operator. Each model's input-output variables and parameters are depicted in Figure 3.4. The first decision unit represents all the mining units in this system. These can be solved in parallel. Production plans for each mine and train requests are the outputs of this unit. All requests from the mines are combined, at an intermediate stage—depicted by the symbol $\bigoplus$ in Figure 3.4, before feeding it as an input to the rail operator decision unit. The weight of each job, which reflects its importance, is also computed at this stage. A higher job weight reflects higher importance. The rail operator finds an optimal schedule of jobs, which is feasible for each of the mines with respect to their production constraints. The *delay* of a job is defined as the difference between the actual completion time and due-date of that particular job. Feedback parameters are train availability and *delay* in request satisfaction for each mine. These are determined from the previous iteration's train schedule at an intermediate step, marked in Figure 3.4 by $\bigotimes$.

The objective of each mine is to minimise the total cost of inventory holding, order placing

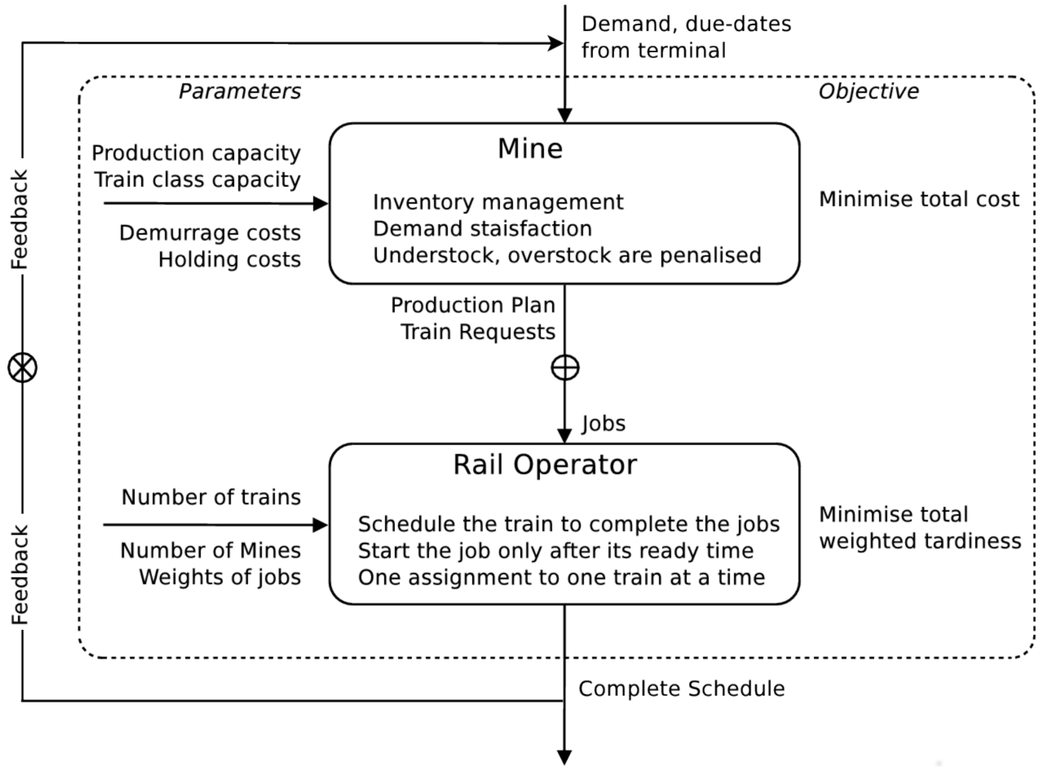


Figure 3.4: Schematic diagram of a two-party coordination approach

and demurrage. At any iteration, every mine considers journey time, loading time, delay in request satisfaction and train availability, in production planning. The rail operator's objective is to minimise the total weighted tardiness of all jobs (requests).

This inter-dependence results in two primary types of *railing conflicts* from the rail operators' point of view: excessive requests or under-utilisation. The latter case results in loss of profits, which is not a desirable outcome for the rail operator. The former case results either in an infeasible schedule or in delivery delays for the mines. In our approach we do not allow an infeasible schedule. Hence, all railing conflicts are assumed to result in delivery delays, which are undesirable from the point of view of the mines. Note that some of these delivery delays might even result in demurrage penalties being levied.

If, for any mine, there are no delays, then, the schedule is considered to be satisfactory (and perhaps even optimum) for the mine. However, if there is a delay in request satisfaction of even one job, the mine has to re-adjust its set of requests and, simultaneously, its production plan. This re-adjustment may mean that the mine requests for a different class of train or a different time slot for the delayed job. Note that this adjustment is necessitated by the fact that the mines do not know the requests of the other mines. Note also that, at the time of specifying requests, mines only specify the train class and not a specific train that they would like.

Most of the times, the mines will prefer the most economical train class with respect to capacity and turn-around time. Due to this, some train classes will be under utilised. Hence, the rail operator also broadcasts the availability of such train classes to the mines. Clearly, this is also a mechanism for the rail operator to overcome its under-utilisation conflict. Hence, the mines may readjust their production and requests, to avoid the delay with frequently requested train classes.

Figure 3.5 shows a general framework for the decomposed and decentralised approaches. At the start of iteration-1, each mine assumes that the rail operator has sufficient number of trains to meet their requests. In step S0, we initialise the feedback parameters to zero.

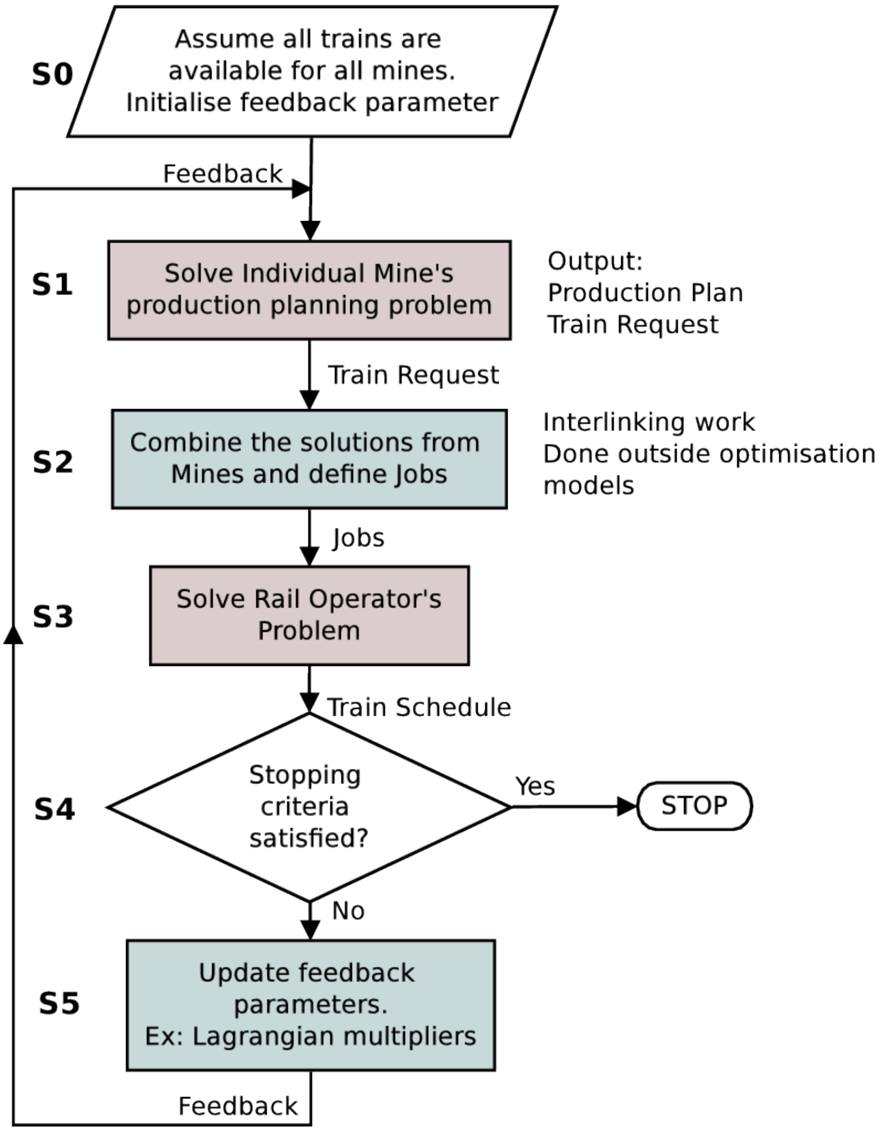


Figure 3.5: Flowchart for coordinated planning and scheduling

Step S1 and S3 solve the decision models for the mines and the rail operator, respectively.

The intermediate steps, S2 and S5 correspond to $\bigoplus$ and $\bigotimes$ in Figure 3.4 respectively. Step S2, prepares a list of jobs, each with a ready time, due-date, and weight. As we mentioned earlier, each mine receives and processes the information on train classes and not specific trains. A particular train class is available only when at least one train in that class is free or available. In step S5, feedback parameters are computed. It might vary based on underlying decomposition techniques.

The output of S1, train requests, may not be feasible for the rail operator to satisfy without any delay. However, if the feedback is provided to the mine, then it may be able to adjust its production schedule to make requests for another free time period and/or train class. If a mine readjusts a previously delayed request to an available time period or train class, then the time period in which it was scheduled previously can be used for other requests (either from the mine under consideration, or other mines). Note that, feedback from the mines is on train class availabilities. Thus, in the following iteration, it is likely that all mines might, simultaneously, adjust their schedules to take advantage of the availabilities that were previously announced as part of the feedback. Meanwhile the output of S3, train schedule, is feasible for all mines and optimal for the rail operator. Each mine's actual cost can be computed from the train schedule. As mentioned above, the actual cost and the optimal cost (objective value) will not be same unless there is no delay in request satisfaction.

3.3.2 Uniqueness of the problem

In this problem, each mine receives multiple orders from the terminal. Each order can be met by a different combination of train classes. In each class, multiple choices are available. There is no one-to-one correspondence between the orders and the train trips. Mostly, more than one train trip is required to meet a particular order. At the same time, a train load, especially the last trip for an order, can cater to more than one order. The due-date of each job is dependent on the order's due-date. However, the mapping between jobs and orders is part of the decision to be made. This indefinite nature of the allocation rules out the chances of modelling this problem as a batch scheduling/makespan minimisation problem. If the order quantity is not a multiple of the train size, then the mines have to either allow overstock at the terminal or pay demurrage. Since, there is penalty for overstock and demurrage; the mines need to consider the earliness and tardiness of all jobs simultaneously. On top of all these constraints, resource constraint makes this problem interesting and challenging. Thus, the problem under consideration is unique in its nature.

3.3.3 Three-party coal supply chains

The two-party models are extended to three-party models by including one more resource manager. In the coal supply chain example, the three parties are the mines, the rail operator and the terminal. The terminal handles the coal arrival from the mines, ship arrival and ship loading. The coal arriving from the mines are stored in large stockpiles. Based on the ship arrival and its orders, the terminal decides how to fill these stockpiles. Then using stackers and reclaimers this coal is moved to required locations. The terminal operations include the handling of these machines, stockpile capacities, and ships.

In this case also, the operations/models of mines and rail operator are similar to the two party case. We assume that the terminal has information regarding the ship arrival at the terminal and their order quantities. The date of arrival is the expected arrival date. The terminal uses their experience and previous records to split it into orders for the mines (producers). We expect that the splitting of ship-order to mine-order is done outside our model. The major decisions combining the mines (producers), the rail operator (resource manager) and the terminal are: (a) production plan, (b) resource allocation, (c) resource schedule and (d) loading schedule for the ship. Table 3.4 summarises the information-sharing between the DMUs in a three-party coal supply chain.

Table 3.4: Information availability in a three-party coal supply chain

Information	**Individual Mine**	**Rail operator**	**Terminal**
Production capacity	Yes	*No*	*No*
Holding and demurrage costs	Yes	*No*	*No*
Demand at terminal	Yes	*No*	Yes
Train (class) capacity	Yes	Yes	Yes
Journey time	Yes	Yes	*No*
Number of trains	*No*	Yes	*No*
Number of mines	*No*	Yes	Yes
Tardiness, earliness and running cost	*No*	Yes	*No*
Terminal capacity	*No*	*No*	Yes
Ship arrival & orders	*No*	*No*	Yes
Ship loading time	*No*	*No*	Yes
Stockyard capacity	*No*	*No*	Yes

Figure 3.6 shows a schematic diagram of the three-party coal supply chain and the inter

connection between the DMUs. This diagram is useful in giving an broad feel of the approach. However, the actual implementation might deviate slightly based on the solution techniques. In a decentralised framework, these DMUs can be solved with distributed computing facilities. The rail operator is the single DMU directly interacts with the terminal and the mines. The rail operator and the terminal have to make similar job-scheduling decisions, but with different objectives. Hence, sometimes it might be logical to combine some components of these DMUs.

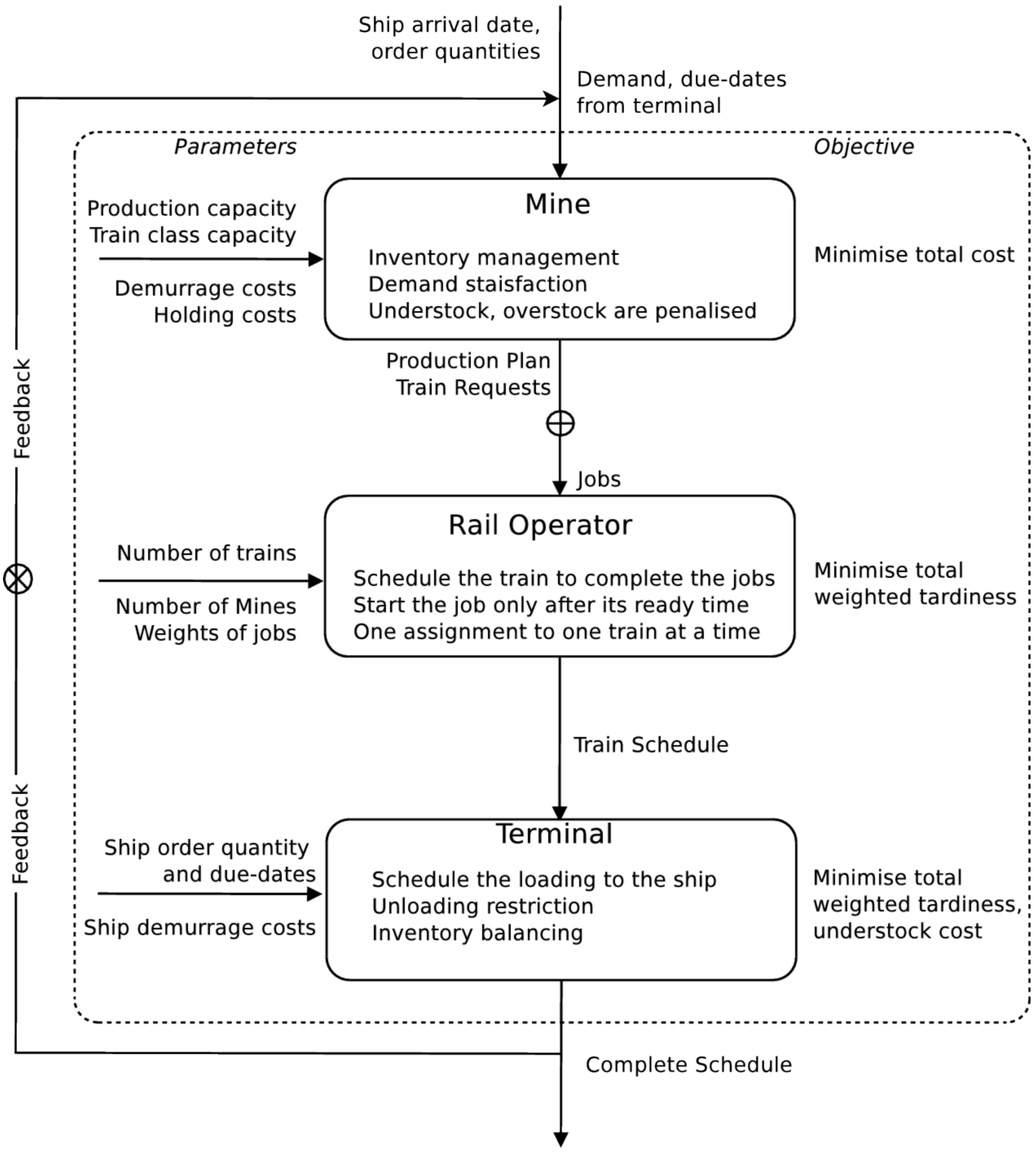


Figure 3.6: Schematic diagram of a three-party coal supply chain

The complexity of the model increases as one more resource manager is added to the two-party model. From a decentralised view, major activities of the DMUs in a three-party coal SC are listed as:

Terminal receives orders from the ship, split the ship-orders to mine-orders, and sent the mine-orders to the mines along with appropriate due-dates. The terminal also has to schedule the unloading at the terminal and the loading to the ships. The objective is to minimise inventory holding and under stock situations.

Mine receives the mine-orders from the terminal and places a request for the resources to the rail operator. The objective is to minimise total cost of production, inventory holding and the request placing.

Rail operator accepts the requests from the mines as jobs and schedules them subject to the resource availability. Either, the rail operator can incorporate the restrictions of the terminal in its job scheduling model, or the rail operator can send their schedule to the terminal for further refinement.

The three-party coal supply chain can be generalised to a manufacturing supply chain which has independent producers, a distributor and a warehouse. The problem structure is identical in both supply chains. Hence, the solution approaches proposed for the three-party coal SC can be extended to other SCs.

3.4 Generic problem description

The modelling approaches developed for the two-party and three-party coordination can be extended to a multi-party case. Consider a model where a set of independent decision-makers are sharing another set of common resources. From Figure 3.7, we can see that producers are independent of each other. Similarly the resource managers are also independent. Each resource acts as a link between the producers. Each decision maker (either resource manager or producer) has their own constraints and decision model. Further, assume that all these decision-makers require some portion of shared resources to complete their activities. For example, electricity, transport facility, computational resources, space, etc. can be considered as common resources. In other way, the constraints of each decision maker have some influence on the utilisation of the shared resources, which, in reality, is the independent decision of the resource managers. This forced dependency makes the decision-makers to coordinate amongst themselves to achieve a better position.

Even though the problem is abstracted for a resource constrained scheduling problem from the coal supply chain, it can be applied in similar resource-scheduling problems. For example, a similar situation is also observed in wine supply chains [124]. Each *grower* typically has contracts with many wineries. Each winery needs certain varieties and may stipulate, from a production point of view, that these varieties are presented to the

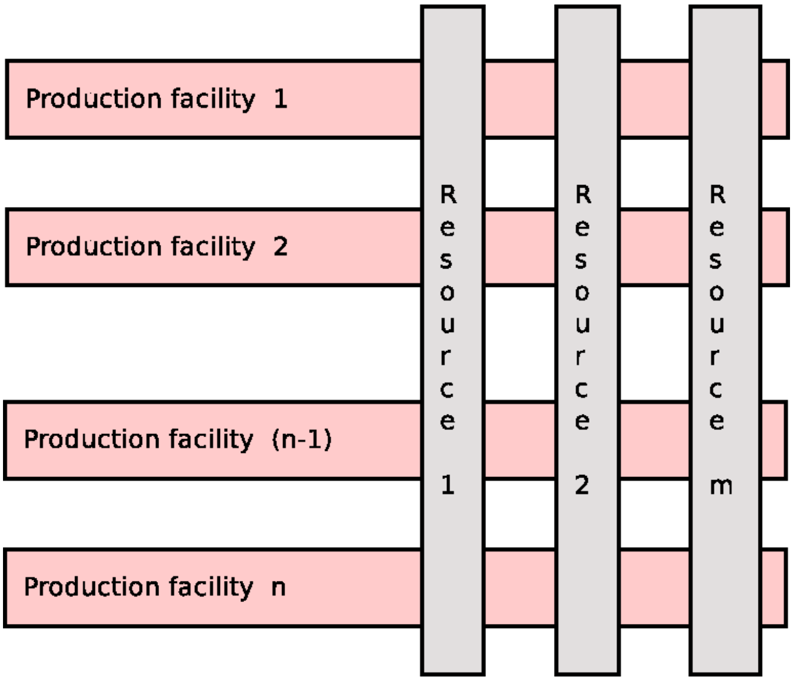


Figure 3.7: A pictorial representation of the generic problem

winery in a preferred sequence. The winery, therefore, needs to coordinate harvesting and transport to minimise wastage, waiting time (and hence, reduce perishability) and to maximise resource utilisation (of the crusher, fermenter, and the like) at the winery. Singh et al. [124] gives a detailed analysis of the coordinated scheduling problems that exist in the wine supply chains in Australia. Similarly, some of the coordination problems illustrated in Section 2.1 also can be formulated as RCSPs with a similar problem structure.

3.4.1 Mathematical representation

The scheduling problem discussed in Section 3.3 can be abstracted to this generic framework. A generic representation of this multi-resource constrained scheduling problem, which includes a group of producers and a group of inter-linking resource managers, can be defined as

$$\textbf{[IM]} \qquad \min \sum_i c_i^T x_i + \sum_j d_j^T x \tag{3.1}$$

$$\text{subject to} \quad A_i x_i \leq b_i \quad \forall\, i \quad (\text{Producer} - i) \tag{3.2}$$

$$R_j x \leq e_j \quad (\text{Resource manager} - j) \tag{3.3}$$

where $x = [x_1, x_2, \ldots, x_n]^T$.

In (3.4), constraints (3.2) and (3.3) are represented in a matrix form.

$$\begin{bmatrix} A_1 & & & & \\ & A_2 & & & \\ & & A_3 & & \\ & & & \ddots & \\ & & & & A_n \\ R_{1,1} & R_{2,1} & R_{3,1} & \dots & R_{n,1} \\ \vdots & \vdots & & & \vdots \\ R_{1,j} & R_{2,j} & R_{3,j} & \dots & R_{n,j} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \\ e_1 \\ \vdots \\ e_j \end{bmatrix} \tag{3.4}$$

where the variables $A, B, \dots$ are used only for representational purpose. It does not directly link with any of the model discussed later.

As we can see that, the matrix has a diagonal structure except in a few last rows. The diagonal entries represent the independent decisions of different DMUs in the supply chain. The last rows represent the resource managers who link these independent DMUs.

In other words, if we relax constraint (3.3), then we can solve the following n independent less complex subsystems.

$$\text{For all } i,\ \min\left(c_i^T x_i + \sum_j {d_j^i}^T x_i\right) \quad \text{subject to } A_i x_i \leq b_i, \quad \text{where } \sum_i {d_j^i}^T x_i \equiv d_j^T x \tag{3.5}$$

A two-party model has two groups of decision-makers. However, it has n producers, a resource manager in it. Therefore, the actual number of independent DMUs in a two-party model is $(n + 1)$. At the same time, it has only one set of linking constraints (in the form of (3.3)). In a three-party case, there are two common resources. For example in the coal supply chain, the terminal and the rail operator act as resource managers. This implies that, a three-party model has $n + 2$ DMUs and two sets of linking constraints. Similarly, multi-party coordination models also can be characterised. Therefore the problems considered in this supply chain are indeed multi-party resource constrained scheduling problem.

The model presented in (3.1) to (3.3) integrates all the independent decision-makers, we refer to it as *the integrated model*, in short, IM. The overall objective of the integrated model is to minimise total cost. Practical instances of such problems tend to be very large-sized. Such problems are complex because complete information (on costs and capacities, say) may not be available to a *central honest broker* that aims to solve the integrated model for all decision-makers. Such information is often competitive and confidential to each decision maker in the supply chain.

3.5 Conclusions

In this chapter, we have introduced coordination models and their importance. The models are classified based on their operational and decision-making aspects. Three major modelling approaches, to tackle supply chain coordination, are introduced in this chapter. The coordination problem in coal supply chains with a single shared resource and multiple resources are discussed with adequate details. The DMUs, information sharing between these DMUs and their interactions are also presented. A framework for distributed decision-making is proposed and the decision-flow in the process is explained with schematic diagrams.

This chapter concludes by proposing a structure of a generic problem and its representation in a matrix form. This means that, any resource constrained scheduling problem, with a similar representation, can be solved using the proposed solution approaches.

The next chapter discusses a decomposed modelling approach, based on LR, to solve the integrated problem which involves multiple independent mines and a common rail operator.

Chapter 4

A Decomposed Approach Based on Lagrangian Relaxation for Two-Party Coordination

The previous chapter explains different modelling approaches used in supply chain coordination. In this chapter, we introduce an integrated planning and scheduling problem motivated by a coal supply chain. In this problem, multiple independent producers need to coordinate with a resource manager to improve their performance. A decomposed solution approach is proposed by separating planning and scheduling decisions. The performance of the algorithm is improved with some effective techniques and supported with an heuristic algorithm to improve the upper bound. At last, the decomposed approach is compared with the integrated approach on a large group of randomly-generated datasets.

Section 3.3.1 provides an overview of the two-party coal supply chain. Even though we explain the two-party coal supply chain with n mines and a single rail operator, it can be generalised to a manufacturing supply chain with n producers and a resource manager. Therefore, we may use the nouns 'mine' and 'producer', 'train' and 'resource' or 'rail operator' and 'resource manager' interchangeably, based on the context. The nouns 'mine' and 'rail operator' are used whenever the example of the coal supply chain is referred to.

The integrated model includes all the constraints from the producers and the resource manager. Practical instances of such problems tend to be very large-sized. Such problems are complex because complete information (for example, on costs and capacities) may not be available to a *central honest broker* that aims to solve the integrated model for all decision-makers. Such information is often competitive in nature and restricted to the knowledge of each decision maker in the supply chain. Thus, for such problems, we need to move away from the use of a single integrated model that solves it; we propose a decomposition approach based on the popular Lagrangian relaxation.

Some parts of this chapter have been published as an article in the International Journal of Production Research [134]

4.1 Background

Grossmann and Biegler [66] review some important MILP decomposition techniques—Benders decomposition, Lagrangian relaxation (LR), cross-decomposition and bilevel decomposition—to deal with large scale optimisation problems. Some of these methods exploit the primal-dual relationship and others depend on the structure of the network. The critical issue in such problems, as highlighted by Grossmann and Biegler [66], is to identify the right decomposition strategy that balances computational gains with the cost of the time and the resources invested. Moon et al. [104] proposed an approach based on evolutionary search to solve a problem of integrated process planning and scheduling. The model considered resource selection and minimisation of makespan. The authors show the advantages of their integrated model over traditional models. Considering these approaches undertaken in the past, we decompose the integrated problem into many sub-problems by relaxing the linking constraint. The proposed iterative scheme is further strengthened with the addition of a few other recent techniques from literature.

Considering the decentralised nature of supply chains, we prefer to decompose the primal structure of the supply chain coordination problem. Cross-decomposition and Bender's decomposition exploit the dual structure of the problem. Therefore we use popular primal decomposition based on Lagrangian relaxation and Column generation techniques.

Lagrangian relaxation is a well-known and widely-used method for the decomposition of a large problem into many smaller, easily solvable problems. A detailed discussion on the primal-dual relationship, feasibility gap and other aspects of the Lagrangian method can be found in [150] and [56]. LR methods are strengthened with many heuristics and approximations developed on the characteristics of the problem. Early applications of the LR, and a systematic, theoretical description are also presented in [60]. Guignard [68] reviewed some of the interesting aspects of LR and the Lagrangian function. The article also discussed some schemes which can potentially improve the bounds obtained by LR.

Initially, the LR method assumed that the dual problem is differentiable. Later, it was extended to solve the non-differentiable dual problem and to compute a primal solution [12]. Barahona and Anbil [14] proposed an extension of the sub-gradient algorithm to update the Lagrangian multipliers. In Bahiense et al.'s [12] words, *"it produces primal as well as dual vectors by estimating the volume below the faces that are active at an optimal dual solution"*. Hence, it is known as the *Volume algorithm* (VA). Singh and Ernst [120] had used VA to strengthen an LR algorithm. VA helps the LR to converge quickly.

Wedelin [148] proposed an algorithm to accelerate the convergence of LR methods. Many interpretations are available for this algorithm. Mason's [102] interpretation and notations are popularly used in the literature. The original algorithm was proposed for a class of large scale 0-1 programming problems, where the constraint matrix consists of 'zero' and 'one' entries. Bastert et al. [17] proposed a generalisation and other improvements to it. This algorithm explores the dual-primal relationship and adds a minor perturbation to the multipliers. It favours variables with smaller reduced-cost to come to feasible region. At the same time, some other variables are discouraged to break the symmetry.

From the formulation given in Section 3.4, it can be observed that resource utilisation constraint (3.3) is the only constraint linking different decision-makers. In the absence of this constraint, the planning problem of individual producers can be separated. Therefore, in Section 4.5, we propose a decomposed optimisation method based on the LR algorithm to solve this problem. Section 4.5.1 explains the features of Lagrangian relaxation, such as, application of VA and WA. Section 4.6 provides an heuristic model and algorithm to compute better upper bounds for the LR scheme, and thus, to improve the speed of convergence. Another motivation for using a decomposed algorithm was, as results in Section 4.7 show, that in several instances CPLEX* (a commercial optimisation suite) was not able to find a single feasible solution even in one hour of CPU time for the above-mentioned model.

4.2 Decomposition approach

Motivated from the two-party coal supply chain presented in Section 3.3.1, let us consider a generic production-distribution supply chain, where n producers are linked with a distributor. Many resource constrained scheduling problems—especially those that involve multiple independent producers—are equivalent to the generic problem considered in Section 3.4. The special structure of the constraint matrix in (3.4) allows us to decompose the overall problem into one problem for each of the n producers *if* we ignore the resource class availability constraint (3.3). This creates n easily solvable production planning problems, provided we have one resource-scheduling problem linking the single resource manager and the (many) producers. Thus, we decompose the problem into two parts:

Production planning Each decision maker plans their production based on their priorities and their objective and places a set of requests to the resource manager for a certain number of resources. In other words, each producer defines a set of jobs with certain properties in order to meet their orders.

Resource scheduling After receiving the requests (jobs) from the producers, the resource manager prepares a schedule based on resource availability. A simple merge

*IBM ILOG CPLEX, url: http://www.ibm.com/software/integration/ optimization/cplex-optimizer/

of requests for the resource utilisation may not be globally feasible with respect to the resource constraint. This problem is equivalent to a job scheduling problem.

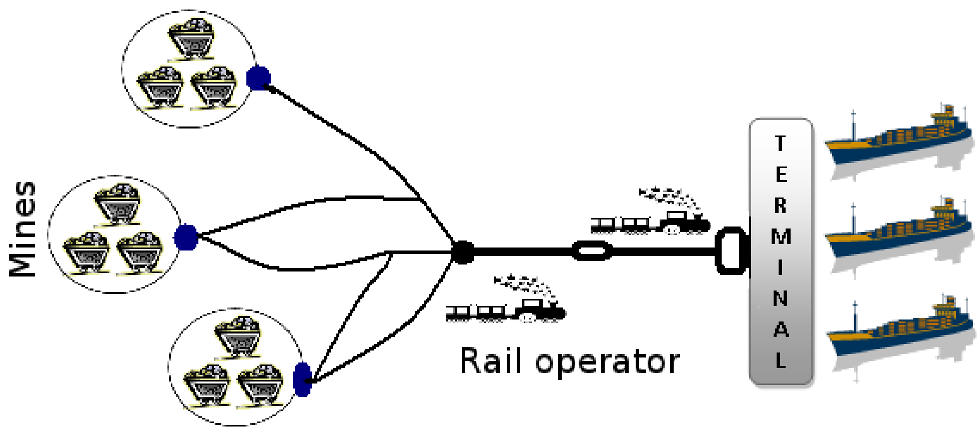


Figure 4.1: A schematic diagram of integrated production-planning and resource-scheduling

A feasible schedule needs to: (i) assign the right combination of resource-classes to producers, and (ii) create a schedule of resource allocations for the resource manager. Each train trip can be considered as a *job*. The orders that are given to the producers need not have one-to-one correspondence with jobs. Usually, more than one job is required to meet a complete order. It is also possible that one job might also satisfy two separate orders. This special order-job relationship means that the problem cannot be modelled as a batch-scheduling/makespan minimisation problem. The penalty for earliness and tardiness implies that the resources should be scheduled close to the due-dates. However, there are a finite number of trains in each class. Also, multiple producers make simultaneous requests for the same class of trains thereby making this problem difficult. What we have is a scheduling problem with different classes of resources; this can be treated as a multi-resource constrained scheduling problem.

There are two due-dates in this problem: (i) the due-date of an order associated with a producer, and (ii) the due-date of a job, which is an internal due-date, between the producer and the resource manager. Similarly, tardiness is also defined with respect to the job as well as the order. The due-date/tardiness of a producer and the resource manager are computed with respect to the order and the job, respectively.

In such problems many independent *decision-makers* (producers or service providers) need to coordinate with a *resource manager*, that provides resources such as trucks, trains, bandwidth or infrastructure to transport/transmit the product to destinations. In such supply chains, producers make planning decisions and the resource manager makes scheduling decisions that affect each of the independent producers. The problem is then one of ensuring minimal schedule-conflict at maximal efficiency.

4.3 Production planning model

We decompose the integrated problem formulated in (3.1) to (3.3) to solve the production planning and resource-scheduling separately. Irrespective of the algorithm, the production planning sub-problem has to be solved separately in all iterations. Therefore, first we describe the production planning sub-problem. We may use the nouns 'mine' and 'producer', 'train' and 'resource' or 'rail operator' and 'resource manager' interchangeably, based on the context. We have developed all the models discussed in the thesis. The article(s) which are used to derive a cut/constraint is cited whenever is required.

Let $i \in \{1, 2, \ldots, I\}$, $t \in \{0, 1, \ldots, T\}$, $u \in \{1, 2, \ldots, U\}$ and $w \in \{1, 2, \ldots, W\}$ be the indexes of producers, time periods, orders and train (resource) classes, respectively. Let $\mathcal{P}_i$ denote the production planning sub-problem of i^{th} mine. The solution space corresponding to $\mathcal{P}_i$ is denoted by $\mathcal{S}_i$.

Parameters

A_w^t cost incurred for requesting a train (resource), of class w, at time t

C^t tardiness /demurrage cost of an order at time t

O^t overstock cost per unit quantity, at the terminal, at time t

H^t inventory holding cost per unit quantity, at the mine, at time t

F_u due date of order u

Q_u u^{th} order quantity

D^t cumulative demand at time t

P production capacity

B inventory holding capacity at the mine

V_w capacity of the trains in class w

K_w number of trains in class w

R_w^0 forward travel time required for the trains in class w to reach a mine from the terminal

R_w^1 loading time required for the trains in class w

R_w^2 return travel time required for the trains in class w, from a mine to the terminal

L_w $= R_w^1 + R_w^2$, time required for loading and return travel to the terminal.

Decision variables

θ^t inventory level of the producer by the end of time t

σ^t amount produced in period t

τ^t over-stock level of the product at the terminal by the end of time t

ψ^t binary variable, equals to 1 if any order is tardy at time t. Zero indicates no order is tardy

η_w^t total number of trains from class w requested on or before time t.

A delivery order u from the terminal can be seen as an ordered pair of due date (F_u) and quantity (Q_u). Without loss of generality, we assume that $F_u < F_{u+1}$ and u^{th} order must be met before satisfying the demand for $(u+1)^{\text{th}}$ order. The cumulative demand for a producer at time t is defined as $D^t = \sum_{\{u|F_u \le t\}} Q_u$. For example, if producer i has three orders $\{(45, 10500), (89, 9300), (120, 11800)\}$, then the cumulative demand will be defined as:

$$D^t = \begin{cases} 0 & \text{if } 0 \le t < 45 \\ 10500 & \text{if } 45 \le t < 89 \\ 19800 & \text{if } 89 \le t < 120 \\ 31600 & \text{if } 120 \le t \end{cases}$$

Similarly, the demurrage cost, C^t, will be defined as:

$$C^t = \begin{cases} 0 & \text{if } 0 \le t < 45 \\ C_1 & \text{if } 45 \le t < 89 \\ C_2 & \text{if } 89 \le t < 120 \\ C_3 & \text{if } 120 \le t \end{cases},$$

where C_1, C_2, C_3 are constant.

Objective: The overall objective, Z_i, is to minimise the total system cost. This includes the cost of holding the inventory at the mines, the over-stock cost at the terminal, demurrage cost and total cost of requesting trains.

$$\min\ Z_i = \sum_t \left[\theta^t H^t + \tau^t O^t + \psi^t C^t + \sum_w (\eta_w^t - \eta_w^{t-1}) A_w^t \right] \tag{4.1}$$

Constraints: Equations (4.2) to (4.12) completely define the constraints of the model. Total number of resources requested on or before time t is non-decreasing.

$$\eta_w^t \ge \eta_w^{t-1} \qquad \forall\ w, t \tag{4.2}$$

The amount produced at any time t is not more than its production capacity.

$$\sigma^t \le P \qquad \forall\ t \tag{4.3}$$

The inventory balancing constraint: Inventory at time t can be computed by subtracting the supply at time t from the sum of the previous inventory and the production at time t.

$$\theta^t = \theta^{t-1} + \sigma^t - \sum_w (\eta_w^t - \eta_w^{t-1}) V_w \qquad \forall\ w, t \tag{4.4}$$

The inventory at the end of the period t cannot be more than the holding capacity of the producer.

$$\theta^t \leq B \qquad \forall\ t \tag{4.5}$$

If a resource of class w has to reach the terminal at time t, then it should reach the mine at $t - L_w$. Hence, the cumulative supply at time t can be expressed as $\sum_w \eta_w^{t-L_w} V_w$. If it is more than the producer's demand, then it has an overstock.

$$\sum_w \eta_w^{t-L_w} V_w - D^t \leq \tau^t \qquad \forall\ t \tag{4.6}$$

$(u-1)^{\text{th}}$ order must be delivered before the due-date of the u^{th} order, F_u. This also implies that, at any time t, only one order for a producer can be tardy at the most. In other words, cumulative supply at time F_u should be not less than the demand at time F_{u-1}.

$$\sum_w \eta_w^{F_u - L_w} V_w \geq D^{F_{u-1}} \qquad \forall\ u \tag{4.7}$$

If the cumulative supply at time t is less than its demand then the mine needs to pay a demurrage cost.

$$\psi^t \geq 1 - 1/D^t \times \sum_w \eta_w^{t-L_w} V_w \qquad \forall\ t \geq F_1 \tag{4.8}$$

All orders must be met by the end of the planning horizon.

$$\sum_w \eta_w^T V_w \geq D^T \tag{4.9}$$

Only one train can load at the mine at a given time period.

$$\sum_w (\eta_w^t - \eta_w^{t-R_w^1}) \leq 1 \qquad \forall\ t \tag{4.10}$$

The total number of resources of class w, being utilised at any time t, cannot exceed the total number of trains in that class.

$$(\eta_w^{t+R_w^0} - \eta_w^{t-L_w}) \leq K_w \qquad \forall\ w, t \tag{4.11}$$

Boundary conditions and scope.

$$\eta_w^0 = \sigma^0 = \theta^0 = \tau^0 = \psi_u^T = 0; \theta^t, \tau^t, \sigma^t \geq 0;\ \psi_u^t \in \{0,1\};\ \eta_w^t \in \mathbb{Z}^+. \tag{4.12}$$

Valid inequalities: The constraints mentioned above are necessary to represent the model. A few more additional integer cuts and bounds have been generated and used to tighten the Mixed Integer Programming (MIP) formulation. These valid inequalities are redundant in the MIP formulation, but, active in the linear programming relaxation of the MIP. Hence, an MIP solver should take less time to solve a problem that contains these additional constraints.

The cumulative supply at time t cannot be more than the total production. This means that:

$$\sum_w \eta_w^t V_w \leq tP \qquad \forall\ t$$

Due to holding costs at the mine and terminal, the producer does *not* want to supply more than D^T. Since (i) partial loading and order mixing are not permitted and (ii) D^T need not be a linear combination of train sizes, we need to allow a margin, $V_{\max}$, in the final cumulative supply. Hence, $\sum_w \eta_w^t V_w \leq D^T + V_{\max}$, where $V_{\max} = \max_w\{V_w\}$. From these two relations, we define an upper bound X^t for the cumulative supply.

$$\sum_w \eta_w^t V_w \leq X^t\ \forall\ t,\ \text{where}\ X^t = \min\{tP, D^T + V_{\max}\} \tag{4.13}$$

Since the supply is bounded, overstock also has to be bounded.

$$\tau^t \leq X^t - D^t \qquad \forall\ t \tag{4.14}$$

If a producer overstocks at the terminal then it cannot incur demurrage.

$$\psi^t \leq 1 - \tau^t/(X^t - D^t) \qquad \forall\ t \tag{4.15}$$

Constraint (4.13) is similar to a Knapsack constraint and therefore can be further tightened with the integer cuts proposed by Atamtürk [10]. Let Λ_w be the upper bound of η_w^t which satisfies (4.13) after fixing all other variables to zero. The set $C \subset \{1, 2, \ldots, N\}$ is a 'cover' if $\sum_{w\in C}(\Lambda_w V_w - X^t) \geq 0$. Then a stronger cut can be derived as

$$\sum_{w\in C}(\Lambda_w - \eta_w^t) \geq \lceil X^t/\overline{V}\ \rceil \quad \forall\ t \text{ where } \overline{V} = \max_{\{w\in C\}} V_w \text{ and } \Lambda_w = \lfloor X^t/V_w \rfloor. \tag{4.16}$$

Similarly we can use an integer cut from constraint (4.8). Since both ψ and η are integer variables, this cut will help us to improve the relaxed solution.

$$(1-\psi^t)\lceil D^t/V_{\min}\rceil \leq \sum_w \eta_w^{t-L_w}\lceil V_w/V_{\min}\rceil \qquad \forall\ u, F_u \leq t < F_{u+1} \tag{4.17}$$

$$\text{where } D^t = \sum_{\{u|F_u\leq t\}} Q_u \text{ and } V_{\min} = \min_w\{V_w\}.$$

The demand changes only at the due-date of orders. Once the demand for an order was met in an interval, it stays as 'met' till the due-date of the next order. This gives

$$\psi^t \leq \psi^{t-1} \qquad \forall\ u, F_u < t < F_{u+1}. \tag{4.18}$$

The constraints (4.1)-(4.18) are applicable for individual producers. Hence, the decision variables and the parameters corresponding to producer i are used with a subscript of i. For example P_i, H_i, η_{iw}^t, are used in the context of the integrated problem. Since the train is utilised across the mines, constraint (4.11) can be further tightened as:

$$\sum_i(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-L_w}) \leq K_w \qquad \forall\ w, t \tag{4.19}$$

4.4 The integrated model

An integrated model (IM) for the two-party case, similar to the generic representation discussed in Section 3.4, has a collection of all sub-problems with an additional linking constraint. It can be expressed as:

$$[\mathbf{IM}] \qquad \min \sum_i Z_i(s_i) \tag{4.20}$$

$$\text{subject to} \qquad \sum_i \chi_{iw}^t(s_i) \le K_w, \quad \forall\, w, t \tag{4.21}$$

$$s_i \in \mathcal{S}_i \quad \forall\, i \tag{4.22}$$

where s_i corresponds to a schedule (column) of each producer from the solution pool $\mathcal{S}_i$, which satisfies constraints (4.2) to (4.18). $\chi_{iw}^t(s_i)$ represents the number of active resources (trains) from the class w at time t for the producer i. Constraint (4.22) and constraint (4.19) represent the same resource constraint. The integrated model mentioned above is referred to as IM.

4.5 Lagrangian relaxation

Singh and Ernst [120] had used a Lagrangian relaxation-based algorithm strengthened with concepts of the Volume Algorithm [14]. The main advantage of the Volume algorithm is that it stabilises the sub-gradient method used in traditional Lagrangian relaxation schemes. Instead of directly using the current violations, Volume algorithm uses a convex combination with the previous violations. It helps the Lagrangian relaxation to converge quickly.

The complicated and interlinking train resource constraint (4.19) is relaxed as it allows us to decompose the integrated planning-scheduling problem for each of the independent producers, with a modified train request cost. The relaxed problem becomes,

$$\begin{aligned} Z(\lambda) = \min \sum_i \sum_t & \left[\theta_i^t H_i^t + \tau_i^t O_i^t + \psi_i^t C_i^t + \sum_w (\eta_{iw}^t - \eta_{iw}^{t-1}) A_{iw}^t \right] \\ & + \sum_w \sum_t \lambda_w^t \left(\sum_i \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-L_w} \right) - K_w \right) \end{aligned} \tag{4.23}$$

$$\textbf{subject to} \quad (4.2) - (4.10), (4.12) - (4.18).$$

where λ_w^t (≥ 0) is the Lagrangian multiplier that corresponds to the resource constraint at time t for the resource class w. Then the objective (4.1) for i^{th} producer can be updated as:

$$Z_i(\lambda) = \min \sum_t \left(\theta_i^t H_i^t + \tau_i^t O_i^t + \psi_i^t C_i^t + \sum_w (\eta_{iw}^t - \eta_{iw}^{t-1}) \bar{A}_{iw}^t \right) \tag{4.24}$$

$$\text{where } \bar{A}_{iw}^t = A_{iw}^t + \sum_{u=t-R_w^0}^{t+L_w-1} \lambda_w^u$$

Note that, even though the resource constraint (4.19) is relaxed, a disaggregated version of the constraint can still be added in this relaxed formulation for individual producers (see constraint (4.11)). This ensures that the requests placed by the individual producers will be feasible with respect to the overall resource constraint.

Then the sub-problem of i^{th} producer can be stated as

$$[\mathcal{P}_i] \qquad \left.\begin{array}{ll} \min Z_i(\lambda) & (4.24) \\ \text{subject to } (4.2) - (4.18). & \end{array}\right\} \tag{4.25}$$

In Algorithm 1, we present a modified version of the Lagrangian relaxation algorithm presented in [120]. It is strengthened with the Volume Algorithm [14] and an heuristic based on the Wedelin Algorithm [148].

The Lagrangian relaxation method based on sub-gradient optimisation exhibits slow convergence, especially when it has a small step size [142]. Hence, the following features were introduced to improve the convergence. In rest of the report, we refer to the Algorithm 1 as LR.

4.5.1 Features of the decomposed algorithm

The Lagrangian multiplier (λ) and the step (ρ) are critical parameters in Algorithm 1. Therefore, it is very important for the algorithm to manage them without getting trapped in the local optimum. Hence, we used the following techniques to control their variations and role:

Volume algorithm: The parameter $\gamma_{\max}$ in the Volume algorithm controls the influence of current violations in the Lagrangian algorithm and helps to minimise resource constraint violations. When $\gamma_{\max} = 1$ the algorithm converts to the traditional Lagrangian relaxation algorithm which only uses the violations in current iterations to update the multipliers. On the other hand, a $\gamma_{\max}$ value of 0.65 implies that a value of 65%, at the most, is given to the violations in the current iteration and the remaining value is given to violations over previous iterations. In every iteration, Volume algorithm picks the best γ which minimise the linear combination of violations in current and previous iterations. The

Algorithm 1 (LR) A decomposed algorithm based on Lagrangian relaxation

Initialise $k = 0, \rho = 0.1, LB^* = 0,\ UB^* = \infty,\ gap = \infty, rGap = -\infty,\ \nu = 0,$
    $\lambda^0 = (\lambda_w^{t(0)}) = 0,\ \lambda^* = \lambda^{(0)}, TL = 3600,$
    $S = (S_w^t) = 0,$ /* *History of violations* */
    $\mathcal{G} = \phi$ /* *Solution pool* */
    $C_l = 0,$ /* *Lower bound improvement counter* */
    $eTime = 0.$ /* *Elapsed time of the algorithm* */
while $(eTime < TL) \wedge (gap \geq 0.001)$ **do**
    Solve individual producer's production-planning problem $\mathcal{P}_i$, given in (4.25). Let $\{\hat{\eta}_{iw}^t\}\ \forall\ i, w, t$ be the solution from $\mathcal{P}_i$.
    Find the violations for each resource class w at time t, $\Phi_w^t = K_w - \sum_i \left(\hat{\eta}_{iw}^{t+R_w^0} - \hat{\eta}_{iw}^{t-L_w}\right).$
    Set the lower bound, $LB^{(k)} = \sum_i Z_i(\lambda^{(k)}) - \sum_w \left(K_w \cdot \lambda_w^{(k)}\right).$
    Compute $UB^{(k)}$ using the Algorithm 3 (See Section 4.6).
    $UB^* = \min\{UB^*, UB^{(k)}\}$
    if $(LB^* < LB^{(k)})$ **then**
        $\lambda^* = \lambda^{(k)};\ LB^* = LB^{(k)};\ \rho = \min\{1.3\rho, 2\};\ C_l = 0$
    else
        $\rho = 0.9\rho;\ C_l = C_l + 1$
    if $(\rho < 0.01) \wedge (TL - eTime > 300) \wedge (rGap \neq gap)$ **then**
        $\lambda^{(k)} = \lambda^*;\ rGap = gap$
    if $(\rho < 0.005) \wedge (TL - eTime > 300)$ **then**
        $\rho = 2$
    if $(C_l \geq 10)$ **then**
        $\nu = 4;\ C_l = 0.$
    if $(\nu > 0)$ **then**
        $\gamma_{\max} = 0.9;\ \nu = \nu - 1$
    else
        $\gamma_{\max} = 0.65$
    for (each resource class w) **do**
        Let γ^* be the solution that minimises $||\gamma\Phi_w + (1-\gamma)S_w^t||$.
        Set $\gamma = \begin{cases} 1 & \text{if } k = 0 \\ \gamma_{\max}/10 & \text{if } \gamma^* < 0 \\ \gamma_{\max} & \text{if } \gamma^* \geq \gamma_{\max} \\ \gamma^* & \text{otherwise} \end{cases}$
        Update $S_w^t = \gamma\Phi_w + (1-\gamma)S_w^t$
        The Lagrangian multipliers are adjusted as
        $$\lambda_w^{t(k+1)} = \max\left\{0, \lambda_w^{t(k)} - \rho(UB^* - LB^*)\frac{S_w^t}{||S_w^t||^2}\right\}$$
        Use Wedelin based Algorithm 2 (see Section 4.5.1) to update the multipliers.
    $gap = (UB^* - LB^*)/UB^*$
    $k = k + 1$

original article describing the Volume Algorithm [14] does not provide clear direction for the selection of this parameter. The value $\gamma_{\max} = 0.65$ is suggested in [120].

Steps (16)-(25) of Algorithm 1 implement the Volume algorithm. The increase in $\gamma_{\max}$, mentioned in step (19) helps the algorithm to recover from a local optima by increasing the influence of violations in the current iteration. Once $\gamma_{\max}$ is increased to 0.9, the same value is used for the next four iterations and then reset to 0.65. The value of 0.9 for $\gamma_{\max}$ was selected based on preliminary experiments with different values ranging from 0.2 to 1.0.

Wedelin algorithm: We present Algorithm 2 inspired by the Wedelin Algorithm [148] to update the Lagrangian multipliers. Originally, a Lagrangian multiplier is defined for each resource class w at time t. To apply the customisation for each producer, we define λ_{iw}^t as the Lagrangian multiplier used for producer i's sub-problem. The output, λ_{iw}^t, of Algorithm 2 is used instead of λ_w^t in constraint (4.24).

Algorithm 2 A procedure based on Wedelin's algorithm to update the λ's

Input: $w, t, \Phi_w^t, \lambda_w^t, K_w, C_l$, $\{\hat{\eta}_{iw}^0, \ldots, \hat{\eta}_{iw}^T\}$ $\forall$ i. /* *The optimal solution from* $\mathcal{P}_i$ */
Output: λ_{iw}^t

Compute $\chi_{iw}^t = (\hat{\eta}_{iw}^{t+R_w^0} - \hat{\eta}_{iw}^{t-L_w}) \quad \forall\ i, t.$
/* χ_{iw}^t *tells whether a resource of class* w *is servicing for producer* i *at time* t */

Compute the perturbation interval $[a, b]$ based on the lower bound improvement counter, C_l. Where

$$[a,b] = \begin{cases} [0.1, 0.2] & \text{if } 10 \leq C_l < 20 \\ [0.2, 0.5] & \text{if } 20 \leq C_l < 30 \\ [0.5, 1.0] & \text{if } 30 \leq C_l \\ [0.0, 0.0] & \text{otherwise} \end{cases}$$

Initialise $\lambda_{iw}^t = \lambda_w^t$ for all producer i
if $(\Phi_w^t \geq 0) \vee (\Phi_w^{t-1} < 0) \vee (a = b = 0)$ **then**
    return /* *No modifications to the Lagrangian multipliers.* */
Compute the perturbation $\epsilon = 0.01(a - (b - a)\ \Phi_w^t / K_w)$
for (each producer i) **do**
    if $(\chi_{iw}^{t-1} > 0)$ **then**
        $\beta = (1 - \epsilon)$ /* *Favour producer* i */
    else
        $\beta = (1 + \epsilon)$ /* *Discourage producer* i */
    $t' = t$
    while $(\chi_{iw}^{t'} > 0)$ **do**
        $\lambda_{iw}^{t'} = \beta \lambda_w^{t'}$; $t' = t' + 1$

Resetting multipliers: It is possible for the Lagrangian algorithm to get stuck in a particular direction without any improvement in the lower bound. The algorithm tries to find a different and possibly better direction by reducing the step size. It is also possible that even step size reduction may not improve the bound. If this is the case, we reset the lambda ($\lambda^{(k)}$) to the best available lambda (λ^*) and assist it to search in a known descent direction (see steps 12 and 13 of Algorithm 1).

Resetting step size: In some cases, the step size may be too small to escape from the influence of the local optima. Therefore, we reset the step size to its maximum value if there are atleast 300 seconds of cpu time remaining for the Algorithm 1 (see steps 14 and 15).

4.6 Upper bound computation

Among other factors, the convergence of the Lagrangian Algorithm (1) mentioned above depends on the quality of the upper bound found in various iterations. The sub-problems from each of the iterations suggest a feasible production schedule and the expected resource combinations for individual producers. It is unlikely that a simple amalgamation of the individual producers' solutions 'as-is' will produce a globally feasible schedule. In this section, we, therefore, propose an MILP-based procedure to compute an upper bound using the information on production and resource combinations for individual producers. Note that the solution from producers' sub-problem is already feasible with respect to all but the resource constraint (4.19). Therefore, the aim of the MILP presented below is to find an optimal schedule which satisfies the resource constraint (4.19) with values of some of the decision variables from individual producers' sub-problems as fixed. Clearly, this gives an upper bound for the integrated model described in Section 4.4. Specifically, the MILP formulation considers the following values of decision variables from each producer as input,

- *Production schedule*, $\hat{\nu} = \{\hat{\nu}_i^0, \hat{\nu}_i^1, \ldots, \hat{\nu}_i^T\}$
- *Train class combinations*, $\hat{\eta} = \{\hat{\eta}_{iw_1}^T, \hat{\eta}_{iw_2}^T, \ldots, \hat{\eta}_{iw_N}^T\}$
- *Order demurrage*, $\hat{\psi} = \{\hat{\psi}_i^0, \hat{\psi}_i^1, \ldots, \hat{\psi}_i^T\}$

Since the number of trips with trains in a class for a producer is fixed, we can *a priori* calculate the number of round-trips to this producer from the terminal. We will call each of these trips a *job*. A job j can be represented as a pair (M_j, T_j), where M_j is the producer and T_j is the resource class associated with the job. Other properties of job j such as travel time, quantity delivered, processing time and loading time are inherited from the resource class, T_j. For example, the amount of coal delivered by a job j is $V_j = V_{T_j}$, and loading time of a job $L_j = L_{T_j}$. We also define the set $\mathcal{J}^i = \{j|M_j = i\}$ as the set of all jobs associated with producer i. We refer to this job-based model as **UBM** in the rest of the text. MILP formulation of UBM is presented below:

Define, $X_i^t = \sum_{t'=0}^{t} \hat{\nu}_i^{t'}$ is the cumulative production at the producer i at time t; and z_j^t is a binary variable equal to 1 if the resource for job j arrives at the terminal by time t. 0, otherwise.

The overall objective of this model is to minimise the total system cost. This includes the cost of inventory at the mines and the terminal, demurrage cost and total cost of requesting trains over all the producers. The objective of UBM and IM (4.1) are identical. The objective function is

$$[\textbf{UBM}] \qquad \min \sum_t \sum_i \left(\theta_i^t H_i^t + \tau_i^t O_i^t + \psi_i^t C_i^t + \sum_{j \in \mathcal{J}^i} (z_j^t - z_j^{t-1}) A_i^t \right) \tag{4.26}$$

subject to the constraints:

The inventory holding at mine i at time t is defined as,

$$\theta_i^t \;=\; X_i^t \;-\; \sum_{j \in \mathcal{J}^i} z_j^{t+L_j} V_j \quad \forall\, i, t. \tag{4.27}$$

Cumulative supply cannot be more than the production.

$$\sum_{j \in \mathcal{J}^i} z_j^{t+L_j} V_j \le X_i^t \qquad \forall\, i, t \tag{4.28}$$

An additional tightening integer cut can be derived from the previous constraint (4.28).

$$\sum_{j \in \mathcal{J}^i} z_j^{t+L_j} \left\lceil \frac{V_j}{V_{\min}} \right\rceil \le \left\lceil \frac{X_i^t}{V_{\min}} \right\rceil \qquad \forall\, i, t \tag{4.29}$$

where $V_{\min} = \min_{\{j \in \mathcal{J}^i\}} V_j$.

$$\tag{4.30}$$

If the cumulative supply by a producer i at time t is more than its demand then the producer has an over-stock. This constraint is equivalent to constraint (4.6) of the IM.

$$\tau_i^t \ge \sum_{j \in \mathcal{J}^i} z_j^t V_j - D_i^t \qquad \forall\, i, t \tag{4.31}$$

If the cumulative supply by a producer i at time t is less than its demand then the producer needs to pay demurrage cost.

$$\psi_i^t \ge 1 - \sum_{j \in \mathcal{J}^i} z_j^t V_j / D_i^t \qquad \forall\, i, t \tag{4.32}$$

Similar to (4.17), additional tightening integer cuts are derived from constraint (4.32).

$$(1 - \psi_i^t) \left\lceil \frac{D_i^t}{V_{\min}} \right\rceil \le \sum_{j \in \mathcal{J}^i} z_j^t \left\lceil \frac{V_j}{V_{\min}} \right\rceil \quad \forall\, i, u, F_{i,u} \le t < F_{i,u+1} \tag{4.33}$$

Constraints (4.15) and (4.18) are also can be strengthened similarly.

An order must be satisfied before the due-date of the next order.

$$\sum_{j \in \mathcal{J}^i} z_j^{F_{i,u}} V_j \geq D_i^{F_{i,u-1}} \qquad \forall\, i, u \tag{4.34}$$

Once the job is completed it stays as completed.

$$z_j^t \geq z_j^{t-1} \qquad \forall\, j, t \tag{4.35}$$

All jobs must be completed before the end of the horizon.

$$z_j^T = 1 \qquad \forall\, j \tag{4.36}$$

Not more than one train can load at any mine.

$$\sum_{j \in \mathcal{J}^i} (z_j^{t+L_j} - z_j^{t+R_j^1}) \leq 1 \qquad \forall\, i, t \tag{4.37}$$

The total number of trains of a class w running at any time t cannot exceed its maximum availability.

$$\sum_{j | T_j = w} (z_j^{t+R_j^0+L_j} - z_j^t) \leq K_w \quad \forall\, w, t \tag{4.38}$$

If, for any two jobs j_1 and j_2, $M_{j_1} = M_{j_2}$ and $T_{j_1} = T_{j_2}$, then the two jobs will introduce symmetrical solutions in the model. We, therefore, introduce the following constraint to break this symmetry.

$$z_{j_2}^t \leq z_{j_1}^{t+L_{j_1}-L_{j_2}} \qquad \forall\, t, j_1 < j_2 \tag{4.39}$$

Note that constraint (4.39) will not affect the optimal solution. Indeed, if in the optimal solution for some t, $z_{j_2}^t > z_{j_1}^{t+L_{j_1}-L_{j_2}}$, then, given that the two jobs have the same train class and mine, we can easily relabel j_1 as j_2 and vice versa, without affecting the solution.

Let $\mathcal{J}' \subset \mathcal{J}^i$ such that

$$\sum_{j \in \mathcal{J}'} V_j \ < D_i^t \ \leq \sum_{j \in \mathcal{J}'} V_j + \overline{V}$$

where $\overline{V} = \max_{\{j \in \mathcal{J}^i \setminus \mathcal{J}'\}} V_j$. It implies that the total quantity delivered by the jobs in $\mathcal{J}'$ is not sufficient to meet the demand of D_i^t at time t. Therefore, to avoid the demurrage, at least one job should be done from the set $\mathcal{J}^i \setminus \mathcal{J}'$. Thus,

$$\sum_{j \in \mathcal{J}^i \setminus \mathcal{J}'} z_j^t + \psi_i^t \geq 1 \quad \forall\, i, t \tag{4.40}$$

Due to the symmetry-breaking constraint (4.39), constraint (4.40) can be further tightened. Let $\mathcal{J}'' \subseteq \mathcal{J}^i \setminus \mathcal{J}'$, such that it contains exactly one job from each train class in $\mathcal{J}^i \setminus \mathcal{J}'$ and among all the jobs of the same train class in $\mathcal{J}^i \setminus \mathcal{J}'$, $\mathcal{J}''$ contains the job with the smallest index. More formally, $\mathcal{J}''$ must satisfy the two following conditions:

1) For any two jobs $j_1, j_2 \in \mathcal{J}''$, $T_{j_1} \neq T_{j_2}$ whenever $j_1 \neq j_2$

2) For every $j \in \mathcal{J}^i \setminus \mathcal{J}'$, $\exists\, j_1 \in \mathcal{J}''$ such that $j_1 \leq j$ and $T_{j_1} = T_j$

Then the constraint (4.40) can be tightened as

$$\sum_{j \in \mathcal{J}''} z_j^t + \psi_i^t \geq 1 \quad \forall\ i, t \tag{4.41}$$

For example, consider $\mathcal{J}^i = \{1, 2, 3, 4, 5\}$ where $T_1 = T_2 = w_1$, $T_3 = T_4 = T_5 = w_2$, $V_{w_1} = 3000, V_{w_2} = 5000$ and $\mathcal{J}' = \{3\}$ defined for a demand of 6000 at t. Then, constraint (4.40) gives

$$z_1 + z_2 + z_4 + z_5 + \psi_i^t \geq 1. \tag{4.42}$$

The symmetry breaking constraint (4.39) enforces 1 to complete before 2 and 4 to complete before 5 and $\mathcal{J}'' = \{1, 4\}$. Hence, with respect to constraint (4.41), constraint (4.42) can be tightened as

$$z_1 + z_4 + \psi_i^t \geq 1.$$

If an order is tardy in a sub-problem then it will be definitely tardy in the integrated problem.

$$\psi_i^t = 1 \qquad \forall\ i, t \ \text{ if } \ \hat{\psi}_i^t = 1 \tag{4.43}$$

Boundary conditions and scope,

$$\theta_i^0 = 0, \tau_i^0 = 0, \psi_i^T = 0, z_j^0 = 0, z_j^t, \psi_i^t \in \{0, 1\}, \theta_i^t, \tau_i^t \geq 0 \tag{4.44}$$

The Algorithm 3 computes the upper bound for the integrated model in Algorithm 1. As the production schedule for every producer in UBM is fixed, it is possible that in the

Algorithm 3 An algorithm to compute the upper bound

Input: $k, \mathcal{G},\ (\hat{\nu}, \hat{\eta}, \hat{\psi})$ /* *Solution set* */
Output: $UB^{(k)}$

if $(\hat{\nu}, \hat{\eta}, \hat{\psi}) \in \mathcal{G}$ **then**
  return /* *This solution was previously evaluated* */
else
  $\mathcal{G} = \{(\hat{\nu}, \hat{\eta}, \hat{\psi})\} \cup \mathcal{G}$
if $(k = 0)$ **then**
  If needed, adjust the horizon T to accommodate all train trips to get a feasible solution.
Solve UBM described in Section 4.6 with a time limit of 120 seconds.
Evaluate the solution of UBM from previous step via the IM model. Let $UB^{(k)}$ be the objective value from this evaluation.

optimal solution of UBM, producers' production can be delayed to further reduce any inventory holding cost at the production unit. Therefore, at the end of this algorithm, we evaluate this solution using IM. We also improve the run-time of the algorithm by keeping the history, $\mathcal{G}$, of previous solution sets and avoid re-evaluation of a pre-existing solution set.

In the initial iterations, the gap between the optimal solution from the integrated model and the solution obtained from this model may be large. However, the LR scheme adjusts its production towards the optimal production schedule very quickly and hence the computed upper bound will become better.

4.7 Computational experiments

The decomposed optimisation model LR was compared with the integrated model IM via computational experiments on 240 randomly-generated instances. This section shows the properties of data instances, experimental settings and results.

4.7.1 Data generation

The randomly-generated instances were bundled in eight series with 30 instances per series. Each series represents a scenario with $5, 6, 7, 8, 9, 10, 12$ or 15 mines (producers). These series have the following common properties.

- The production capacity (P_i) of all producers is taken as 400 tonnes per period, where one period is an hour.
- The inventory capacity (B_i) at the mine is 20000 tonnes.
- Objective coefficients: $H_i^t = 1, F_i^t = 3, C_i^t = 50000, A_{iw}^t = 100$ for producer i and resource class w.
- For series with 5 and 6 producers, the resource (train) classes used are 3000, 5400 and 7200 tonnes. For every other series, a resource class of 8400 tonnes is used additionally.
- The loading hours (R_w^1) for resource classes $w =$ 3000, 5400, 7200 and 8400 is 1, 2, 3, and 4, respectively.

The above mentioned properties are common to all data series and the data instances. All instances in a series will have the same number of producers (I), trains (K_w) and train classes (N). Moreover, the forward (R_w^0) and return travel times (R_w^2) for trains in a class w are also constant. Table 4.1 summarises the main properties of each data series.

In addition, the following properties are generated randomly in every data instances,

- the number of orders, $\bar{N}_i \sim U(1, 4)$

Table 4.1: Properties of the data series

I	5	6	7	8	9	10	12	15
N	3	3	4	4	4	4	4	4
T	150	150	150	200	200	200	200	200
K_w	[2, 1, 1]	[1, 2, 1]	[3, 2, 1, 1]	[1, 2, 2, 1]	[3, 2, 1, 1]	[3, 2, 1, 2]	[3, 2, 2, 1]	[3, 2, 3, 2]
$R_w^0 = R_w^2$	[5, 6, 7]	[5, 6, 7]	[5, 6, 7, 7]	[5, 5, 7, 7]	[5, 5, 7, 7]	[5, 6, 7, 7]	[5, 5, 7, 7]	[5, 5, 6, 6]

- the order quantity, $Q_{i,u} \sim 5000 + 100U(0, 100)$
- the order due-dates, $F_{i,u} \sim F_{i,u-1} + 10 + U(0, T/\bar{N}_i)$

where $U(a, b)$ represents the uniform integer random number generated between a and b. The above settings result in an average of 2.5 orders per producer, an average of 10000 tonnes per order and about five trips per train. The experimental data and scripts are made available to the public through github. The url is `https://github.com/annuct/phd/`.

The order quantity and due-dates play a critical role in the complexity of the problem. Through uniformly-generated random inputs, we make sure that a broad spectrum of problem instances are covered for a series. Moreover different series present ample opportunities to study the scheme's performances with respect to the number of partners.

4.7.2 Experimental settings

IBM ILOG CPLEX† is a commercial tool for solving optimisation problems. It has the functionality to handle linear programming (LP), network flow, quadratic programming (QP), quadratically constrained programming (QCP) and mixed-integer programming (MIP) problems. CPLEX provides an additional facility to access its Application programming interface (API) through concert technology. Concert technology provides a set of C++, Java, and .NET class libraries offering an API that includes modelling facilities to allow the programmer to embed CPLEX optimisers in respective applications. We have developed a Java application to use CPLEX with concert technology.

All computational experiments were done with CPLEX 12.1 on a 64-bit server machine‡. In IM, CPLEX was terminated either at 0.1% of relative gap or with a CPU time limit of one hour, whichever came first. For a fair comparison, the same termination criteria were

†IBM ILOG CPLEX, url: http://www.ibm.com/software/integration/optimization/cplex-optimizer/
‡Configuration: 64-bit, 16 core, 2.93 GHz Intel Xeon(R) X7350 processor

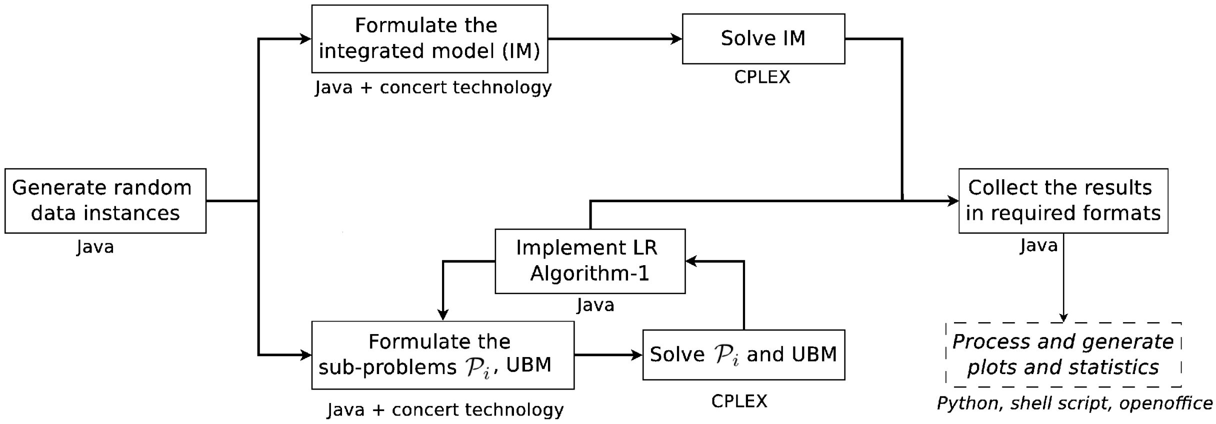


Figure 4.2: An outline of the experimental setup and tools used

also used for the LR method. In LR, UBM was terminated after 120 seconds or with a zero percentage relative gap.

The following CPLEX parameters were fixed for our experiments.

1. *IloCplex.DoubleParam.EpGap* = 0.01, acceptable relative objective gap
2. *IloCplex.DoubleParam.TiLim* = 3600, maximum CPU runtime in seconds
3. *IloCplex.IntParam.MIPEmphasis* = 4, emphasise finding hidden feasible solutions.

4.7.3 Performance measures

As the optimal objective value depends on randomly-generated demand and due-dates, it is not possible to compare the solution approaches by its objective function's face value. Therefore, the relative performance of the presented solution approaches is compared by Student's t-test. For lower bound comparison, the relative performance ratio is defined as

$$LBR(LR, IM) = \frac{(LB_{LR} - LB_{IM})}{\max(LB_{LR}, LB_{IM})}. \tag{4.45}$$

A positive lower bound ratio (LBR) means that LR is better as it has a higher lower bound. Similarly, for the upper bound ratio (UBR) is defined as

$$UBR(LR, IM) = \frac{(UB_{LR} - UB_{IM})}{\min(UB_{LR}, UB_{IM})}. \tag{4.46}$$

Here, a negative UBR means that LR is better as it has a lower upper bound.

In addition to these performance ratios, we compare LR and IM with their relative gap and runtime. In the following section, we have presented the overall results, series-wise results and group-wise results using confidence intervals, cumulative frequency plots, box plots (quantile plots) and summary tables.

4.7.4 Experimental results

Since over all the instances, the average order per producer and the average trips per train are about the same, results in this section will show that an increase in the number of producers alone makes this problem challenging. Indeed, as the number of producers increases, the system will have more overlapping due-dates and will be forced to make more numbers of trips in the same planning horizon. This makes it increasingly difficult for IM to optimally allocate train resources. The complexity explosion is also reflected in the results, since, in more than 40% of instances, IM failed to find even a feasible solution.

Figure 4.3 and Figure 4.4 show the estimated mean and 95% confidence interval from the Student's t-test for the performance ratios. Interval boundaries and the mean are marked with thick 'red' cross lines.

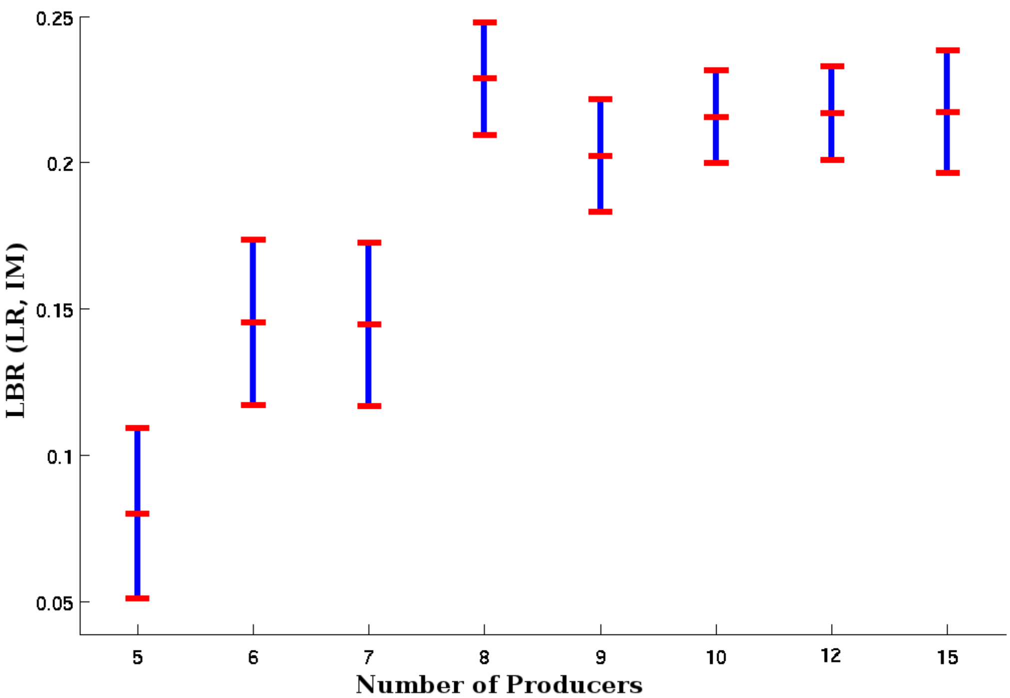


Figure 4.3: 95% confidence interval for the LBR (LR, IM)

Figure 4.3 and Figure 4.4 clearly demonstrate the dominance of LR scheme over IM and results can be summarised as follows:

1. The lower bounds obtained by the LR scheme were always at least 5% better than the lower bounds found by IM. The difference was much more significant in instances with a larger number of producers. For example, in instances with 10, 12 and 15 producers, the quality of the lower bounds found by LR was at least 20% better than that of IM.

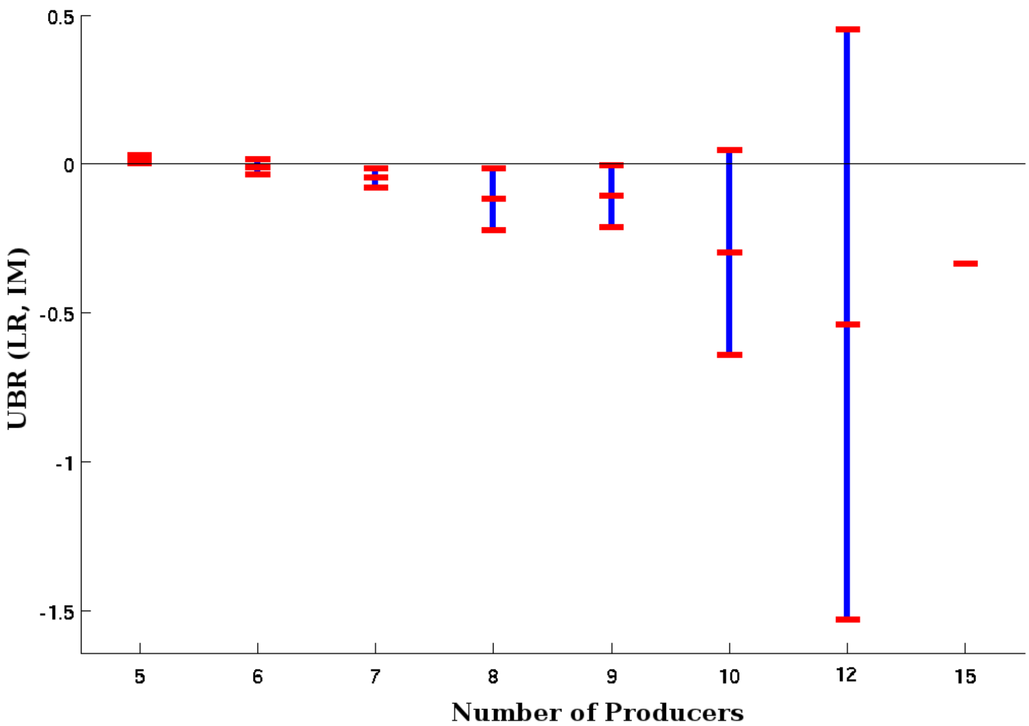


Figure 4.4: 95% confidence interval for the UBR (LR, IM)

2. The upper bound obtained by IM is only slightly better than the upper bound obtained from LR for the series with five producers. In this case, the difference is only marginal since a 95% confidence interval for the ratio is $[0.0007, 0.0293]$.

3. For instances with six or more producers, LR always found better upper bounds than IM. This difference was much more significant as the number of producers increased.

Figure 4.5 shows the overall cumulative frequency plot for both the ratios. Few interesting points, where the ratio hits at 5%, 10%, 90% and 95% are also marked. The LBR is positive in 95% of instances. This implies that the lower bound computed using LR is higher than the lower bound computed using IM. The UBR is less than 0.03 in 90% instances. Thus, the upper bound obtained from LR is either better or close to the bound obtained by IM. There are instances where the improvement in the upper bound is more than 35% and the improvement in the lower bound is more than 30%.

In every series, the lower bound ratio is computed from 30 observations. However, less than 11 observations were available for the upper bound ratio in the series with 9, 10, 12 and 15 producers. Therefore, based on IM's performance and to undertake a more detailed analysis of the results, we further classify the data instances into four groups.

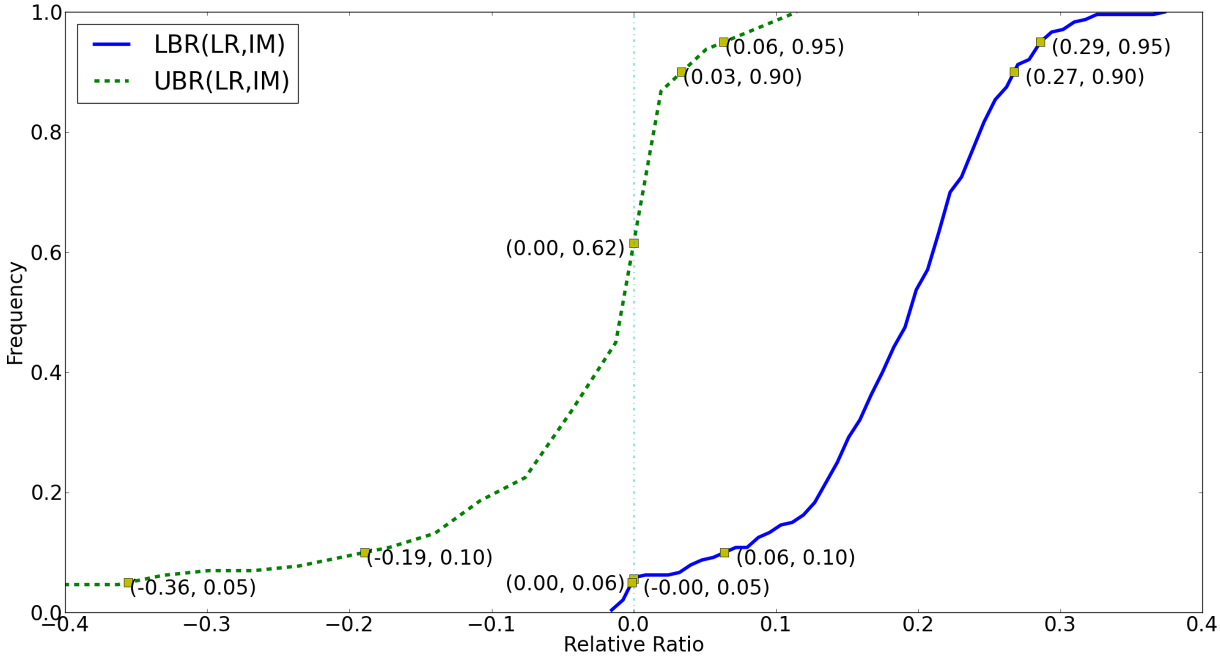


Figure 4.5: Cumulative frequency plot for LBR and UBR

Group	**Description**
Easy	Final gap obtained by the IM is less than 10%
Medium	Final gap obtained by the IM is between 10% and 20%
Hard	If the IM finds a feasible solution and the final gap is greater than 20%
Very Hard	IM could not find a feasible solution.

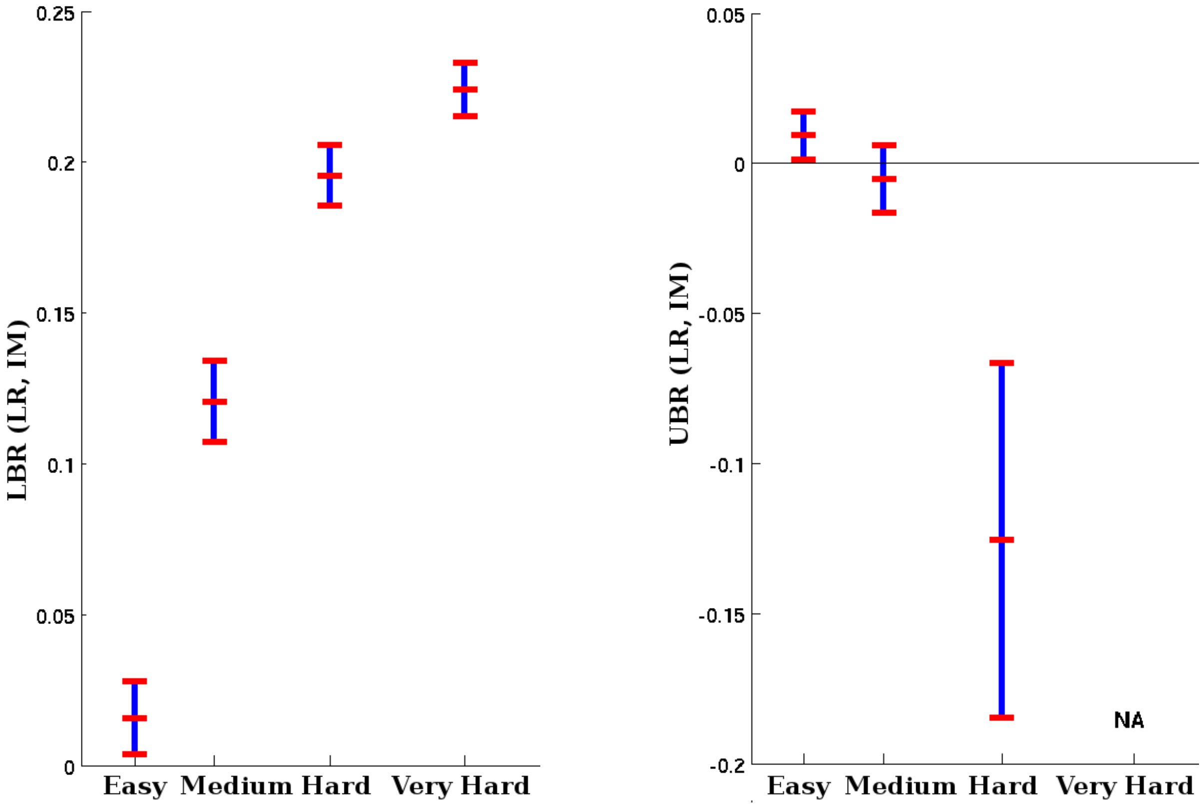
Table 4.2 summarise distribution of the instances and their properties in each group. The table clearly indicates that as the number of producers increases it becomes increasingly difficult for IM to find any feasible solution or to converge to a solution within a reasonable gap.

Table 4.2: Properties of the data groups

# producers (I)	5	6	7	8	9	10	12	15			
Group	Number of data instances								Total	§Orders	§Trips
Easy	14	5	5	–	–	–	–	–	24	10	18
Medium	7	8	6	5	2	1	1	–	30	14	26
Hard	9	17	17	13	8	7	3	1	75	16	30
Very Hard	–	–	2	12	20	22	26	29	111	22	41
Total	30	30	30	30	30	30	30	30	**240**		

§= Average number of, # = number of

The 95% confidence interval from Student's t-test for different groups is presented in Figure 4.6. Once again, the gap between IM and LR has visibly increased and demonstrates



(a) Lower bound

(b) Upper bound

Figure 4.6: 95% confidence interval for the relative difference for different groups

the advantage of decomposed optimisation scheme proposed in this chapter. The results show that:

1. The average improvement of LR over IM in the lower bound is 2, 12, 19 and 22% for the groups easy, medium, hard and very hard, respectively.
2. Only in 'easy' instances could IM find upper bounds better than the LR method; the difference was, however, only marginal.
3. The upper bound found by LR is on an average better by 12.5% in the 'hard' group—which includes almost 1/3 of instances.
4. In the 'very hard' group, IM could not find any feasible solution. Therefore, the upper bound ratio is not available. However, the LR scheme could find feasible solutions with a median gap of less than 10% in 1800 seconds (Figure 4.7(d)).

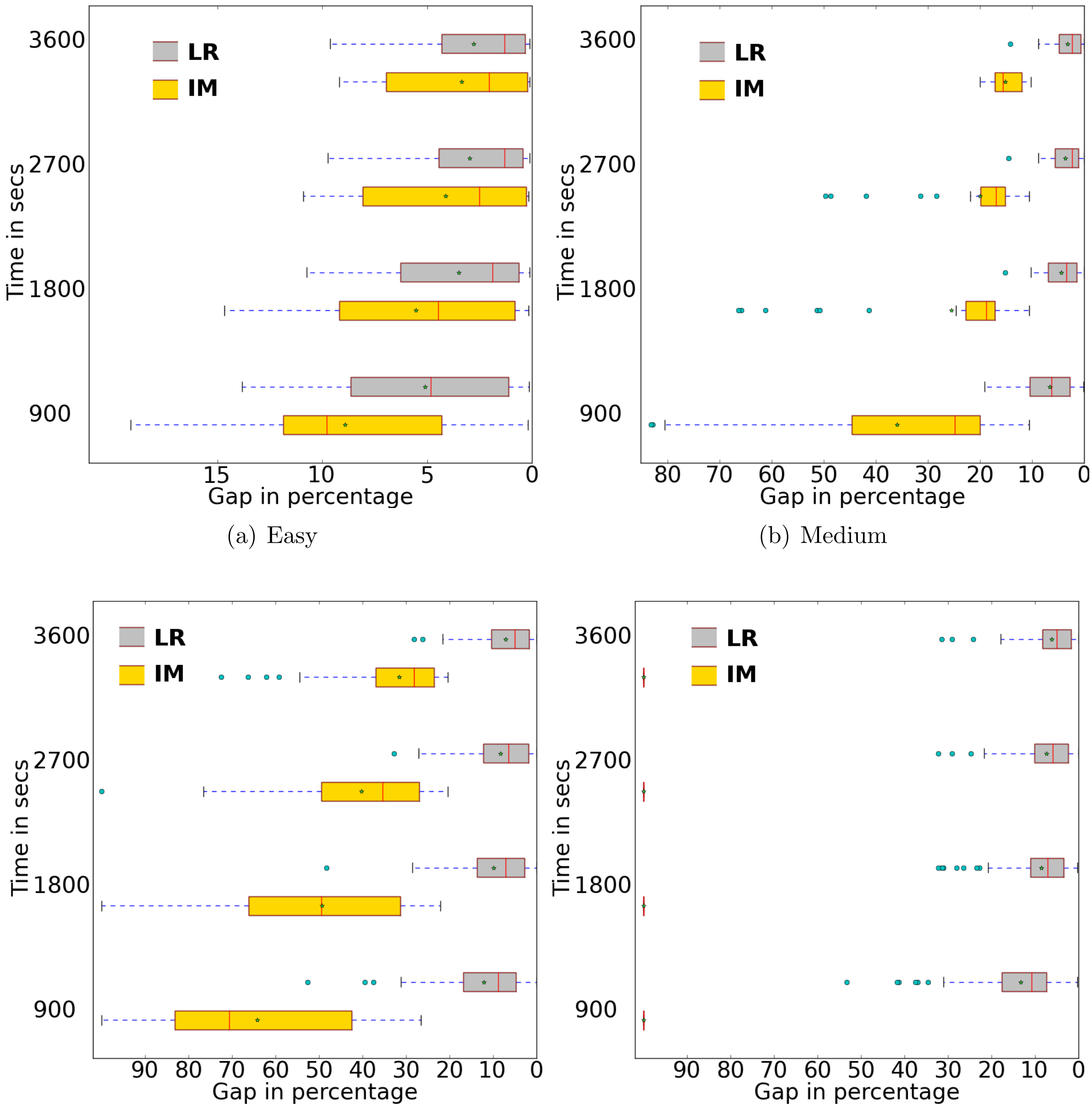
Simply comparing the final results does not completely convey the performance of both methods. Therefore, we also compare the *gap* obtained from LR and IM at four different time periods. The gap is defined as $(UB - LB)/UB$. Table 4.3 presents the median gap computed for each series. Figure 4.7 shows the median, 1/4 and 3/4 quartile of gap at 900, 1800, 2700 and 3600 seconds. The solid box represents the gap obtained from LR and the other box is for IM. The mean gap is marked with a black dot and outliers with a red 'plus' (+) symbol. In Table 4.3 and Figure 4.7(d), the gap of IM is taken as 1 (or 100%) in the absence of any feasible solution.

Table 4.3: Median of the relative gap in percentage

	Integrated model (IM)				Decomposed model (LR)			
	Time in seconds				Time in seconds			
# Producers	900	1800	2700	3600	900	1800	2700	3600
5	18.03	13.65	11.94	10.51	9.75	6.21	5.46	4.54
6	34.28	26.05	23.47	22.10	12.61	10.14	7.51	6.92
7	55.50	28.48	25.93	22.85	6.99	4.47	3.11	2.61
8	87.02	74.04	59.95	38.97	6.89	5.32	4.01	2.99
9	100	100	100	100	7.94	5.83	4.94	3.44
10	100	100	100	100	7.36	4.27	2.79	1.83
12	100	100	100	100	15.54	10.07	9.07	7.98
15	100	100	100	100	9.38	4.96	3.78	3.03
Easy	9.90	5.10	2.96	2.28	2.10	1.21	0.90	0.73
Medium	25.57	18.96	17.09	15.66	6.68	3.44	2.48	2.21
Hard	70.68	49.44	35.41	28.11	10.00	7.28	6.46	5.21
Very Hard	100	100	100	100	10.73	7.09	5.99	5.18
Overall	87.25	74.50	61.64	41.05	8.92	6.25	5.15	4.02

The following observations can be made from Figure 4.7 and Table 4.3:

1. The gap of LR is consistently better in all groups and series at all time periods. LR has less than 17% gap in 229 out of 240 instances.
2. For the series with more than nine producers, IM is intractable because the median gap is 100%.
3. For the series with 9, 10, 12 and 15 producers, LR's median gap is 3.44%, 1.83%, 7.98% and 3.03%, respectively. It shows the strength of LR in instances with a large number of producers.
4. LR gets feasible solutions very quickly. In 900 seconds, the median gap obtained from the LR is less than 7% in easy and medium groups and 11% in hard and very



(a) Easy (b) Medium

(c) Hard (d) Very Hard

Figure 4.7: Relative gap at different time points for different groups

hard groups.

5. During the period of 1800 to 3600 seconds, the reduction in the gap of LR is very minimal (on an average, 1% per 900 seconds). As part of ongoing research, stronger constraints, which will act as cuts, will be explored in order to improve the convergence of LR.

6. In Table 4.3, as the number of producers increases, the gap obtained from IM increases non-linearly. At the same time, LR shows stable behaviour, irrespective of this increase. This clearly demonstrates the strength of the decomposed algorithm over integrated models.

To summarise, LR has better lower bound in 99%, upper bound in 90% and gap in 98% of instances. In our scheme, the production plan and train combination computed for the lower bound, guides the upper bound. However, IM is benefited from independent branching carried out by the CPLEX solver for the lower bound and upper bound. The guided upper bound computation helps LR to bring down the gap very quickly. The median gap from 240 instances of IM is 41% and that of LR is 4%. In short, the above computational experiment confirms the effectiveness of the LR scheme to give solutions very quickly especially if the number of producers increases.

4.8 Conclusions

In this chapter, an integrated planning-scheduling problem motivated by the coal-supply chain in Australia is considered. The integrated production-planning and resource-scheduling problem is formulated. We propose a decomposition scheme to split the production planning and resource-scheduling decisions based on Lagrangian relaxation. Mathematical programming models for the sub-problems, additional constraints to improve the LP relaxation, Lagrangian function and the decomposed algorithm are also discussed in this chapter. The proposed algorithm is improved with the Volume algorithm and the Wedelin algorithm. The strength of the algorithm is demonstrated by comparing its performance with an MILP, via extensive computational experiments. The results show that the decomposed algorithm found significantly better lower and upper bounds than MILP.

This chapter shows that the LR-based decomposition is a powerful computational technique to solve the integrated problems which have a diagonal structure (see Section 3.4). The integrated models require all the DMUs to share complete information and make decisions simultaneously, by solving a single MILP. Therefore, the decomposed models cannot be directly applied on a decentralised supply network. In such situations, decomposed models are preferable and easy to execute. In this approach, individual decision-making units have the flexibility to make the decisions separately and the feedback is given through the Lagrangian multipliers. In the sense of information visibility and controls, both of these models are single-operator models.

At present, two major decisions, production planning and resource-scheduling, are considered in this model. This study can be extended by including other DMUs. The convergence of the distributed scheme will be further strengthened by improvements to Lagrangian relaxation, Volume algorithm and Wedelin algorithm. This algorithm can be extended by considering partial information sharing and multiple objectives for different DMUs.

Even though the LR algorithm performed better than the integrated model, the speed of convergence is low in larger problems. The convergence of the LR algorithm is slower since it does not have any history of solutions computed in any iteration other than the current one. Therefore, other decomposition algorithms such as column generation and Bender's decomposition can be attempted on the same framework.

Chapter 5

A Decomposed Approach Based on Column Generation for Two-Party Coordination

The previous chapter (and [134]) presents a decomposition algorithm based on Lagrangian relaxation (LR). It highlights the advantages of a decomposed decision-making approach over an integrated one. Even though the LR algorithm performed better than the integrated model, it had a slow convergence rate for larger problems. Hence, column generation (CG) algorithms are explored to solve the integrated problem. The main advantage of CG over LR is that while LR uses only one set of previous history, CG records all feasible columns from every iterations to construct globally feasible columns. Hence, CG converges faster than LR. A combination of multiple stabilisation methods is explored to improve the performance of the CG algorithm. We discuss and customise a few techniques that improve the computational efficiency of the traditional CG algorithm.

In this chapter, we develop an approach using a customised column generation (CG) scheme, to solve an integrated production planning and resource scheduling problem. In the production planning stage, each producer defines a partial allocation of resources. This means that resource allocation and its scheduling can be carried out sequentially. Thus, the original problem is decomposed into smaller solvable sub-problems to solve production planning and resource scheduling separately. A feasible production plan and corresponding resource allocation are considered together as a column and are stored in the solution pool. The master problem will then pick the best columns, from the solution pool, which minimise the overall cost—subject to the resource availability constraint. This method is computationally efficient compared to the Lagrangian relaxation method because it stores and manages multiple solutions at the same time.

Section 5.1 provides a summary of related work. Section 5.2 explains decomposition and the details of the column generation algorithm. In order to compute globally feasible solutions, two heuristics algorithms are added in Section 5.3. In Section 5.4, the basic CG implementation is strengthened with a few additional techniques. Section 5.5 gives

Some parts of this chapter have been published as an article in the European Journal of Operational Research [137] an overview of the datasets and performance measures. Section 5.6 summarises detailed results and comparisons of different modelling approaches.

5.1 Related work

Decomposition is an established idea to solve a large problem by splitting it into solvable sub-problems. It is ideal for the problems which has a group of independent decision blocks connected with a few linking constraints.

Dantzig and Wolfe [45] introduced an efficient decomposition algorithm for a special class of problems. Today, Dantzig-Wolfe decomposition (DWD) is a widely-applied solution approach in solving large optimisation problems. In a good number of research articles, CG-DWD is preferred as the solution technique. All of these show that the CG scheme is effective in solving large problems. Wentges [149] introduced a weighted DWD and presented some results on the convergence of this scheme under certain conditions. Pessoa et al. [110] discuss a stronger convergence of this algorithm. The major issues concerned with DWD are slow convergence and stabilisation of dual prices. Amor et al. [5] and Vanderbeck and Wolsey [143] analyse some of the latest (dual price) stabilisation techniques. Please read Desrosiers and Lübbecke [51] for an introduction to CG and its key features. Pochet and Wolsey [111] discuss different production planning models and MILP formulations. They suggest many practical reformulation techniques and polyhedral results to strengthen the model and solve it effectively.

5.2 Column generation scheme

The Dantzig-Wolfe decomposition (DWD) allows us to express all points in the feasible region of a linear program as a linear combination of extreme points and extreme rays. For large problems, it is impossible to identify *all* extreme points/rays. Therefore, in DWD, we start with one point and expand the collection of vertices iteratively. This chapter presents a customised column generation (CG) scheme to solve the two-party coordination problem, discussed in Chapter 4.

A CG algorithm has two components: *master problem* (MP) and sub-problems. The master problem (MP) is the train scheduling problem which does not have any production considerations. The production planning problem, described in Section 4.3, is considered in the sub-problems. The solutions (schedules) of each producer form the columns of the MP, from which a globally feasible solution is obtained. However, this MP can be very large. Therefore, we consider a restricted MP which has a smaller subset of the columns.

The sub-problems iteratively generate new columns for the MP using the dual prices from the restricted MP. The different solutions to the MP imply different allocations of trains to the mines.

5.2.1 Master problem

The master problem selects the best column for each producer which does not violate the resource availability constraint and minimises total system cost. Index c denotes a column.

Define,

V_{ic} = Value (cost) of column c of i^{th} producer. For each column c, there exists a solution in $\mathcal{S}_i$. Hence, V_{ic} can be computed with the objective function (4.1).

χ^t_{icw} = Number of servicing trains of class w for the i^{th} producer with respect to column c at time t.

x_{ic} is a binary variable, = 1 if the column c of producer i is used, 0 otherwise.

The parameters V_{ic} an χ^t_{icw} are precomputed for each column. x_{ic} is the only decision variable of the MP. Then, the master problem is

[MP]

$$\mathcal{V}_{MP} = \min \sum_{i,c} V_{ic} x_{ic} \tag{5.1}$$

$$\text{subject to} \quad \sum_{c} x_{ic} = 1 \qquad \forall\ i \tag{5.2}$$

$$\sum_{i,c} \chi^t_{icw} x_{ic} \le K_w \qquad \forall\ w, t \tag{5.3}$$

$$x_{ic} \in \{0, 1\} \tag{5.4}$$

The objective (5.1) minimises the total value of selected columns. Constraint (5.2) ensures that exactly one column will be selected for each producer. All the selected columns must satisfy the linking constraint (5.3). Each of the columns satisfies the corresponding producer's constraints. Therefore, the optimal solution to the master problem will be optimal for the system and feasible for all producers and the resource manager. In the relaxed master problem (RMP) which relax the integer constraint (5.4) is

[RMP]

$$\mathcal{V}_{MP} = \min \sum_{i,c} V_{ic} x_{ic} \tag{5.1}$$

$$\text{subject to} \quad (5.2), (5.3) \text{ and}$$

$$0 \le x_{ic} \le 1. \tag{5.5}$$

At an iteration k, the number of columns will be finite and restricted to a subset of all the feasible columns. The restricted relaxed master problem at iteration k is represented as RMP^k.

5.2.2 A decomposed algorithm based on the column generation

The column generation algorithm solves the relaxed master problem iteratively. Initially, it starts with a finite set of feasible columns and then, it iteratively adds more columns. The penalty for violating the resource constraint is communicated to the sub-problems through the dual prices computed from the master problem.

Let λ_w^t be the dual price of resource constraint for a class w at time t, then the sub-problems are solved with the modified LR objective function (4.24). Algorithm 4 illustrates a basic decomposed approach that is developed based on traditional iterative column generation schemes and DWD.

Algorithm 4 A basic column generation algorithm

Initialise $LB^* = 0, UB^* = \infty, k = 0, TL = 3600$secs, $\epsilon = 10^{-5}, eTime$=elapsed time.
Initialise solution pool $\mathcal{S}^0$ with a feasible solution (column).
while $(eTime < TL)$ **do**
    Solve the RMP^k with solution pool $\mathcal{S}^k$ and the corresponding dual variables be $\lambda^{(k)}$.
    For each producer i, solve $\mathcal{P}_i$ given in (4.25) with $\lambda^{(k)}$.
    Compute $LB^{(k)} = \sum_i Z_i(\lambda^{(k)}) - \sum_w \left(K_w \cdot \lambda_w^{(k)}\right)$.
    Update solution pool with all new columns.
    if $(LB^k > LB^*)$ **then**
        Set $LB^* = LB^k$.
    Call the heuristics presented in Section 5.3 to generate globally feasible columns.
    Remove columns which are non-basic and do not contribute to the optimal solution of RMP for ten continuous iterations.
    if $((\mathcal{V}_{RMP}(\mathcal{S}^k) - LB^*)/\mathcal{V}_{RMP}(\mathcal{S}^k) < \epsilon)$ **then**
        Evaluate the integer version of the master problem to identify the best-known upper bound UB^* and corresponding integer solution.
        break */* Prices are stable*/*
    $k = k + 1$
Report LB^*, UB^* and $gap = (UB^* - LB^*)/UB^*$.

The objective of the CG algorithm is to compute a better lower bound for the relaxed master problem. The lower bound from the IM and the CG will be the same if the integer relaxation of the problem was being solved. However, in this case, it is an MILP problem; therefore, the quality of the bound also depends on the cuts and how deep the branch and bound search is able to reach. The valid inequalities (4.12) to (4.18) are introduced to tighten the MILP formulation of $\mathcal{P}_i$. It helps us to achieve better bounds.

In every iteration k, LB^k is less than or equal to $\mathcal{V}_{RMP}$. Once both values are equal, sub-problems will not add any new columns to the solution pool. Therefore, the algorithm will not guarantee good integer solutions. Section 5.3 illustrates heuristic procedures to achieve a better upper bound by generating more numbers of globally feasible columns from the solutions of the RMP^k.

5.3 Heuristics to generate feasible columns

The sub-problem $\mathcal{P}_i$ given in (4.25) create columns for the i^{th} producer. It is unlikely that a simple amalgamation of these columns 'as-is' will produce a globally feasible solution. At the same time, the rate of convergence of this algorithm is dependent on quality of globally feasible solutions we have in the solution pool. Therefore, in this section, we propose two heuristics procedures to generate globally feasible columns.

(I) The *Job scheduling model*, where the rail operator attempts to derive a feasible solution to the problem with inputs from all the producers,

(II) The *Leader-follower mechanism*, where each producer attempts to make a better schedule after fixing certain columns (partial schedules) for all the other producers.

In these heuristics, importance is given to global feasibility rather than to cost minimisation. Heuristic (I) and Heuristic (II) receive inputs from the solutions of the RMP, which consists of many partial schedules for each producer. Since these heuristics are customised to generate globally feasible columns, it improves the upper bound UB^*.

5.3.1 Job scheduling model

A combination of the solutions from the sub-problems need not satisfy the resource constraint (5.3) of the master problem. Therefore, the rail operator uses the available information from the sub-problem solution and attempts to prepare a globally feasible schedule which minimises deviation from expected schedules. The rail operator, upon receiving the requests from the producers, is aware of the resource class combination and due-dates. Hence, the rail operator defines the train trips (jobs) required by each class at this stage. Each job has three stages: (i) forward journey from the terminal to the mine; (ii) loading; and (iii) return journey with full load to the terminal and unloading at the terminal. We define the additional variables required for the job scheduling model:

Parameters

j index of job

M_j mine associated with job j

T_j train class associated with job j

R_j^0 forward travel time required for the job j

R_j^1 loading time required for the job j

R_j^2 return travel time required for the job j

L_j $= R_j^1 + R_j^2$, time required for loading and return travel for the job j

F_j due date of job j

W_j^T tardiness weights of the job j, which reflect the importance of the job. Higher weight implies higher importance

W_j^E earliness weights of the job j

C_j^t weighted tardiness of job j at time t, $C_j^t = W_j^T \max\{t - F_j, 0\}$ for all j, t

G_j^t weighted earliness of job j at time t, $G_j^t = W_j^E \max\{F_j - t, 0\}$ for all j, t

K_w number of trains with class w

Decision Variables

$$z_j^t = \begin{cases} 1 & \text{if the job } j \text{ is completed on or before time } t \\ 0 & \text{otherwise.} \end{cases}$$

Each job j can be represented as a pair of the producer, M_j, and the corresponding resource class, T_j, that is, $job(j) = (M_j, T_j)$. The properties such as R_j^k, L_j are inherited from the corresponding resource class. In other words, $L_j = L_{T_j}$ and so on. Suppose a producer makes a request for a train from the class w at time t', then $\eta_w^{t'} - \eta_w^{t'-1} = 1$. Therefore, the corresponding job is expected to arrive at time t'. Hence, its due date is $F_j = t' + L_j$.

The weights W_j^E, W_j^T and costs C_j^t, G_j^t are decided by the resource manager. In this section, the properties such as due-date, earliness, tardiness and weight are discussed only with respect to the jobs. These properties differ from similar ones in the orders received from the terminal by the mines.

Objective: The objective of this job scheduling model is to minimise the total weighted tardiness and earliness. This guides the model to schedule the jobs as close to their due-dates as possible. If the earliness cost is greater than the tardiness cost, jobs will be advanced only if it is required to maintain feasibility.

$$\min Z_R = \sum_j \sum_{t>0} \left[C_j^t(z_j^t - z_j^{t-1}) + G_j^t(z_j^t - z_j^{t-1}) \right] \tag{5.6}$$

Constraints: Equations (5.7)-(5.13) denote the constraints of the model.
The cumulative supply from any producer should be less than its total possible production.

$$\sum_{j|M_j=i} z_j^{t+L_j} V_j \leq tP_i \qquad \forall\, i, t \tag{5.7}$$

At any time, there should not be more than one job for loading at any production location.

$$\sum_{j|M_j=i} (z_j^{t+L_j} - z_j^{t+R_j^2}) \leq 1 \quad \forall\, i, t \tag{5.8}$$

Once a job is completed, it stays as completed.

$$z_j^t \geq z_j^{t-1} \quad \forall\, j, t \tag{5.9}$$

All jobs must be completed at the end of the planning horizon.

$$z_j^T = 1 \quad \forall\, j \in \mathcal{J} \tag{5.10}$$

The number of trains in any class is limited. This is equivalent to constraints (3.3), (4.19) and (5.3).

$$\sum_{j|T_j=w} (z_j^{t+R_j^0+R_j^1+R_j^2} - z_j^t) \leq K_w \quad \forall\, w, t \tag{5.11}$$

Every producer must satisfy $(u-1)^{\text{th}}$ order before the due-date of the u^{th} order.

$$\sum_{j|M_j=i} z_j^{F_{i,u}} V_j \geq D^{F_{i,u-1}} \qquad \forall\, i, u \tag{5.12}$$

Scope and boundary conditions.

$$z_j^0 = 0;\ z_j^t \in \{0, 1\} \quad \forall\, j, t \tag{5.13}$$

In general, this model will give us a feasible solution by advancing or delaying jobs, unless the due-dates are very close, and hence, constraint (5.12) is violated. Once the job schedules are computed, the corresponding columns for each producer and their values can be computed easily.

5.3.2 Leader-Follower mechanism

As stated earlier, the resource constraint (5.3) is the only link between the producers. If we know the complete (or partial) schedule of all the other producers, then each producer can have a better understanding of the resource availability. Hence, a particular producer can customise their schedule to obtain a globally feasible solution. In this case, constraint (4.11) can be modified as

$$(\eta_w^{t+R_w^0} - \eta_w^{t-L_w}) \leq \bar{K}_w^t \qquad \forall\ w, t \tag{5.14}$$

where, $\bar{K}_w^t = K_w -$ (Trains reserved by other producers at time t). Solving the sub-problem with constraint (5.14) will restrict the solution space and will increase the possibility of getting globally feasible columns. To achieve this, columns are randomly selected from the solution pool of the RMP^k for a few producers thus solving the job scheduling for others. Hence, it will be restricted to a small neighbourhood of feasible solutions of the RMP. Adding these extra columns back to the pool will improve the UB^*.

Figure 5.1 shows the pictorial representation of CG Algorithm 4. The sub-problem and the RMP are used to compute the limits on the lower bound. Finding better upper bound is equivalent to computing better globally feasible solutions. Here we have three alternatives: (i) Master problem, (ii) Job scheduling model and (iii) Leader-Follower mechanism. The RMP computes the dual-prices of the resource utilisation constraints and provides feedback to the producers on their column generation.

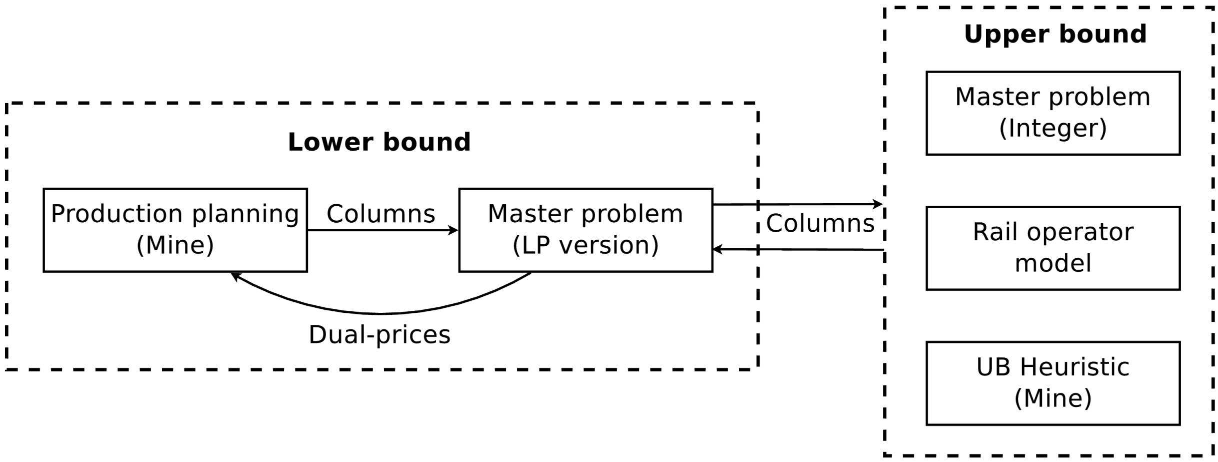


Figure 5.1: Column generation framework

5.3.3 Integer solutions

We may get good integer solutions directly from the heuristics described in Section 5.3. Another option is to solve the integer version of the MP, in which, the model tries to find optimal columns from the available pool. There is no guarantee that the optimal integer solution will be one among the optimal solutions of the RMP; some non-basic columns of the RMP can also be in the optimal integer solution. To maintain a reasonable problem size, the CG algorithm removes a few non-contributing columns from the solution pool (see step 11 of Algorithm 4). Hence, there is a chance of missing some good candidates for the optimal integer solution. To avoid this problem, we can have two versions of the solution pool $\mathcal{S}$: one for the RMP and one for the integer MP. Unless there are some memory restrictions, all columns from any iteration can be stored. However, solving the integer MP with all columns is not advisable because of the size of the solution pool. Once the prices are stable, new columns will not be added to the solution pool. In this case, we can filter the columns for the integer MP based on the current bounds, such that, they are within a neighbourhood of the relaxed solution.

A basic column generation Algorithm 4 was implemented and tested on several randomly generated datasets. More details of the datasets and results are presented in Section 5.5. Even though CG has shown significant progress compared to the centralised model, there are multiple strengthening methods that can and must be explored. Section 5.4 explains some of them.

5.4 Strengthening techniques for the CG algorithm

The literature and computational experiments suggest that the traditional CG algorithm can be improved significantly. In this section, we discuss two stabilisation methods and an improvement technique to strengthen the algorithm.

5.4.1 Stabilisation

The major drawbacks of the basic CG scheme, as mentioned by Amor et al. [5], are (a) slow convergence and (b) high oscillation of dual-prices. Many stabilisation methods are proposed in the literature to overcome these problems. The reader can refer to Lübbecke and Desrosiers [96], Amor et al. [5] and Vanderbeck and Wolsey [143] for recent reviews. The stabilisation of the dual price is centred on what is referred to as *stability centre*. In any iteration, the stability centre is the current best-known dual-price. Hence, we try to get dual prices close to the current stability centre. The concept of stability centre will minimise very high deviations in dual prices and will eventually improve the quality of the solution. In our proposed scheme, we use the weighted Dantzig-Wolfe decomposition method discussed in [51] and [110], along with a two-piece stability term motivated from [5]. The weighted scheme stabilises the dual prices at the sub-problem level. The stability term added to the objective helps to stabilise the master problem. By introducing both methods, we aim to minimise the instability of the scheme and achieve the optimal state in a fewer number of iterations.

Let $\bar{\lambda}$ be the stability centre. To stabilise the dual price, we penalise the difference between new dual prices λ and $\bar{\lambda}$. Fenchel's conjugate of $\mathcal{D}(\lambda - \bar{\lambda})$ is added to the objective of the RMP as a stability term (see [5]). Therefore, the RMP can be updated as:

$$\mathcal{V}_{RMP} = \min \sum_{i,c} V_{ic} x_{ic} - \bar{\lambda} \cdot y + \mathcal{D}^*(y) \tag{5.15}$$

subject to

$$\sum_{c} x_{ic} = 1 \qquad \forall\ i \tag{5.16}$$

$$\sum_{i,c} \chi^t_{icw} x_{ic} - y^t_w \leq K_w \qquad \forall\ w, t \quad \text{(dual variables } \lambda\text{)} \tag{5.17}$$

$$0 \leq x_{ic} \leq 1,\ 0 \leq y^t_w \tag{5.18}$$

where, $y = (y^t_w)$ corresponds to the additional resource provided for the class w at time t and the stability term is defined as,

$$\mathcal{D}^*(y) = \sum_{w,t} g(y^t_w), \qquad \text{where } g(\nu) = \begin{cases} \nu M_1 & \text{if } 0 \leq \nu \leq \mu, \\ \mu M_1 + M_2(\nu - \mu) & \text{if } \nu > \mu, \\ 0 & \text{otherwise.} \end{cases} \tag{5.19}$$

and M_1, M_2 are constants such that $M_1 < M_2$.

To understand the role of the stability function, preliminary experiments were conducted where the CG scheme was executed with four different stability functions.

Case-1 : $\mu = 0.5, M_2 = LB^* k/20, M_1 = M_2/10$

Case-2 : $\mu = 1, M_2 = LB^* k/20, M_1 = M_2/10$

Case-3 : $\mu = 0, M_2 = LB^* k/20$ (Single piece)

Case-4 : $\mu = 0.5, M_2 = LB^* k/20, M_1 = M_2/20$

where k is the iteration number. The preliminary experiments were conducted on 20 data sets from each data series (See Section 4.7.1 on details about the data set generation).

Figure 5.2 shows a comparison of the upper bound obtained from these four cases, normalised with the minimum UB. On the x-axis, the upper bound is taken after dividing it with the best-known value from these four methods. On the y-axis, the cumulative frequency is marked. It can be observed from Figure 5.2 that the trends are quite similar

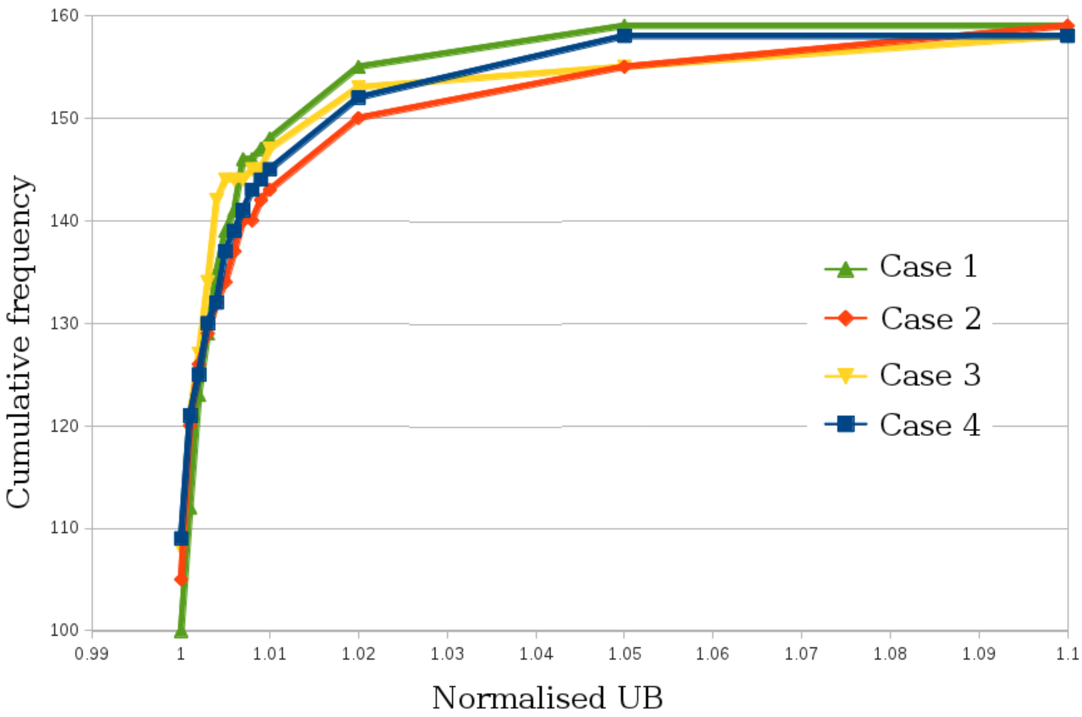


Figure 5.2: Upper bound comparison with different stability functions

for all four cases, with case-1 having some relative advantage. Therefore, case-1 is chosen for further improvements to the CG scheme.

5.4.2 Multiple columns

In general, finding all possible solutions of each sub-problem is very difficult. The traditional CG algorithm considers only one optimal solution from a sub-problem. However, Huisman et al. [79] suggest that faster convergence may be achieved by adding more columns. The MP complexity and the additional time and computational resource required are the bottlenecks in using this option directly. A commercial solver, CPLEX 12.1, is used to perform the computational experiments. Instead of storing a single solution from each iteration, multiple solutions can be recorded. Step 7 of Algorithm 4 can be modified to record all the solutions reported by the CPLEX solver. CPLEX has the ability to generate more solutions, although at the cost of significant time and memory in each of the sub-problems and the MP. Hence, the advantage of iterative improvement is lost. Therefore, all computational experiments were performed with the settings mentioned in Section 4.7.2. Step 11 removes a few non-contributing columns and maintains the size of the MP.

5.4.3 Improved CG algorithm

The basic CG algorithm is strengthened with stabilisation techniques and the improvement features discussed in Section 5.4.2. An improved version of it is presented in Algorithm 5. In Algorithm 5, we apply stabilisation in two stages:

Algorithm 5 An improved CG algorithm (CG)

1: Initialise $\bar{\lambda} = 0, LB^* = 0, UB^* = \infty, k = 0, \alpha = 0.5$, Time Limit $TL = 3600$, $\epsilon = 10^{-5}$, $\mu = 0.5, eTime =$ elapsed time in seconds.
2: Initialise the solution pool $\mathcal{S}^0$ with a feasible solution.
3: **while** $(eTime < TL)$ **do**
4:     Solve the RMP^k, given in (5.15)–(5.18), with solution pool $\mathcal{S}^k$ and the corresponding dual variables be $\lambda^{(k)}$.
5:     Set $M_2 = (k/20)LB^*$ and $M_1 = M_2/10.0$.
6:     Set $\hat{\lambda} = \alpha\lambda^{(k)} + (1-\alpha)\bar{\lambda}$.
7:     For each i, solve $\mathcal{P}_i$ with $\hat{\lambda}$.
8:     Compute $LB^{(k)} = \sum_i Z_i(\lambda^{(k)}) - \sum_w \left(K_w \cdot \lambda_w^{(k)}\right)$.
9:     Update solution pool with all the new columns.
10:     **if** $(LB^k > LB^*)$ **then**
11:         Set $LB^* = LB^k$.
12:         $\bar{\lambda} = \hat{\lambda}$ */* Update the stability centre*/*
13:     Call the heuristics presented in Section 5.3 to create globally feasible columns.
14:     Remove columns which are non-basic and do not contribute to the optimal solution of the RMP for 10 consecutive iterations.
15:     **if** $((\mathcal{V}_{RMP}(\mathcal{S}^k) - LB^*)/\mathcal{V}_{RMP}(\mathcal{S}^k) < \epsilon)$ **then**
16:         Evaluate the integer version of the MP to identify the best-known upper bound UB^* and integer solution.
17:         **break** */*Since the bounds are close enough, there is scope to improve the dual prices*/*
18:     $k = k + 1$
19: Report LB^*, UB^* and $gap = (UB^* - LB^*)/UB^*$.

(i) Stabilisation at a sub-problem level is implemented by using a linear combination of previous and current dual prices (see step 6) and

(ii) Stabilisation at the master problem level is provided by adding a stabilisation term to the objective of the MP (see Section 5.4.1).

Apart from stabilisation techniques, we implemented a feature to store and use multiple solutions (columns) at each iteration (see step 9 of Algorithm 5). Column management, which was discussed in Section 5.3.3, is implemented in step 11 of Algorithm 5. With these two factors in consideration, Table 5.1 describes four different versions of Algorithm 5. The abbreviations 'SS' and 'MS' represent a single solution and multiple solutions, respectively.

Table 5.1: Different CG algorithms and their properties

Version	Stabilisation	Solution(s)	Tag	RMP references
1	No	Single	CG – Non-stabilised SS	(5.1)-(5.3) and (5.5)
2	Yes	Single	CG – Stabilised SS	(5.15)-(5.18)
3	No	Multiple	CG – Non-stabilised MS	(5.1)-(5.3) and (5.5)
4	Yes	Multiple	CG – Stabilised MS	(5.15)-(5.18)

5.5 Computational experiments

In this section we compare the column generation algorithm with the integrated model over a set of realistic problem instances. For convenience, the iterative column generation Algorithm 5 is represented as 'CG'.

This integrated model is a collection of all sub-problems with an additional linking constraint. It can be expressed as,

$$[\mathbf{IM}] \quad \min \left\{ \sum_i Z_i(s_i) \;\middle|\; \sum_i \chi^t_{iw}(s_i) \le K_w, \ \forall\, w, t; \ s_i \in \mathcal{S}_i, \ \forall\, i \right\} \tag{5.20}$$

where s_i corresponds to a schedule (column) of each producer from the solution pool $\mathcal{S}_i$, which satisfies constraints (4.2) to (4.18) for all i. $\chi^t_{iw}(s_i)$ represents the number of active resources (trains) from the class w at time t for the producer i. The above integrated model is referred to as 'IM'. The IM is implemented and solved using CPLEX 12.1.

In Section 4.7, we had used 240 randomly generated instances used for a comparison between LR and the IM. The same datasets are used for the results presented in this section. The IM and CG schemes were terminated when the optimal solution was found or when a CPU time limit of one hour was reached. An additional 0.1% tolerance was permitted in the final optimality gap for the IM and 0.005% for the lower bounds in CG (see step 15 of Algorithm 5). However, sub-problems and job scheduling models were terminated at optimality. The lower bound of CG corresponds to the objective value of the RMP and the upper bound is computed from the integer MP. Therefore, we allow a 2% optimality gap in CG. Similar termination criteria are used for the LR scheme.

As mentioned in Section 4.7.3, the optimal objective value of the problem depends on randomly generated demand and due-dates. Therefore, it is not easy to capture the impact of the solution approaches on different datasets by comparing objective function face values. Therefore, a normalised performance measure is developed based on the Student's t-test. Similar to (4.45) and (4.46), the relative performance ratio for comparing the lower bounds and upper bounds obtained from CG and the IM are defined as

$$\text{LBR (CG, IM)} = \frac{LB_{CG} - LB_{IM}}{\max(LB_{CG}, LB_{IM})} \qquad \text{and} \tag{5.21}$$

$$\text{UBR (CG, IM)} = \frac{UB_{CG} - UB_{IM}}{\min(UB_{CG}, UB_{IM})}. \tag{5.22}$$

A positive LB ratio (LBR) means that CG is better, as it has a higher lower bound. Similarly for the upper bound, a positive UB ratio (UBR) means that the IM is better. In some cases UBR cannot be computed since the IM was not able to compute a single feasible solution within the time limit. However, the lower bound computation is very quick, and hence, LBR is available for all problem instances. Estimated mean and 95% confidence interval (CI) for the same ratios are computed using the Student's t-test.

5.6 Results and discussions

The results are summarised in two sections. Section 5.6.1 shows the impact of stabilisation of CG algorithms and we choose the best one for further analysis. In Section 5.6.2, we compare the results from the best CG algorithm with the results from the IM and the LR algorithm discussed in the previous chapter.

5.6.1 Impact of stabilisation and improvements

The following results compare the four different versions of the CG algorithm given in Table 5.1. Figure 5.3 and Table 5.2 compare the relative gap and execution time of different CG versions computed on 240 problem instances.

We analyse the impact of two features: (i) multiple-solutions (MS) and (ii) stabilisation. MS algorithms have smaller relative gaps than their SS equivalent versions. For example, in 91.3% of data instances, the non-stabilised SS scheme has less than 10% relative gap. However, the equivalent MS scheme has 95.8% of data instances with less than 10% relative gap. Similarly 95.4% is improved to 96.7% in the stabilised version. Having multiple solutions helps to improve the upper bound (integer solution). Sometimes, in SS, the CG-based algorithm may converge to a common lower bound with a worse integer solution. In the long run both approaches may yield the same solution. However, in a

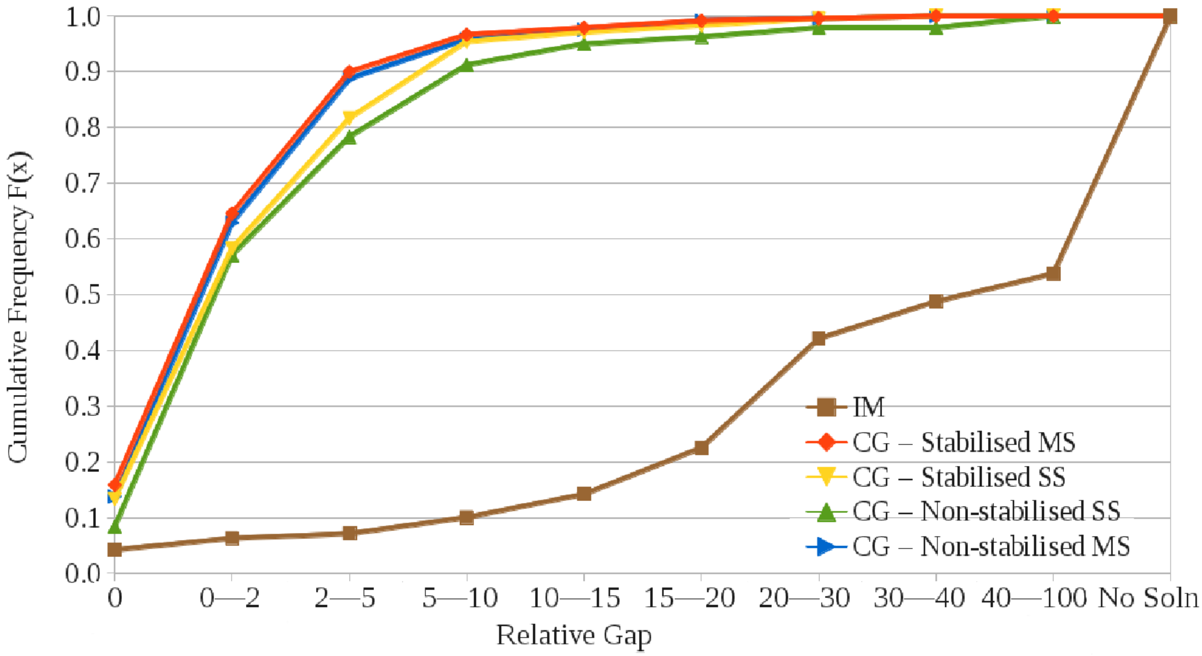


Figure 5.3: Comparison of the relative gaps obtained from different CG algorithms

Table 5.2: CPU run time comparison of different versions

Model	Non-stabilised				Stabilised			
	Single-solution		Multiple-solution		Single-solution		Multiple-solution	
# Producers	Mean	Median	Mean	Median	Mean	Median	Mean	Median
5	1819	1783	2022	1704	2231	2859	1795	1433
6	2339	2481	2407	2535	2569	3318	2033	1875
7	1844	1698	1045	1044	1840	1750	1228	1251
8	2227	2603	1612	1291	2347	2955	1808	1518
9	2286	2699	1513	1274	2738	3470	1806	1490
10	2335	3145	1877	1530	2460	2551	2122	1832
12	3154	3582	2604	3229	3273	3602	2843	3303
15	2796	3362	2314	2360	2935	3539	2838	3137

given time limit, the stabilised versions have smaller gaps. The relative gap computed by any CG algorithm is significantly better than the gap obtained by the IM. When we compare the mean and median execution time, a stabilised MS is relatively better than any of the other models. In Table 5.2, the clustering around time limit, $TL = 3600$ seconds is due to one of the termination criteria on the maximum CPU run time.

The best approach from among the four versions is the multiple solutions version with the stabilised column generation scheme. A reduction of one-to-five percentage in relative gap is observed by the application of stabilisation techniques in the 240 datasets. These problem sets have only 150/200 time periods and three to four orders per production unit. Therefore, the advantage will be significantly higher in large problems.

Decomposition algorithm based on LR

Chapter 4 (and [134]) proposed a distributed solution approach based on Lagrangian relaxation to solve this problem. Algorithm 1 was strengthened with the Wedelin Algorithm [148] and the Volume Algorithm [14]. This iterative algorithm is referred to as 'LR' in the rest of the chapter. The same datasets and termination criteria were used to test LR and CG algorithms.

5.6.2 Comparison of the decomposed models and the integrated model

From the above runs, we identify that the stabilised multiple solutions method is the best among all CG versions. For convenience this version, from now on, will be referred to as 'stabilised CG', or in short, CG from now on, unless otherwise stated. Some of the results and part of the discussions comparing LR with the IM are presented in the previous chapter (see Section 4.7.4). This section emphasises the new results and comparisons with the CG scheme.

Figure 5.4 and Figure 5.5 compare LBR and UBR calculated with the stabilised CG, LR and the IM, using equations (4.45), (4.46), (5.21) and (5.22). The CI boundaries and the mean are marked with thick lines. The observations are summarised below:

1. The UBR and LBR increase as the number of producers increases. The trend is identical in both, LR and CG, schemes. The highest LBR is observed for the series with eight producers in both cases. The highest UBR is obtained for the series with twelve producers in both cases.
2. On an average, LBR for CG is higher than that of LR by 1-2%. However, the UBR varies up to 8% since CG get better upper bounds than LR.
3. The lower bounds obtained by CG are always at least 6% better than the lower bounds found by the IM. The difference was much more significant in instances with more than seven producers; the quality of lower bounds found by CG was at least 20% better.
4. For a series with five or six producers, the IM, LR and CG, were able to converge to a solution that is close to the optimal in the given time. Hence, the UBR is close to zero for these series.

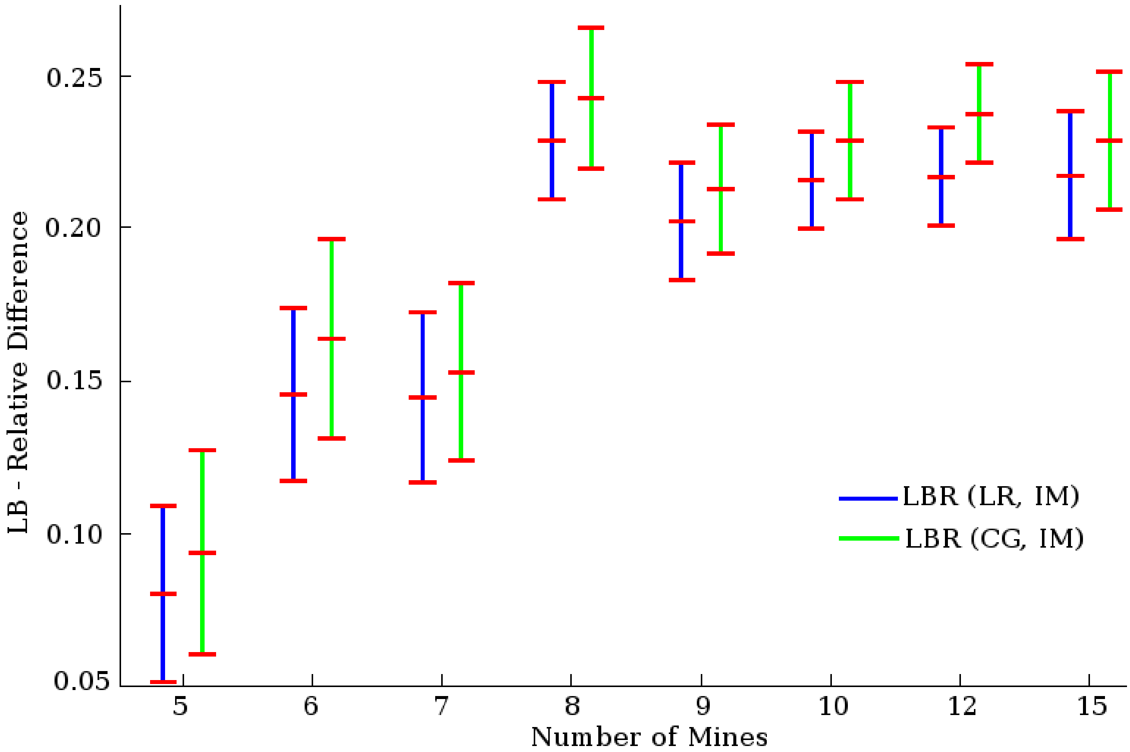


Figure 5.4: 95% confidence interval for the relative difference in the lower bound

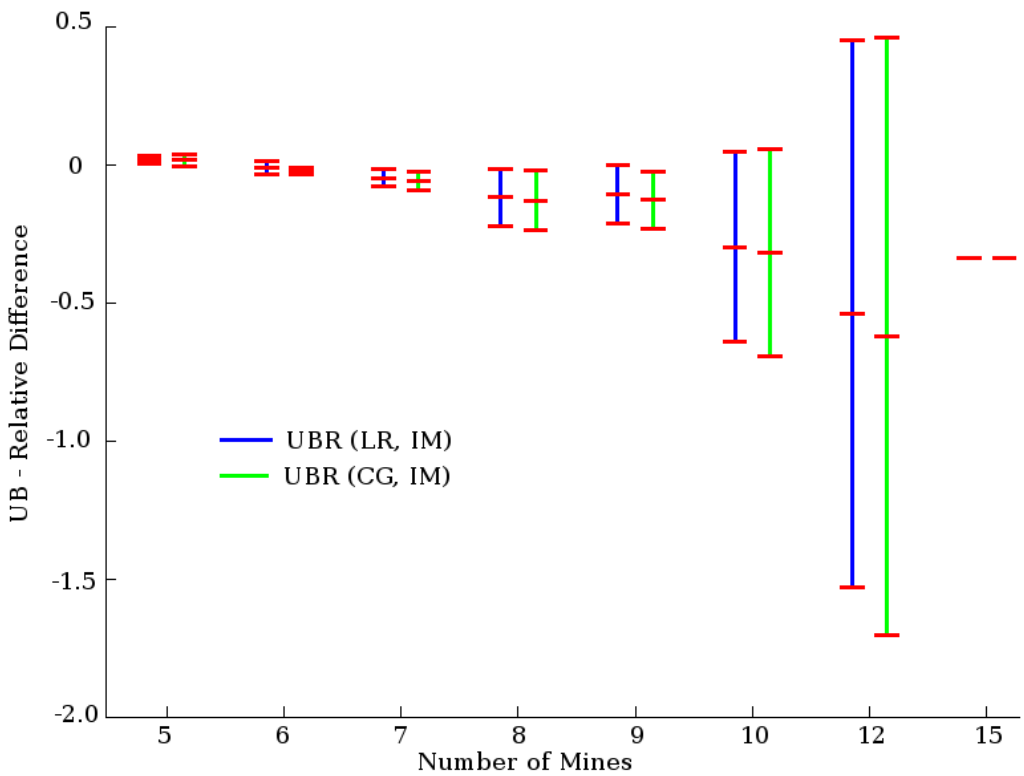


Figure 5.5: 95% confidence interval for the relative difference in the upper bound

5. The CI for the UBR with twelve producers is relatively large. In 26 out of 30 instances; the IM could not find the solution. Therefore, the CI has been computed

from the remaining four valuesthe possible reason for the larger interval.

The UBR and LBR increases as the number of producers increases. Figure 5.4 and Figure 5.5 show that the distributed algorithm, specially CG algorithm, is a computationally preferable solution approach than the IM.

Table 5.3: Comparison of the mean LBR and UBR

	# Producers →	5	6	7	8	9	10	12	15
LBR	(LR, IM)	0.08002	0.14533	0.14448	0.22862	0.20223	0.21553	0.21671	0.21723
	(CG, IM)	0.09361	0.16364	0.15273	0.24237	0.21289	0.22873	0.23734	0.22846
UBR	(LR, IM)	0.01502	-0.01124	-0.04828	-0.11903	-0.10914	-0.29863	-0.54168	-0.33725
	(CG, IM)	0.01505	-0.02333	-0.06203	-0.13051	-0.12948	-0.31871	-0.62244	-0.33859

Table 5.3 compares the mean LBR, UBR from the two decomposed approaches. Figure 5.6 shows the cumulative frequency plot for the ratios. The trends for both decomposed models are same. As we can see from Table 5.3 and Figure 5.6, the LBR from CG is always higher than that from LR. This implies that the CG scheme is able to get a tighter lower bound than that of LR. The same is true for the upper bound too. For the instances with greater than seven producers, the mean LBR is greater than 20% and the mean UBR is greater than 10% for both distributed algorithms. It indicates the advantage of the distributed model over the integrated model.

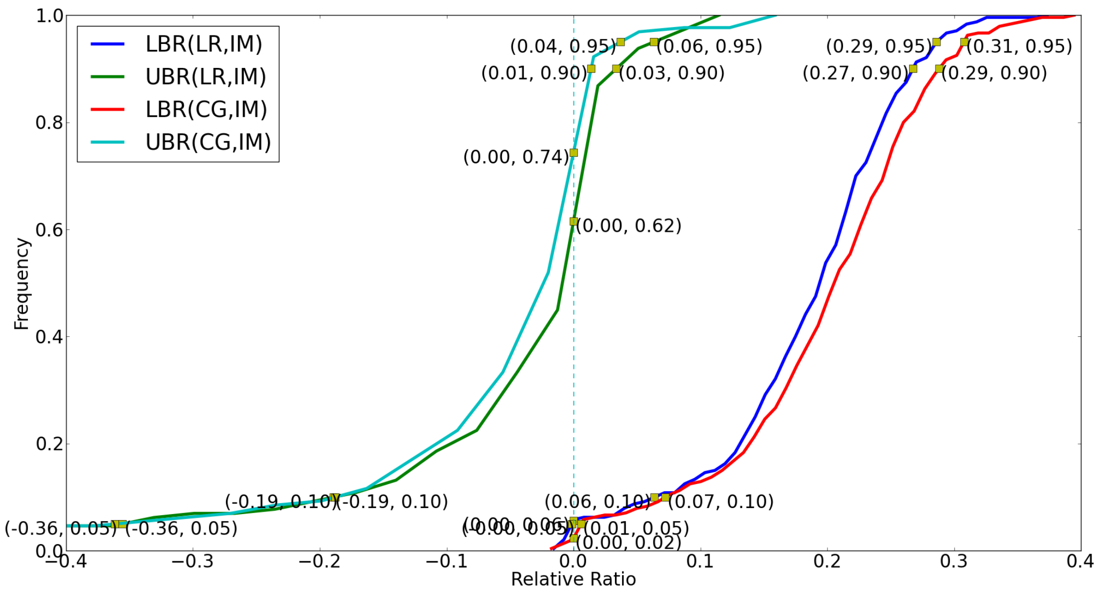


Figure 5.6: Cumulative frequency plot for LBR and UBR

Figure 5.7 exhibits the mean and median of the relative ratios computed for each data series and each data groups (see Section 4.7.4 and Table 4.2). From this figure, we can see that, as the number of players increases the complexity of the problem also increases. The improvement in the LBR shows a steady behaviour and the average is approximately 20%. However, the benefit in the upper bound computation is heavily depends on the problem complexity (see the UBR of 10 and 12 mine series and the UBR of the 'hard' group). Since IM could not find any single solution for the 'very hard' problems, UBR value cannot be computed for this group.

Figure 5.8 shows the histogram of the relative gap observed for the IM, LR and CG on 240 data sets. Some key observations are,

1. CG outperforms LR and the IM in all cases with significant differences. The number of cases versus the relative gap for all the approaches is summarised below:

Relative Gap %	**CG**	LR	IM
< 2	155/240	80/240	15/240
< 5	216/240	134/240	17/240
< 10	232/240	198/240	24/240

2. When CG achieves a gap of less than 5% in 89% of the cases, the IM could not find a single solution in 111 (46%) of the cases. Hence, the IM will be ineffective in larger problems.
3. CG reaches 99% cumulative frequency with a relative gap of less than 20% and LR achieves the same with a relative gap of less than 30%.
4. Figure 5.8 shows that the CG scheme converges faster than the LR scheme.

Table 5.4 compares the CPU run time of IM, LR and CG. It is evident that CG has a clear advantage over LR and the IM in terms of run time. The clustering around the time limit, $TL = 3600$ seconds is due to one of the termination criteria of fixing the maximum CPU run time to one hour.

5.7 Conclusions

In this chapter, we study a resource constrained scheduling problem (RCSP) with multiple producers and a single linking constraint. The integrated problem is decomposed and solved with column generation (CG) techniques and compared with a similar Lagrangian relaxation (LR) approach discussed in Chapter 4. The decomposed approach has many advantages over the integrated modelling approach in terms of the quality of solutions

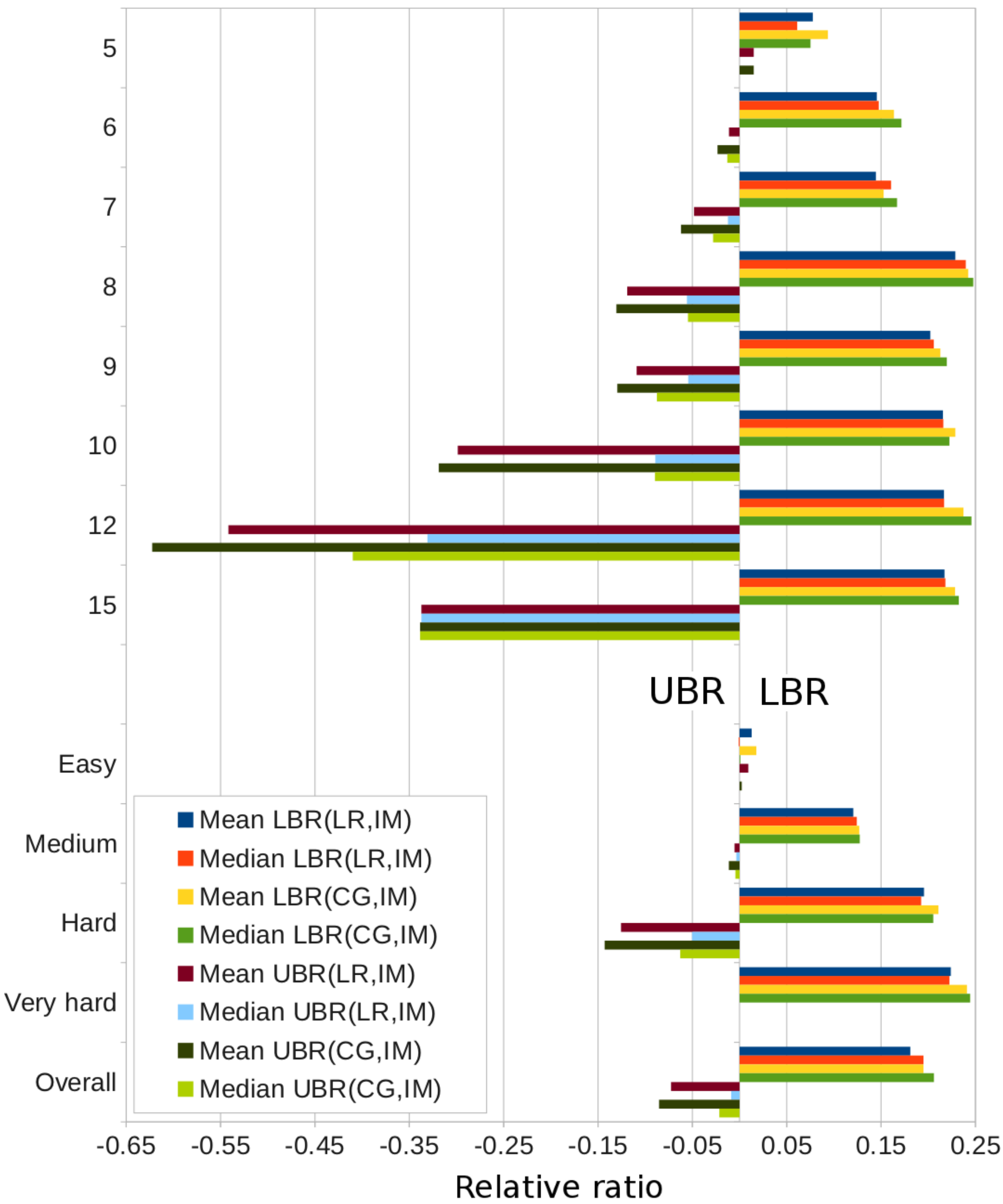


Figure 5.7: Mean and median of the relative ratios computed from the decomposed approaches

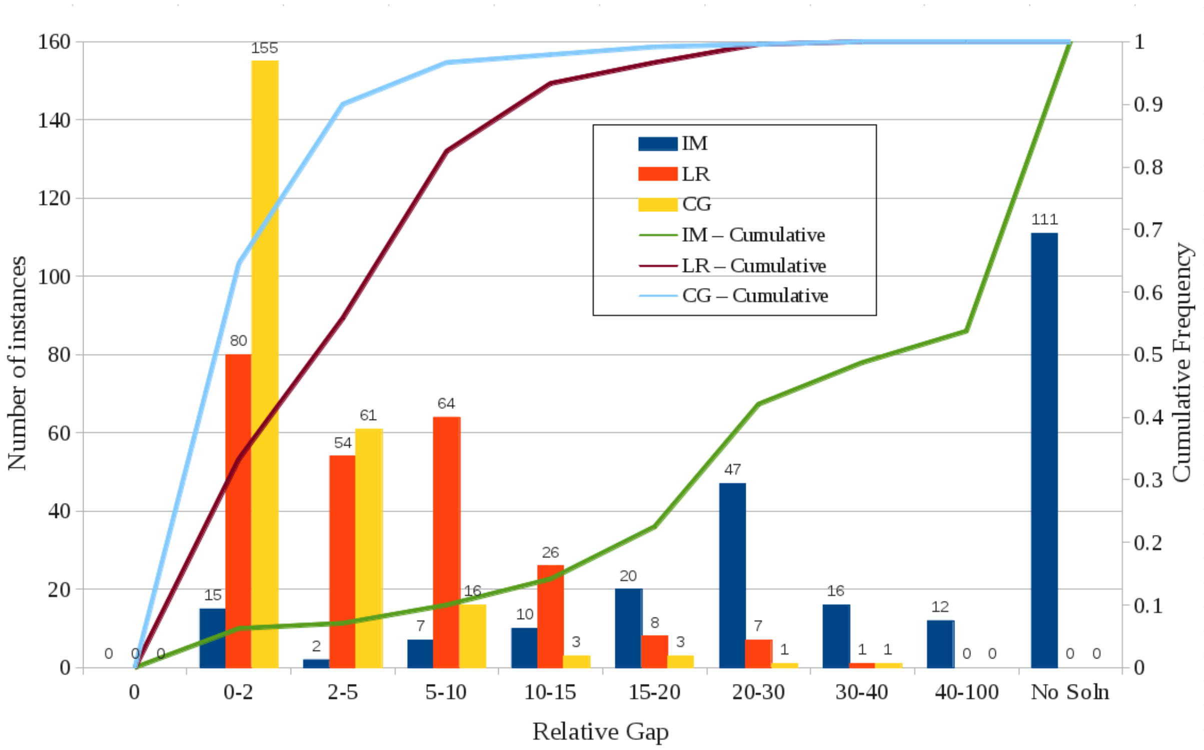


Figure 5.8: Relative gap comparison of the IM and CG

Table 5.4: CPU run time comparison of IM, LR and CG

	IM				LR				CG			
# Producers	Min	Max	Mean	Median	Min	Max	Mean	Median	Min	Max	Mean	Median
5	41	TL	2911	TL	950	TL	3444	TL	21	TL	1795	1433
6	1666	TL	3498	TL	2978	TL	TL	TL	202	TL	2033	1875
7	369	TL	3383	TL	TL	TL	TL	TL	114	3541	1228	1251
8	TL	TL	TL	TL	357	TL	3424	TL	307	TL	1808	1518
9	TL	TL	TL	TL	2058	TL	3572	TL	249	TL	1806	1490
10	TL	TL	TL	TL	2867	TL	TL	TL	284	TL	2122	1832
12	TL	TL	TL	TL	TL	TL	TL	TL	691	TL	2843	3303
15	TL	TL	TL	TL	TL	TL	TL	TL	697	TL	2838	3137

Time limit, $TL = 3600$ seconds

and computational time. Computational experiments show that the CG outperforms LR and the IM. Although the methods that have been developed in this chapter are tested within the context of a coal supply chain, it can be extended to other general RCSPs in contexts such as airline, wine, automobile manufacturing and the services industry.

Two major decisions are considered in our approach: *production planning* and *resource scheduling*. The upper bound of the distributed model can be further strengthened by the proposed heuristics. Different stability functions can also be explored to further strengthen the performance of CG. In future research, we plan to develop new algorithms for cases with partial information-sharing and also for cases where there are multiple objectives.

In a *truly decentralised decision-making* there should be a 'negotiation mechanism' supported with an incentive scheme. In that sense, ours is not a truly decentralised decision-making framework. In our approach, the negotiation protocol is only partially implemented using the CG algorithm, which is (in a sense), 'controlled' by the 'honest broker' rail operator. We made this assumption simply so that we could converge to a solution, which is in fact a very difficult real-world problem. Further research will need to include proper negotiation mechanisms and ensure that the rail operator is no longer a 'de facto honest broker'.

Chapter 6

A Decentralised Approach for Two-Party Coordination

Chapters 4 and 5 discuss decomposition approaches for the two-party case. The decomposition models can be viewed as single-operator decentralised models, where the single-operator uses centralised information to update the bounds and the multipliers. As an alternative, we look at a *truly decentralised decision-making* approach which comprises of (*i*) multiple stakeholders, (*ii*) information asymmetry, (*iii*) conflict in objectives, and (*iv*) a negotiation protocol.

The problem under consideration has information asymmetries, conflict in the objectives among the partners. The decomposed framework proposed in the previous chapters can be upgraded to a decentralised framework by introducing an honest broker. A decentralised framework would also have a negotiation scheme. The honest broker who manages the negotiation decides whether a decision is a wrong or right and manages the compensatory schemes for bad solutions that any one producer receives. In our models, the rail operator acts as an honest broker and addresses the negotiation partially.

Decentralised decision-makers are influenced and guided by the individual players/owners. The objective of the individual players might conflict/be different from that of the other players in the supply chain. This might lead to inefficient operations from the supply chain point of view. If there are multiple producers/resource managers, then it is difficult to align the objectives and priorities of all decision-makers to form a single objective. Pareto optimal solutions are more suitable in the handling of such multi-objective problems. This means that the supply chain's best solution will be the one which cannot be improved without worsening at least one player's performance. There are different issues in incorporating coordination in a decentralised supply chain: an expectation mismatch between the DMUs and conflicting objectives are some of them.

Certain portions of this chapter are ready to be submitted to the International Journal of Production Economics [135]

Information-sharing is one of the key factors in designing a successful coordination mechanism [40, 34, 153, 47, 87]. The methods to capture the information and provide feedback to the various players, and the medium through/by which such information is shared are also becoming increasingly pertinent to the successful operation of these decentralised supply chains. In any supply chain, different types of information are shared at various levels through diverse DMUs. Associating a quantifiable value with this multi-dimensional information is not easy. Our aim is to conceptualise the *value of information* and provide a definition for it and then to study the impact of it in designing a coordination mechanism for a decentralised supply chain. Even if the DMUs are willing to share the information, mechanisms are needed to filter out false information. This can be achieved only if the player does not get any benefit by provided false information. In our analysis, we also assume that the DMUs are truthful and willing cooperate, with minimal information sharing.

Section 6.1 outlines decentralised supply chain coordination and Section 6.2 explains the main components of the coordination. Section 6.3 defines the value of the information and Section 6.5 presents the computational experiments and their results.

6.1 Decentralised supply chain coordination (DSCC)

Chapter 3 (or [133]) classified decentralised models broadly into two levels: (*i*) operational level and (*ii*) decision-making level. A supply chain can be operated by a single or multiple operators. The decisions of a single-operator supply chain can be taken either with a *centralised decision-making* model or with a *decentralised decision-making* model. However, multi-operator supply chains cannot be solved with a centralised decision-making model. Single-operator decentralised models are often solved with decomposition or other similar techniques. Under a single operator, lack of information-sharing and the presence of a conflicting objective may not be present. Decomposed approaches can be extended to decentralised approaches by reducing the information sharing and minimising the role of a central coordinator. Information-sharing plays a key role in successful supply chain coordination [153, 47, 59, 41, 34] A detailed literature review on the supply chain coordination and information sharing is presented in Section 2.2.1 and Section 2.2.5, respectively. In this chapter, we focus on multi-operator supply chain problems which are solved using decentralised approaches.

The focus of this chapter is to explain the role of information-sharing in decentralised supply chain coordination (DSCC). Quantifying the usefulness of information is also illustrated. Understanding the different components and features of DSCC allows us to design better coordination procedures for decentralised SCs. We explain DSCC with an example from a coal supply chain.

6.1.1 A coal supply chain example

Let us consider a coal supply chain example, in which we have two groups of DMUs. One group is for mines/producers and the second for a rail operator/resource manager. Figure 6.1 is a schematic diagram of a two-party model. The first group of decision-makers has multiple sub decision-makers internally, corresponding to individual mines. Each is independent of the other, but they all have identical structures and operations.

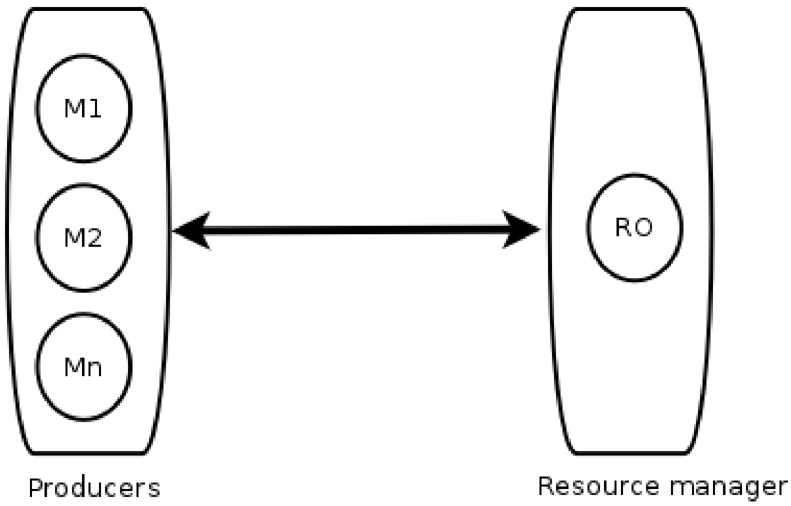


Figure 6.1: Schematic diagram of a two-party supply chain

Only two independent decisions have to be made in this supply chain: (1) resource allocation of trains to the producers and (2) scheduling of requested resources/trains. Many other dependent decision variables such as the production plan, the overstock status, the inventory level and the like, can be computed using these independent decision variables. An elaborate discussion on this supply chain and the mathematical models for each of the DMUs are presented in Chapter 4. An overview of this is provided below:

Production Planning

The producer receives the orders from the terminal and prepares their production plan subject to production constraints, resource class properties and their objective costs. The mathematical model for production planning is presented in Section 4.3. The objective of this model is to minimise the total production cost. The production plan can be prepared by considering (or not) the resource information. If it is prepared without considering the properties of the resource-class (this means, resource availability is assumed to be infinite), then the decisions might be far away from a globally realisable or feasible solution. If we plan the production along with appropriate resource utilisation, then we can compute all other dependent variables with respect to this. In a typical situation, resource allocation and resource scheduling create a bigger bottleneck than the production constraints. Production planning, without knowing resource-class properties, might increase the inventory stocked at different places.

Resource Scheduling

In a completely independent situation, the resource manager expects the producer to report the inventory level. Then, the resource manager is expected to make decisions on the resource-class allocation and its scheduling. As we mentioned earlier, if we completely separate resource-class allocation from the producer, it might be very difficult to bridge the gap between the decisions of the producer and the resource manager.

Figure 6.2 illustrates the interrelation between the producer and the resource manager. The resource constraint connects different producers with the resource manager. The detailed mathematical models for production planning and resource/job scheduling are presented in Section 4.3 and Section 5.3.1, respectively.

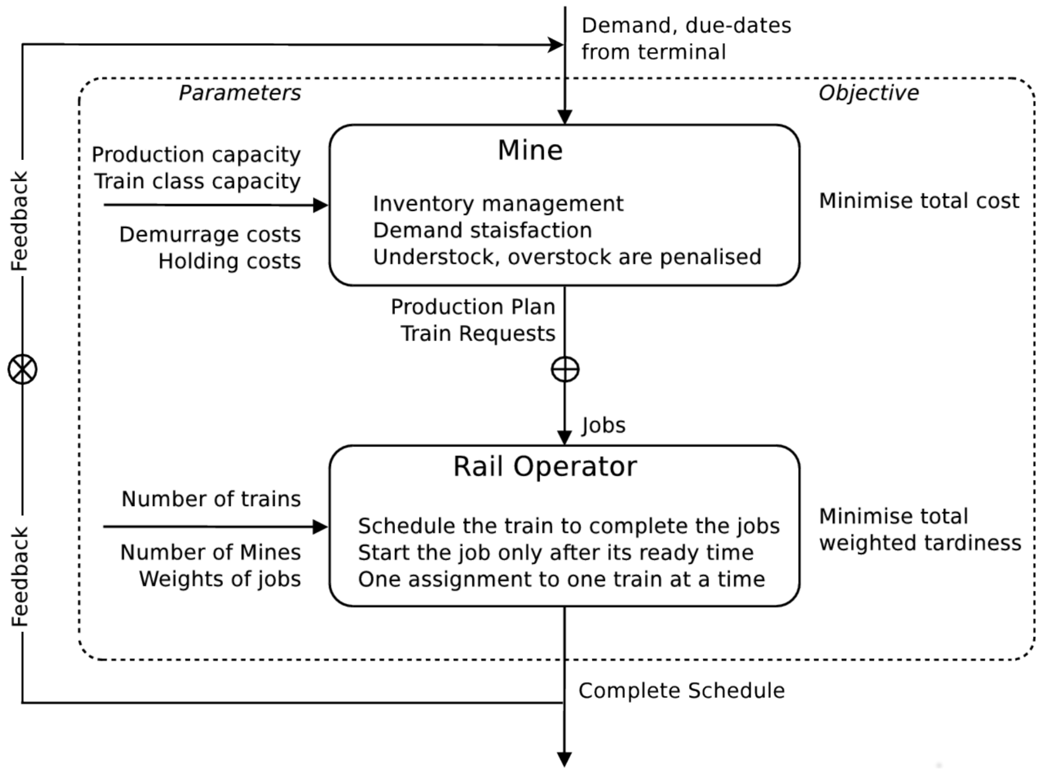


Figure 6.2: Schematic diagram of a two-party coordination model

In the two-party decomposed models discussed in the previous chapters, the rail operator's objective was not considered explicitly. However, here, the rail operator has a direct contribution to the objective of the integrated model. Let ξ_w^t be the running cost of a train of class w at time t. Then the total running cost (with respect to the resource requests, η) would be

$$\sum_t \sum_w \left[\xi_w^t \sum_i \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-R_j^1-R_j^2} \right) \right] \tag{6.1}$$

Or the running cost can be expressed in terms of the jobs (j) as,

$$\sum_t \sum_j \left[\xi_w^t \sum_i \left(z_j^{t+R_j^0+R_j^1+R_j^2} - z_j^t \right) \right] \tag{6.2}$$

Then objective (5.6) can be updated as

$$\min Z_R = \sum_j \sum_{t>0} \left[(C_j^t + G_j^t)(z_j^t - z_j^{t-1}) \right] + \sum_t \sum_j \left[\xi_j^t \sum_i \left(z_j^{t+R_j^0+R_j^1+R_j^2} - z_j^t \right) \right] \tag{6.3}$$

The objective of the integrated model (4.4) can also be updated accordingly.

6.2 Main components of DSCC

This section discusses four main components of DSCC. Coordination among the DMUs in a decentralised supply chain can be studied by characterising these components.

6.2.1 Decisions makers and decisions

In a supply chain, there are multiple DMUs and multiple decisions. Some decisions are explicitly related and some others are implicitly related to each other. Coordination helps us to avoid redefining similar variables in different DMUs. In a well-defined DSCC, the decisions and the decision-makers should be grouped appropriately. This means that the coordination mechanism must help the decision variables to use the information from other DMUs to compute their decisions.

Based on the different ways of assigning the decisions to decision-makers, different coordination mechanisms can be developed. Common decisions can be made by any of the common DMUs. For example, in a producer-distributor supply chain, production plan, resource (distributing facility/vehicle) allocation and resource scheduling are the major decisions to be taken. Resource allocation is a decision of the resource manager, which depends heavily on the producer's plan and orders. It can be done by the resource manager as well as the producer. If the producer can propose a complete or partial resource allocation, it will generate a globally-feasible (close-to-optimal) solutions for the supply chain. Similarly, if the DMUs can share the common decisions, then coordination between the DMUs will be strong. Through common decision variables, a DMU is able to give some feedback or influence other DMUs in a favourable way.

6.2.2 Information-sharing

The coordination efforts are directly linked with the level of information-sharing. Without information-sharing, it is impossible to consider the preferences/feedback from the other part of the supply chain. In a win-win situation, DMUs might be willing to share a fraction of their private information in order to improve their own performance. When a DMU wants to coordinate with the other DMUs, it needs sufficient information to ensure global feasibility. For example, if the resource manager plans to advance some of its vehicle schedules, then it is necessary to know the production capacities to avoid overall infeasibility. On the other hand, the producer can make better plans if resource availability and other properties are known to them.

The level of information-sharing can vary from supply chain to supply chain. The producer can share the production plan, inventory level, order-quantity, due-dates, resource allocation, and objective costs. The resource manager's critical information is resource class properties, resource availability, and objective costs. If there is no information-sharing, the coordination will be tough. It might require many iterations to even find a globally-feasible solution.

False reporting is another issue in information-sharing. It will be difficult to find out whether a DMU reports its true features or false ones to have better utility. The concept of *mechanism design* is useful in addressing such situations (see [75], [22]). We have not considered false reporting in this thesis. We assume that DMUs report only *true* information. The medium or the way in which information is shared is not addressed here. Our focus is the quality of the information shared and the assigning of a value to it. Different DSCC models are compared by the value of the information and other performance measures.

6.2.3 Multiple objectives

Many independent DMUs exist in a decentralised supply chain. In some cases, their different objectives can conflict or be similar or have different scales and units. So, it is important to consider these multiple objectives carefully. In general, a centralised approach expects to have a single supply chain objective by taking a combination of these individual objectives. In a decentralised approach, we have additional freedom to give importance to the individual player's objective.

In the coal example, producers assume that the objectives are the same or different—minimise the sum of the inventory holding cost, demurrage and order placing costs, minimise late/early deliveries, on-time order satisfaction, and maximise profit. At the same time, the resource manager may have objectives—minimise total tardiness/earliness of the requests (jobs), minimise makespan, minimise running cost, maximise utilisation and the like.

There are multiple performance functions or utility functions in a DSCC to define global optimality. Hence, the concept of Pareto optimality describes the equilibrium in DSCC. The performance of DMUs cannot be improved from a Pareto optimal solution without deteriorating the performance of some other DMU. There can be more than one Pareto optimal solution. Another popular approach is to use a weighted sum approach to make a single objective from all these separate objective functions. Marler and Arora [100] claim that the weighted sum method is extensively used for multi-objective problems. There are some interesting connections between the weighted sum method and Pareto optimality. If all of the weights are positive then minimum of the weighted sum is always Pareto optimal [63]. Marler and Arora [100] provide a comprehensive discussion on the weight selection and how to maximise the effectiveness of the weighted sum method. The authors conclude their discussion by pointing out a few scenarios in which the weighted sum method may not be adequate.

6.2.4 Coordination mechanism

Different coordination mechanisms can be designed based on the level of information-sharing and decisions grouping. It can be iterative or sequential or with/without feedback. The integrated model is executed only once. However, the algorithms based on LR and CG are iterative. The feedback on the linking constraints is given through the multipliers and they are updated over these iterations. In a production-distributor supply chain, many coordination mechanisms are possible. The following is not an exhaustive list. However, a few mechanisms are listed here to convey a broad idea of the variety of that is available or possible:

1. Solve the production planning model and report the requests for resources. Then the resource manager can adjust a few requests to make a globally-feasible schedule. This is a non-iterative two-step process.

2. The producers report their inventory level and the resource manager is required to work out the resource allocation and scheduling. This can be done iteratively/non-iteratively.

3. The producers request certain resource allocation and the resource manager schedules these requests and sends a feedback to the producer. It is an iterative process.

4. The resource manager allots the resources to a producer one after the other. In this setup, the interaction between the producers is almost nil.

5. The resource manager allots the resources using an MILP model for all the producers simultaneously. In this case, it is assumed that producers are willing to share some information with other producers.

These models can be refined by customising them for specific supply chains. The mechanism mentioned above can be repeated with different information-sharing set-ups. In general, if the iterative procedures are built on the foundation of Lagrangian relaxation or any other decomposition, then convergence is guaranteed. The proposed decentralised scheme is developed on an iterative algorithm: Lagrangian relaxation (LR). In this algorithm, the producer allocates the resources and the rail operator does the scheduling of the limited resources. Our interest lies in analysing similar coordination mechanisms, under different information-sharing situations and computing the value of their information.

6.3 Value of information

From the previous section, it is clear that information-sharing is very important to achieve successful coordination within a decentralised supply chain. However, before starting the sharing of information, it is important to address the following important questions:

(i) What are the critical bits of information that can affect the final solution?

(ii) How much of this information needs to be shared?

(iii) How can we quantify the usefulness of an information?

(iv) Is there a mechanism for sharing information and does this mechanism protect the privacy and security of the data?

In any supply chain, each DMU will always be interested in gaining knowledge on aspects of other DMUs' decisions that affect them. On the other hand, since the DMUs are independent and more or less autonomous; they do not wish to share their private and competitive information with other DMUs. For example, in the coal supply chain, all the mines compete for the same trains (resources), at same time. Therefore, it is not wise (or possible) for the mines to share their production plans with other mines that compete for the same rail resources. In some cases, sharing some of the information is really helpful in situations where there is a mutual benefit for the partners. For example, if the rail operator knows the production capacity then it may ask some of the mines to align the production schedule to avoid possible conflicts with the scheduling of the trains. Therefore, it is very critical to identify key pieces of information which *can* be shared and also pieces of information that assist in making an improvement.

Quantifying the impact of information in a supply chain is non-trivial [91]. Some information may only have a short-term impact while some other piece of information may have direct or indirect long-term impacts [86]. The same information could yield different benefits in different supply chains. In the literature, the *value of information* is defined as a relative ratio of the performance measure, both with and without sharing the information (see [47, 153, 59]). In this chapter, we use a similar definition for the value of the information.

In this model, we consider two pieces of information that can be shared between players in a coal supply chain: (i) production capacity and (ii) resource availability. Table 6.1 summarises the properties of the mathematical models and their dependencies. We define the *value of information* as a relative ratio of the performance measure (utility), both with and without sharing that information. For example, the value of sharing information 'a' with respect to the lower bound on the objective value will be defined as,

$$\mathcal{V}_{LB}(a) = \frac{LB_{\bar{a}} - LB_a}{\max\{LB_{\bar{a}}, LB_a\}} \tag{6.4}$$

where $LB_{\bar{a}}$ is the best lower bound computed without sharing information 'a' and LB_a is the best lower bound computed with sharing information 'a'. A similar definition can be used for other performance measures.

Table 6.1: A summary of the decision-making models of the mines and the rail operator

	Mine	**Rail Operator**
Input	Orders from the terminal	Train request from the mines
Decisions	Production plan and train numbers of different sizes	Feasible train schedule
Objective	Minimise total cost of production and inventory	Minimise operating and tardiness cost
Critical information	Production capacity	Number of trains of different sizes

6.4 Decentralised decision-making approaches

Two popular decomposition approaches—the Lagrangian relaxation (LR) and the DWD-column generation (CG)—were presented in the previous chapters. In these two iterative algorithms, the lower and upper bounds are updated by a central player who has access to all the required information. The CG algorithm performs better that the LR in terms of the bounds and the convergence. The basic building block in the CG algorithm is the column with a value associated with it. If there are only two players, then we cannot compute the value of a column without revealing the actual values. Hence, we cannot extend the CG approach to a two-party decentralised approach. Therefore, in this case, we propose an approach based on LR.

Algorithm 1 presented in Chapter 4 assumed that the information is centralised. In this chapter, we extend the decomposed approach to a decentralised one by removing this assumption. Algorithm 1 presents an updated Lagrangian relaxation algorithm, strengthened with the Volume Algorithm [14] and the Wedelin Algorithm [148]. Therefore, the proposed algorithm in this chapter also includes the Volume algorithm to stabilise dual-prices, and an heuristic based on the Wedelin algorithm to break the symmetry in updating the dual prices.

In the coal supply chain example, independent mines are linked with a single rail operator, who has a fixed number of trains to meet the requests from the mines. Therefore, in the absence of any restrictions on the number of trains available, there are separate planning problems for each mine and a train scheduling problem for the rail operator. On the other hand, if the mines create train requests without knowing the number of trains, that are available for the operation, then, merely combining the requests from the mines to create feasible train schedules need not be feasible for the rail operator. Therefore, we use an algorithm based on Lagrangian relaxation to bridge the gap between the train schedule requested by the mines and a feasible train schedule for the rail operator. The rail operator gives feedback to the mines in the form of Lagrangian multipliers. It will eventually minimise the gap between the requests and the actual schedule. The multipliers, also known as dual-prices, represent the 'cost' of using a train of a particular size at a particular time. In the next iteration, the mines take these multipliers into account while creating a new set of train requests.

Figure 6.3 shows the framework for a similar two-party case proposed in [133]. At the start of iteration-1, each mine assumes that the rail operator has a sufficient number of trains to meet its requests. Therefore, in step 'S0', we initialise the feedback parameter to zero. Then the decisions are made, iteratively.

The decision models of the mines (see Section 4.3) and the rail operator (see Section 5.3.1) are solved in Step S1 or S3, respectively. In step S2, the requests from the mines are combined as inputs (jobs) for the rail operator model. In Step S4, the lower and upper bounds are computed from the solutions of different decision models. Step S5 and S6

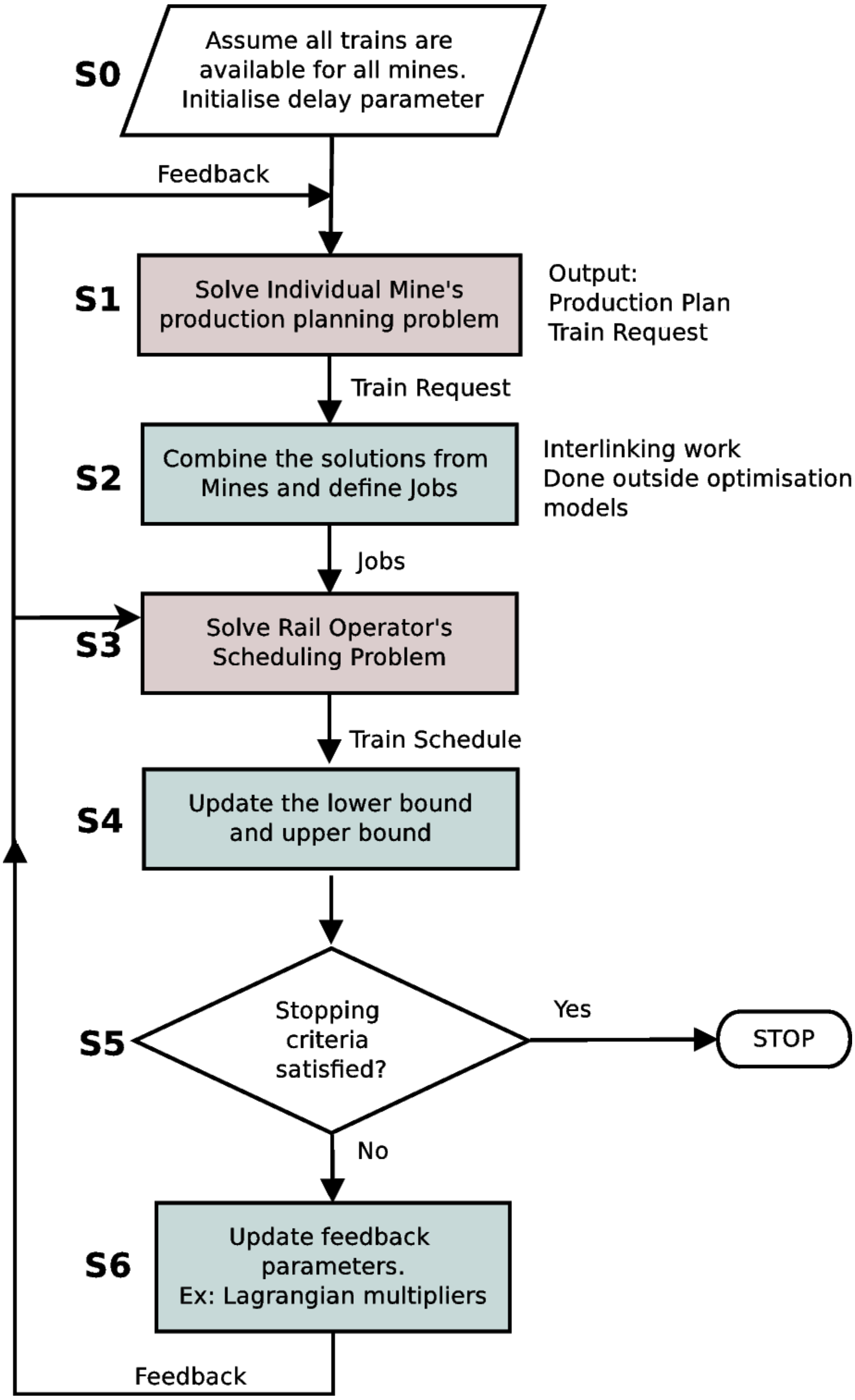


Figure 6.3: Flowchart for coordinated planning and scheduling

are then used to check the termination criteria and to update the Lagrangian multipliers, respectively.

In Step S4, the model discussed in Chapter 4 uses centralised methods and heuristics to compute the bounds. However, in the next sub-section, we propose an approach that computes the bounds without any central player or information.

Let λ_w^t be the Lagrangian penalty coefficient of resource class w at time t, then the planning sub-problems are solved with a Lagrangian objective (4.24)+(6.3), is given as,

$$Z(\lambda) = \min \Bigg[\Bigg(\sum_i (\text{Objective of producer-}i) + \text{Objective of the resource manager} \Bigg) \\ - \sum_{w,t} \lambda_w^t (\text{Violation of resource constraint of class } w \text{ at time } t) \Bigg]. \tag{6.5}$$

Algorithm 6 describes an iterative LR algorithm to solve the DSCC problem. It is a modified version of Algorithm 1. Since our focus is the value of information, here, we provide an outline of the algorithm, for the completeness.

Algorithm 6 An iterative algorithm based on LR for DSCC

Input: Order quantity with due-date from the terminal
Output: Production plan and train schedule

Initialisation
while (Elapsed Time < Time Limit) $\wedge$ ($gap \geq 0.001$) **do**
  Solve i^{th} sub-problem given in Section 4.3 with an updated objective (6.5).
  Find the current violations for each resource class.
  Set the lower bound using secure-sum, $LB^{(k)}$.
  Compute $UB^{(k)}$ using any decentralised method.
  Update UB^*, LB^*, if necessary.
  Take linear combination of current violation and previous violations.
  Update the Lagrangian multipliers.
  Use Wedelin based algorithm to update the multipliers.
  $gap = (UB^* - LB^*)/UB^*$
  $k = k + 1$

There are some differences in the implementation of Algorithm 6 based on different levels of information-sharing. If the producer knows the resource availability then the feasibility of the set of requests is higher. At least there will not be any resource violation for the same producer. There is also a similar modification for the upper bound computation based on the knowledge of production capacity.

The traditional LR is a decomposed centralised approach. Normally, the lower bound (LB) and the upper bound (UB) in each iteration are computed by a centralised coordinating player/agent. If an LR algorithm needs to be converted to a *truly* decentralised one, we need to compute the lower bound and the upper bound using decentralised methods. There need not be a central player who has all information about the producer and the resource manager. The following subsections suggest procedures to compute LB and UB using decentralised methods.

6.4.1 Decentralised methods to compute the bounds

Step S4 in Figure 6.3 is the key step of our decentralised approach where the lower and upper bounds are computed without using any central information.

Lower bound

The lower bound in a Lagrangian relaxation algorithm is computed as the sum of the objective cost of the mines and the rail operator and a penalty term for resource constraint violations. The Lagrangian function (6.5) can be evaluated in a decentralised way using the *secure-sum* method, proposed by [42]. Singh and O'Keefe [121] use this method to compute the lower bounds in a Lagrangian relaxation algorithm without exchanging any private information. In the secure-sum method, the sum is initialised with a random number and passed to the first mine, who then adds its objective term to this sum. The first mine then passes this sum to the second mine and so on. Finally, the initial random number can be subtracted to get the sum, and hence, the lower bound can be computed without knowing the individual objective terms. This process can be implemented using a distributed system without sharing the individual objective costs with other mines.

Upper bound

Based on the level of information-sharing, the upper bound can be computed using different models. Even though there are multiple coordination mechanisms, we use only one method to compute the upper bound to compare the value of the information. There are some differences in the output, based on the availability of information. Algorithm 7 exhibits the iterative procedure to compute the upper bound. If the production capacity is not known to the resource manager, then a job can be scheduled only after it is ready

Algorithm 7 An algorithm to compute better upper bound

Input: Jobs={Request for trains (resources) with due-date}, Production capacity
Output: Train schedule

```
Solve job scheduling model presented in Section (5.3.1) with the objective (6.3)
/* The frequency of the following while-loop can be reduced to once in 10 iterations or
so to save the total execution time. */
while Select a mine randomly do
    Apply Leader-follower model discussed in Section (5.3.2).
    if there is no improvement then
        Break
Report the actual cost of the schedule and production plan
```

time. Otherwise a job can be scheduled earlier, if it is feasible to produce the required quantity by that time. In step (3), the leader-follower improvement assures that the solution reaches at some *Pareto equilibrium*. That means we reach at a local equilibrium point where we cannot improve further without decreasing some DMU's performance.

6.4.2 Convergence

The proposed decentralised algorithm is based on LR. Therefore, convergence is guaranteed through the convergence of the underlying LR scheme. However, the speed of the convergence depends on the way in which UB and Lagrangian multipliers are updated. The quality of the solutions is expected to be lower than those from the centralised and decomposed algorithms as the level of information-sharing is very minimal in a decentralised framework. However, our main argument is that the decentralised methods are close enough to decomposed models and better than centralised one in terms of convergence.

In the proposed solution approach, the two main-pieces of information are (*i*) production capacity and (*ii*) resource constraint. Therefore, we can have four different versions of the decentralised algorithm. The impact of two sections of information in a decentralised modelling environment can be computed by comparing different performance measures. Table 6.2 lists four versions of the algorithm along with a centralised and integrated model. The centralised and integrated model M0 is a single optimisation model solved directly with CPLEX. This is used only to benchmark the decentralised schemes.

We analyse the impact of two sections of information in a decentralised modelling environment. Using Algorithm 6, we have four different variants for comparison.

Table 6.2: Different versions of the decentralised algorithm

Version	Information-sharing	Production capacity	Resource constraint
M0	Centralised and Integrated model		
M1	Complete	Shared	Shared
M2	No	Not shared	Not shared
M3	Partial	Shared	Not shared
M4	Partial	Not shared	Shared

6.5 Computational results

In Chapters 4 and 5, we have used 240 random problem instances for a comparison between the decomposed approach based on LR, and the integrated model. The same datasets, presented in Section 4.7.1, are used for the results presented here. The 240 instances are arranged in eight series, each corresponding to a set with the same number of mines. The centralised model and all other sub-DMUs are solved the same experimental setup mentioned in Section 4.7.2.

Since the optimal cost of a dataset depends on randomly-generated order quantity and its due-date, we use the value of information to compare the approach with multiple data instances. Different models are compared with respect to their lower bound (LB), upper bound (UB), relative gap (RG) and the number of iterations (NI) needed to converge/execute in a given time limit. For convenience, we identify four sets of information with the following letters.

i: Integrated model with complete information

c: Decentralised model with complete information sharing

p: Decentralised model which shares only the production capacity

r: Decentralised model which shares only the resource (Train) availability information.

For example, the LB from a decentralised approach Mk, $(k = 1, \ldots, 4)$ can be compared with the LB from the integrated approach M0 using equation (6.4) as,

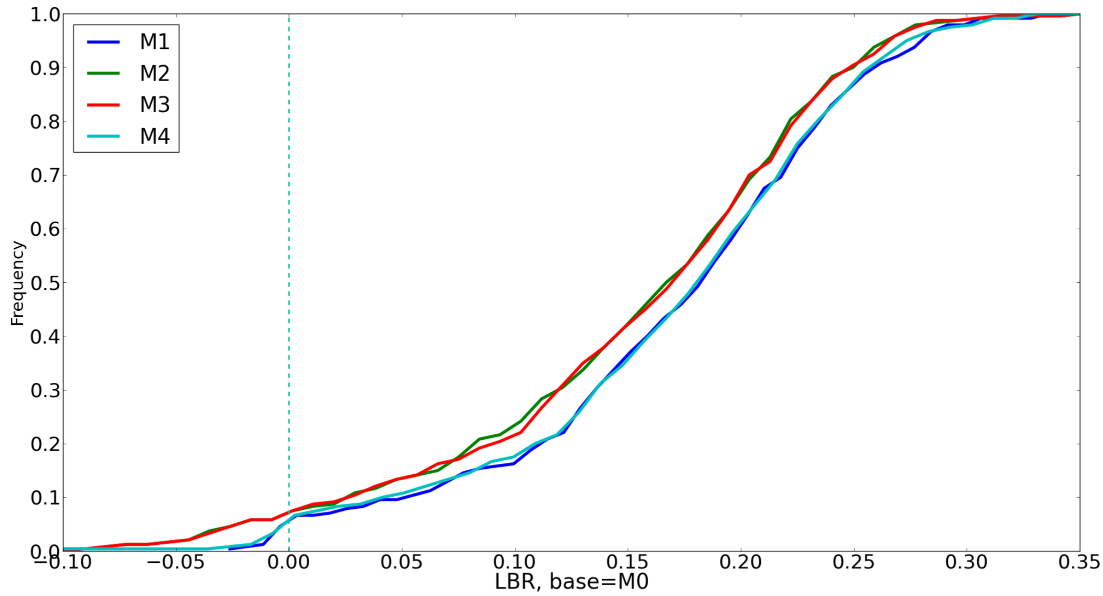
$$\mathcal{V}_{LB}(i) = \frac{LB_{\bar{i}} - LB_i}{\max(LB_{\bar{i}}, LB_i)} \tag{6.6}$$

where $\bar{i} = Mk, k = 1, \ldots, 4$ and $i = M0$. Similarly $\mathcal{V}_{LB}(p), \mathcal{V}_{LB}(r), \mathcal{V}_{UB}(.)$, $\mathcal{V}_{RG}(.)$ and $\mathcal{V}_{NI}(.)$ can also be defined using equation (6.4). Please note that 'maximum' is best for the lower bound and 'minimum' is best for the upper bound, the relative gap, and the number of iterations. This is similar to the ratios, LBR and UBR, defined in the previous chapters.

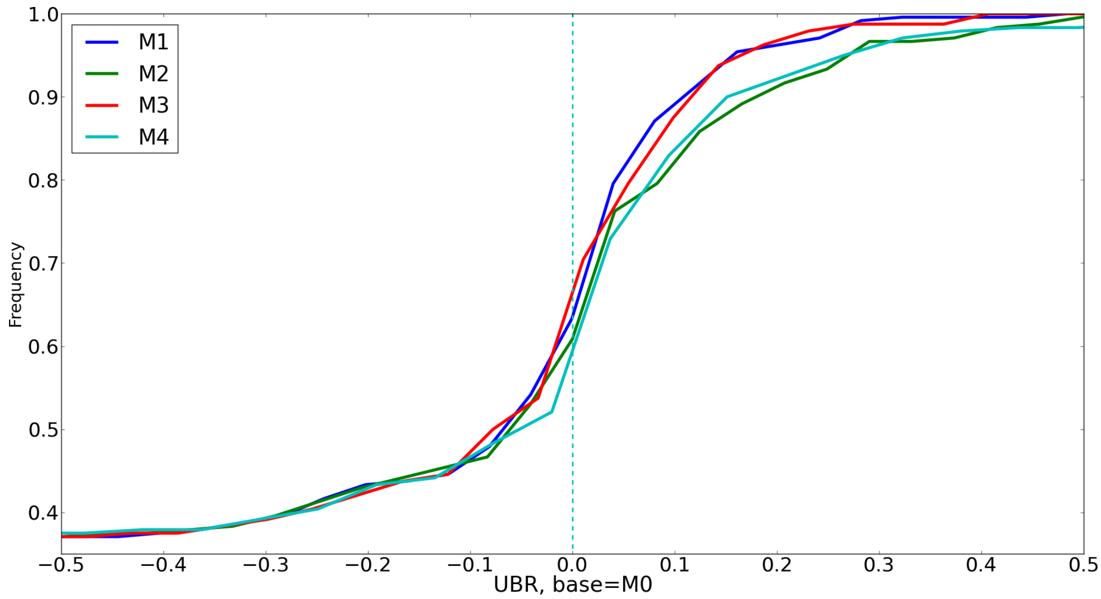
A positive $\mathcal{V}_{LB}(i)$ means that Mk is better, since it has a higher lower bound. Similarly for the upper bound, a positive $\mathcal{V}_{UB}(i)$ means that M0 is better. In some cases $\mathcal{V}_{UB}$ cannot be computed since the centralised model (M0) could not compute a single feasible solution in the given time limit. However, lower bound computation is very quick, and so, $\mathcal{V}_{LB}$ is available for all problem instances. These measures can be used to compare the overall performance.

6.5.1 Decentralised model versus centralised model

Figure 6.4 shows a comparison of all decentralised models with the centralised model. In Figure 6.4, $\mathcal{V}_{LB}(i)$ and $\mathcal{V}_{UB}(i)$ are plotted for all $k = 1, \ldots, 4$.



(a) Lower bound: $\mathcal{V}_{LB}(i)$



(b) Upper bound: $\mathcal{V}_{UB}(i)$

Figure 6.4: Comparison of the decentralised models with the centralised model

The model Mk is better with respect to both the bounds if and only if

$$\mathcal{V}_{LB}(i) > 0 \text{ and } \mathcal{V}_{UB}(i) < 0. \tag{6.7}$$

As seen in Figure 6.4, the decentralised models outperform the centralised one in 90% of the instances in LB comparison. Of that, more than 50% of data instances shows that the improvement is more than 20%. The $\mathcal{V}_{UB}$ shows that decentralised models are better in nearly 60% of the cases in a given time. If we run the experiments for a longer duration, then the percentage might go up. In 33% of the cases, the centralised model was not able to find a single feasible solution. It is interesting to see that 'no information-sharing' (M2) can also produce a better lower bound along with M3. The cases where the resource constraint is shared, M2 and M4, have a better upper bound. M3 and M2 display similar behaviours in Figure 6.4(a). However, M4 has inferior performance in LB computation. The upper bounds from M3 and M1, M2 and M4 are closer. Figure 6.5 shows the frequency of the relative gap observed with M0 and M1 from 240 data instances.

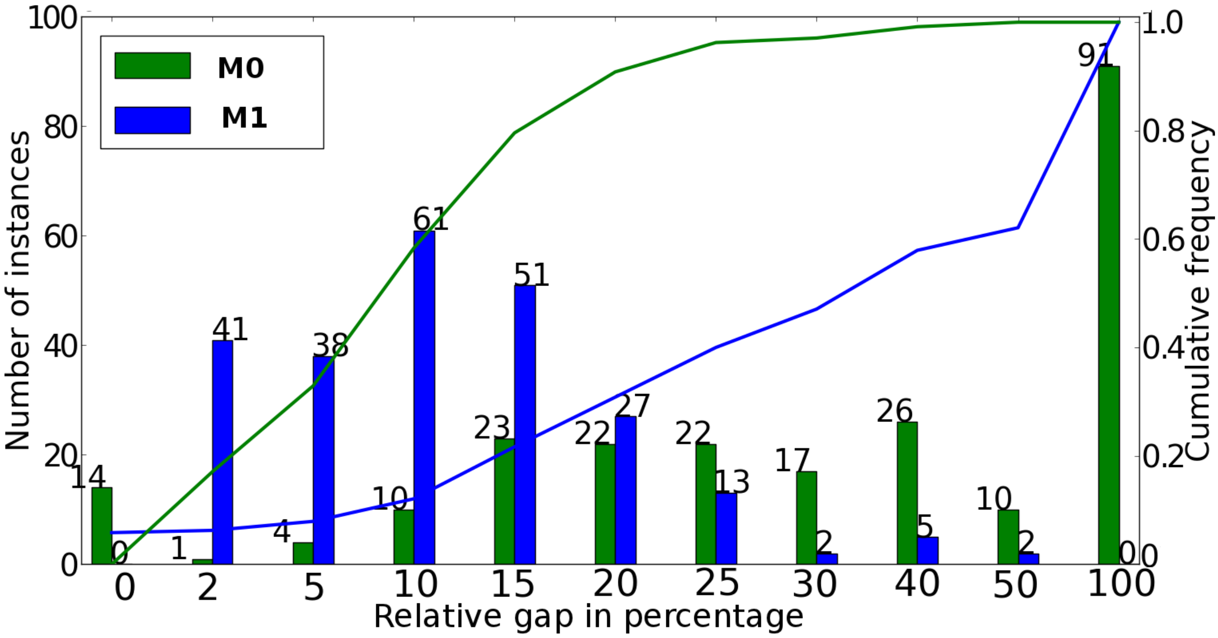


Figure 6.5: Distribution of relative gap and its cumulative frequency

6.5.2 Value of information

The value of an information is defined as a relative ratio (equation (6.4)) for all performance measures. In the previous section, we have seen the comparison of decentralised models with the centralised model using $\mathcal{V}_{LB}$ and $\mathcal{V}_{UB}$.

We have two pieces of vital information (i) production capacity and (ii) resource constraint. Production capacity can be analysed both in the presence and absence of a resource constraint and vice versa. In the pair (M2, M3), the resource constraint is not shared and production capacity is the only varying component. So, the pair (M2, M3) can be used to compare the benefit of sharing production information while resource information is not shared. Thus, the comparison of (M2, M3) and (M4, M1) is used to observe the impact of the production capacity; the comparison of (M2, M4) and (M3, M1) is for the resource availability (see Table 6.2). These pairs can be expressed in terms of the value of information as:

$$\begin{aligned}
(Mk, M0) &\equiv \mathcal{V}(i) \quad \text{and} \quad (M2, M1) \equiv \mathcal{V}(c = p + r) \\
(M2, M3) &\equiv \mathcal{V}(p|\bar{r}) \quad \text{and} \quad (M4, M1) \equiv \mathcal{V}(p|r) \\
(M2, M4) &\equiv \mathcal{V}(r|\bar{p}) \quad \text{and} \quad (M3, M1) \equiv \mathcal{V}(r|p)
\end{aligned}$$

In other words,

$$\begin{aligned}
\mathcal{V}(p|\bar{r}) &= \text{Value of production capacity while the resource constraint is private} \\
\mathcal{V}(p|r) &= \text{Value of production capacity while the resource constraint is public} \\
\mathcal{V}(r|\bar{p}) &= \text{Value of resource constraint while the production capacity is private} \\
\mathcal{V}(r|p) &= \text{Value of resource constraint while the production capacity is public} \\
\mathcal{V}(p + r) &= \text{Value of sharing resource constraint and the production capacity}
\end{aligned}$$

When we compute the value of information $a|b$, there is a change only in the first information a. For example, $\mathcal{V}_{LB}(a|b) = \dfrac{LB_{\bar{a}|b} - LB_{a|b}}{\max\{LB_{\bar{a}|b}, LB_{a|b}\}}$ and other measures are defined similarly.

Figure 6.6 provides the overall comparison of two decentralised models: a full information-sharing model, M1, and a no information-sharing model, M2. In nearly 80% of the cases, M1's lower bound is better than M2's. It is very intuitive that the information-sharing will help us to find a better lower bound. Interestingly, in the case of the upper bound, it is not the same. Moreover, the results show that in 40% of the cases no information-sharing is better than full information-sharing. The combined information is represented by $c = p + r$. Hence, LBR(M2,M1) is equal to $\mathcal{V}_{LB}(c)$.

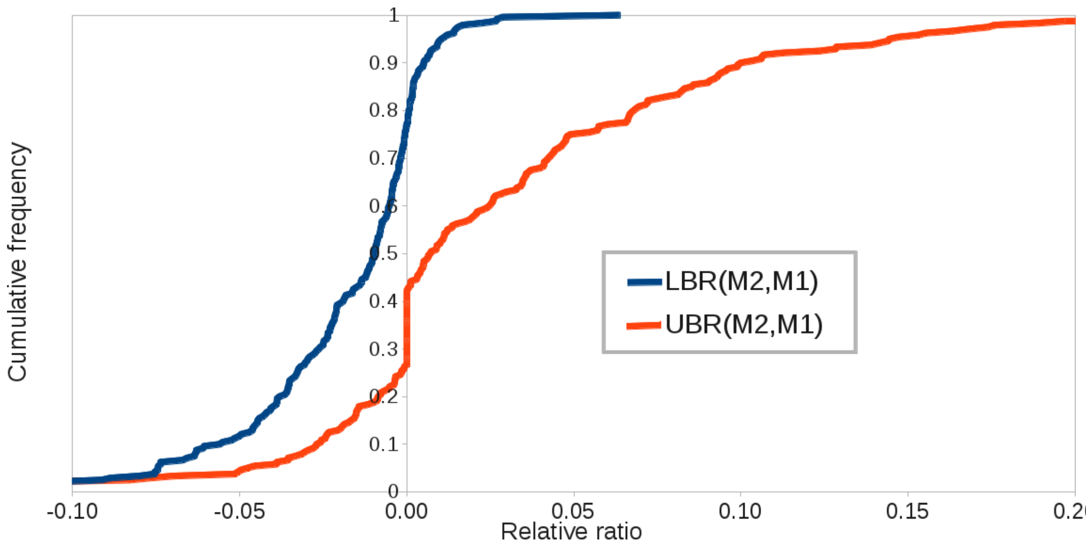
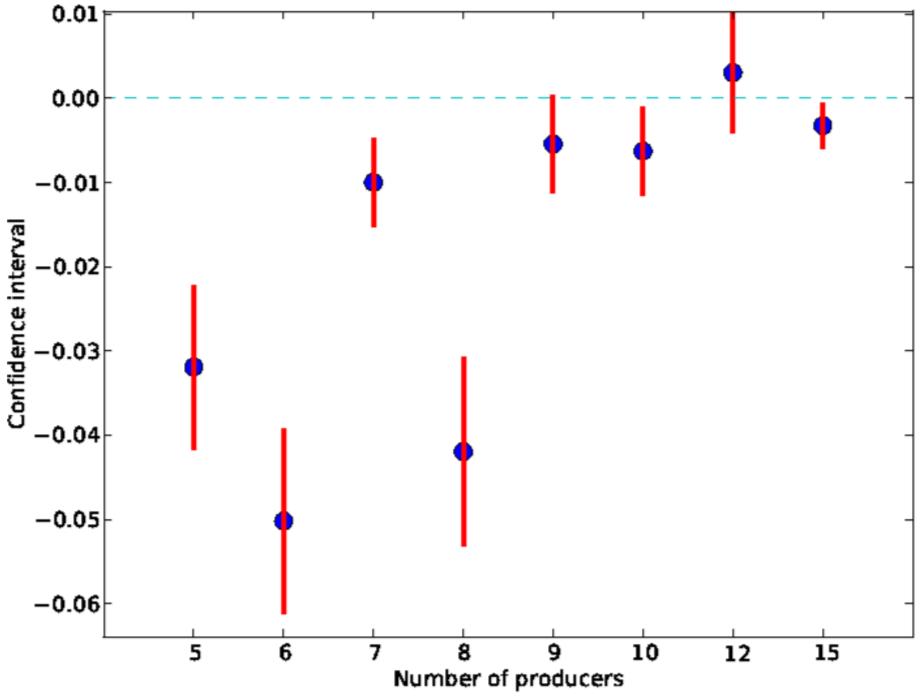
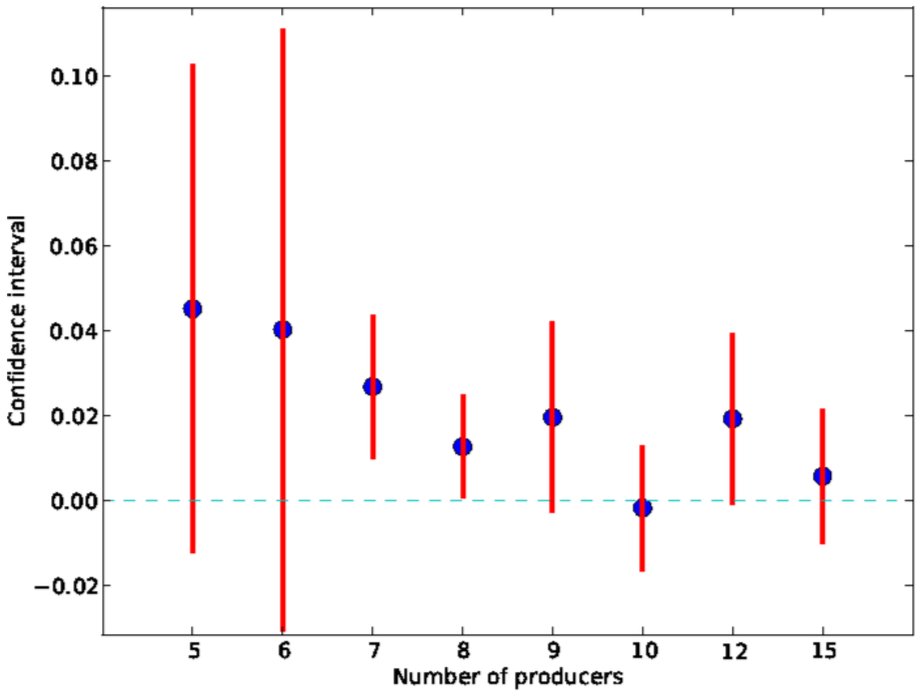


Figure 6.6: Comparison of decentralised models with full information (M1) and no information (M2)



(a) Lower bound: $\mathcal{V}_{LB}(c)$=LBR (M2, M1)



(b) Upper bound: $\mathcal{V}_{UB}(c)$= UBR (M2, M1)

Figure 6.7: Confidence interval comparison of decentralised models

Figure 6.7 compares the value of the information for each series with a 95% confidence interval. The interval is marked with red lines and the mean is marked with a blue circle. The mean LBR, being negative in all cases except for the twelve-producer case, implies that information-sharing is necessary to have tighter lower bounds. The mean UBR is positive/ close to zero in all cases. This means that the M2 upper bounds are equal to or higher than that of M1. This shows that information-sharing helps us to reduce the upper bound. However, due to the wider span of differences, Figure 6.6 shows a slight advantage of M2 over M1 in some cases.

Box-plots for the LBR and UBR are presented at four different time periods. The box highlights the entries from the first quartile to the third quartile and the median is marked with a red line inside the box. The mean is marked with the symbol '*'. In some cases, it is outside the box. The performance measures are computed using linear interpolation because the values were not available exactly at $t = 900, 1800, 2700$ and 3600 seconds.

Production capacity

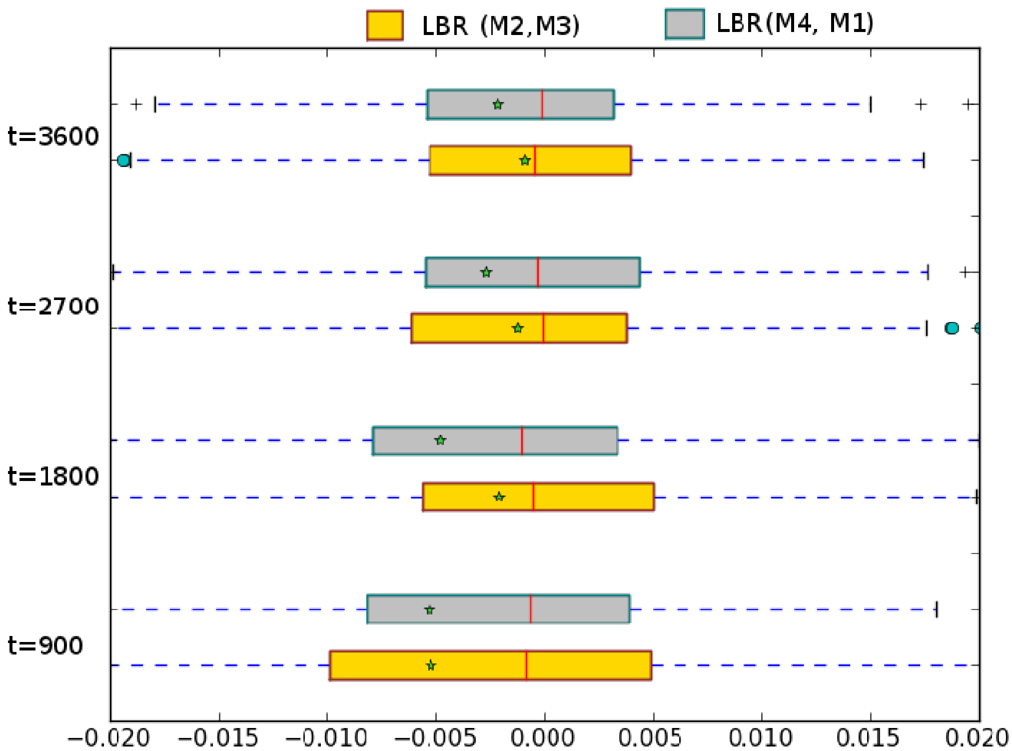
In Figure 6.8, the value of the information is compared using different performance measures. The impact of 'Production capacity' is examined by comparing M2 (Not shared) against M3 (Shared) under the assumption that the resource constraint is not shared in both cases. A similar case, where the resource availability is shared, can be observed by comparing M4 (Not shared) with M1 (Shared). In other words, the yellow box compares the value of production capacity information when the resource constraint is not shared and the grey box represents the same when the resource constraint is shared.

Except in Figure 6.8(a), a positive ratio means that sharing production-capacity information is better. In other words, while the information is shared, the model is able to get a smaller upper bound and relative gap. For the lower bound case, a negative ratio means the higher lower bound is achieved when the information is shared. The information regarding resource availability does not seem to boost the performance when the production capacity is shared. There is no significant difference in the number of iterations. The box-plots, as time progresses may not bring a steep rise/decline since the ratios are computed with respect to the best value available at that time.

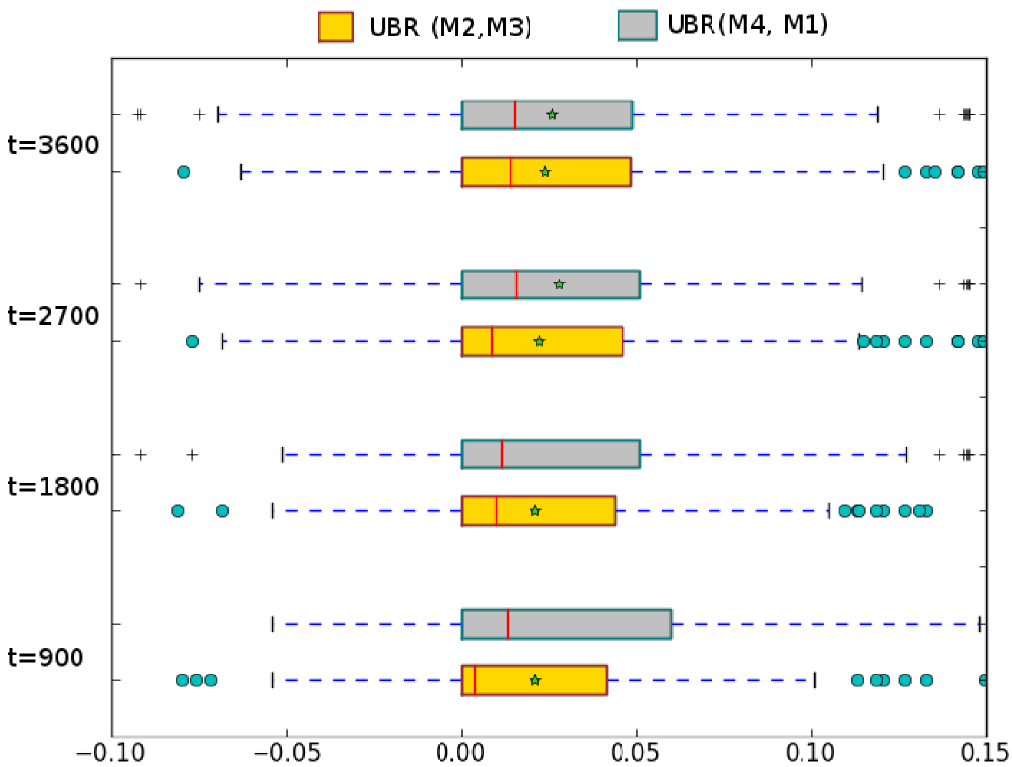
Resource Constraint

Figure 6.9 captures the value of the resource constraint with respect to different performance measures. As before, a negative ratio for the LB and a positive ratio for all other measures mean that the resource availability sharing has some advantage. For the resource constraint information, M2 is compared with M4 and M3 is compared with M1.

The LBR comparison shows that resource availability has a significant advantage. The upper bound and the relative gap are reduced significantly when the production information is also shared (See (M3,M1) case). The difference in the number of iterations is smaller when production information is shared.



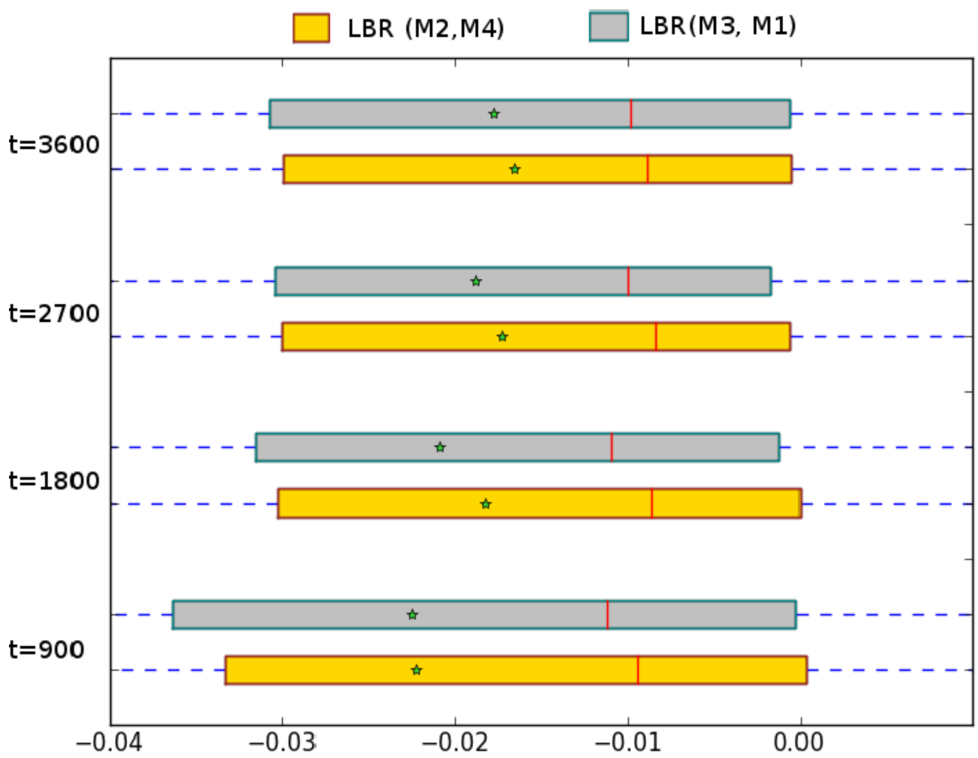
(a) Lower bound



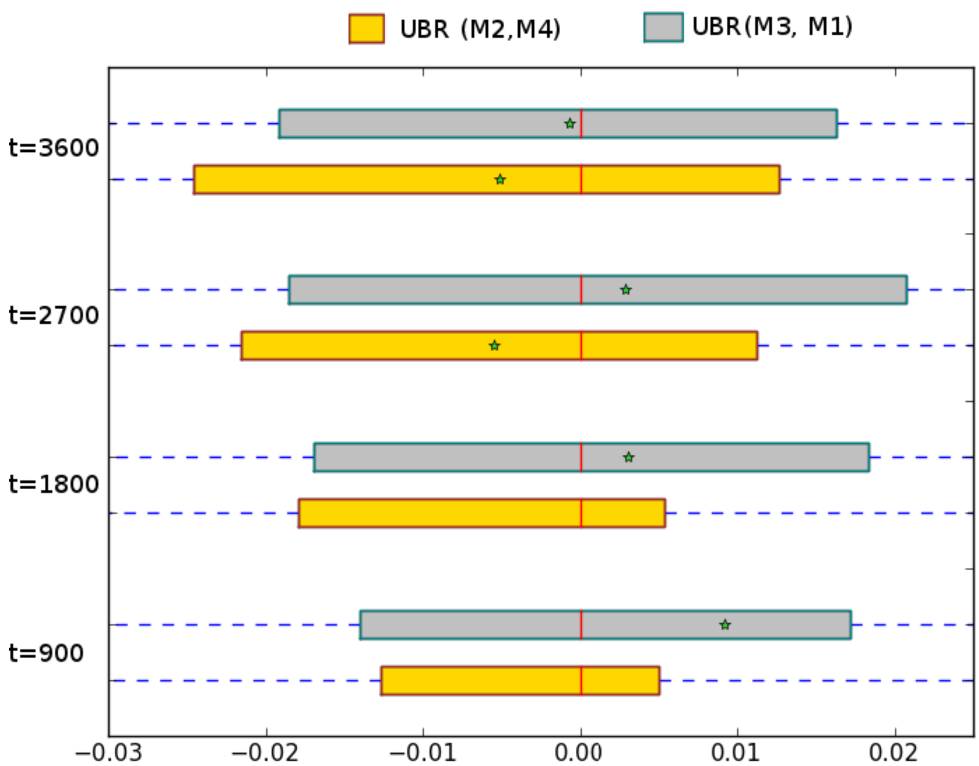
(b) Upper bound

Figure 6.8: Value of sharing the production capacity information

Figure 6.10 show the 95% confidence interval of the performance measures computed using Student's t-test. In this figure, all five variants are plotted together. The performance



(a) Lower bound



(b) Upper bound

Figure 6.9: Value of sharing the resource constraint

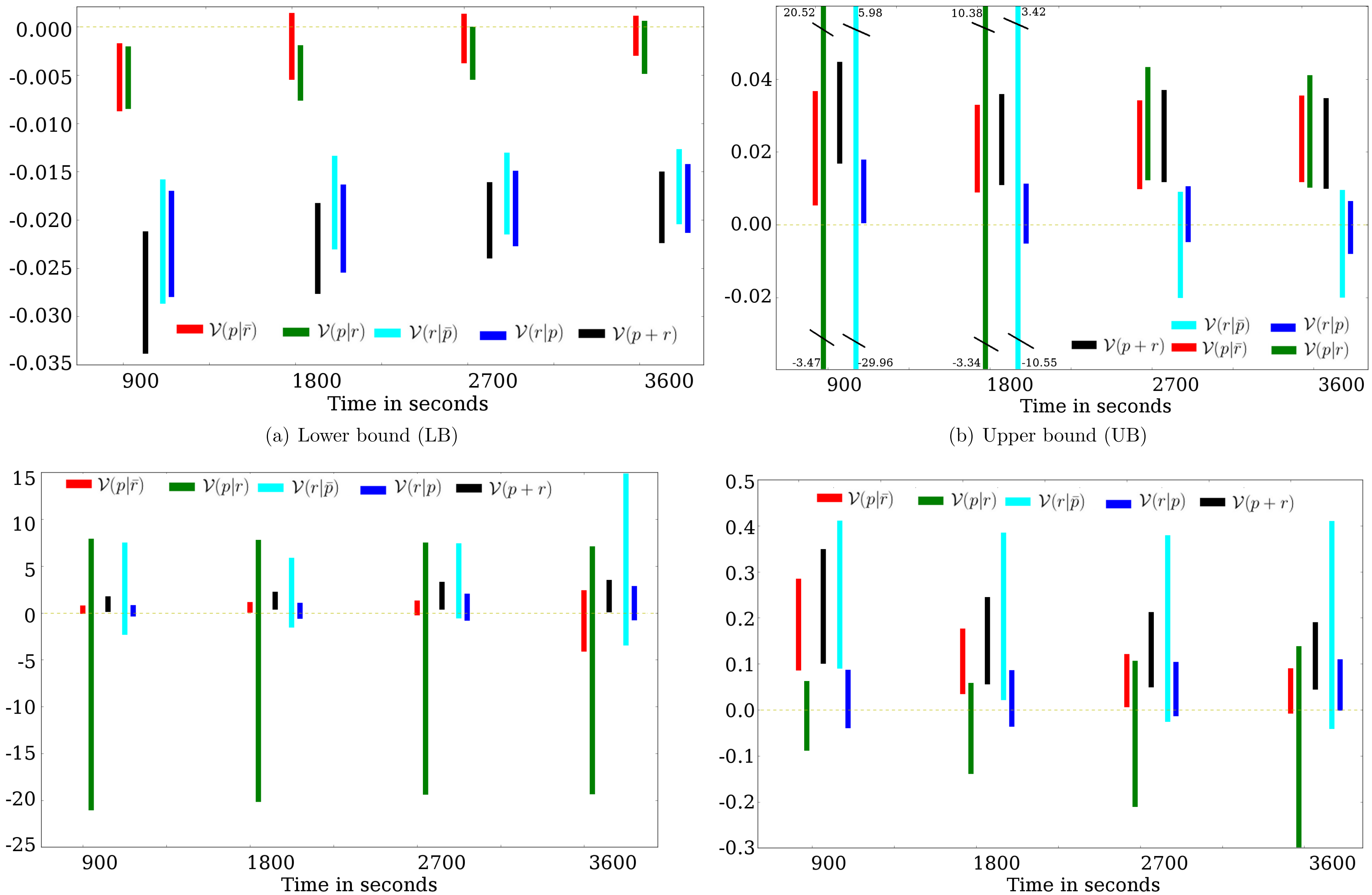
measures are computed using linear interpolation since the values were not available exactly at t = 900, 1800, 2700 and 3600 seconds.

Table 6.3 compares the central measures of the ratios and the gap obtained from different information-sharing scenarios.

Table 6.3: Mean and median comparison of different information-sharing settings

	LBR		UBR		Relative Gap	
	Mean	Median	Mean	Median	Mean	Median
$\mathcal{V}(p\|\bar{r})$	-0.0009	-0.0004	0.0236	0.0137	-0.8460	0.1477
$\mathcal{V}(p\|r)$	-0.0021	-0.0001	0.0256	0.0149	-6.1296	0.1943
$\mathcal{V}(r\|\bar{p})$	-0.0165	-0.0089	-0.0052	0.0000	5.7061	0.0583
$\mathcal{V}(r\|p)$	-0.0178	-0.0098	-0.0007	0.0000	1.0258	0.1427
$\mathcal{V}(p+r=c)$	-0.0187	-0.0098	0.0223	0.0081	1.7947	0.3876

The results show that the lower bounds of the iterative algorithm can be significantly improved by sharing necessary information. The overall comparison using the confidence intervals shows that resource availability information is more critical than production capacity information. The performance measures obtained at four different time points show the gradual change in the behaviour. Figure 6.10(a) shows that resource availability has a higher impact than production capacity information in tightening the lower bound. The confidence intervals in Figure 6.10(b) exhibit that sharing the production information reduces the upper bound. However, the resource information does not have much influence in improving the upper bound. As seen in Figure 6.10(d), the model requires a lesser number of iterations when the second information is shared, compared to that in the non-shared case. In general, we can see that as time progresses, the confidence interval stabilises.



(a) Lower bound (LB)

(b) Upper bound (UB)

(c) Relative gap (RG)

(d) Number of iterations (NI)

Figure 6.10: 95% confidence interval at different times for $\mathcal{V}_{RG}$ and $\mathcal{V}_{NI}$

6.6 Conclusions

In this chapter, the decentralised supply chain coordination was characterised and its components were discussed in detail. A decentralised iterative algorithm based on the Lagrangian relaxation was developed to compare the value of information-sharing. To make this algorithm 'decentralised', a lower bound was computed using a secure-sum method and the upper bound was computed with a decentralised heuristic. Using a coal supply chain as an example, the impact of sharing two vital pieces of information, (*i*) production capacity and (*ii*) resource constraints, are also examined. With different levels of information-sharing, four different versions of a decentralised model and the centralised model were compared. Our computational experiments showed that the decentralised methods are better or similar to the centralised model without sharing all of the information and a common model.

The impact of information in supply chain coordination was compared using different performance measures. The results and analyses show that the lower bounds of the iterative algorithm can be significantly improved by sharing the necessary information. An overall comparison using the confidence intervals shows that resource availability information is more critical than production capacity information. The performance measures obtained at different time points show a gradual change in the behaviour.

In this chapter and the previous chapters, our focus was a two-party coal supply chain. In the next chapter, we will move to a three-party case in which the terminal operations are also considered.

Chapter 7

A Decentralised Approach for Three-Party Coordination

In Chapter 6, we discussed a decentralised approach for two-party coordination. Here, we extend it to a three-party case. In a three-party coal supply chain, we consider the operations of the terminal along with the mines and the rail operator. The two-party models discussed in the previous chapters had only one shared resource manager. However, in the three-party case, there are two resource managers to link independent producers. The critical resources in the three-party coal supply chain are: (i) the limited number of trains managed by the rail operator; *and* (ii) the limited number of unloading slots at the terminal. In general, this can be viewed as a multi-resource constrained scheduling problem (RCSP) with two shared resources. As the number of linking constraints increases, it becomes difficult to decompose the decisions of different decision-making units (DMUs) in the supply chain. In the two-party case, the feedback from the resource manager to the DMUs was given in the form of one set of Lagrangian multipliers. However, in the three-party case, we need to have two sets of multipliers—each one corresponding to the linking constraint of the two resource managers.

Another analogy is to see the two-party coal supply chain as a bi-level system; in which case the three-party scenario becomes a multi-level system. Dempe [50] introduced the concept of bi-level programming. In this approach, decisions are made hierarchically at two levels, without any direct dependency. The proposed decentralised method is different from the hierarchical model. We have a set of independent DMUs at one level, which can be separated by relaxing the linking constraints. These DMUs can be solved and the requests for the resources also can be placed in parallel. Hence, we can solve the resource scheduling problem—simultaneously or hierarchically—for both the resource managers.

Chapters 4 and 5 present decomposed approaches for the two-party case based on Lagrangian relaxation (LR) and the column generation (CG), respectively. From an

Certain portions of this chapter are ready to be submitted to an International Journal [136] elaborated computational experiment (see Section 5.6.2), we learned that CG algorithms perform better than LR algorithms. The decentralised approach discussed in Chapter 6 considered only two parties—the mines and the rail operator (see Appendix A.1). So, we could not employ the CG algorithm since there was no possible decentralised technique, such as *secure-sum*, to find the value of a column. In this chapter, we have three parties—one set of independent producers and two shared resources. Therefore, we use a secure-sum method to compute the value of a column and solve the three-party case with the CG algorithm.

In this chapter, we explain the three-party coordination problem and the decentralised solution approach with the help of an example from the coal supply chain. Similar problems are seen in other supply chains too. In the three-party coal supply chain, the rail operator and the terminal act as common resource managers. Here, the mines make a set of requests for the trains, called as jobs, which can be scheduled by any of the resource managers. As seen in Figure 7.1, the rail operator and the terminal need to act on the job-scheduling problem simultaneously, rather than solve it hierarchically. Based on the level of information-sharing, there can be some differences in the decision-making process.

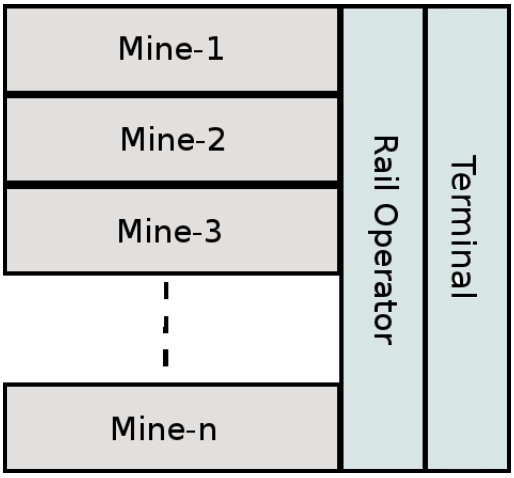


Figure 7.1: Interaction of DMUs in a three-party coordination problem

The aim of this chapter is to show the impact of an additional player and the role of information in the decentralised decision-making.

Section 7.1 gives a recap of the coal supply chain problem with some additional information regarding the third player. Section 7.3 presents an integrated model updated with a third player. A three-party decentralised approach and the CG-based algorithm are presented in Sections 7.4 and 7.5, respectively. Section 7.6 describes computational experiments and a comprehensive discussion of the results.

7.1 An expanded coal supply chain example

Three independent decision-makers—the mines, the rail operator and the terminal have been considered in this coal supply chain. They have their own decisions that need to be optimised. Some of the decisions are interlinked. The operations of the mines and the rail operator in this three-party case is same as in the two-party case (See Section 3.3.1 for the details).

The terminal handles ship arrival and loading and inventory management at the stockpiles. The terminal is connected to multiple mines with a common, shared rail operator. During train scheduling, the rail operator needs to consider some of the constraints of the mines and the terminal. The sequence of major decision-making in this three-party decentralised coal supply chain can be visualised as:

Step-1: The terminal receives orders from the ship with an expected delivery date

Step-2: The terminal splits these ship-orders and passes them to the mines in smaller quantities along with suitable due-dates

Step-3: The mines make production plans based on available information and request the rail operator to release a certain class of trains to ship the coal from the mines to the terminal

Step-4: The rail operator makes a suitable plan based on these requests, train availability and the restrictions of the terminal.

In a decentralised environment, the decision-makers in the supply chain do not have comprehensive information about their partners. The objective function of each partner is considered by them as private information. A quick overview of the DMUs is presented below.

The Mines have to independently plan their production. Information such as the production capacity, objective costs, order quantity and the like, are critical and private to the mines. Some of the information regarding the train class and its properties are shared between the mines and the rail operator. However, the mines do not have any information regarding the actual number of trains available with the rail operator and the unloading restriction at the terminal. The objective of this DMU is to minimise the total cost of inventory holding, the demurrage, and the cost of order placing.

The Rail Operator connects the mines and the terminal. It has access to certain information, such as the loading time at the mines and the unloading restriction at the terminal. However, it is unaware of the actual order of the mines or the terminal, the demurrage, and the production/inventory-holding capacity. The objective of the DMU is to minimise total weighted tardiness and the running cost.

The Terminal consolidates the supply from different mines. It does not bother about the production capacity of individual mines or the travelling time of trains. The only concern of the terminal is to get the required quantity of coal before the arrival of the ships, subject to the terminal holding capacity and unloading restrictions. The objective of this DMU is to minimise the total cost of the inventory holding and the under stock.

If we consider the overall system as a black box, then the input to this system is the ship-orders and the output is the train schedule and the production plan. A ship-order is converted into a set of mine-orders and passed to the mines with appropriate due-dates. Based on the mine-orders and the resource class properties, the mines can request for the resources (that is, the trains and the unloading slots at the terminal). Such requests are considered as *jobs*. The overall scheduling is dependent on the jobs created intermediately by the mines. Once the mines create jobs, the resource managers—the rail operator and the terminal—can give their feedback to the mines to alter their decisions to make their requests globally acceptable (feasible).

Figure 7.2 shows a schematic diagram of the operations in a three-party coal supply chain. The figure exhibits an iterative sequential decision-making process. In some cases, if the information regarding the terminal is available to the rail operator, the decision-making for the rail operator and the terminal can be merged. This implies that the resource scheduling considering both the resource managers can be done together.

As mentioned earlier in Section 3.1, here as well, we can see the presence of information asymmetry amongst players in the modelling and operational framework. The decisions in this example are influenced by the objective function of the DMUs. In some instances, the different objectives can conflict or be similar or have different scales and units. Therefore, an integrated approach is not applicable in such supply chains. However, we use a conceptual integrated model for benchmarking our decentralised approaches. The integrated model is nothing but a model that combines the constraints and objectives of all the DMUs. The integrated model also helps us to explain decomposition and distributed decision-making.

7.2 Mathematical models

In this section, we present the mathematical models used in the decentralised decision-making approach. There are changes to a few variables compared to the models used

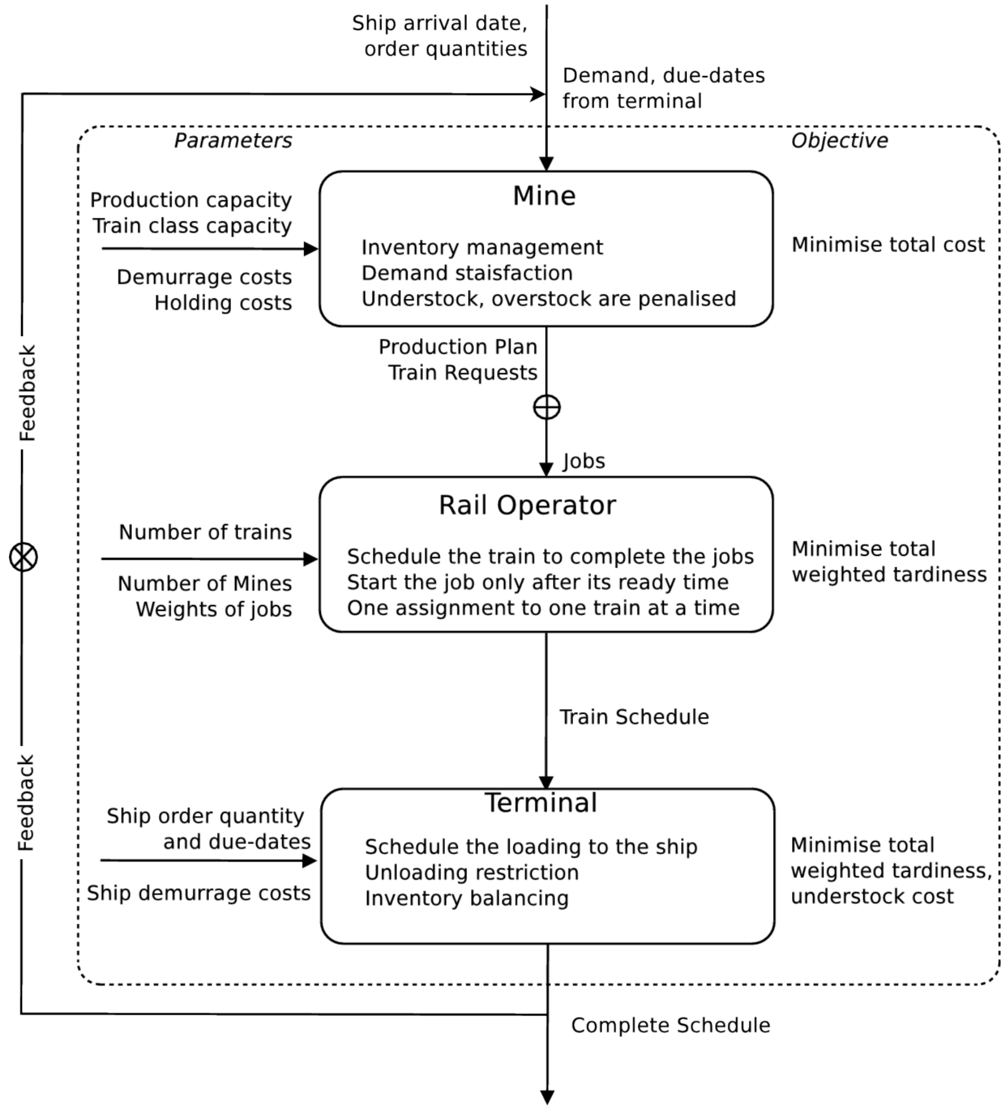


Figure 7.2: Schematic diagram of a three-party coal supply chain

in the two-party case. The mathematical models for the three party case assumes five segments in a train trip: (i) journey to the mines; (ii) loading at the mine; (iii) journey to the terminal; (iv) unloading at the terminal; and (v) journey back to the garage. It is noted that in the two-party case, only the first three segments were considered in a train trip. The cumulative supply, train availability and the like, are modified to include this change.

In Figure 7.3, the variable y_j^t represents the decision that indicates that job j has delivered the coal at the terminal, which is equivalent to z in the rail operator's decision model for the two-party case (see Section 5.3.1). Therefore, the production planning model presented in Section 4.3 and the job-scheduling model presented in Section 5.3.1 have to be updated accordingly. The decision-making model of the terminal is a job-scheduling model, similar to that of the rail operator, but with a different objective function.

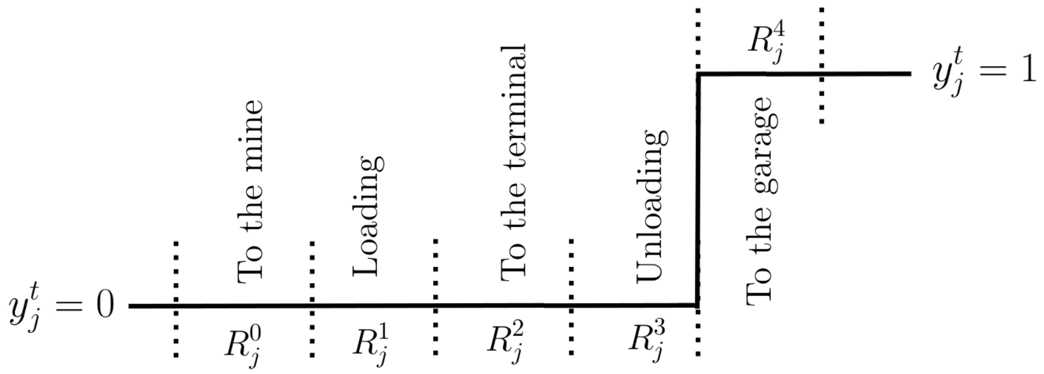


Figure 7.3: Different stages in a garage-to-garage trip

In the next sections, the decision-making models for each DMU are presented in detail.

7.2.1 Production planning model

Most of the parameters, the decision variables and the constraints given in Section 4.3 are used as is. The updated parameters and the constraints are listed below.

Let $i \in \{1, 2, \ldots, I\}$, $t \in \{0, 1, \ldots, T\}$, $u \in \{1, 2, \ldots, U\}$ and $w \in \{1, 2, \ldots, W\}$ be the indices of producers, time periods, orders and train (resource) classes, respectively.

Parameters:

- R_w^0 forward travel time required for the trains in class w to reach a mine from the garage,
- R_w^1 loading time required for the trains in class w at the mines,
- R_w^2 return travel time required for the trains in class w, from a mine to the terminal,
- R_w^3 unloading time required for the trains in class w,
- R_w^4 return travel time required for the trains in class w, from the terminal to the garage,
- L_w $= \sum_{v=1}^{3} R_w^v$, total time required to pick the cargo from the mine and to deliver at the terminal.

The overall objective, Z_i, is to minimise the total system cost. This includes the cost of holding inventory at the mines, the overstock cost at the terminal, demurrage cost and the total cost of requesting trains.

$$\min \ Z_i = \sum_t \left[\theta^t H^t + \tau^t O^t + \psi^t C^t + \sum_w (\eta_w^t - \eta_w^{t-1}) A_w^t \right] \tag{4.1}$$

subject to the constraints (4.2) - (4.10) and (4.12) - (4.18).

In a decentralised setting, the mines do not have information regarding train availability. This means that, equivalent to (4.11), there is no resource availability constraint at the mines, unless it is mentioned explicitly.

7.2.2 Train scheduling

The key constraint of the rail operator is that the total number of trains of a class w running at any time t cannot exceed the total number of trains in that class (see constraint (4.19)). This means

$$\sum_i (\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v}) \leq K_w \qquad \forall\ w, t \tag{7.1}$$

where η_{iw}^t is the total number of trains from class w requested on or before time t by the producer i. Suppose constraint (7.1) is a public information, then a relaxed resource constraint can be added for each DMU. This means that, DMU-i can have an additional constraint, similar to (4.11).

$$(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v}) \leq K_w \qquad \forall\ i, w, t \tag{7.2}$$

A resource request (η_{iw}^t) can be seen as a job with a release date and a due-date. An independent job-scheduling model, equivalent to the one mentioned in Section 5.3.1, is possible for train scheduling. Then, the main decision variable in this model is

$$z_j^t = \begin{cases} 1 & \text{if the job } j \text{ is delivered at the terminal on or before time } t, \\ 0 & \text{otherwise.} \end{cases}$$

The relation between η_{iw}^t and z_j^t can be represented as $T_j = w, M_j = i$ (see Section 5.3.1 for more details). Hence, the resource constraint (7.1) can be rewritten in terms of z as

$$\sum_{j|T_j=w} \left(z_j^{t+\sum_{v=0}^3 R_j^v} - z_j^{t-R_j^4} \right) \leq K_w \quad \forall\ w, t. \tag{7.3}$$

If a train from class w is requested to reach the mine i at time t', then $\eta_{iw}^{t'} - \eta_{iw}^{t'-1} = 1$ and this implies that some job j (such that $T_j = w$ and $M_j = i$) has to start at $t' - R_j^0$ and deliver it at $t' + L_j$. Or equivalently,

$$\left(z_j^{t'+L_j} - z_j^{t'+L_j-1} \right) = 1. \tag{7.4}$$

The objective 7.5 of the rail operator is to minimise the running cost and the total weighted tardiness and earliness.

$$\min Z_R = \sum_j \sum_{t>0} (\bar{C}_j^t + \bar{G}_j^t)(z_j^t - z_j^{t-1}) + \sum_t \sum_w \left[\xi_j^t \sum_{j|T_j=w} \left(z_j^{t+\sum_{v=0}^{3} R_j^v} - z_j^{t-R_j^4} \right) \right] \tag{7.5}$$

where $\bar{C}$ and $\bar{G}$ are the weighted tardiness cost and the weighted earliness cost, respectively. Do note that the early delivery of a job happens only if the production information is shared with the rail operator.

In a decentralised framework, we can see a request as a job with due-dates as specified by the mines. However, in the integrated problem, the due-date of a job is specified internally as a joint decision. Therefore, in the integrated model, all jobs are completed on time. Hence, the contribution of (7.5) in the objective of the integrated model can be written in terms of η as

$$\min \ Z_R' = \sum_t \sum_w \left[\xi_w^t \sum_i (\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^{4} R_w^v}) \right] \tag{7.6}$$

7.2.3 Terminal operations

Assumptions: The terminal has information regarding the ship arrival at the terminal and ship-order quantities. The date of arrival is the expected arrival date. We represent the order quantity and due-date pair with (Υ_l, Φ_l). The terminal uses its experience and previous records to split the ship-order into smaller orders for the mines (producers). Appropriate due-dates for the mines will be computed with respect to the ship-orders and the terminal constraints. The order pair for the mine is (Q_u, Fu). We assume that the splitting of ship-order to mine-order is done outside our decision-making model.

The job-scheduling model of the rail operator and the terminal are similar. The decision variable, z represents the status of a job in the rail operator model, and similarly, y in the terminal model. This means that, we can retain the same job set, formed from the requests of the mines, for the rail operator and the terminal. However, the objectives of both these models are different. The terminal model has an additional decision variable to compute the lateness in meeting the ship-orders. The lateness penalty is computed based on the short quantity (under-stock) to meet a particular order.

Let us define

Parameters:

Υ_l $\quad l^{\text{th}}$ ship-order quantity

Φ_l $\quad$ Expected date of arrival (due-date) of l^{th} ship-order

Ψ^t Cumulative quantity of the cargo loaded into ships by time t

S_j Release (ready) time of job j. This means that a sufficient quantity is ready at the mine for loading job j

F_j Due-date of job j

$\bar{B}^t$ Under-stock cost at time t

$\bar{D}_l$ Actual delivery date of an order l

K' Upper limit on the number of trains which can unload at the same time.

Decision Variables:

τ^t Inventory level at time t, we assume the initial stock is $\tau^0 = 0$.

$$y_j^t = \begin{cases} 1 & \text{if the job } j \text{ is completed (delivered) on or before time } t, \\ 0 & \text{otherwise.} \end{cases}$$

χ^t Under-stock quantity at time t

Objective: The objective of this job-scheduling model is to minimise the total weighted tardiness and earliness of the jobs, the cost of inventory holding and the late delivery of ship-orders. This guides the model to schedule the jobs as close to their due-dates as possible. Kindly note that, there are two types of tardiness in this model. One is with respect to the job (j)'s due-date (F_j) and the second one is with respect to the ship-order l's due-date (Φ_l).

$$\min \sum_j \sum_{t>0} (\bar{C}_j + \bar{G}_j^t)(y_j^t - y_j^{t-1}) + \sum_t \left(O^t \tau^t + \bar{B}^t \chi^t \right) \tag{7.7}$$

subject to:

Inventory balancing constraint.

$$\tau^t = \tau^{t-1} + \sum_j y_j^t V_j - (\Psi^t - \Psi^{t-1}) \qquad \forall\, t \tag{7.8}$$

Cumulative supply cannot exceed the demand quantity.

$$\Psi^t \leq \sum_{\{l|t \leq \Phi_l\}} \Upsilon_l \qquad \forall\, t \tag{7.9}$$

Unloading restriction: the maximum number of simultanoues unloading is limited.

$$\sum_j (y_j^{t+R_j^3} - y_j^t) \leq K' \qquad \forall\, j, t \tag{7.10}$$

Once the job is completed, it stays as completed.

$$y_j^{t-1} \leq y_j^t \qquad \forall\, j, t \tag{7.11}$$

All jobs must be completed at the end of the planning horizon.

$$y_j^T = 1 \qquad \forall j \tag{7.12}$$

A job (train) cannot reach the mine before its 'ready time'.

$$y_j^{t+L_j} = 0 \qquad \forall\ t < S_j, j \tag{7.13}$$

All orders should be met at the end of the planning horizon.

$$\Psi^T = \sum_l \Upsilon_l \tag{7.14}$$

Demand can be expressed as the sum of the actual supply and the under-stock quantity.

$$\Psi_t + \chi^t = \sum_{\{l|\Phi_l \leq t\}} \Upsilon_l \qquad \forall\ t \tag{7.15}$$

Boundary conditions.

$$\tau^t \geq 0, y_j^0 = 0, \Psi^t \geq 0, \chi^t \geq 0 \tag{7.16}$$

The decision-making model of the terminal does not distinguish between the individual mines. The main concern of the terminal is to get appropriate accumulative quantity to meet the ship-orders.

Most of the constraints mentioned above can be rewritten in terms of the decision variables from other models. For example, Constraint (7.10) can be written in terms of η as

$$\sum_{i,w}(\eta_{iw}^{t-R_w^1-R_w^2} - \eta_{iw}^{t-R_w^1-R_w^2-R_w^3}) \leq K' \qquad \forall t \tag{7.17}$$

The decision variables z_j^t and y_j^t can be interpreted in a similar way. Both the rail operator and the terminal models accept the same set of jobs.

Similar to the rail operators' objective, the first component in (7.7) does not have any contribution in the objective of the integrated model. The second component can be directly added to the integrated model's objective.

$$\min Z_T' = \sum_t \left(O^t \tau^t + \bar{B}^t \chi^t\right) \tag{7.18}$$

The under-stock quantity χ^t can be split for each mine and written as $\chi^t = \sum_i \chi_i^t$.

7.3 The three-party integrated model

The integrated model (equivalent to the IM in Section 4.4) can be formed by combining the above DMUs. The three-party integrated model is denoted by IM-3 in the rest of the thesis. The objective of IM-3 has three components from three DMUs, specifically, constraints (4.1), (7.6) and (7.18). The cost of overstock considered in (4.1) is an internal cost in the integrated model, which is paid by the mines to the terminal. However, in the terminal model, this is considered again as an inventory holding cost. Therefore, we exclude the overstock cost from the objective of the mines and consider it in the objective of the terminal. Therefore, the contribution of the mines to the objective of the IM-3 is

$$\min \ Z_i' = \sum_t \left[\theta^t H^t + \psi^t C^t + \sum_w (\eta_w^t - \eta_w^{t-1}) A_w^t \right] \tag{7.19}$$

There is a relation between the demand D_i^t and the ship-orders (Υ_l, Φ_l). The cumulative demand from all the mines at the due-date of k^{th} order (that is, $t = \Phi_k$) is equal to the sum of the ship-orders until that time.

$$\sum_i D_i^{\Phi_k} = \sum_{l=1}^{k} \Upsilon_l \qquad \forall\ k \tag{7.20}$$

Then the complete integrated model (IM-3) can be represented as follows:

$$\min Z = \sum_i Z_i' + \sum_t \sum_w \left[\xi_w^t \sum_i \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v} \right) \right] + \sum_t \left(O^t \tau^t + \bar{B}^t \chi^t \right) \tag{7.21}$$

$$= \sum_i \left[Z_i + \sum_t \sum_w \xi_w^t \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v} \right) + \sum_t \bar{B}^t \chi_i^t \right] \tag{7.22}$$

subject to the constraints (4.2)-(4.18) for each i, and

$$\sum_i (\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v}) \leq K_w \qquad \forall\ w, t \tag{7.1}$$

$$\sum_{i,w} (\eta_{iw}^{t-R_w^1-R_w^2} - \eta_{iw}^{t-R_w^1-R_w^2-R_w^3}) \leq K' \qquad \forall t \tag{7.17}$$

$$\sum_i D_i^{\Phi_k} = \sum_{l=1}^{k} \Upsilon_l \qquad \forall\ k \tag{7.20}$$

As we mentioned earlier, the integrated model is created to benchmark our decentralised scheme and to explain the decentralised approach as a Lagrangian relaxation of this integrated problem.

7.4 A three-party decentralised approach

In this section, we propose a decentralised decision-making approach based on column generation. A decentralised framework proposed for two-party coordination (see Figure 6.3) is updated for the three-party case as follows.

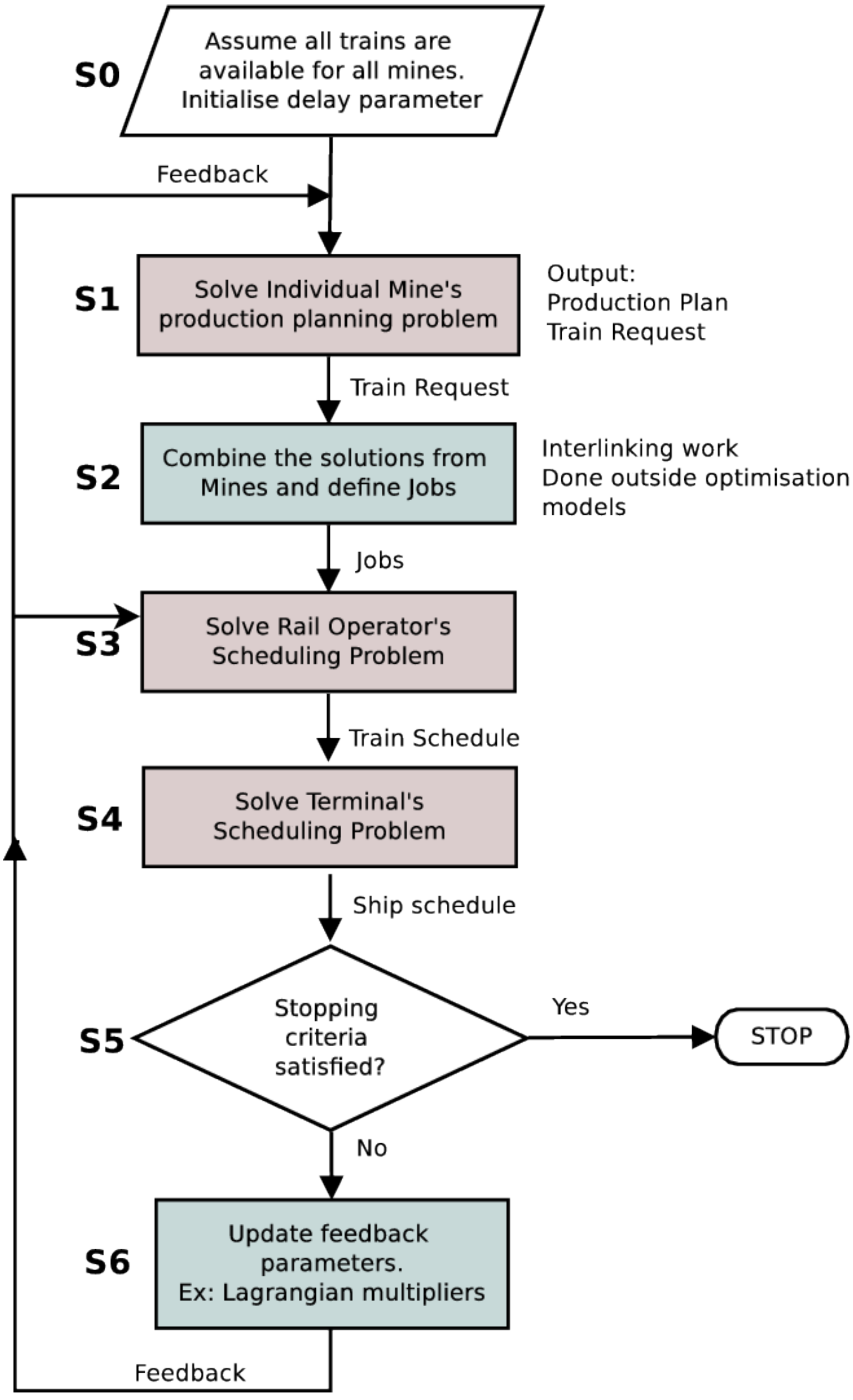


Figure 7.4: Flowchart for a three-party coordination approach

Section 5.2 explains the basic concepts and framework of column generation schemes. We need to decompose the decisions into easily solvable sub-problems and formulate a master problem to link these sub-problems. As we see from the mathematical models, the train availability constraint of the rail operator and the unloading restriction at the terminal are the bottlenecks in coordinating the decisions of multiple mines. If we relax these linking constraints (7.1 and 7.17), then the production planning problems can be distributed. In Chapter 5, we have seen a popular decomposition technique—column generation (CG). Here also, the decomposition approach based on the CG technique is adapted. Then, production-planning and resource-scheduling are considered at the sub-problem and the master problem levels, respectively.

Let λ and μ be the corresponding Lagrangian multipliers of (7.1) and (7.17), then the Lagrangian objective function of Z can be written as

$$\begin{aligned} LRR(Z,\lambda,\mu) = & \sum_i \left[Z_i + \sum_t \sum_w \xi_w^t \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v} \right) + \sum_t \bar{B}^t \chi_i^t \right] + \\ & \sum_t \sum_w \lambda_w^t \left(\sum_i \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v} \right) - K_w \right) + \\ & \sum_t \mu^t \left(\sum_{i,w} \left(\eta_{iw}^{t-R_w^1-R_w^2} - \eta_{iw}^{t-R_w^1-R_w^2-R_w^3} \right) - K' \right) \end{aligned} \tag{7.23}$$

$$LRR(Z,\lambda,\mu) = \sum_i \bar{Z}_i - \sum_t \left[\sum_w \lambda_w^t K_w + \mu^t K' \right] \tag{7.24}$$

$$\begin{aligned} \text{where} \quad \bar{Z}_i(\lambda,\mu) = & Z_i + \sum_w \sum_t (\xi_w^t + \lambda_w^t) \left(\eta_{iw}^{t+R_w^0} - \eta_{iw}^{t-\sum_{v=1}^4 R_w^v} \right) + \\ & \sum_t \mu^t \left(\sum_w \left(\eta_{iw}^{t-R_w^1-R_w^2} - \eta_{iw}^{t-R_w^1-R_w^2-R_w^3} \right) \right) + \bar{B}^t \chi_i^t \end{aligned} \tag{7.25}$$

In an iterative implementation of the CG algorithm, the lower bound is computed from the Lagrangian objective function (7.24). An upper-bound to the optimal objective value and an upper-bound to the lower-bound are computed from the master problem and their relaxed master problem, respectively.

7.4.1 Master problem

Let x_{ic} represents the c^{th} column from mine-i and V_{ic} be the cost equivalent to $\bar{Z}_i(0,0)$. Then the master problem is

$$[\mathbf{MP}] \qquad \mathcal{V}_{MP} = \min \sum_i \sum_c V_{ic} x_{ic} \tag{7.26}$$

$$\text{subject to} \qquad \sum_c x_{ic} = 1 \qquad \forall i \tag{7.27}$$

$$\sum_{i,c} (\bar{\eta}_{iwc}^{t+R_w^0} - \bar{\eta}_{iwc}^{t-\sum_{v=1}^4 R_w^v}) \le K_w \qquad \forall\, w,t \tag{7.28}$$

$$\sum_{i,c,w} (\bar{\eta}_{iwc}^{t-R_w^1-R_w^2} - \bar{\eta}_{iwc}^{t-R_w^1-R_w^2-R_w^3}) \leq K' \qquad \forall t \tag{7.29}$$

$$x_{ic} \in \{0, 1\} \tag{7.30}$$

where, $\bar{\eta}_{iwc}^t$ is the cumulative resource utilised by mine-i in column c at time t.

In our proposed solution approach, comprehensive information is not shared between the DMUs. Each decision-maker has its own private objective costs and some private information. The rail operator acts as an *honest broker* to connect the mines and the terminal. As we see in Chapter 5, if there a centralised player to manage the DMUs, then the CG algorithm provides good solutions. However, in the decentralised environment, we do not expect similar information-sharing and central coordination. So, there will be a loss in the quality of the solution.

7.5 Decentralised column generation algorithm

As we discussed in the previous section, a master problem is defined with the train availability constraint of the rail operator and the unloading constraint at the terminal. The sub-problem for producer-i ($\mathcal{P}_i$) is min $\bar{Z}_i(\lambda, \mu)$ subject to (4.2)-(4.18). In the relaxed master problem (RMP), the variable x_{ic} is relaxed to take any value between zero and one (similar to CG model in Section 5.2.1). Algorithm 8, shown below, presents a CG-based algorithm (similar to Algorithm 5 in Section 5.4.3) for three party coordination. The three-party decentralised decision-making approach presented in Algorithm 8 is referred to as CG-3.

Even though the overall procedure is the same as in Algorithm 5, Algorithm 8 does not require any central information, especially, in computing the bounds and in updating the multipliers. Details of decentralised computation are provided in Section 7.5.2. In chapters 4 and 5, we discussed many strengthening techniques and methods for stabilisation for two-party decomposed approaches. Some of them are implemented in Algorithm 8 too. They are,

1. Volume algorithm [14] to stabilise the dual prices. For example, instead of directly taking λ, a stability centre $\bar{\lambda} = \alpha\bar{\lambda} + (1-\alpha)\lambda$ is considered.
2. In step 8, instead of adding one column from each mine, more than one column is added for faster convergence (see [79]).
3. To manage the solution pool of RMP, the non-contributing columns are removed (see step 15 of Algorithm 8).
4. It is difficult to create globally-feasible solutions directly by combining the columns

(requests) from the mines. The job-scheduling model (combining models given in Section 7.2.2 and Section 7.2.3) allows the delay of some of the requests from the mines in order to get feasible solutions.

Algorithm 8 A CG-based algorithm for three-party coordination (CG-3)

Initialise the bounds, counters, etc.
Initialise the solution pool $\mathcal{S}^0$ with a feasible solution (column).
while (Elapsed time $< TimeLimit$) **do**
    Solve the RMP^k with a solution pool $\mathcal{S}^k$ and compute the dual variables $\lambda^{(k)}$ and $\mu^{(k)}$.
    For each i, solve $\mathcal{P}_i$ with $\lambda^{(k)}$ and $\mu^{(k)}$.
    Pass the columns to the rail operator and the terminal and compute the objective cost using the secure-sum method. /* *This step does not require any central information.* */
    Compute $LB^{(k)} = LRR(\lambda^{(k)}, \mu^{(k)})$ (see equation (7.24)).
    Update the solution pool with all new columns from $\mathcal{P}_i$.
    Solve the job-scheduling model of the rail operator and the terminal to generate a few more globally feasible columns. /* *This step does not require any central information.* */
    if ($LB^k > LB^*$) **then**
        Set $LB^* = LB^k$.
    if $((\mathcal{V}_{RMP}(\mathcal{S}^k) - LB^*)/\mathcal{V}_{RMP}(\mathcal{S}^k) < \epsilon)$ **then**
        Evaluate the integer version of the master problem to identify the best-known upper bound UB^* and corresponding integer solution.
        break /**Prices are stable**/
    Remove columns which are non-basic and do not contribute to the optimal solution of RMP for ten continuous iterations.
    Use Volume Algorithm to update the multipliers.
    $k = k + 1$
Report LB^*, UB^* and $gap = (UB^* - LB^*)/UB^*$.

Figure 7.5 shows a pictorial representation of Algorithm 8.

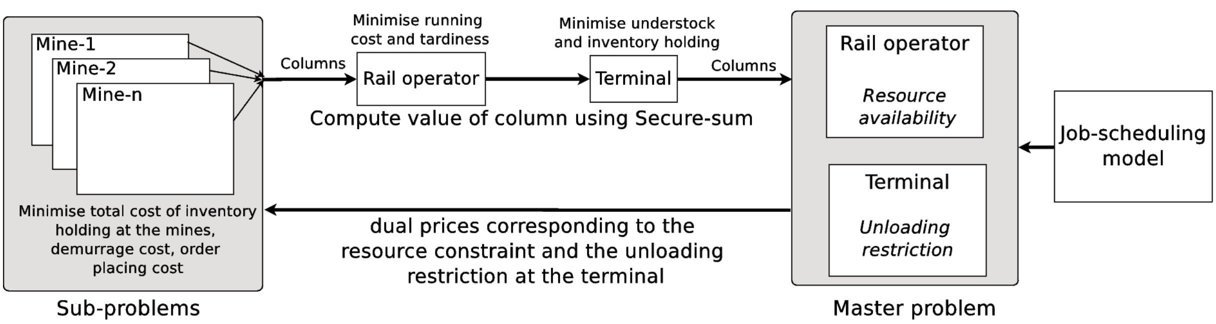


Figure 7.5: A decentralised algorithm for three-party coordination

7.5.1 Information sharing

In the previous chapter, we have seen that information-sharing has an impact on the performance of decentralised approaches. In a decentralised environment, information regarding the objective of each DMUs is private to them. Figure 7.6 shows the visibility of some of the important information in the three-party coal supply chain. The information shared between the 'Mine' and 'Terminal' is given in the first cell. The information shared between 'Mine' and 'Mine' is its private information. Similarly, other cells also can be explained.

Figure 7.6: Information visibility in a decentralised three-party coal supply chain

	Terminal	**Rail operator**	**Mine**
Mine	Mine-orders Demurrage cost	Properties of train classes Request for trains (jobs) Journey time	Objective function Production capacity
Rail operator	Train class properties Unloading restriction Number of mines	Objective function Number of trains	
Terminal	Objective function Ship-orders Ship loading time		

In this coal SC, the critical information of the mines is their objective costs and the information regarding the production capacity. The rail operator announces the details of the train classes, such as volume and journey time, but not the actual number of trains in each class. The unloading capacity at the terminal is a bottleneck in the three-party case. It is known only to the terminal and the rail operator.

Even in a decentralised framework, minimal information is required to share with the dependent DMUs to cooperate and achieve better performance for the supply chain. For example in the coal SC, some properties of train-classes need to be shared with the mines and the terminal to avoid infeasibilities at their end.

7.5.2 Decentralised methods to compute the bounds

As the objective costs are private, we need to use the secure-sum method discussed in Section 6.4.1 to compute the value of a column and the lower bound, LB^k. The mines create a column (request) based on the multipliers received from the master problem and their own constraints and then pass it to the rail operator to add its contribution to the value of the corresponding column and then to the terminal. We need to have at least three units to perform the secure-sum. This is one of the reasons that the CG-algorithm is not used in a two-party decentralised model.

The CG-based decomposition Algorithm 5, uses two heuristics to create globally-feasible columns (see Section 5.3). In this three-party decentralised case, we cannot directly reuse the Leader-follower (see Section 5.3.2) heuristic, since the train availability information is private to the mines. However, the job-scheduling model can be used by combining the resource constraints of the rail operator and the terminal. It does not require any private information.

Since we have these two methods to compute the bounds in a decentralised manner along with the distributed execution of the sub-problems, we assert that the proposed approach is a decentralised one in its execution. It requires significantly less information-sharing with no centralised controls when compared to the equivalent decomposed/integrated approaches. In general, decomposition methods are expected to produce better results compared to the results from an integrated model. However, we cannot expect the same from the proposed decentralised method, since it does not have comprehensive information-sharing between the partners. Therefore, we expect the decentralised method to perform very close to or better than the integrated model. The main attraction of the decentralised method is that it can produce reasonably good solutions with very minimal information sharing.

Figure 7.5 exhibits a decentralised approach based on the column generation algorithm, where the value of a column is computed using the secure-sum method. This means that the Lagrangian objective function (7.24) is not solely computed by the mines. If we assume that the objective costs are also shared between the DMUs, then we can have a decomposed version of the CG algorithm. Kindly note the following facts in the decomposed implementation.

1. The sub-problem of the mines are solved with the objective (7.25), which includes the objective terms of the rail operator and the terminal.
2. The objective cost computed using a sub-problem is the value of the column. The master problem is solved directly using the columns along with its value.
3. The secure-sum is used only to compute the sum from the individual sub-problems.

We denote the decentralised CG implementation for the three-party case as 'CG-d' and the decomposed implementation as 'CG-c' in the rest of the chapter. In a sense, CG-c finds solutions which achieves $\min(Z'_i + Z'_R + Z'_T)$. However, in the CG-d, the individual components of each DMU are minimised separately. Hence, in the decentralised case, the objective cost would be $\min(Z'_i) + \min(Z'_R) + \min(Z'_T)$. This is the main difference between the decentralised version and the decomposed version. Therefore, the bounds computed using the decomposed version will be better. The aim of this chapter is to demonstrate the modelling and implementation of a decentralised approach which can be later generalised to a multi-party case. The decomposed version is provided only for the purpose of comparison.

7.6 Computational experiments

The objectives of the computational experiment are to understand the effect of an additional player and the impact of the decentralisation in distributed decision-making. Therefore, in this section, we compare (i) the decentralised model with the integrated model (7.3); (ii) the decentralised model with the decomposed version; and (iii) the three-party decomposed with the two-party decomposed model. All these comparisons are performed on a new randomly generated dataset. Since there is a new DMU added to the problem, we cannot reuse the previous dataset used for the two-party case. Hence, the results presented in this section are nowhere presented in the previous chapters.

The terminal receives the orders from the ship with the expected due-dates. Based on this input, the terminal splits the ship-orders and passes them to the mines. At present, the splitting and assigning is done outside the model. The mine orders are inputs to the model. However, the jobs are internally defined by the mines. Hence, it is a part of the decision to be made. Therefore, randomly generated data instances have information regarding ship-orders, mine-orders, train class properties, objective coefficients and other parameters.

7.6.1 Dataset generation for the three-party model

We have created 100 data instances in four series. Each series corresponds to the instances with the number of mines = 6, 9, 12 and 15. For a fair comparison, the following parameters are fixed in all data instances.

- The production capacity (P_i) of all producers is taken as 400 tonnes per period, where one period is an hour.
- The inventory capacity (B_i) at the mine is 20000 tonnes.
- Objective coefficients: $H_i^t = 1, F_i^t = 3, C_i^t = 50000, A_{iw}^t = 100, \xi_w^t = V_w/100$, $\bar{B}^t = 100$ for producer i and resource class w.
- The length of the planning horizon $T = 200$.

- Train class properties

Number of trains	3	2	3	2
Capacity in tonnes	3000	5400	7200	8400
$R^1_w = R^3_w$	2	3	3	4
$R^0_w = R^2_w$	5	6	7	8
R^4_w	1			
No. of data instances	25	25	25	25

- The capacity of a vessel is 75000 tonnes.
- The average production per mine is 25000 tonnes. This means that the number of vessels (ships) is the number mines divided by three.

The due-dates for l^{th} ship-order are computed using the following relation,

$$\Phi_l = \Phi_{l-1} + \frac{140}{\text{Number of ships}} * \mathbf{Unif}(0.8, 1.0) \quad \forall\, l \qquad \text{where } \Phi_0 = 0 \tag{7.31}$$

If there are I mines, then a ship-order can be split across all the mines. The average mine-order is 10000 units. The mine order quantity is computed by the formula $Q_u = 5000 + 100.\mathbf{Unif}(0, 100)$. For example, in a six-mine instance, a ship-order of 75000 tonnes can be split as $12600 + 5100 + 8200 + 10200 + 7200 + 9000 + 6100 + 9200 + 7400$, which is assigned to six mines. If the number of mine-orders is not equal to the number of mines, then it can be assigned randomly. Ideally, the due-date for these mine-orders (F_u) should be Φ_l. However, due to unloading and stocking restrictions, the terminal may ask some of the mines to deliver the cargo earlier. In our experiments, three time units are reserved as a buffer time between the deliveries.

The ship-orders and mine-orders are the only varying parameters across different data instances in a series. This means that, we can analyse the influence of the number of DMUs by comparing the results from the different series.

The following abbreviations are used in this section.

CG-d Three-party decentralised model based on CG

CG-c Three-party decomposed model based on CG

IM-3 The integrated model discussed in Section 7.3

CG Two-party decomposed approach based on CG

IM The integrated model for the two-party case

We reuse the two-party models CG and IM, presented in Chapter 5, to illustrate the impact of an additional player. In other words, the performance of CG and IM are compared again with the new dataset to benchmark the performance of the three-party algorithms. For convenience, the implementations based on column generation for the three-party case, CG-c and CG-d, are collectively referred to as 'CG-3'.

Since the objective costs are not directly comparable, the relative ratio of the bounds are computed and compared. The comparison between IM and IM-3 and CG-2 and CG-3 will show the impact of an additional player. At the same time, the comparison of CG-c and CG-d will show the role of decentralisation.

The integrated models and the DMUs used in the CG algorithms are solved with CPLEX 12.1. The decentralised approaches are implemented using a Java application in combination with CPLEX with Concert Technology (see Section 4.7.2 for more details). In IM-3 and IM, CPLEX was terminated either at 0.1% of relative gap or with a CPU time limit of one hour, whichever came first.

Performance measures

The relative ratios—lower bound relative ratio (LBR) and upper bound relative ratio (UBR)—are used to compare the bounds from different approaches. For example, the bounds from the CG-based decentralised model (CG-3) and the integrated model (IM-3) are compared with

$$LBR(CG\text{-}3, IM\text{-}3) = \frac{LB_{CG\text{-}3} - LB_{IM\text{-}3}}{\max\{LB_{CG\text{-}3}, LB_{IM\text{-}3}\}} \quad \text{and} \tag{7.32}$$

$$UBR(CG\text{-}3, IM\text{-}3) = \frac{UB_{CG\text{-}3} - UB_{IM\text{-}3}}{\min\{UB_{CG\text{-}3}, UB_{IM\text{-}3}\}} \tag{7.33}$$

A positive LBR(CG-3, IM-3) implies that the lower-bound from the CG-3 is higher than that from the IM-3, which is preferable. Similarly, a negative UBR means that the upper-bound computed using the CG-3 is better (smaller) than that from the IM-3.

Figure 7.7 compares the decomposed models for the two-party and the three-party cases. Out of 100 instances, a single solution was not found in 70 and 47 instances using IM and IM-3, respectively. Figure 7.7 captures the effect of an additional player in the performance of the decomposed approaches. The common trend is that the decomposed approaches are better than the integrated model in both the cases. The two-party LBR and UBR are better than the equivalent ratios from the three-party case, while, in most of the instances, UBR(CG,IM) is negative and UBR(CG-3,IM-3) is positive in nearly 15% of the instances. This implies that upper bound computation becomes more difficult as

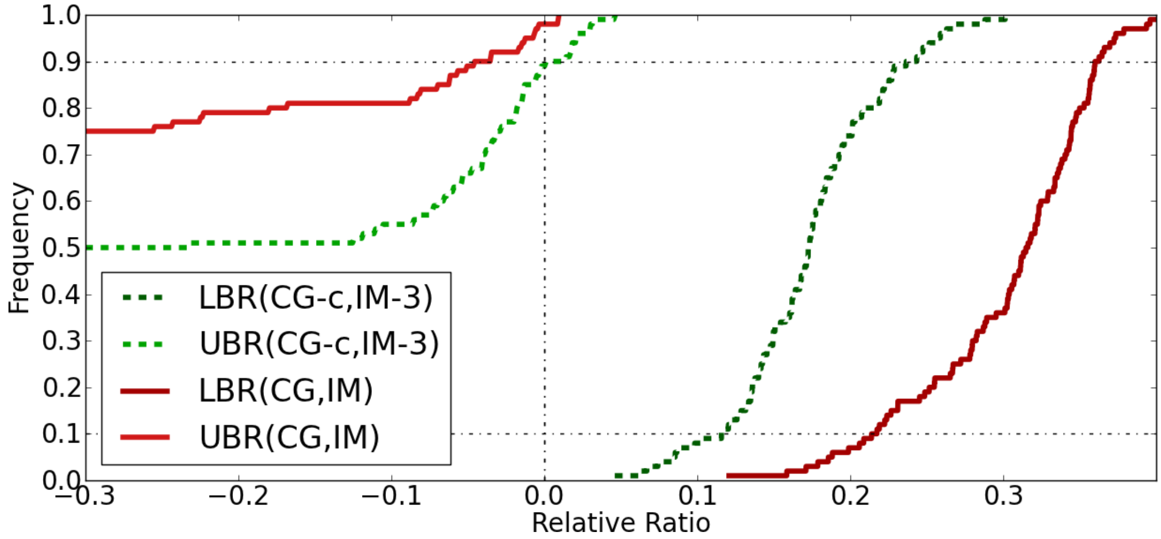


Figure 7.7: Comparison of the relative ratios for the two-party and three-party decomposed models

the number of players increases. The gap in the LBRs is around 20% in most of the cases. However, the gap in UBRs is approximately 5%.

Figure 7.8 shows the change due to decentralisation and information-sharing. As expected, the performance directly depends on the level of information-sharing. Ideally, the LBR has to be positive and the UBR has to be negative to indicate that the three-party approaches perform better than the integrated models. The gap in LBRs is roughly 10-15% and the gap in UBRs is less than 2%. In comparison with IM-3, CG-d achieves a lower upper-bound in 80% of the instances and a higher lower-bound in 92% of the instances.

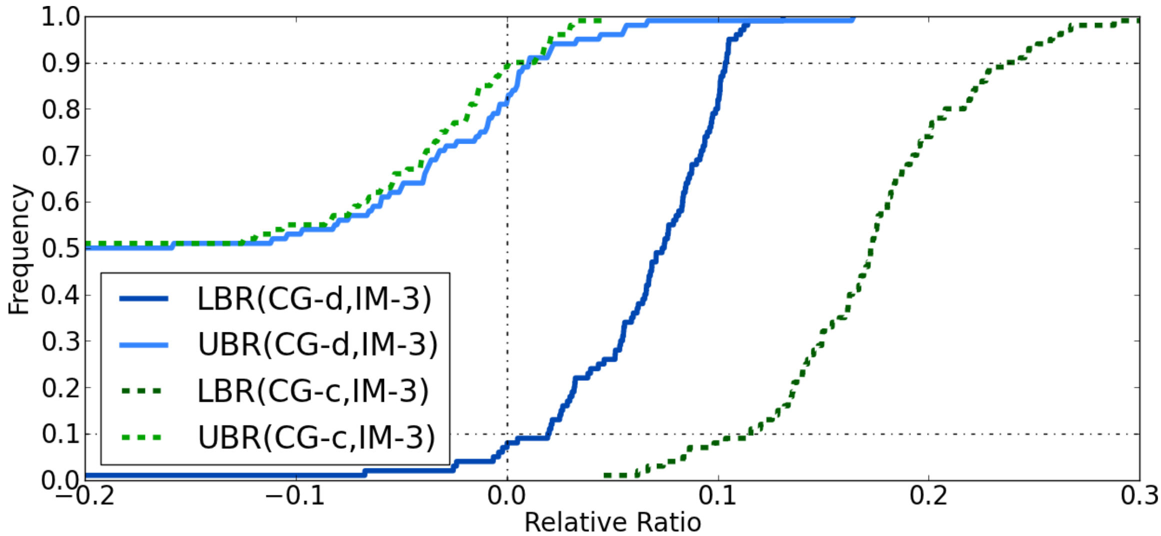


Figure 7.8: Comparison of the relative ratios for the three-party models

Table 7.1 presents the median of LBR and UBR computed for each series. If the IM-3 model cannot find any solution, then the UBR cannot be computed. The LBR is expected to increase and the UBR is expected to decrease as the number of mines increases. The expected behaviour is observed in Table 7.1, except in two shaded cells. One hour was insufficient for LBR(CG-d, IM-3) to compute better lower bound as the number of mines were high. In a twelve-mine series, we could compute UBR(CG,IM) only for one data instance. The overall median shows that the improvement in the LBR is almost five times of the improvement in the UBR.

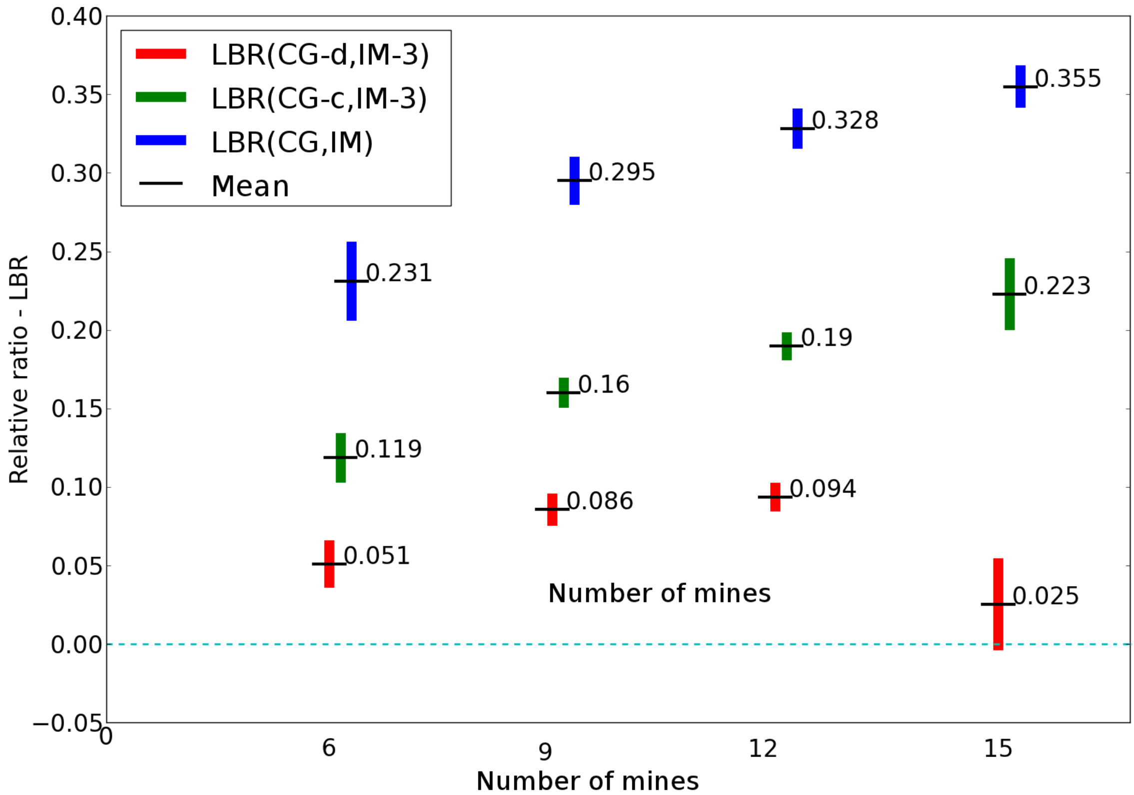
Table 7.1: Median of the relative ratios

	LBR			**UBR**		
# Mines	(CG-d, IM-3)	(CG-c, IM-3)	(CG, IM)	(CG-d, IM-3)	(CG-c, IM-3)	(CG, IM)
6	0.0535	0.1258	0.2224	0.004	-0.013	-0.0567
9	0.0838	0.1618	0.2952	-0.0393	-0.0597	-0.886
12	0.1	0.1874	0.3337	-0.4465	-0.4428	-0.6775
15	0.0299	0.2279	0.3566	Not Available		
Overall	0.0741	0.1725	0.3154	-0.0154	-0.0320	-0.0660

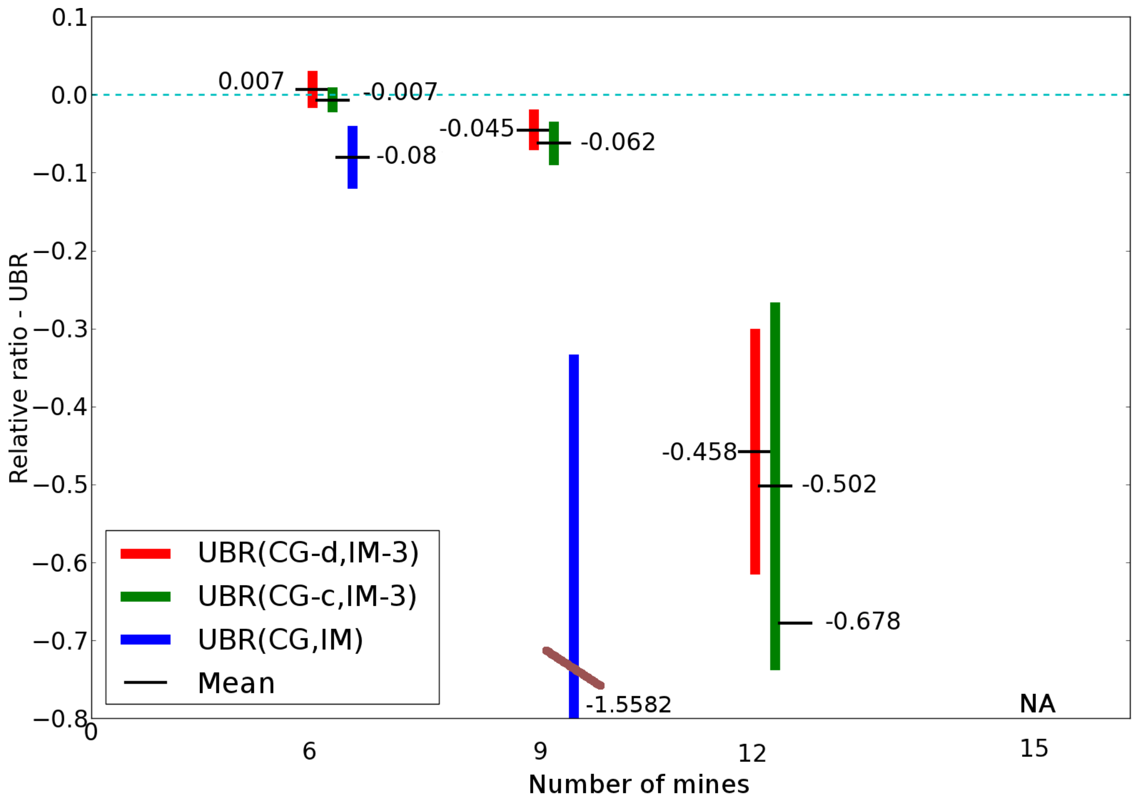
Figure 7.9 shows the 95% confidence interval computed using the Student's t-test. The mean is marked with a small black horizontal line. It shows that the performance and the number of mines are proportional. Since the number of available upper-bounds decreases, the CI of UBR becomes weak, and hence, the interval is large. The observations made from Figure 7.8 and Table 7.1 can be cross-verified with the observations from Figure 7.9.

Figure 7.10 presents the cumulative relative gap comparison of three-party models. The distributed decision-making models perform better than the integrated model. In some cases, IM-3 could not find a single feasible solution. Such cases are included with 100% relative gap. The model CG-d, achieves less than 20% relative gap in more than 50% of the instances. At the same time, 25% and 75% of instances had less than 20% gap in the IM-3 and CG-c, respectively. The performance of CG-d can be improved if we have a more efficient decentralised method to compute the upper bound.

As the results show, the proposed decentralised approach based on CG is very efficient and quick in bringing down the relative gap between the bounds. Once the lower bound from the relaxed master problem (RMP) and the Lagrangian objective function are the same, the algorithm will not add any new columns. At that point, we will have an optimal solution from the RMP. We need to have better branching algorithms to compute integer solutions close to this solution.



(a) Lower bound ratio: LBR



(b) Upper bound ratio: UBR

Figure 7.9: 95% confidence interval for the relative ratios

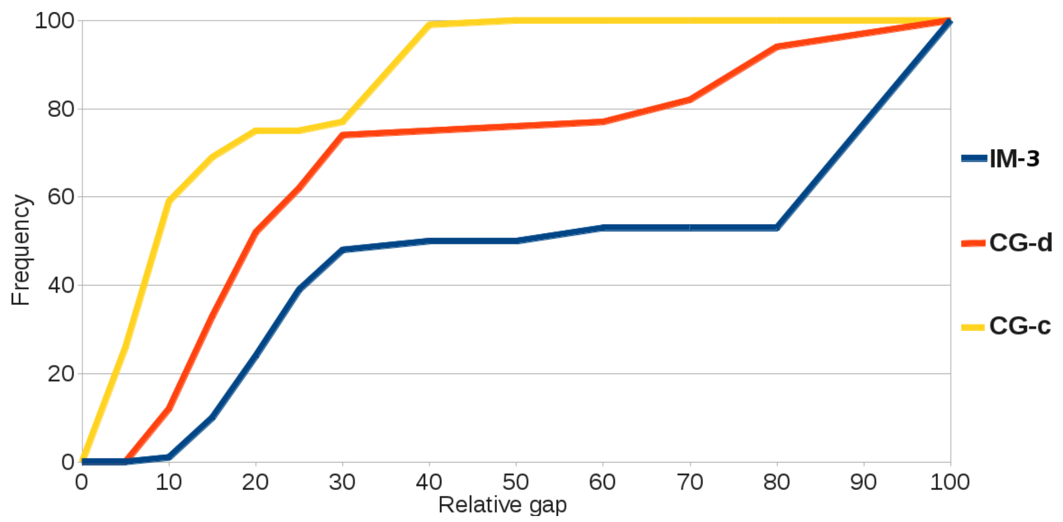


Figure 7.10: Cumulative relative gap frequency diagram

These results show that the addition of a new player has reduced the performance of the algorithm. Again, the lack of information sharing further reduced the performance. CG-d, has the least information-sharing and its performance is not good as compared to the other decomposed approach. However, we should accept the fact that the decentralised approaches could produce *acceptable* results with minimal information-sharing and with no central coordinator. In any case, complex IM-3 can be replaced with CG-d.

7.7 Conclusions

In this chapter, we consider a three-party coordination problem by including one more resource manager to the two-party case. A decentralised decision-making approach based on CG is proposed to address this problem. Some of the strengthening methods discussed in the previous chapters are included in the proposed algorithm. The main challenge in designing a decentralised approach based on CG is computing the value of a column, updating multipliers and generating globally feasible columns. We have also discussed the mathematical models for the DMUs and decentralised methods to compute the bounds.

Similar to all other models, three-party distributed decision-making models are also compared with the integrated model using an intense computational experiment. The results show that the decentralised model could achieve better or equal solutions compared to that from the integrated model with significantly less information and interaction.

The combination of the iterative procedure based on CG and the secure-sum method assures a better lower bound in almost all instances, compared to that of the integrated model, with significantly less information-sharing and no centralised controls. However, in some cases, the quality of the upper bound is not good. Therefore, future research should focus on developing better decentralised mechanisms to compute the upper bound.

Chapter 8

General Multi-party Decentralised Coordination Scheme

We have been looking at alternative modelling approaches for supply chain coordination. We started with a two-party model motivated by the coal supply chains in Australia. The integrated production planning and resource scheduling problem is solved using two decomposed algorithms based on LR and CG, respectively. Further, we reduced the information-sharing and the role of the central player/coordinator in these models. Thus, we propose a decentralised approach based on LR for two-party coordination, where there is only one shared resource. Later, we extended this approach to a three-party case using a CG algorithm which has two shared resources. This chapter highlights a few generalisation methods and proposes a framework for practical implementation of different modelling approaches.

8.1 Generalisation to multi-party cases

It is observed based on the problem and model structure, the two party model (discussed in Chapters 4, 5 and 6) and the three-party model (discussed in Chapter 7) belong to the same family. Therefore, a multi-party coordination problem can also be mapped to the same family. The generic representation presented in Section 3.4, abstracts a coordination problem in a multi-party case. The decentralised decision-making model based on decomposition techniques can be extended to the multi-party case as well, with relevant customisation. The number of multipliers might grow as the number of linking constraints increases.

Figure 8.1 is a schematic representation of multi-party cases. The producers are independent of each other and and so are the resource managers. However, all resource managers, potentially, link all the producers.

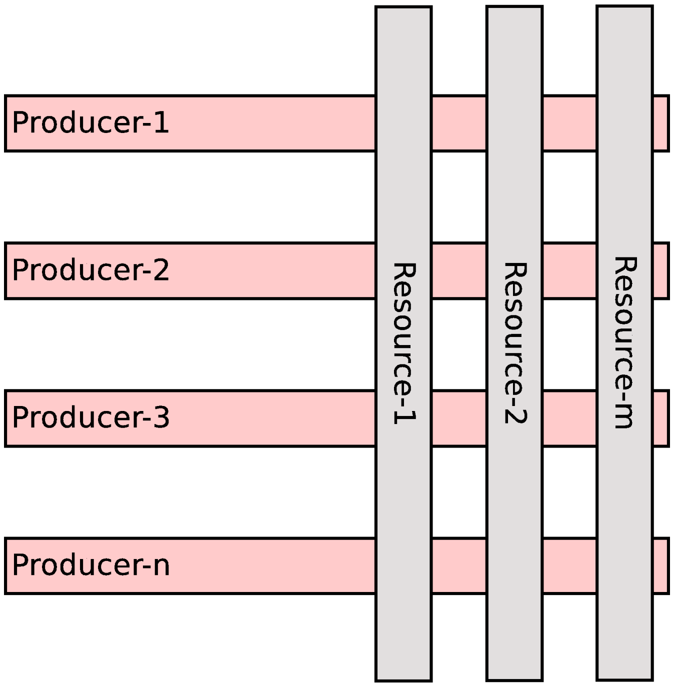


Figure 8.1: A schematic of multi-party coordination models

The mathematical representation of a generalised problem with multiple DMUs (reproduced from Section 3.4) is given below:

$$\min Z = \sum_i c_i^T x_i + \sum_j d_j^T x \tag{3.1}$$

$$\text{subject to} \qquad A_i x_i \leq b_i \quad \forall i \qquad (\text{Producer} - i) \tag{3.2}$$

$$R_j x \leq e_j \quad \forall j \qquad (\text{Resource manager} - j) \tag{3.3}$$

where $x = [x_1, x_2, \ldots, x_n]^T$. The linking constraints (3.3) need to be relaxed to separate the decisions of each producer. Then, the relaxed problem would be

$$\min Z = \sum_i c_i^T x_i + \sum_j d_j^T x + \sum_j \lambda_j (R_j x - e_j) \tag{8.1}$$

$$\text{subject to} \qquad A_i x_i \leq b_i \qquad \forall i \qquad (\text{Producer} - i) \tag{3.2}$$

The problem for each individual producer can be stated as:

$$\min Z_i = c_i^T x_i + \sum_j (\bar{d}_j^i + \lambda_j \bar{R}_j^i) x_i \tag{8.2}$$

$$\text{subject to} \quad A_i x_i \leq b_i$$

where $\bar{d}_j^i$ and $\bar{R}_j^i$ are the contributions of producer-i such that $\sum_i \bar{d}_j^i x_i = d_j x$ and $\sum_i \bar{R}_j^i x_i = R_j x$. Then, the overall objective can be expressed as $Z = \sum_i Z_i - \sum_j \lambda_j e_j$.

The objective function $c_i^T x_i$ and $d_j^T x$ are private information of the producer-i and the resource manager j, respectively. Moreover, Constraint (3.3) is also a private information to the resource manager j. This decomposition is computationally efficient only when the DMUs share their objective cost. Otherwise, the Z_i in the objective (8.2) cannot be minimised as is, using decomposed models. This implies that, in a decentralised decision-making environment, we attempt to compute $\tilde{Z}_i$ (see (8.3)) instead of 'min Z_i'.

$$\tilde{Z}_i = \min c_i^T x_i + \min \sum_j \bar{d}_j^i x_i + \min \lambda_j \bar{R}_j^i x_i \tag{8.3}$$

Such summations can be computed using secure-sum without revealing the exact cost. This is one of the major differences between decomposed models and decentralised models. It is assumed to publicly share some of the resource properties to enable coordination between independent producers and resource managers. As illustrated in the previous chapter, multiple variants of the decentralised approach are possible by assuming different levels of information-sharing.

Decentralised decision-making is closer to real life industrial instances with some inefficiencies [18]. The real challenges in designing a decentralised decision-making model are, the ability to (*i*) compute better bounds in order to tighten the search space, (*ii*) handle multiple objectives, (*iii*) design a negotiation scheme and proper incentive mechanisms, and (*iv*) ensure information security and privacy. Our proposed models address all of these challenges to some extent. However, these models can be further improved with relevant customisation.

8.2 Generic decision-making framework

The three-party decentralised approach provides a framework for decentralised approaches for multi-party cases. Based on the experience of implementing different decomposed and decentralised models, we propose a framework, in Figure 8.2, for the multi-party case. Such approaches can be implemented using distributed computing facilities. The arrows in Figure 8.2 represent the flow of information and decisions. The decisions are made in two stages. In the first stage, each producer generates columns—which are the building blocks of globally feasible solutions—and the value of columns is computed by the *secure-sum* method. In the second stage, the best combination of columns will be selected such that it is feasible for all the DMUs. If there is any infeasibility or a chance for improvement, then feedback on resource utilisation is given in terms of the dual-prices of complex linking-constraints.

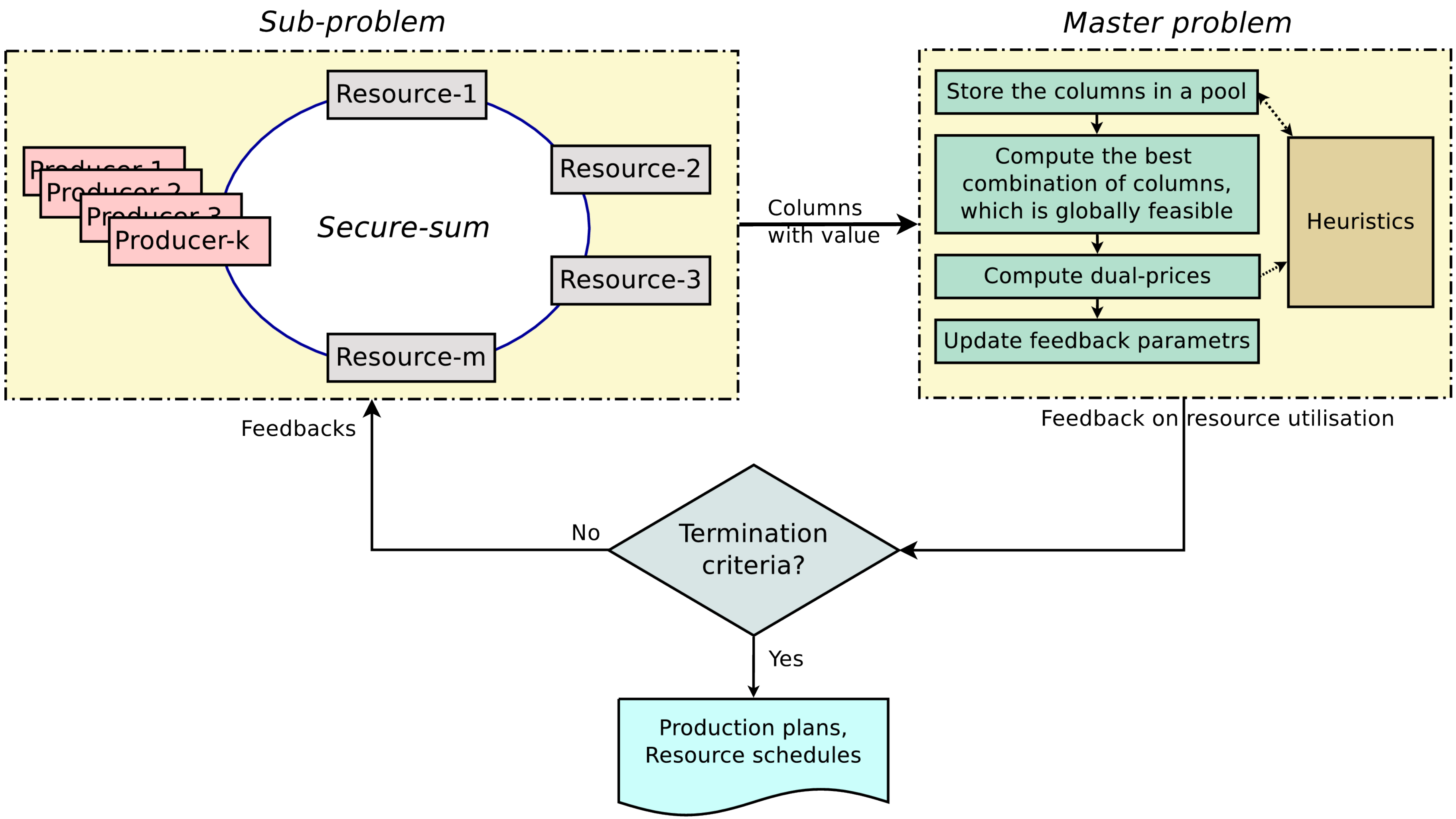


Figure 8.2: A decentralised decision-making framework for multi-party cases

Suppose the system has partial information-sharing then either there can be a third-party coordinator, who does not have any decision-making power, or one of the players/parties may act as a coordinator, if needed. We observed that in the decomposed and decentralised approaches, the speed of convergence directly depends on the quality of the bounds computed. Therefore, heuristics developed exclusively based on the features of the underlying supply chain can help us to compute better bounds and speeds up the convergence of the decision-making process.

Section 6.2 presents four main components—(*i*) decisions makers and decisions; (*ii*) information-sharing; (*iii*) multiple objectives; and (*iv*) coordination mechanism—of the decentralised supply chain coordination. All of them have significant role in the multi-party case also. Based on the structure of the underlying supply chain, the decisions and decision makers can be grouped efficiently. In other words, it helps us to identify the inputs and outputs of the individual decision-making units in a supply network.

In a *truly* decentralised coordination model, we should assign higher priority for qualitative features/measures such as negotiation, information-sharing, overall policy, satisfaction of individual DMUs, service level, sustainability and the like—along with the quantitative objectives such as total cost, revenue/profit, utilisation, and the like.

8.3 Practical implementation

It is observed that information-sharing is a key factor in designing a successful coordination mechanism [40, 34]. The methods to capture the information and provide feedback to the various players is quite vital. Further, the medium through which such information is shared is also becoming increasingly pertinent to the successful operation of these decentralised supply chains. Delays in information availability (and processing) and information distortion need to be, therefore, significantly reduced. However, these new modes of operation come with an overload of information security and privacy [155], which need to be taken into account when developing new frameworks. The Internet is one such framework, which allows partners to share the information instantaneously and to process it at remote locations. Cloud computing is another popular distributed computing technology that could enable remote computations while also enabling decentralised decision-making. Distributed computing environments potentially enable partners to share their information and computational resources in an economical manner as long as the technology is also able to ensure security and privacy to all the participating players.

The medium through which information is shared between players has also changed significantly over the years. Traditionally, much of the information that was shared was through the medium of paper (with appropriate signatures to ensure that delegated authorities share the information that was required). More recently, information is shared through the medium of fax, email, Internet programs and telephone. Sometimes satellite communication devices have also been used for information-sharing [119]. The growth in the use of the Internet has also allowed the supply chain to share information amongst players.

Distributed computing facilities could assist in information-sharing too. It empowers the DMUs to use better software and computational infrastructure while making decisions. Unlike the centralised case, the DMUs have more control on the decisions to be made. It allows us to eliminate needless delays and ensures no-loss information-sharing. On the other hand, however, there are security and privacy concerns with the use of distributed computing environments for information sharing [155]. Distributed computing eases the overall decision-making process without compromising the independence of the DMUs while also supporting coordination amongst them.

Toka et al. [140] discuss distributed computing as a meaningful technology that contributes to the SCM by providing infrastructure, platform and computational resources for the whole supply chain network via the Internet. Figure 8.3 shows the basic architecture required for such a distributed computing framework. Wu et al. [152] state that business process complexity, an entrepreneurial culture and information system's compatibility and application functionality significantly affect a firm's intention to adopt distributed computing technologies to support its supply chain operations.

The benefits of distributed computing are: (*i*) it offers real-time access to information, (*ii*) it is a very economical option for information transfer and (*iii*) heavy computation can be done in remote servers, if required; the DMUs, therefore, do not need to own expensive, heavy-duty solvers.

Zissis and Lekkas [155] identify some of the concerns associated with distributed computing, including trust, privacy, availability, integrity and physical and network security. The client has to trust the server (service provider) that the private information will be protected and since it is without any alterations, it will be made available to the client. The service provider has to ensure that only authorised clients will have access to restricted information. In addition, the service provider has to protect the physical systems and network.

Let us visualise the decentralised framework for a coal supply chain implemented using distributed computing facilities. Each of the mines and the train operators will have local servers that are connected via some service provider. Each individual mine's problems will be solved locally and results, once collected by the service provider, will be sent to

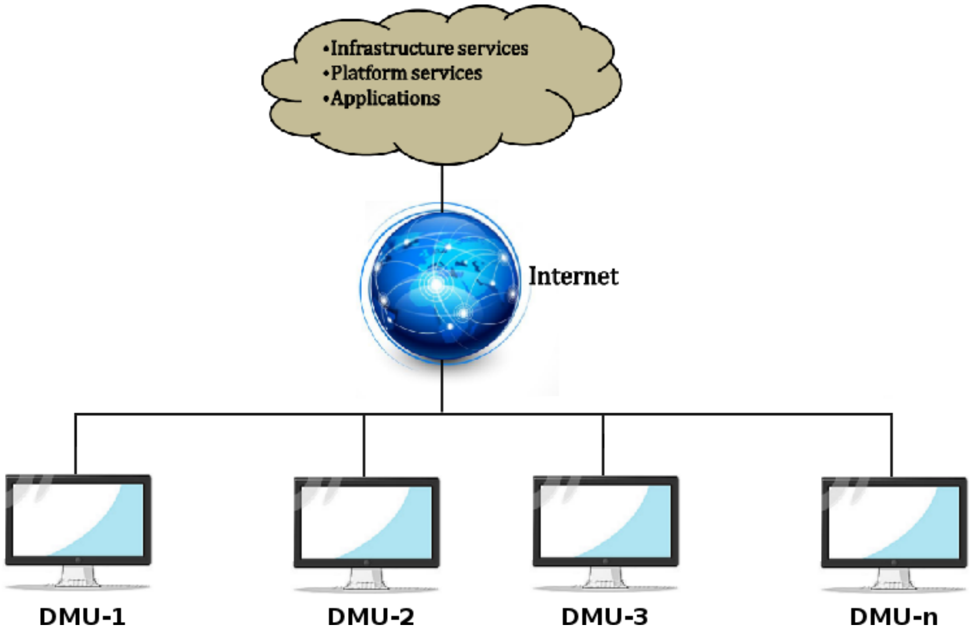


Figure 8.3: Distributed computing architecture (Inspired from [101])

the train operator server for processing. Once processed, the train operators sends the feedback to the individual mines through the distributed computational infrastructure.

As we explained earlier, *secure-sum* can be implemented to compute the lower bound. Similarly, decentralised heuristics are developed to compute the upper bound. Based on information availability, there will be some changes in the framework. Most of the critical information and decisions are handled by the DMUs themselves. Only minimal and required information is shared through the secure channel provided by the service provider. Therefore, the proposed decentralised method ensures data privacy and security. The interface and data communication channels are part of the distributed computing facility provided by the service provider. If the service provider is a third party, it does not make decisions or influence the DMUs. In some cases, one of the DMUs can provide distributed computing service to others. In our example, the rail operator acts as an *honest coordinator* to provide the service to the mines and the terminal.

8.4 Conclusions

This chapter shows that our solution approaches can be extended to a multi-party case with any number of common resources. In this thesis, we have introduced decentralised approaches based on different decomposed techniques. It helps the solution approaches to develop around the existing framework and converge to some equilibrium in reasonable time. We have explored only two popular decomposition techniques, LR and CG. However, decentralised approaches can be built on any other decomposition techniques, or concepts like agent-based modelling, and multi-level programming. In our proposed approaches, only one kind of feedback through penalty costs was analysed. Nonetheless, other feedback mechanisms can also be explored for decentralised decision-making approaches.

Chapter 9

Conclusions and Future Directions

In this thesis, we explore alternative approaches for supply chain coordination using decomposed and decentralised decision-making models. This is motivated by a real-world mining example, which is generalised to a multi-resource constrained scheduling problem (MRCSP). We start with a two-party coordination problem, which involves multiple producers and a resource manager, and extend it to a three-party case by considering one more resource manager. We have explored three different modelling approaches to solve the MRCSP. This chapter concludes by proposing a framework for decentralised decision-making approaches for multi-party coordination problems and listing some directions for future research.

Although the methods that have been developed in this thesis are validated within the context of a coal supply chain, it can be extended to other general RCSPs in contexts such as airline, wine, automobile manufacturing and also the services industry.

9.1 Conclusions

We conducted an extensive literature review on supply chain coordination models and different solution approaches. An overall review of the related literature and a few motivating examples from different industries are presented in Chapter 2. In addition, the topic of discussion in every chapter is supported with latest and relevant literature.

The supply chain coordination models are classified based on their operational and decision-making aspects. We focused on the decentralised decision-making models (DDMs) which is governed by single or multiple operators. To address the DDMs, three major modelling approaches—(*i*) integrated approach, (*ii*) decomposed approach and (*iii*) decentralised approach—are identified in Chapter 3. The coordination problem in coal supply chains with a single shared resource and multiple resources are introduced with adequate details. The DMUs, information sharing between these DMUs and their interactions are also presented. The chapter concludes by proposing a structure of a generic problem and its representation in a matrix form. Any resource constrained scheduling problem, with a similar representation, can be solved using the proposed solution approaches.

We have analysed a coordinated production-planning and resource-scheduling problem that exists between a set of independent producers and a common resource manager. A coal supply chain coordination problem is considered as an example. The DMUs, their decisions and their interconnections are analysed. Mathematical programming models for each of these DMUs have been formulated and an integrated problem is also proposed, which is equivalent to an RCSP with a single shared resource.

A decomposition approach based on the *Lagrangian relaxation* (LR) is presented in Chapter 4. The proposed decomposition scheme splits the production planning and resource-scheduling decisions. This approach is strengthened using the Volume algorithm, the Wedelin algorithm and an heuristic to compute the upper bound. Chapter 4 explains the framework for two-party coordination, the mathematical programming models and the algorithms, using the coal supply chain example. A process to benchmark the distributed approaches against the integrated model using computational experiments is also illustrated. It includes data generation, the experimental design, implementation, and the analysis with different performance measures.

Since the speed of convergence of the LR algorithm for large problems is not as good as is for the small problems, Dantzig-Wolfe decomposition together with a *column generation* (CG) technique was explored further to solve the two-party coordination problem. In this algorithm, we have used many strengthening techniques, including two levels of stabilisation. The two-party decomposition approach based on CG (Chapter 5) outperforms the LR-based approach significantly. Different performance measures are used to compare the decomposed approaches against the integrated model.

From elaborate computational experiments we conclude that the distributed decision-making approach is preferable to the integrated approach. For a quick comparison, Figure 9.1 shows the relative gap distribution observed for the two-party decomposed approaches, LR (see Section 4.7.4 for detailed results) and CG (see Section 5.6.2 for detailed results). The overall objective in these experiments is to minimise the total cost of the mines (the rail operator's objective was not considered in this case). The decomposed approaches compute a better solution than the integrated approach, with a relative gap less than 20% in almost all cases, with the CG approach outperforming the LR approach.

The mines and the rail operator possess different sets of information. Hence the decision model has an inherent information asymmetry. The objectives of (each of) the mines and

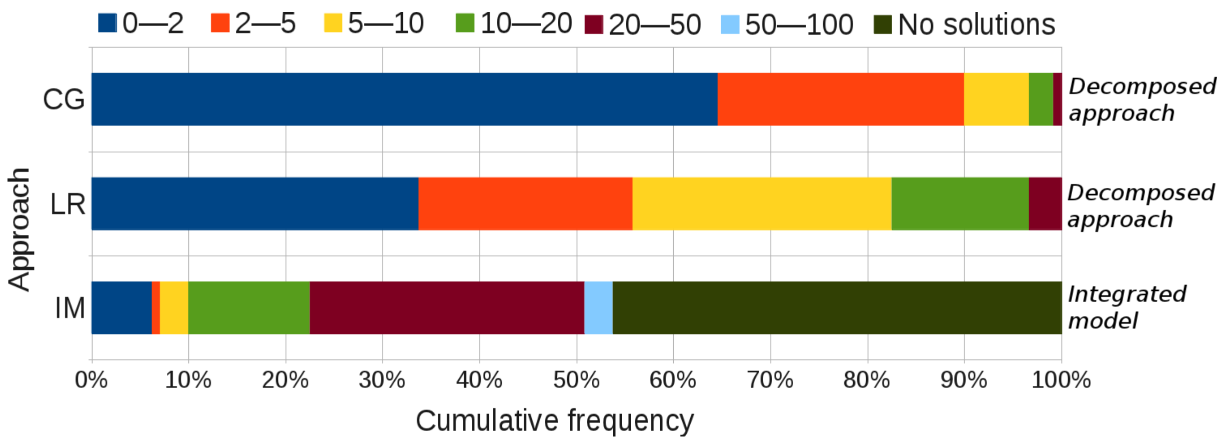


Figure 9.1: Distribution of relative gap (in percentage) observed for the two-party decomposed approaches

the rail operator are different. Every mine expects the rail operator to send a train to their mine to load the coal and deliver it ontime to the terminal. However, the rail operator may have different priorities and constraints. Hence there is a conflict in the objectives of the players. Decomposed approaches are not *truly* decentralised because there is no 'negotiation mechanism' in these models. In our approach, the negotiation protocol is partially implemented using the CG algorithm which is (in a sense) 'controlled' by the 'honest broker' rail operator. It allows the DMUs to be independent without revealing all of the information. The decomposed approaches expect the DMUs to share dependent information, with the DMUs managed/controlled by a central coordinator. Therefore, we further reduced the information sharing and the role of the central coordinator to develop *decentralised* approaches. Ideally, a decentralised method should have a negotiation and incentive scheme to assist the DMUs to make better decisions quickly.

Information-sharing plays a key role in decentralised decision-making. The role of information-sharing and other factors in two-party coordination problems is discussed in Chapter 6. Based on this analysis, we developed a decentralised approach which has limited access to the information and does not require any central coordinator. In a decentralised approach the lower bound and upper bound should be computed using decentralised methods. The *secure-sum* method has been used to compute the lower bound so that no player will be able to find out the true objective cost of other players, but will be aware of the total supply chain's costs. We have developed a few heuristics which does not require any central information. Such heuristics are used to improve the upper bound. In the two-party decentralised case, the number of players is two and hence, the value of a column cannot be computed without revealing the actual cost. Hence, the CG algorithms cannot be deployed for the two-party decentralised case. Therefore, LR approach is extended to develop the decentralised approach for the two-party case. The impact of the decentralisation and the value of two critical pieces of information in the coal supply chain—(*i*) production capacity and (*ii*) resource availability—are analysed and compared with different performance measures. The results of computational experiments show that the lower bounds of the iterative algorithm can be significantly improved by sharing the necessary information. The overall comparison using the confidence intervals shows that resource availability information is more critical than production capacity information. Figure 9.2 presents a quick comparison of the relative gap obtained from different decentralised approaches for the two-party case. In Figure 9.2, M0 refers to the integrated model, which minimises the total cost of the mines as well as the running cost of the rail operator. M1 to M4 are decentralised approaches with different levels of information-sharing. See Section 6.5 for detailed results.

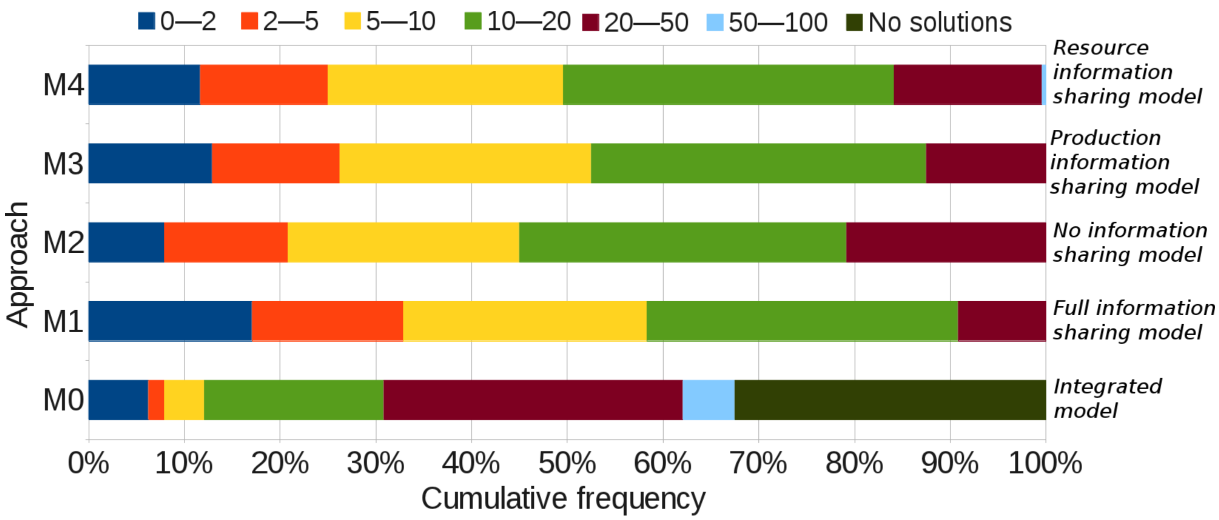


Figure 9.2: Relative gap distribution observed for the two-party decentralised approaches

Even though the coordination problem with a single resource manager is complex and difficult, we extend the decentralised approach for the three-party case (Chapter 7) by adding one more common resource (the terminal to the coal supply chain). This is required as a step to generalise the decentralised decision-making approach for larger problems which has multiple shared resources. In the three-party case, there are two resource managers to link independent producers. The critical resources in the three-party coal supply chain are: (*i*) the limited number of trains managed by the rail operator; *and* (*ii*) the limited number of unloading slots at the terminal. The proposed decentralised scheme has two sets of multipliers—each one corresponding to the linking constraint of the two resource managers. The main challenges in designing a decentralised approach based on CG are computing the value of a column, updating multipliers and generating globally feasible columns. Similar to all other models, the three-party distributed decision-making models are also compared with the integrated model using an intense computational experiment. Figure 9.3 shows an overall comparison of the three-party approaches, with

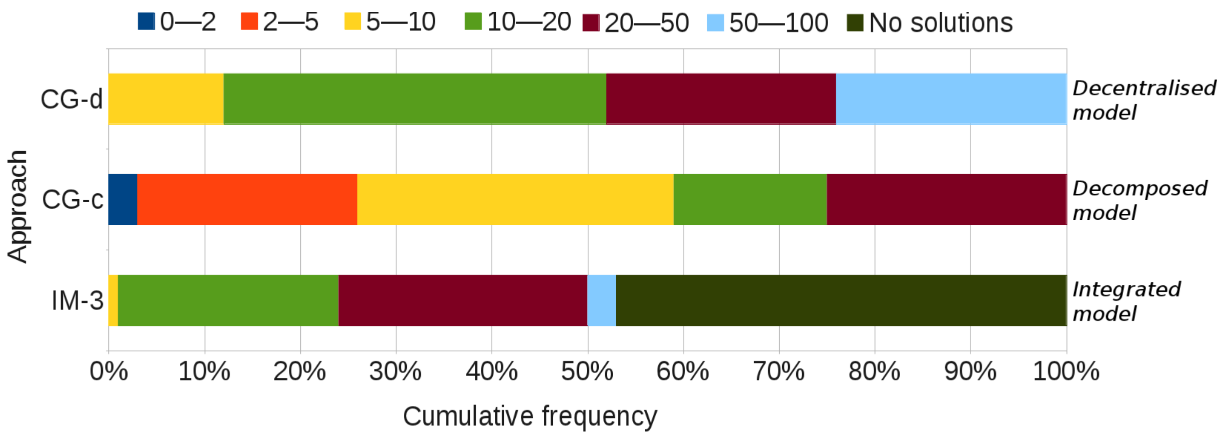


Figure 9.3: Relative gap distribution observed for the three-party approaches

respect to the relative gap. The objectives of all the three DMUs were considered in these models. The computational results highlight the impact of an additional player and the value of information-sharing. The results show that the decentralised model could achieve better or equivalent solutions compared to that of the integrated model with significantly less information and interactions.

We have modelled the coal industry problem as close to reality as possible. Therefore, one can see that the production planning model discussed in Section 4.3 has two sets of constraints. One set is just sufficient to represent the model. The second set includes a few more additional integer cuts and bounds which are used to tighten the MIP formulation. Similarly, the job-scheduling models are also presented with all the constraints and decisions. The decomposition algorithms LR and CG are also strengthened adequately.

Decomposition is a proven technique for factorising large structured optimisation problems. The novelty of our research is in developing decentralised decision-making approaches based on well-established decomposition techniques. In comparison with non-viable integrated approaches, proposed decentralised approaches are efficient and practically implementable. The players in a supply chain have to share some amount of information to achieve coordination. The decentralised approach does not require any additional information. Also, it protects the autonomy of all the players. Therefore, it is suitable for industries which are looking for a coordination approach that can deliver quality solutions in a reasonable amount of time.

To conclude, we have proposed a scalable and robust, decentralised framework of decision-making for a multi-party supply chain, that is a better alternative to the integrated approach. It requires only minimal information sharing between the players and guarantees convergence by means of the underlying decomposition algorithm. The approach can be used even as the level of coordination (information-sharing) improves. The proof of the concept has been demonstrated using a large and complex multi-party coal supply chain.

9.2 Contributions

The major contributions in the thesis are:

1. *A classification, framework and analysis of supply chain coordination models* (see Chapter 3). The models are classified as centralised and decentralised based on their aspect of decision-making. A framework for distributed decision-making is proposed and the decision-flow in the process is explained with schematic diagrams.

2. *Decomposed decision framework based on Lagrangian relaxation and column generation* (see Chapters 4 and 5). The speed of convergence of the decomposed approach based on LR is improved by the Volume algorithm and the Wedelin algorithm. The CG-based approach also includes many strengthening techniques and two levels of stabilisation.

3. *Decentralised decision-making approaches for a multi-party supply chain* (see Chapters 6 and 7). These approaches are useful when the supply chain does not have any central coordinator and prefers minimal information-sharing between the players.

4. *An iterative implementation scheme for distributed decision making.* The computationally implementable schemes for all proposed distributed decision-making approaches are presented with all the details such that it can be reused and/or extended in the future.

5. *Detailed mathematical programming models for the integrated problems, production planning and resource scheduling in a coal supply chain, and supporting heuristics.*

6. *Demonstration of customisable information-sharing in the decentralised approach.* The performance of a decentralised approach is directly proportional to the level of information-sharing.

7. *Computational framework, data instances, benchmarking results, and the like, can be used for future analysis.* A set of 240 data instances and a set of 100 instances are available for the two-party and three-party case, respectively. Also, the computational framework is robust enough to handle multi-party cases and large number of data instances.

8. *A work flow for practical implementation of decentralised approach, for a multi-party supply chain, using distributed computational facilities.* Such distributed facilities allow us to implement decentralised decision-making approaches economically in industries.

9.3 Future research directions

The solution approaches discussed in this thesis are developed around the coal supply chain. It can be extended and customised for other supply chains which have a similar structure such as the airline industry, the wine industry and the like.

A combination of CG and LR is often discussed in literature as an efficient procedure to overcome the weaknesses of both the algorithms. It might be an interesting idea to explore it along with other decomposition algorithms such as Bender's decomposition, multi-level programming or a hybrid version supported with some meta-heuristic algorithms. The performance of the CG algorithm can be further improved if we have a better method to compute integer solutions, very quickly, from the solutions of the relaxed master problem. We have used heuristic approaches with LR and CG to improve the quality of the solutions (see sections 4.6 and 5.3). Since our focus was on developing a general framework, we did not spend much time in solving a particular industrial problem. Based on case-to-case, we can improve the solution techniques with case-specific heuristics.

At present, our model deals with homogeneous systems. For example, all the mines had similar structure, costs, and decision variables. The research work can be further extended to include conflicting objectives, and Pareto optimality. Several other interesting questions can be pursued: What if there are more than one resource manager to support identical services? How can medium to long-term contracts be designed and operated between rail operators and mines? What if there are different product types involved? Will it be possible to include ship berthing, loading and unloading decisions too? What if players are not honest in their dealings? It will be interesting to develop negotiation protocols for coordination in a heterogeneous system.

In this thesis, the proposed decision-making approaches dealt with the overall performance of the supply chain. It is important to recognise individual partners' gain/loss in such large coordination problems. The individual performances versus the overall performance can be studied under different negotiation protocols and incentive schemes. The negotiation and incentives motivate the DMUs to be a part of the coordination even if they are not at their best.

Information-sharing plays a key role in decentralised supply chain coordination. It directly affects the underlying mathematical models, negotiation protocols and the incentive schemes. Therefore, theoretical studies on understanding the role of information and quantifying its value are much needed. Mechanism design concepts can be further explored to capture it.

All the solution approaches discussed in this thesis are mainly developed for industrial use. So, it might be logical to improve the performance by using modern computational facilities. Some components in the proposed algorithms can be executed in parallel to reduce run-time and to speed up convergence. Our solution techniques will be computationally more efficient if we convert them into a program which makes use of multi-core, multi-threaded architecture.

Appendices

A.1 Secure-sum

Secure-sum is a method to compute multi-party sum without revealing the individual value. This is proposed by Clifton et al. [42]. A practical implementation of the secure-sum method in a Lagrangian relaxation algorithm is presented in [121].

Let $P_1, P_2, \ldots P_n$ be the players. Then, the objective of the *secure-sum* method is to compute $V = \sum_k V_k$. Every player wanted to perform the summation without revealing its contribution to the other players. Then the secure-sum computation is as follows:

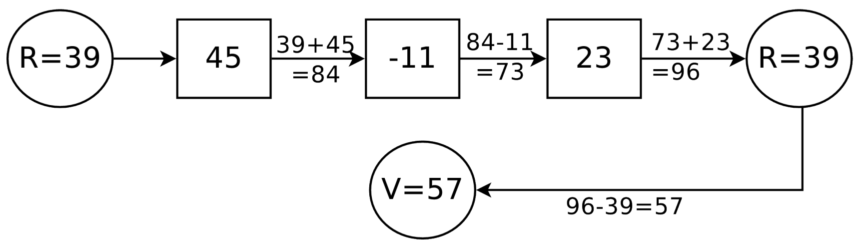
Algorithm 9 Secure-sum computation

Initialise the value with a random number, $V = R$
for $k = 1 \ldots n$ **do**
    $V = V + V_k$
$V = V - R$
Report the value V

Let us consider an example with three players.

Player (k)	1	2	3	Value
V_k	45	-11	23	57

The following figure shows the secure-sum computation. The arrows show the sequence of operations.



In the algorithm, player k knows the input $R + \sum_{\{j<k\}} V_j$ and the final sum $\sum_j V_j$. Moreover, the sequence in which the values are computed can also be a confidential information.

Case-1 $n = 2$**:** If there are only two players, then a player can compute the contribution of the other player by subtracting its contribution from V, where $V = V_1 + V_2$.

Case-2 $n \geq 3$**:** If there are three or more players, then a player k knows V, V_k and $V - V_k$ only. Therefore, it is impossible to compute the exact contribution of the other players.

A.2 Using CPLEX Concert technology in Java

Concert Technology allows a Java application to call IBM© ILOG CPLEX* directly. The Java interface supplies a rich means for the user to use Java objects to build optimisation models†.

The class `IloCplex` implements the Concert Technology interface for creating variables and constraints. It also provides functionality for solving Mathematical Programing (MP) problems and accessing solution information.

Example: Let n be an odd positive integer. Consider the knapsack problem

$$\min\left\{x_{n+1}\middle|2x_1+\ldots+2x_n+x_{n+1}=n, x\in\{0,1\}^{n+1}\right\}$$

A CPLEX model is given below (*Knapsack.java*).

```java
import ilog.concert.*;
import ilog.cplex.*;

public class Knapsack {
  public static void main(String[] args) {
    try {
      int N=4;
      IloCplex cpx = new IloCplex();          // Create an object of IloCplex
      IloNumVar[] X  = cpx.boolVarArray(N+1); // Create boolean array with size N
      for(int i=0; i< N+1; i++)
        X[i].setName("X"+(i+1));             // Give variable names X1, X2, .. , X(N+1)
      IloNumExpr temp_sum = cpx.numExpr();    // A temporary variable to compute the sum
      for(int i=0; i< N; i++){
        int coeff = (i==N-1? 1: 2);
        temp_sum = cpx.sum(temp_sum,cpx.prod(coeff, X[i]));
      }
      cpx.addEq(temp_sum, N).setName("MyConstraint");   // Equality constraint
      cpx.addMinimize(X[N]);                  // Objective is to minimise X(N+1)
      cpx.exportModel("Knapsack.lp");         // Export the model in LP format
      if (cpx.solve() ) {                     // Solve the cplex model
        cpx.output().println("Solution status = " + cpx.getStatus());
        cpx.output().println("Objective value  = " + cpx.getObjValue());
        cpx.output().print("n="+N+",   X* = ");
        double[] val = cpx.getValues(X);
        for (int j = 0; j < N+1; ++j)  cpx.output().print( val[j] + ", " );
      }
      cpx.end();                              // End the CPLEX session
      System.out.println("\n\n*** END ***");
    }
    catch (IloException e) {
      System.err.println("Concert exception '" + e + "' caught");
    }
  }
}
```

*URL: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

†Source: `http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r2/topic/ilog.odms.cplex.help/Content/Optimization/Documentation/CPLEX/_pubskel/CPLEX118.html`

A.3 Useful Python functions

Package	Function	Description
Python	`open(fname,mode)`	Opens a file to read/write
	`range(n)`	Generates a list of n values from 0 to $n-1$
	`readlines()`	Reads until EOF and returns a list containing the lines
numpy	`arange(a,b,s)`	Generates samples from a to b in steps of s
	`average(array)`	Computes the average of an array
	`linspace(a,b,n)`	Generates n-linearly spaced samples between a and b
	`round(a,b)`	Rounds the value a, to b decimal points
random	`randint(a,b)`	Return a random integer x such that $a \leq x < b$
	`random()`	Return a random floating point number in the range [0,1)
	`shuffle(a)`	Shuffles the elements in the list a
scipy	`interpolate.interp1d`	1-D interpolation function
	`sqrt(x)`	Computes square root of x
	`stats.t.ppf(alpha,df)`	Percent point function (inverse of cdf)
	`stats.ttest_1samp(values)`	Calculates the t-test for the mean of one group of values
	`std(array)`	Computes the standard deviation of an array
statsmodels	`distributions.ECDF`	Return the Empirical CDF of an array as a step function
matplotlib.pyplot	`boxplot(array)`	Makes a box and whisker plot
	`hist(Y,bins)`	Plots the histogram of the given list Y
	`hold(Boolean)`	If it is true, then the subsequent commands will be added to the current figure
	`plot(X,Y)`	Plots the values Y against X
	`savefig(filename)`	Saves the figure into eps/png format
	`text(x,y,'text')`	Adds a text in the figure at location (x, y)
matplotlib.patches	`Polygon(box_coordinates)`	Draws a 2-D polygon

The above functions are used in Python 2.7.

A.4 Student's t-test and confidence interval

Student's t-test is used to compare different performance ratios obtained from two methods/models. Let us define the following:

$M1$	First method
$M2$	Second method
(X_i)	Observations from the first method
(Y_i)	Observations from the second method
$\bar{X}$	Mean value of the observation from the first method
$\bar{Y}$	Mean value of the observation from the second method
n	Number of observations

Assumption: The paired differences $(X_i - Y_i)$ are independent and identically normally distributed.

In this case, our null hypothesis is H0: $\bar{X} - \bar{Y} = 0$ and the alternate hypothesis is H1: $\bar{X} - \bar{Y} \neq 0$. Then, the t statistic to test whether the means are different is

$$t = \frac{\bar{X} - \bar{Y}}{S_n/\sqrt{n}}$$

where S_n is the sample variance of $(X - Y)$, computed using n observations.

The confidence interval for $\bar{X} - \bar{Y}$ is

$$t_{\alpha/2,n-1}\frac{S_n}{\sqrt{n}} \quad \leq \quad \bar{X} - \bar{Y} \quad \leq \quad t_{1-\alpha/2,n-1}\frac{S_n}{\sqrt{n}}.$$

References

[1] ABARES. *Australian mineral statistics 2011 - March quarter.* Australian Bureau of Agricultural and Resource Economics and Sciences, 2011. URL `http://data.daff.gov.au/brs/data/warehouse/pe_abares99010607/AMS11.2_Marchqtr_2011_lowres_REPORT.pdf`.

[2] F. B. Abdelaziz and S. Mejri. Decentralised bilevel model for shared inventory management. *Production Planning & Control*, 24(8-9):684–701, 2013. doi: 10.1080/09537287.2012.666857.

[3] B. Abdul-Jalbar, J. Gutierrez, J. Puerto, and J. Sicilia. Policies for inventory/distribution systems: The effect of centralization vs. decentralization. *International Journal of Production Economics*, 81-82:281–293, 2003. doi: 10.1016/S0925-5273(02)00361-4.

[4] M. Albrecht. *Supply chain coordination mechanisms: New approaches for collaborative planning.* Lecture Notes in Economics and Mathematical Systems. Springer, 2009. ISBN 978-3-642-02833-5. URL `http://books.google.co.in/books?id=7Cp-dS8VpIYC`.

[5] H. M. T. B. Amor, J. Desrosiers, and A. Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009. doi: 10.1016/j.dam.2008.06.021. URL `http://www.sciencedirect.com/science/article/pii/S0166218X08002916`.

[6] G. Anandalingam. A mathematical programming model of decentralized multi-level systems. *The Journal of the Operational Research Society*, 39(11):1021–1033, 1988. URL `http://www.jstor.org/stable/2583201`.

[7] Arshinder, A. Kanda, and S. G. Deshmukh. Supply chain coordination: Perspectives, empirical studies and research directions. *International Journal of Production Economics*, 115(2):316–335, Oct. 2008. doi: 10.1016/j.ijpe.2008.05.011.

[8] K. Arshinder, A. Kanda, and S. G. Deshmukh. A review on supply chain coordination: Coordination mechanisms, managing uncertainty and research directions. In T.-M. Choi and T. C. E. Cheng, editors, *Supply Chain Coordination under Uncertainty*, International Handbooks on Information Systems, pages 39–82. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-19257-9_3.

[9] A. Arya, S. Minner, and T. Pfeiffer. Editorial special issue on “operations research and accounting in supply chain coordination”. *OR Spectrum*, 35(4):935–936, 2013. doi: 10.1007/s00291-013-0352-5.

[10] A. Atamtürk. Cover and pack inequalities for (mixed) integer programming. *Annals of Operations Research*, 139(1):21–38, 2005. doi: 10.1007/s10479-005-3442-1.

[11] A. Baboli, J. Fondrevelle, R. Tavakkoli-Moghaddam, and A. Mehrabi. A replenishment policy based on joint optimization in a downstream pharmaceutical supply chain: centralized vs. decentralized replenishment. *The International Journal of Advanced Manufacturing Technology*, 57(1-4):367–378, 2011. doi: 10.1007/s00170-011-3290-x.

[12] L. Bahiense, N. Maculan, and C. Sagastizábal. The volume algorithm revisited: Relation with bundle methods. *Mathematical Programming*, 94(1):41–69, 2002. doi: 10.1007/s10107-002-0357-3.

[13] B. K. Bahinipati, A. Kanda, and S. G. Deshmukh. Coordinated supply management: Review, insights, and limitations. *International Journal of Logistics Research and Applications*, 12(6):407–422, 2009. doi: 10.1080/13675560802476382.

[14] F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000. doi: 10.1007/s101070050002.

[15] J. F. Bard and N. Nananukul. The integrated production—inventory—distribution—routing problem. *Journal of Scheduling*, 12(3):257–280, 2009. doi: 10.1007/s10951-008-0081-9.

[16] M. H. Bassett, J. F. Pekny, and G. V. Reklaitis. Decomposition techniques for the solution of large-scale scheduling problems. *AIChE Journal*, 42(12):3373–3387, 1996. doi: 10.1002/aic.690421209.

[17] O. Bastert, B. Hummel, and S. de Vries. A generalized wedelin heuristic for integer programming. *Informs Journal on Computing*, 22(1):93–107, Jan. 2010. doi: 10.1287/ijoc.1090.0328.

[18] E. Belavina and K. Girotra. The benefits of decentralized decision-making in supply chains. *INSEAD Working Paper No. 2012/79/TOM*, Sept. 2012. doi: 10.2139/ssrn.2141214. URL `http://www.insead.edu/facultyresearch/research/doc.cfm?did=50277`.

[19] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962. doi: 10.1007/BF01386316.

[20] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[21] G. R. Bitran, S. Gurumurthi, and S. L. Sam. Emerging trends in supply chain governance. Sloan Research Paper 4590-06, MIT, 2006. URL `http://mitsloan.mit.edu/pdf/ups.pdf`.

[22] T. Börgers. *An introduction to the theory of mechanism design.* University of Michigan, 2013. URL `http://www-personal.umich.edu/~tborgers/TheoryOfMechanismDesign.pdf`.

[23] R. Borndorfer, U. Schelten, T. Schlechte, and S. Weider. A column generation approach to airline crew scheduling. *ZIP report 05-37*, 2005. URL `www.zib.de/Publications/Reports/ZR-05-37.pdf`.

[24] M. Boudia. Coordination of production planning and distribution. *4OR: A Quarterly Journal of Operations Research*, 6(1):93–96, 2008. doi: 10.1007/s10288-007-0041-9.

[25] M. Boudia and C. Prins. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *European Journal of Operational Research*, 195(3): 703–715, 2009. doi: 10.1016/j.ejor.2007.07.034.

[26] M. Boudia, S. Dauzere-Peres, C. Prins, and M. A. O. Louly. Integrated optimization of production and distribution for several products. In *International Conference on Service Systems and Service Management*, volume 1, pages 272–277, 2006. doi: 10.1109/ICSSSM.2006.320625.

[27] J. Bracken and J. T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973. URL `http://www.jstor.org/stable/169087`.

[28] P. Brandimarte and M. Calderini. A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling. *International Journal of Production Research*, 33(1):161–181, 1995.

[29] P. Brucker and S. Knust. *Complex scheduling.* Springer, 2006.

[30] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41, 1999. doi: 10.1016/S0377-2217(98)00204-5.

[31] G. P. Cachon. *Supply chain coordination with contracts*, volume 11, pages 229 – 339. Elsevier, Amsterdam, 2003.

[32] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli. Solving a resource allocation problem in wireless mesh networks: A comparison between a cp-based and a classical column generation. *Networks*, 55(3):221–233, 2010. doi: 10.1002/net.20367.

[33] P. Chandra and M. L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517, 1994. doi: 10.1016/0377-2217(94)90419-7.

[34] C.-L. Chen, B.-W. Wang, and W.-C. Lee. Multi-objective optimization for a multi-enterprise supply chain network. *Industrial & Engineering Chemistry Research*, 42(9):1879–1889, 2003. doi: 10.1021/ie0206148.

[35] J.-M. Chen and T.-H. Chen. The multi-item replenishment problem in a two-echelon supply chain: the effect of centralization versus decentralization. *Computers & Operations Research*, 32(12): 3191–3207, 2005. doi: 10.1016/j.cor.2004.05.007.

[36] X. Chen and D. Simchi-Levi. Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case. *Operations Research*, 52(6):887–896, 2004. doi: 10.1287/opre.1040.0127.

[37] T. W. Chien. Determining profit-maximizing production/shipping policies in a one-to-one direct shipping, stochastic demand environment. *European Journal of Operational Research*, 64(1):83–102, 1993. doi: 10.1016/0377-2217(93)90010-K.

[38] M. Chiu and G. Lin. Collaborative supply chain planning using the artificial neural network approach. *Journal of Manufacturing Technology Management*, 15(8):787–796, 2004. doi: 10.1108/17410380410565375.

[39] E. Choi and D.-W. Tcha. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(7):2080–2095, 2007. doi: 10.1016/j.cor.2005.08.002.

[40] S. Chopra and P. Meindl. *Supply chain management: Strategy, planning, & operation.* Pearson Prentice Hall, 2007.

[41] W. H. J. Chu and C. C. Lee. Strategic information sharing in a supply chain. *European Journal of Operational Research*, 174(3):1567–1579, 2006. doi: 10.1016/j.ejor.2005.02.053.

[42] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4(2):28–34, 2002. doi: 10.1145/772862.772867.

[43] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand. *Decomposition techniques in mathematical programming.* Engineering and Science Applications. Springer, 2006.

[44] J.-F. Cordeau, G. Stojkovic, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388, 2001. doi: 10.1287/trsc.35.4.375.10432.

[45] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8 (1):101–111, 1960. URL `http://www.jstor.org/stable/167547`.

[46] P. P. Datta and M. G. Christopher. Information sharing and coordination mechanisms for managing uncertainty in supply chains: A simulation study. *International Journal of Production Research*, 49(3):765–803, 2011. doi: 10.1080/00207540903460216.

[47] L. B. Davis, R. E. King, T. J. Hodgson, and W. Wei. Information sharing in capacity constrained supply chains under lost sales. *International Journal of Production Research*, 49(24):7469–7491, 2011. doi: 10.1080/00207543.2010.535037.

[48] F. I. Dehayem Nodem, J. P. Kenne, and A. Gharbi. Hierarchical decision making in production and repair/replacement planning with imperfect repairs under uncertainties. *European Journal of Operational Research*, 198(1):173–189, 2009. doi: 10.1016/j.ejor.2008.09.001.

[49] M. R. Delavar, M. Hajiaghaei-Keshteli, and S. Molla-Alizadeh-Zavardehi. Genetic algorithms for coordinated scheduling of production and air transportation. *Expert Systems with Applications*, 37 (12):8255–8266, 2010.

[50] S. Dempe. *Foundations of bilevel programming.* Kluwer Academic Publishers, 2002.

[51] J. Desrosiers and M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 1–32. Springer US, 2005. doi: 10.1007/0-387-25486-2_1.

[52] C. Dhaenens-Flipo. Spatial decomposition for a multi-facility production and distribution problem. *International Journal of Production Economics*, 64(1-3):177–186, 2000. doi: 10.1016/S0925-5273(99)00056-0.

[53] M. Ebadian, M. Rabbani, S. A. Torabi, and F. Jolai. Hierarchical production planning and scheduling in make-to-order environments: Reaching short and reliable delivery dates. *International Journal of Production Research*, 47(20):5761–5789, 2009. doi: 10.1080/00207540802010799.

[54] S. S. Erenguc, N. C. Simpson, and A. J. Vakharia. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 115(2):219–236, June 1999.

[55] M. L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981. doi: 10.1287/mnsc.27.1.1.

[56] M. L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 50(12):1861–1871, 2004. doi: 10.1287/mnsc.1040.0263.

[57] P. Forget, S. D'Amours, J.-M. Frayret, and J. Gaudreault. Study of the performance of multi-behaviour agents for supply chain planning. *Computers in Industry*, 60(9):698–708, 2009. doi: 10.1016/j.compind.2009.05.005.

[58] F. Fumero and C. Vercellis. Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33(3):330–340, 1999.

[59] M. Ganesh, S. Raghunathan, and C. Rajendran. The value of information sharing in a multi-product, multi-level supply chain: Impact of product substitution, demand correlation, and partial information sharing. *Decision Support Systems*, Online:–, 2013. doi: 10.1016/j.dss.2013.01.012.

[60] A. Geoffrion. Lagrangean relaxation for integer programming. In M. Balinski, editor, *Approaches to Integer Programming*, volume 2 of *Mathematical Programming Studies*, pages 82–114. Springer Berlin Heidelberg, 1974. doi: 10.1007/BFb0120690.

[61] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20(5):822–844, 1974. doi: 10.1287/mnsc.20.5.822.

[62] M. Gnoni, R. Iavagnilio, G. Mossa, G. Mummolo, and A. D. Leva. Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. *International Journal of Production Economics*, 85(2):251–262, 2003. doi: 10.1016/S0925-5273(03)00113-0.

[63] A. Goicoechea, D. R. Hansen, and L. Duckstein. *Multi-objective Analysis With Engineering and Business Applications*. Wiley, New York, 1982.

[64] S. K. Goyal and S. G. Deshmukh. Integrated procurement-production systems: A review. *European Journal of Operational Research*, 62(1):1–10, 1992. doi: 10.1016/0377-2217(92)90172-6.

[65] I. E. Grossmann. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE Journal*, 51(7):1846–1857, 2005. doi: 10.1002/aic.10617.

[66] I. E. Grossmann and L. T. Biegler. Part ii. future perspective on optimization. *Computers & Chemical Engineering*, 28(8):1193–1218, 2004. doi: 10.1016/j.compchemeng.2003.11.006.

[67] GS1. Wine supply chain traceability: EAN-UCC standards application guideline, 2005. URL `http://www.gs1.org/docs/GS1_0003.pdf`.

[68] M. Guignard. Lagrangean relaxation. *Top*, 11(2):151–200, 2003. doi: 10.1007/BF02579036. URL `http://link.springer.com/article/10.1007%2FBF02579036`.

[69] O. Guyon, P. Lemaire, E. Pinson, and D. Rivreau. Cut generation for an integrated employee timetabling and production scheduling problem. *European Journal of Operational Research*, 201(2):557–567, 2010. doi: 10.1016/j.ejor.2009.03.013.

[70] A. Y. Ha and S. Tong. Contracting and information sharing under supply chain competition. *Management Science*, 54(4):701–715, Apr. 2008. doi: 10.1287/mnsc.1070.0795.

[71] D. C. Hall and C. Saygin. Impact of information sharing on supply chain performance. *The International Journal of Advanced Manufacturing Technology*, 58(1-4):397–409, 2012. doi: 10.1007/s00170-011-3389-0.

[72] E. W. Hans, W. Herroelen, R. Leus, and G. Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega*, 35(5):563–577, 2007. doi: 10.1016/j.omega.2005.10.004.

[73] A. N. Haq, P. Vrat, and A. Kanda. An integrated production-inventory-distribution model for manufacture of urea: A case. *International Journal of Production Economics*, 25(1-3):39–49, 1991. doi: 10.1016/0925-5273(91)90129-H.

[74] S. Hartmann and D. Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14, 2010. doi: 10.1016/j.ejor.2009.11.005. URL http://www.sciencedirect.com/science/article/pii/S0377221709008558.

[75] B. Heydenreich, R. Müller, and M. Uetz. Games and mechanism design in machine scheduling - an introduction. *Production and Operations Management*, 16(4):437–454, 2007. doi: 10.1111/j.1937-5956.2007.tb00271.x.

[76] M. I. Hohn. Relational supply contracts—optimal concessions in return policies for continuous quality improvements. *Lecture Notes in Economics and Mathematical Systems*, 629, 2010. doi: 10.1007/978-3-642-02791-8_2.

[77] G. Q. Huang, J. S. K. Lau, and K. L. Mak. The impacts of sharing production information on supply chain dynamics: A review of the literature. *International Journal of Production Research*, 41(7):1483–1517, 2003. doi: 10.1080/0020754031000069625.

[78] M. Hugos. *Supply chain coordination*, pages 183–211. John Wiley & Sons, Inc., 2011. doi: 10.1002/9781118386408.ch6.

[79] D. Huisman, R. Jans, M. Peeters, and A. P. M. Wagelmans. Combining column generation and lagrangian relaxation. In G. Desaulniers, J. Desrosiers, and M. Solomon, editors, *Column Generation*, pages 247–270. Springer US, 2005. doi: 10.1007/0-387-25486-2_9.

[80] T. J. R. Johnson. *An algorithm for the resource-constrained project scheduling problem*. PhD thesis, Massachusetts Institute of Technology (MIT), 1967. URL http://dspace.mit.edu/handle/1721.1/10584.

[81] H. Jung and B. Jeong. Decentralised production-distribution planning system using collaborative agents in supply chain network. *International Journal of Advanced Manufacturing Technology*, 25 (1-2):167–173, 2005. doi: 10.1007/s00170-003-1792-x.

[82] P. Kaminsky and D. Simchi-Levi. Production and distribution lot sizing in a two stage supply chain. *IIE Transactions*, 35(11):1065–1075, 2003. doi: 10.1080/07408170304401.

[83] A. P. Kanyalkar and G. K. Adil. Aggregate and detailed production planning integrating procurement and distribution plans in a multi-site environment. *International Journal of Production Research*, 45(22):5329–5353, 2007. doi: 10.1080/00207540600632174.

[84] A. Kazemi, M. H. F. Zarandi, and S. M. M. Husseini. A multi-agent system to solve the production-distribution planning problem for a supply chain: A genetic algorithm approach. *International Journal of Advanced Manufacturing Technology*, 44(1-2):180–193, 2009.

[85] J. D. Kelly and D. Zyngier. Hierarchical decomposition heuristic for scheduling: Coordinated reasoning for decentralized and distributed decision-making problems. *Computers and Chemical Engineering*, 32(11):2684–2705, 2008. doi: 10.1016/j.compchemeng.2007.08.007. URL http://www.sciencedirect.com/science/article/pii/S0098135407002128.

[86] S. W. Kim. An investigation on the direct and indirect effect of supply chain integration on firm performance. *International Journal of Production Economics*, 119(2):328–346, 2009. doi: 10.1016/j.ijpe.2009.03.007.

[87] A. Kovács, P. Egri, T. Kis, and J. Váncza. Inventory control in supply chains: Alternative approaches to a two-stage lot-sizing problem. *International Journal of Production Economics*, 143 (2):385–394, 2013. doi: 10.1016/j.ijpe.2012.01.001.

[88] D. M. Lambert and M. C. Cooper. Issues in supply chain management. *Industrial Marketing Management*, 29(1):65–83, 2000. doi: 10.1016/S0019-8501(99)00113-3.

[89] L. Lasdon and R. Terjung. An efficient algorithm for multi-item scheduling. *Operations Research*, 19(4):946 – 969, 1971.

[90] C. Lee and S. Park. Chebyshev center based column generation. *Discrete Applied Mathematics*, 159(18):2251–2265, 2011. doi: 10.1016/j.dam.2011.08.009. URL `http://www.sciencedirect.com/science/article/pii/S0166218X11003076`.

[91] Y. H. Lee and S. H. Kim. Optimal production-distribution planning in supply chain management using a hybrid simulation-analytic approach. In *Proceedings of Winter Simulation Conference, 2000*, volume 2, pages 1252–1259, San Diego, CA, USA, 2000. Society for Computer Simulation International. doi: 10.1109/WSC.2000.899093.

[92] L. Lei, S. Liu, A. Ruszczynski, and S. Park. On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):940–955, 2006. doi: 10.1080/07408170600862688.

[93] J. Lenstra, A. R. Kan, and P. Brucker. Complexity of machine scheduling problems. In B. K. P.L. Hammer, E.L. Johnson and G. Nemhauser, editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 343–362. Elsevier, 1977. doi: 10.1016/S0167-5060(08)70743-X.

[94] J. T. Lin and Y.-Y. Chen. A multi-site supply network planning problem considering variable time buckets- A TFT-LCD industry case. *The International Journal of Advanced Manufacturing Technology*, 33(9):1031–1044, 2007. doi: 10.1007/s00170-006-0537-z.

[95] M. Lohatepanont and C. Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19–32, 2004. doi: 10.1287/trsc.1030.0026.

[96] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53 (6):1007–1023, 2005. doi: 10.1287/opre.1050.0234.

[97] V. A. Mabert and M. A. Venkataramanan. Special research focus on supply chain linkages: Challenges for design and management in the 21st century. *Decision Sciences*, 29(3):537–552, 1998. doi: 10.1111/j.1540-5915.1998.tb01353.x.

[98] A. S. Manne. A linear programming model of the u. s. petroleum refining industry. *Econometrica*, 26(1):67–106, 1958. URL `http://www.jstor.org/stable/1907384`.

[99] C. T. Maravelias and C. Sung. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering*, 33(12):1919–1930, 2009. doi: 10.1016/j.compchemeng.2009.06.007.

[100] R. T. Marler and J. S. Arora. The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862, 2010. doi: 10.1007/s00158-009-0460-7.

[101] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi. Cloud computing—the business perspective. *Decision Support Systems*, 51(1):176–189, 2011. doi: 10.1016/j.dss.2010.12.006.

[102] A. J. Mason. Elastic constraint branching, the wedelin/carmen lagrangian heuristic and integer programming for personnel scheduling. *Annals of Operations Research*, 108(1-4):239–276, 2001. doi: 10.1023/A:1016023415105.

[103] L. Monostori, J. Váncza, and S. R. T. Kumara. Agent-based systems for manufacturing. *CIRP Annals - Manufacturing Technology*, 55(2):697–720, 2006. doi: 10.1016/j.cirp.2006.10.004.

[104] C. Moon, Y. H. Lee, C. S. Jeong, and Y. Yun. Integrated process planning and scheduling in a supply chain. *Computers & Industrial Engineering*, 54(4):1048–1061, 2008. doi: 10.1016/j.cie.2007.06.018.

[105] J. Mula, D. Peidro, M. Díaz-Madroñero, and E. Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204 (3):377–390, 2010. doi: 10.1016/j.ejor.2009.09.008.

[106] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, New York, NY, USA, 2007.

[107] H. Osman and K. Demirli. Integrated safety stock optimization for multiple sourced stockpoints facing variable demand and lead time. *International Journal of Production Economics*, 135(1):299 – 307, 2012. doi: 10.1016/j.ijpe.2011.08.004.

[108] N. Papadakos. Integrated airline scheduling. *Computers & Operations Research*, 36(1):176–195, 2009. doi: 10.1016/j.cor.2007.08.002.

[109] Y. B. Park. An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, 43(6):1205–1224, 2005. doi: 10.1080/00207540412331327718.

[110] A. Pessoa, E. Uchoa, M. P. de Aragao, and R. Rodrigues. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, 2:259–290, 2010. doi: 10.1007/s12532-010-0019-z.

[111] Y. Pochet and L. A. Wolsey. *Production planning by mixed integer programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[112] R. Rodriguez-Rodriguez, P. P. Gonzalez, and R. Leisten. From competitive to collaborative supply networks: A simulation study. *Applied Mathematical Modelling*, 35(3):1054–1064, 2011. doi: 10.1016/j.apm.2010.07.050.

[113] S.-J. Ryu, K. Nakayama, and H. Onari. Strategic supply chain coordination scheme under cloud computing environment considering market uncertainty. *International Journal of Business Strategy*, 11(3), 2011. URL `http://connection.ebscohost.com/c/articles/67681755`.

[114] G. K. Saharidis. Supply chain optimization: Centralized vs decentralized planning and scheduling. In P. Li, editor, *Supply Chain Management*. InTech, 2011. doi: 10.5772/15860.

[115] A. M. Sarmiento and R. Nagi. A review of integrated analysis of production-distribution systems. *IIE Transactions*, 31(11):1061–1074, 1999. doi: 10.1023/A:1007623508610.

[116] H. Selim, C. Araz, and I. Ozkarahan. Collaborative production-distribution planning in supply chain: A fuzzy goal programming approach. *Transportation Research Part E: Logistics and Transportation Review*, 44(3):396–419, 2008. doi: 10.1016/j.tre.2006.11.001.

[117] N. Shah, G. K. D. Saharidis, Z. Jia, and M. G. Ierapetritou. Centralized-decentralized optimization for refinery scheduling. *Computers & Chemical Engineering*, 33(12):2091–2105, 2009. doi: 10.1016/j.compchemeng.2009.06.010.

[118] W. Shen, Q. Hao, H. J. Yoon, and D. H. Norrie. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics*, 20(4):415–431, 2006. doi: 10.1016/j.aei.2006.05.004.

[119] T. M. Simatupang and R. Sridharan. The collaborative supply chain. *International Journal of Logistics Management*, 13(1):15–30, 2002. doi: 10.1108/09574090210806333.

[120] G. Singh and A. T. Ernst. Resource constraint scheduling with a fractional shared resource. *Operations Research Letters*, 39(5):363–368, 2011. doi: 10.1016/j.orl.2011.06.003.

[121] G. Singh and C. M. O'Keefe. Decentralised shared resource constraint scheduling with confidentiality protection. *Operations Research Letters*, 2013. URL `www.optimization-online.org/DB_FILE/2013/02/3769.pdf`.

[122] G. Singh and R. Weiskircher. Collaborative resource constraint scheduling with a fractional shared resource. *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, 2:359–365, 2008. doi: 10.1109/WIIAT.2008.47.

[123] G. Singh and R. Weiskircher. A multi-agent system for decentralised fractional shared resource constraint scheduling. *Web Intelligence and Agent Systems*, 9(2):99–108, Apr. 2011. doi: 10.3233/WIA-2011-0208.

[124] G. Singh, S. Dunstall, G. Eitzen, A. T. Ernst, M. Horn, M. Krishnamoorthy, B. Owens, and R. Weiskircher. On coordination and scheduling in wine supply chains. In *5th International Conference on Operational Research for Development (ICORD), Jamshedpur, India*, pages 219–229, 2005.

[125] G. Singh, D. Sier, A. T. Ernst, O. Gavriliouk, R. Oyston, T. Giles, and P. Welgama. A mixed integer programming model for long term capacity expansion planning: A case study from the hunter valley coal chain. *European Journal of Operational Research*, 220(1):210–224, 2011. doi: 10.1016/j.ejor.2012.01.012.

[126] S. K. Sinha. *Service level contracts for supply chains.* PhD thesis, Indian Institute of Technology Bombay (IITB), India, 2008.

[127] H. Stadtler. A framework for collaborative planning and state-of-the-art. *OR Spectrum*, 31(1): 5–30, 2009. doi: 10.1007/s00291-007-0104-5.

[128] M. Steinrücke. An approach to integrate production-transportation planning and scheduling in an aluminium supply chain network. *International Journal of Production Research*, 49(21):6559–6583, 2011. doi: 10.1080/00207543.2010.528461.

[129] J. Swaminathan, S. Smith, and S. Sadeh. Modelling the dynamics of supply chains: A multi-agent approach. *Decision Sciences*, 29(3):607–632, 1998. URL `http://www.cs.cmu.edu/afs/cs/user/sfs/www/papers/dsj04.pdf`.

[130] A. Taghipour and J.-M. Frayret. Mutual adjustment search with incentive for supply chain planning coordination. *International Journal of Computer Integrated Manufacturing*, 25(10):946–962, 2012. doi: 10.1080/0951192X.2011.646307.

[131] Z. Tao, Y. Wang, Z. Wu, and J. Hu. Rough approximation based decentralized bi-level model for the supply chain distribution problem. In J. Xu, J. A. Fry, B. Lev, and A. Hajiyev, editors, *Proceedings of the Seventh International Conference on Management Science and Engineering Management*, volume 242 of *Lecture Notes in Electrical Engineering*, pages 787–800. Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-642-40081-0_67.

[132] J. R. Tebboth. *A computational study of Dantzig-Wolfe decomposition.* PhD thesis, University of Buckingham, United Kingdom., 2001. URL `http://www.blisworthhouse.co.uk/OR/Decomposition/tebboth.pdf`.

[133] A. Thomas, M. Krishnamoorthy, G. Singh, and J. Venkateswaran. Collaborative services provision in production-logistics-scheduling: A coal industry example. In *2nd International Conference on Services in Emerging Markets (ICSEM-2011), IIT Bombay, India.* Service Research and Innovation Institute (SRII), 2011.

[134] A. Thomas, G. Singh, M. Krishnamoorthy, and J. Venkateswaran. Distributed optimisation method for multi-resource constrained scheduling in coal supply chains. *International Journal of Production Research*, 51(9):2740–2759, 2013. doi: 10.1080/00207543.2012.737955. URL `http://www.tandfonline.com/doi/abs/10.1080/00207543.2012.737955`.

[135] A. Thomas, M. Krishnamoorthy, G. Singh, and J. Venkateswaran. Coordination and the value of information in a decentralised producer-distributor supply chain. *Submitted to the International Journal of Production Economics*, 2014.

[136] A. Thomas, M. Krishnamoorthy, J. Venkateswaran, and G. Singh. Decentralised decision-making in a multi-party supply chain. *Submitted to the International Journal of Production Research*, 2014.

[137] A. Thomas, J. Venkateswaran, G. Singh, and M. Krishnamoorthy. A resource constrained scheduling problem with multiple independent producers and a single linking constraint: A coal supply chain example. *European Journal of Operational Research*, 236(3):946–956, 2014. doi: 10.1016/j.ejor.2013.10.006. URL `http://www.sciencedirect.com/science/article/pii/S0377221713008084`.

[138] D. J. Thomas and P. M. Griffin. Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1–15, 1996. doi: 10.1016/0377-2217(96)00098-7.

[139] T. Timm and A. Blecken. A method for the hierarchical planning of the structure, dimension and material requirements of manufacturing systems. *International Journal of Production Research*, 49 (12):3431–3453, 2011. doi: 10.1080/00207543.2010.495735.

[140] A. Toka, E. Aivazidou, A. Antoniou, and K. Arvanitopoulos-Darginis. Cloud computing in supply chain management: An overview. In D. Graham, I. Manikas, and D. Folinas, editors, *E-Logistics and E-Supply Chain Management: Applications for Evolving Business*. Idea Group Inc (IGI) Global, 2013. doi: 10.4018/978-1-4666-3914-0.

[141] P. Toktas-Palut and F. Ulengin. Coordination in a two-stage capacitated supply chain with multiple suppliers. *European Journal of Operational Research*, 212(1):43–53, 2011. doi: 10.1016/j.ejor.2011.01.018. URL `http://www.sciencedirect.com/science/article/pii/S0377221711000464`.

[142] R. N. Tomastik and P. B. Luh. The facet ascending algorithm for integer programming problems. In *Decision and Control, 1993., Proceed. of the 32nd IEEE Conference on*, volume 3, pages 2880–2884, Dec. 1993. doi: 10.1109/CDC.1993.325724.

[143] F. Vanderbeck and L. A. Wolsey. Reformulation and decomposition of integer programs. In M. Junger, T. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 431–504. Springer, Berlin, 2010. doi: 10.1007/978-3-540-68279-0_13.

[144] V. A. Varma, G. V. Reklaitis, G. E. Blau, and J. F. Pekny. Enterprise-wide modeling and optimization - an opportunity of emerging research challenges and opportunities. *Computers & Chemical Engineering*, 31:692–711, 2007. doi: 10.1016/j.compchemeng.2006.11.007.

[145] J. Venkateswaran. *Production and Distribution Planning in Dynamic Supply Chains using Multi-Resolution Hybrid Models*. PhD thesis, The University of Arizona, 2005.

[146] C. J. Vidal and M. Goetschalckx. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, 98(1):1–18, 1997. doi: 10.1016/S0377-2217(97)80080-X.

[147] G. L. Wang, L. Lin, and S. S. Zhong. Task-oriented hierarchical planning and scheduling for discrete manufacturing enterprise. *Applied Mechanics and Materials*, 10-12:114–120, 2008. doi: 10.4028/www.scientific.net/AMM.10-12.114.

[148] D. Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57(1):283–301, 1995. doi: 10.1007/BF02099703.

[149] P. Wentges. Weighted dantzig-wolfe decomposition for linear mixed-integer programming. *International Transactions in Operational Research*, 4(2):151–162, 1997. doi: 10.1111/j.1475-3995.1997.tb00071.x.

[150] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*. Wiley-Interscience, 1 edition, Nov. 1999.

[151] D. Wu and M. Ierapetritou. Hierarchical approach for production planning and scheduling under uncertainty. *Chemical Engineering and Processing: Process Intensification*, 46(11):1129–1140, 2007. doi: 10.1016/j.cep.2007.02.021.

[152] Y. Wu, C. G. Cegielski, B. T. Hazen, and D. J. Hall. Cloud computing in support of supply chain information system infrastructure: Understanding when to go to the cloud. *Journal of Supply Chain Management*, 49(3):25–41, 2013. doi: 10.1111/j.1745-493x.2012.03287.x.

[153] M.-M. Yu, S.-C. Ting, and M.-C. Chen. Evaluating the cross-efficiency of information sharing in supply chains. *Expert Systems with Applications*, 37(4):2891–2897, 2010. doi: 10.1016/j.eswa.2009.09.048. URL `http://www.sciencedirect.com/science/article/pii/S095741740900829X`.

[154] H. Zhou and W. C. Benton-Jr. Supply chain practice and information sharing. *Journal of Operations Management*, 25(6):1348–1365, 2007. doi: 10.1016/j.jom.2007.01.009.

[155] D. Zissis and D. Lekkas. Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3):583–592, 2012. doi: 10.1016/j.future.2010.12.006.

[156] N. Zribi, I. Kacem, A. El Kamel, and P. Borne. Assignment and scheduling in flexible job-shops by hierarchical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(4):652–661, July 2007. doi: 10.1109/TSMCC.2007.897494.

Acknowledgments

In the long way to achieve my dream, many people were actively involved in different roles. Frankly speaking, I will not be able to thank them appropriately in a few words or sentences. Unlike other graduate courses, if you compare PhD program with a journey, it is mostly self driven and evolves over time. There is no guarantee that whatever you do during PhD will always be in the right track. Often, you have to retrace and take a new route. Therefore, it is important to have an appropriate vehicle, sufficient fuel, right directions, an encouraging environment and self confidence. I thank the Almighty for giving me the strength and the ability to undertake this long journey to pursue my dream.

It is my family who inspired me to dream high. I am indebted to my parents (Thomas & Vimala) for providing me the right education and mindset. In my PhD journey, they provided me the appropriate vehicle. I am proud to have a wonderful brother, Vinu, who is my motivation, guide, and acted as a traffic signal at critical junctions. Jincy, my wife, joined with me in the last phase of PhD, and helped me bring a timely completion to my journey. I owe her an apology for forcing her to read thesis chapters in the place of love letters and spend time in PhD lab instead of movie halls/beaches. I take this opportunity to thank my parents, brother, wife and my little princess–Isah. I would also like to acknowledge the support and help offered by in-laws and other family members.

I have been fortunate to have such profoundly knowledgeable and energetic supervisors to guide me throughout my research work. I take this opportunity to thank them for their knowledge and support. I am indebted to Prof. Mohan Krishnamoorthy, Prof. Jayendran Venkateswaran and Dr. Gaurav Singh for their patience, active supervision, and professional and personal advices. They showed me the right direction to complete my journey.

I used to often wonder at Prof. Mohan Krishnamoorthy's energy to manage the CEO (of IITB-Monash Research Academy) and the supervisory roles without any compromise. He used to find sufficient time to sit with his students even in the late evenings to motivate and encourage them. He insisted me to aim high, pushed for timely deliveries and helped me hone my professional skills. Often I could not match my hard work with his passion. I respect and admire his talent and passion towards hard work.

Prof. Jayendran Venkateswaran always supported and inspired me to explore the possibilities of a wider research domain. His inputs were really helpful for my research to have different perspectives and hence deliver the outcomes in perfection. More than a supervisor he was like a colleague, a good friend and a great mentor. I express my gratitude for his help and guidance which supported me in difficult times during the course of my PhD.

Dr. Gaurav Singh, a true professional and hardcore researcher, has showed immense patience in helping me define the research problem and to explore diverse solution approaches. He always gave me the freedom to choose a research path, while at the same time, directing it through the right course. During my visit to CSIRO, he gave me all the support to quickly adapt to their environment and allowed me to leverage CSIRO's computational servers for the research work. I express my deepest gratitude to him for his invaluable guidance and support.

Even though my thesis was on 'Supply chain coordination', I often worried how to coordinate three supervisors who have different perspectives about my research problem. I thank my supervisors for spending adequate time for video conferences and email communications and sharing their views which acted as a catalyst to reach at some consensus. This helped me in a great way to focus on the research problems in depth and publish the findings in reputed journals.

My sincere thanks to Dr. Andreas Ernst (CSIRO, Australia) and Prof. Narayan Rangaraj (IITB) for being my Research Progress Committee members. The feedback during my presentations, and the interactions which I had with them were really valuable for me and my thesis. I would also like to thank all the examiners of my thesis for their valuable feedback. I extend my gratitude to Prof. N Hemachandra (Convenor, IEOR) for the informal discussions and the overall support offered to me during the tenure of my PhD.

IITB-Monash Research Academy (IMURA) is an unique attempt in extending research collaboration beyond conventional boundaries. Since we joined in the early stages of formation, we had lots of questions and confusions. I thank the Cross Institutional Program Committee (CIPC) and the staff at IMURA for providing us timely updates, streamlining the processes, supporting our visit to CSIRO, and for timely managing the administrative procedures. I extend my gratitude to all the non academic staff and other professors at IIT Bombay, CSIRO Australia, Monash University and IMURA, for enabling a motivating research atmosphere.

I thank CSIRO for sponsoring my research and offering the high speed computational facilities for the same. I appreciate the efforts of Dr. Andreas Ernst and CSIRO in strengthening the industry-academia collaboration.

I thank Ms Sheba Sanjay, Communications and Training Services Consultant at IMURA, for doing the language review for my thesis and papers. I would also like to acknowledge the help of Aarthi Nandakumar, Jincy Joseph and Jincy P. Manuel to improve the presentation of my thesis.

I would like to acknowledge the support and teaching offered by the faculty members of IEOR. I greatly benefited as a student and as well as a teaching assistant. Under the leadership of Prof. Narayan Rangaraj and Prof. N. Hemachandra, overall research aptitude has improved a lot.

The friends at IEOR PhD lab, aka Chai Buddies[‡], provided the pro active support in all these years. They help me in academic and non-academic matters. I would not have enjoyed my research without Chai Buddies. I keep the following names very close to my heart: Aarthi Nandakumar, Anirban Ghatak, Ketki Kulkarni, Balamurali AR, Manu Gupta, Vikas V Singh, Alok Patel, and Ratnaji Vanga. They were a part of my last five year's life.

Other friends to be mentioned include Subir Rao, Malabika Biswas, Jasvipul Singh, Naveen Nair, Ramkumar Rajendran, Ajay Gopal, Rihab AbdulRazak, Joby Joseph and Mathew P. Mathews. I would always cherish the moments which I spent with them. I thank each and everyone for their support and help. There are so many other people who have influenced me throughout my academic career. I thank all of them.

Date:17-10-2014 Anu Thomas

[‡] http://home.iitb.ac.in/~manu.gupta/chaibuddies.html