Distributed optimisation method for multi-resource constrained scheduling in coal supply chains

10 – Feb – 2012

Anu Thomas
IITB Monash Research Academy,
Industrial Engineering and Operations Research,
IIT Bombay, Powai, Mumbai – 400076, India.
email: anuthomas@iitb.ac.in

Gaurav Singh
CSIRO Mathematics Informatics and Statistics,
Clayton, VIC – 3168, Australia.
email: gaurav.singh@csiro.au

Mohan Krishnamoorthy
IITB Monash Research Academy,
IIT Bombay, Powai, Mumbai – 400076, India.
email: mohank@iitbmonash.org

Jayendran Venkateswaran
Industrial Engineering and Operations Research,
IIT Bombay, Mumbai – 400076, India.
email: jayendran@iitb.ac.in

1 Introduction and Motivation

Modern enterprises operate in a competitive and global supply network. The network structure and inter-dependencies of enterprises means that newer decision models need to be explored to replace traditional sequential decision models. In current business environments, supply chain members cannot operate in complete isolation or be fully dependent on alliances. Therefore, supply networks require decision models that integrate distributed decisions that need to be made. Such integration has been explored at different levels of decision making: strategic, tactical and operational levels [2]. At the same time, there have been attempts to integrate common enterprise operations such as inventory holding and logistics [19, 27]. Integrated approaches to decision modelling are possible. However, this requires substantial information sharing. The models tend to be complex and could involve multiple objectives. Thus, distributed optimisation models are required.

Every enterprise is a collection of many dependent sub-units. For example, in a manufacturing supply chain, procurement, production and distribution decisions are interrelated. The OR literature provides many examples of individual optimisation models for each of these decision making sub-units (see, for example, [32, 34, 24]). However, in arriving at decision models, each of these sub-units makes assumptions on other sub-unit's operations. If these assumptions are not correct, the global system is rendered inflexible due to rapid changes in the environment. To achieve overall optima in the system each of these sub-units should be responsive to the decisions of the system, even if they are greedy to improve their individual objectives [13]. Enterprises in industry sectors such as petroleum, coal, and airlines have very large and complex structures with many inter-dependencies. They need to have long term (strategic) integrated plans and short term (operational) integrated plans. A very small improvement in the performance brings a large profit in these enterprises (see [15, 4, 25]).

This paper addresses an integrated planning and scheduling problem motivated by the coal supply chains in Australia. Integrated decision models are required in this case because of many different players in such systems. Australia is one of the leading coal exporter in the world. According to [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. The old mining areas were close to ports and transport facilities were well established. However a surge in the demand for coal expanded the discovery and production from 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. Coal transport trains are among the longest in the world. Many of them have as many as 6 locomotives and 148 wagons, amounting to a total length of more than 2 kilometres and carrying about 8500 tonnes of coal[1]. However small-sized trains (approx 2800 tonnes) are also used for everyday trips to meet the periodic demand of domestic customers such as power plants and heavy industries[2]. 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*. Around 10 train trips are required to make a stockpile. Generally two or three such stockpiles are required to complete only one a shipload.

Several giant machines like stackers and reclaimers are required to load or unload the coal at the mines and terminals. The unavailability of these machines unnecessarily delays trains

[1] http://www.australiancoal.com.au visited on 10-Oct-2011

[2] http://www.pacificnational.com.au/services/coal.asp visited on 10-Oct-2011

and creates inefficiencies in the system. Many other scheduling complexities such as maintenance schedules of equipment, terminal stockpile capacities and other similar constraints are also present in the coal supply chain scheduling. Due to high cost of infrastructure and investments required for capital equipment, rail network and storage capacity cannot be increased rapidly. Instead, existing facilities need to be utilised optimally. For fairness reasons, long-term contracts also exist between various players in such a network. However, short-term operational scheduling problems are also regularly required to be solved to ensure that the long-term contracts are satisfied.

It is possible to formulate and solve the integrated whole-of-supply chain problem as a single decision-making problem that includes all sub-units of the supply chain. Most scheduling problems are *NP-hard* (see [20, 10]). Therefore a single scheduling model which includes all decision models for each of these sub-units would be an intractable. In general, the non-availability of shared resources creates major bottlenecks in coal production-distribution logistics. Resource constrained scheduling problems (RCSP) are a special class of problems where the partners are required to share the scarce resources like trains, tracks and terminal facilities.

The problem considered in this paper is a large and complex multi-resource constrained scheduling problem. Unlike in traditional RCSP, here we have different options provided by the resource manager to meet the requirement. The key decision of this problem is to identify train trips between the mines and the terminal; we refer to these trips as *jobs*. Every mine will have many jobs that need to be completed. Out of these, a set of jobs will be completed to meet one or more *orders* that need to be supplied to the terminal. The set of jobs that are required to complete a particular order is not known *a priori*. Indeed, that is part of the decision that is required to be made. All mines pick a set of jobs that satisfy their demand requirement (orders). The proposed model considers tardiness and earliness penalties along with operational costs. Problems that include tardiness/earliness are considered amongst the toughest problems in scheduling literature [5, 21]. Multiple orders and the indefinite nature of resource allocation make this problem unique. Given the above description of the problem, the critical decision in our problem is selecting the set of jobs for an order and defining the best schedule for these orders.

2 Related Literature

Integrated planning and scheduling is an active research area. Large industrial problems are solved using scheduling models. Review articles by [32, 34, 27, 23, 24] specifically cover recent advances and trends in production-distribution integrated modelling. [27] consider the integration of the logistic and production functions and discuss its possible benefits in cost saving and efficiency improvement. The review provides a classification on production/distribution/inventory models and inventory/routing problems. Rather than solving the dependent models independently or with a single model, sequential modelling is preferred in many problems. Which is dependent on the level of joint decision making and complexity [27, 8, 11]. The main disadvantage of this approach is that the second player has to wait for the first player's decision and so on. It may not provide the *best* for the second player. [24] provides a review of mathematical programming models for supply chain production and transport planning, in which models are classified based on performance and novelty measures. This review provides a quick look at different modelling approaches such as LP, (M)ILP, NLP, multi-objective, fuzzy programming, heuristics, meta-heuristics, stochastic programming and hybrid models. The authors highlight the need and potential of the integrated production and transport planning. [12] explored an integrated scheduling model of production and distribution operations in the computer and food catering service industries. A heuristic solution approach is used to solve the machine scheduling problem. The authors discussed the value of integration by comparing the proposed algorithm with a sequential approach. There was no inventory holding cost and resource constraint in this problem. Single integrated (centralised) models are often simplified and solved by decomposition of this model into several sub-problems.

[18] discusses an agent based concept to support distributed scheduling in a supply chain where multiple independent and autonomous enterprises operate with partial information sharing. [28] presents a decomposition strategy to split a large centralised refinery scheduling problem into smaller sub-problems such that each sub-problem's solution will be feasible in the overall system. The strategy was supported with case studies. [26] analysed a bilevel decision-making problems exist between the distribution network planning and the production planning. [16] review some important MILP decomposition techniques – such as Benders decomposition, Lagrangian relaxation, cross-decomposition and bilevel decomposition – for dealing with these 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 [16], is to identify the right decomposition strategy that balances the computational gains with the cost of the time and the resources invested. [22] 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. Some of above mentioned decomposition methods and improvement schemes are used in our proposed distributed model.

Resource constrained scheduling problems have continuously attracted and challenged researchers since the term was introduced by [17]. A comprehensive survey of RCSP models and solution methods can be found in [9]. The scheduling problem discussed in this paper is motivated from the problem discussed in [29] and [30]. They looked at a special case of resource constrained scheduling where there are multiple processors who want to execute jobs that each need a fractional amount of an additional resource with an objective to minimise total weighted tardiness.

The problem considered in this paper is a multi-resource-constrained problem with earliness and tardiness as one of the objectives. This problem has several differences with the traditional job scheduling or batch scheduling problem. The main difference is that the definition of a 'job' is not very straight-forward as, unlike traditional scheduling problems, not all the 'jobs' are required to be completed. A service engineering perspective of this problem is presented in [31]. The paper proposed a decentralised and collaborative service provision scheme to achieve the benefits of the centralised model under partial information sharing. To the authors' best knowledge, other than [31], the literature does not include attempts in this direction.

The rest of the paper is formatted as follows. Section 3 explains the specifics of the problem including a description of its unique facets. An MILP formulation for the problem is presented in section 4. Section 5 describes a distributed algorithm based on Lagrangian relaxation, its features and a procedure to compute upper bounds. Results of detailed computational experiment are presented in section 6.

3 A Scheduling Problem in Coal Supply Chains

The coal industry contributes significantly to the Australian economy and energy production. The supply chain is always under the pressure to maximise throughput and minimise inefficiencies. The main partners in this supply chain are *mines*, *rail operators*, *track owners* and *ports/terminals*. Mines produce/excavate coal in anticipation of ship arrivals at the terminal. Rail is the most popular mode of coal transport.

Independent operations such as mining, rail scheduling and track maintenance are carried out by different partners of this supply chain. Each of these partners have full control in their domain and shares necessary information with other partners to work without conflicts. Mines make decisions on production rates, quantity and storage. Rail operators are concerned about the schedule and how it affects utilisation of rolling stock that operates between the various mines and the terminal. The terminal manages ship arrivals, departures and loading. It is likely that some of these users are part of other supply networks too. For example, the same rail operator may provide a service to different mines in another supply network. Similarly, the track operator might lease the track (and the terminal may lease their services under a contract) to another supply network too. Scheduling complexities of this supply chain is discussed in section 1. In this paper, we consider a simplified case where the supply chain has a single terminal (port), one common rail operator and multiple mines. The common rail operator schedules the trains between all the mines and the terminal. The port and other track operational decisions are not considered in this paper, but will be in future research.

The partners in the supply chain will usually have long-term contracts for the production and movement of coal. Some of these contracts are made 6 months ahead of time. However, it is not possible to derive detailed operational schedules based on this information, becuase the demands might only be approximate in the sense that we may not know exactly when ships will arrive. Each of the mines and the rail operator need to engage in short term scheduling to plan their production and utilisation. Train schedules are prepared for shorter periods such as a week or a fortnight. And in such an environment, even if we do not consider dynamic alterations that arise from delays, unplanned maintenance, tide-inflicted changes and other factors, it is still necessary to schedule appropriately so that resources are effectively utilized. Much of this would depend on the previous horizon's operations and outcomes. The mines and the rail operator have to determine their *best schedule* for a particular planning horizon. The problem under consideration is executed at the operational level and is assumed to have deterministic demands (demands are determined at the start of each planning horizon).

Based on the ship arrival and other priorities, the terminal computes the demand for each mine. A due-date and a *demurrage cost* (for keeping a ship waiting) are also associated with every demand. The mines may receive more than one order, each comprising a due date and a quantity. Each mine has to schedule their production subject to production constraints to meet their demands with minimum cost. The common rail operator provides the trains to move the coal out from the mine to the terminal. The rail operator has a pool of trains in different classes. The train classes are defined with respect to the capacity and other features. Mines do not care for individual trains. Whenever they make a request for a certain class of train, the request can be fulfilled by any of the trains available in that class. In a planning week, the total number of available trains in a class is also revised based on maintenance schedule and other priorities. Short-term planning for a week or fortnight accommodates these variations.

Each class of train has specific properties such as load/capacity, journey time and loading time.

Mostly the rail operator uses a set of trains for a region, where almost all mines are identical but independently operated. In other cases, the rail operator may assign a dedicated train and route. At present we ignore the case of dedicated trains. Since we consider the operations in a region, travel time, loading and other operations related with trains are assumed to be dependent only on the class. Loading time is directly proportional to the size of the train. Each train trip has three stages: (a) forward travel from terminal to mine, (b) loading at the mine, and (c) return travel from mine to terminal with the full train-load of coal. Since this is a planning model, we do not allow for waiting time or queuing time between stages or during any stage. A train will be idle only at the terminal if there is no request for that train class from the mines. At present, we consider only the mines and the rail operator in this model. Terminal related constraints are not considered.

The production planning for each mine will include constraints such as inventory balancing, production capacity and order satisfaction. Backlogs are not permitted at 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. Therefore 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. Finally, and most importantly perhaps, late delivery and late satisfaction of a request at the terminal is permitted but penalised. Such late deliveries imply that a ship is waiting in the terminal/port without sufficient coal being available. It is reasonable to assume that every mine can at most delay one order (ship) at any time. The delay 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.

Other than the integrated train scheduling, each mine has independent production planning problem. Therefore the overall scheduling problem can be viewed as distributed planning and integrated scheduling problem. We can either capture this problem in a single *integrated* model or distributed planning models coordinating with each other to find a schedule with feasible, and possibly optimal, resource utilisation.

To understand the complexity of this problem, let us first consider a particular case where all the sub-problems comprise single mine with n requests, all with release date zero and a single train. In this case, finding a "resource feasible" (here the resource is train) schedule which minimise total weighted tardiness is equivalent to $1||\sum wT$ problem, which is known to be NP-hard (see [20]). Therefore, in the general sense, the train scheduling problem and the integrated planning-scheduling problem are NP-hard.

3.1 Uniqueness of the problem

In this problem, each mine receives multiple orders from the terminal. Each order can be met by different combination of train classes. And 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 for more than one order. 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 train size, then the mines have to either allow overstock at the terminal or pay demurrage. Since there is penalty for overstock and demurrage, we 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.

4 Mathematical Model

In this section we present a mixed integer linear programming (MILP) formulation for the problem described in section 3. In the remaining sections, we refer to this model as *integrated* model (IM).

Indexes

i	index of mines $(= 1, 2, \ldots, I)$
τ	index of train classes $(= 1, 2, \ldots, N)$
t	index of time periods $(= 0, 1, \ldots, T)$
k	index of orders $(= 1, 2, \ldots, K_i)$

Parameters

$A^t_{i\tau}$	cost incurred by mine i for requesting a train class τ at time t
C^t_i	tardiness /demurrage cost of mine i (per order) at time t
F^t_i	inventory cost (per unit quantity) at terminal of product from mine i at time t
H^t_i	inventory cost (per unit quantity) at mine i at time t
Δ_{ik}	due date of order k for the mine i
Q_{ik}	k^{th} order quantity of the mine i
D^t_i	cumulative demand of mine i at time t
P_i	production capacity of mine i
B_i	inventory holding capacity of mine i
V_τ	size of the trains in class τ
R_τ	number of trains in class τ
S_τ	forward travel time required for the trains in class τ to reach a mine from the terminal
L_τ	loading time required for the trains in class τ
Ψ_τ	return travel time required for the trains in class τ, from a mine to the terminal
E_τ	$= L_\tau + \Psi_\tau$, time required for loading and return travel to the terminal
G_τ	$= S_\tau + E_\tau$, turn-around time required for a trip

Decision variables

θ^t_i	Inventory level at mine i by the end of time t
ν^t_i	Amount produced by mine i in period t
ω^t_i	Over stock level at the terminal of product from mine i by the end of time t
y^t_i	Binary variable to indicate whether mine i has an order tardy at time t. Zero indicates no order is tardy.
$q^t_{i\tau}$	Total number of trains from class τ requested by the mine i on or before time t.

A delivery order k from the terminal for a mine i can be seen as an ordered pair of due date

(Δ_{ik}) and quantity (Q_{ik}). Without loss of generality we assume that $\Delta_{ik} < \Delta_{ik+1}$ and k^{th} order must be met before satisfying the demand for $(k+1)^{\text{th}}$ order. We define the cumulative demand for a mine i at time t as $D_i^t = \sum_{\{k|\Delta_{ik} \leq t\}} Q_{ik}$. For example, if mine i has 3 orders $\{(45, 10500), (89, 9300), (120, 11800)\}$. Then the cumulative demand will be defined as

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

Similarly, the demurrage cost, C_i^t, will be defined as

$$C_i^t = \begin{cases} 0 & \text{if } 0 \leq t < 45 \\ C_{i1} & \text{if } 45 \leq t < 89 \\ C_{i2} & \text{if } 89 \leq t < 120 \\ C_{i3} & \text{if } 120 \leq t \end{cases},$$

where C_{i1}, C_{i2}, C_{i3} are constant.

Objective

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 mines. This gives

$$[\mathbf{IM}] \qquad \min \sum_i \sum_t \left[\theta_i^t H_i^t + \omega_i^t F_i^t + y_i^t C_i^t + \sum_\tau (q_{i\tau}^t - q_{i\tau}^{t-1}) A_{i\tau}^t \right] \tag{4.1}$$

Constraints

Here we provide all the necessary constraints in the model

1. Total number of train requested by a mine at any time is non-decreasing.

$$q_{i\tau}^t \geq q_{i\tau}^{t-1} \qquad \forall\, i, \tau, t \tag{4.2}$$

2. The amount produced at any time t by a mine is not more than its production capacity.

$$\nu_i^t \leq P_i \qquad \forall\, i, t \tag{4.3}$$

3. The inventory balancing constraint

$$\theta_i^t = \theta_i^{t-1} + \nu_i^t - \sum_\tau (q_{i\tau}^t - q_{i\tau}^{t-1}) V_\tau \qquad \forall\, i, \tau, t \tag{4.4}$$

   The production variable can be computed from this equality constraint and substituted in previous constraint.

4. The ending inventory at any time cannot be more than the holding capacity at the mine.

$$\theta_i^t \leq B_i \qquad \forall\, i, t \tag{4.5}$$

5. If the cumulative supply by a mine i at time t is more than its demand then the mine has an over-stock.

$$\sum_\tau q_{i\tau}^{t-E_\tau} V_\tau - D_i^t \leq \omega_i^t \qquad \forall\, i, t \tag{4.6}$$

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

$$y_i^t \geq 1 - \frac{\sum_\tau q_{i\tau}^{t-E_\tau} V_\tau}{D_i^t} \qquad \forall\, i, t \geq \Delta_{i1} \tag{4.7}$$

7. $(k-1)^{\text{th}}$ order must be delivered before the due-date of the k^{th} order. This implies that at any time t at most one order for a mine can be tardy.

$$\sum_\tau q_{i\tau}^{\Delta_{ik}-E_\tau} V_\tau \geq D_i^{\Delta_{ik-1}} \qquad \forall\, i, k \tag{4.8}$$

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

$$\sum_\tau q_{i\tau}^T V_\tau \geq D_i^T \qquad \forall\, i \tag{4.9}$$

9. Not more than one train can load at any mine.

$$\sum_\tau (q_{i\tau}^t - q_{i\tau}^{t-L_\tau}) \leq 1 \qquad \forall\, i, t \tag{4.10}$$

10. Total number of trains of a class τ running at any time t cannot exceed its maximum availability.

$$\sum_i (q_{i\tau}^{t+S_\tau} - q_{i\tau}^{t-E_\tau}) \leq R_\tau \qquad \forall\, \tau, t \tag{4.11}$$

11. Boundary conditions and scope.

$$q_{i\tau}^0 = 0, \nu_i^0 = 0, \theta_i^0 = 0, \omega_i^0 = 0, y_i^T = 0,\ y_i^t \in \{0,1\}, \theta_i^t, \omega_i^t \geq 0, q_{i\tau}^t \in \mathbf{Z}^+. \tag{4.12}$$

Additional Constraints

In this section we present some extra constraints which can be used to further tighten the model.

1. Cumulative supply at time t cannot be more than the total production. That means

$$\sum_\tau q_{i\tau}^t V_\tau \leq \theta_i^0 + tP_i$$

Due to the holding cost at mine and terminal, the final total supply will not be more than D_i^T. Also, D_i^T need not be a linear combination of train sizes, hence we allow a margin in final supply.

$$\sum_\tau q_{i\tau}^t V_\tau \leq D_i^T + V_{\max}$$

where $V_{\max} = \max_\tau \{V_\tau\}$. From these two constraints, we define an upper bound M_i^t for the cumulative supply. That means,

$$\sum_\tau q_{i\tau}^t V_\tau \leq M_i^t \quad \forall\, i, t \text{ where } M_i^t = \min\{\theta_i^0 + tP_i, D_i^T + V_{\max}\} \tag{4.13}$$

2. Since the supply is bounded, over-stock also has to be bounded.

$$\omega_i^t \leq M_i^t - D_i^t \qquad \forall\, i, t \tag{4.14}$$

3. Demurrage should not be charged if there is an over-stock at the terminal.

$$y_i^t \leq 1 - \frac{\omega_i^t}{M_i^t - D_i^t} \qquad \forall\, i, t \tag{4.15}$$

4. Constraints (4.13) is similar to a Knapsack constraint and therefore can be further tightened with the integer cuts proposed by [3]. Let ub_τ be the upper bound of $q_{i\tau}^t$ which satisfies (4.13) after fixing all other variables to zero. The set $C \subset \{1, 2, \ldots, N\}$ is a 'cover' if $\sum_{\tau \in C} ub_\tau V_\tau - M_i^t \geq 0$. Then a stronger cut can be derived as,

$$\sum_{\tau \in C} (ub_\tau - q_{i\tau}^t) \geq \left\lceil \frac{M_i^t}{\overline{V}} \right\rceil \qquad \forall\, i, t \tag{4.16}$$

where $\overline{V} = \max_{\{\tau \in C\}} V_\tau$ and $ub_\tau = \lfloor M_i^t / V_\tau \rfloor$.

5. Similarly we can use an integer cut from the constraint (4.7). Since both y and q are integer variables, this cut will help us to improve the relaxed solution.

$$(1 - y_i^t) \left\lceil \frac{D_i^t}{V_{\min}} \right\rceil \leq \sum_{\tau} q_{i\tau}^{t - E_\tau} \left\lceil \frac{V_\tau}{V_{\min}} \right\rceil \qquad \forall\, k, \Delta_{ik} \leq t < \Delta_{ik+1} \tag{4.17}$$

where $D_i^t = \sum_{\{k | \Delta_{ik} \leq t\}} Q_{ik}$ and $V_{\min} = \min_\tau \{V_\tau\}$.

6. 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,

$$y_i^t \leq y_i^{t-1} \qquad \forall\, i, k, \Delta_{ik} < t < \Delta_{ik+1}. \tag{4.18}$$

From the above formulation, it can be observed that resource utilisation constraints (4.11) are the only set of constraints that link the mines. In the absence of this constraint, individual mine's production planning problem can be separated. Therefore, in the next section we propose a distributed optimisation method based on Lagrangian relaxation algorithm to solve this problem. In section 5.1, features of the Lagrangian relaxation such as application of the Volume algorithm [6] and Wedelin approximations [35] are explained. Another motivation for using Lagrangian relaxation was, as results in section 6 show, that in several instances CPLEX[3] was not able to find a single feasible solution even in one hour of cpu time for the above-mentioned model.

5 Lagrangian Relaxation

Lagrangian relaxation is a well known and widely used method for the decomposition of large problem into many smaller easily solvable problems. A detailed discussion on the primal dual

[3] IBM ILOG CPLEX Optimizer, url: http://www.ibm.com/software/integration/optimization/cplex-optimizer/

relationship, feasibility gap and other related issues of Lagrangian can be found in [36] and [14]. Lagrangian relaxation methods are strengthened with many heuristics and approximations developed on the characteristics of the problem.

For the model presented in earlier section, the complicating and interlinking train resource constraint (4.11) is relaxed as it allows us to decompose the integrated planning-scheduling problem for each of the independent mines with a modified train request cost. The relaxed problem becomes,

$$
\begin{aligned}
Z(\lambda) = \min \sum_i \sum_t & \left[\theta_i^t H_i^t + \omega_i^t F_i^t + y_i^t C_i^t + \sum_\tau (q_{i\tau}^t - q_{i\tau}^{t-1}) A_{i\tau}^t \right] \\
& + \sum_\tau \sum_t \lambda_\tau^t \left(\sum_i \left(q_{i\tau}^{t+S_\tau} - q_{i\tau}^{t-E_\tau} \right) - R_\tau \right) \qquad (5.1)
\end{aligned}
$$

subject to $(4.2) - (4.10), (4.12) - (4.18)$.

where λ_τ^t (≥ 0) is the Lagrangian multiplier corresponds to the resource constraint at time t for the train class τ. Then the objective (4.1) for i^{th} mine can be updated as,

$$
Z_i(\lambda) = \min \sum_t \left(\theta_i^t H_i^t + \omega_i^t F_i^t + y_i^t C_i^t + \sum_\tau (q_{i\tau}^t - q_{i\tau}^{t-1}) \bar{A}_{i\tau}^t \right) \qquad (5.2)
$$

$$
\text{where } \bar{A}_{i\tau}^t = A_{i\tau}^t + \sum_{u=t-S_\tau}^{t+E_\tau-1} \lambda_\tau^u
$$

Note that, even though the resource constraint (4.11) is relaxed, a disaggregated version of the constraint can still be added in this relaxed formulation for individual mines. This ensures that the solutions found by individual mines will be feasible with respect to the resource constraint. We have,

$$
(q_{i\tau}^{t+S_\tau} - q_{i\tau}^{t-E_\tau}) \leq R_\tau \qquad \forall\, i, \tau, t \qquad (5.3)
$$

[29] had used a Lagrangian relaxation based algorithm strengthened with concepts of the Volume Algorithm [6]. The main advantage with volume algorithm is that it stabilise 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. We present a modified version of the Lagrangian relaxation algorithm presented in [29] below.

The Lagrangian relaxation method based on sub-gradient optimisation exhibits slow convergence especially when it has small step size [33]. Hence the following features were introduced to improve the convergence. In rest of the paper, we refer the algorithm 1 as LR.

5.1 Features of Algorithm 1

Volume Algorithm:

The parameter $\gamma_{\max}$ in the Volume algorithm controls the influence of current violations in Lagrangian algorithm and helps to stabilise the resource constraint violations. When $\gamma_{\max} = 1$ the algorithm converts to 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

Algorithm 1 Lagrangian relaxation method with volume algorithm

1: Initialise $k = 0, \rho = 0.1, LB^* = 0,\ UB^* = \infty,\ gap = \infty, rGap = -\infty,\ \eta = 0,$
$\lambda^0 = (\lambda_\tau^{t(0)}) = 0,\ \lambda^* = \lambda^{(0)}, TL = 3600,$
$S = (S_\tau^t) = 0,$ /* *History of violations* */
$\mathcal{P} = \phi$ /* *Solution pool* */
$C_l = 0,$ /* *Lower bound improvement counter* */
$eTime = 0.$ /* *Elapsed time of the algorithm* */

2: **while** $(eTime < TL) \wedge (gap \geq 0.001)$ **do**

3: Solve individual mine's sub-problem with the updated objective (5.2) and the constraint (5.3). Let $\{\hat{q}_{i\tau}^t\}\ \forall\, i, \tau, t$ be the solution from these sub-problems.

4: Find the violations for each train class τ at time t, $\Phi_\tau^t = R_\tau - \sum_i \left(\hat{q}_{i\tau}^{t+S_\tau} - \hat{q}_{i\tau}^{t-E_\tau}\right)$.

5: Set the lower bound, $LB^{(k)} = \sum_i Z_i(\lambda^{(k)}) - \sum_\tau \left(R_\tau \cdot \lambda_\tau^{(k)}\right)$.

6: Compute $UB^{(k)}$ using the algorithm 3.

7: $UB^* = \min\{UB^*, UB^{(k)}\}$

8: **if** $(LB^* < LB^{(k)})$ **then**

9: $\lambda^* = \lambda^{(k)};\ LB^* = LB^{(k)};\ \rho = \min\{1.3\rho, 2\};\ C_l = 0$

10: **else**

11: $\rho = 0.9\rho;\ C_l = C_l + 1$

12: **if** $(\rho < 0.01) \wedge (TL - eTime > 300) \wedge (rGap \neq gap)$ **then**

13: $\lambda^{(k)} = \lambda^*;\ rGap = gap$

14: **if** $(\rho < 0.005) \wedge (TL - eTime > 300)$ **then**

15: $\rho = 2$

16: **if** $(C_l \geq 10)$ **then**

17: $\eta = 4;\ C_l = 0.$

18: **if** $(\eta > 0)$ **then**

19: $\gamma_{\max} = 0.9;\ \eta = \eta - 1$

20: **else**

21: $\gamma_{\max} = 0.65$

22: **for** (each train class τ) **do**

23: Let γ^* be the solution that minimises $||\gamma\Phi_\tau + (1-\gamma)S_\tau||$.

24: Set $\gamma = \begin{cases} 1 & \text{if } k = 0 \\ \gamma_{\max}/10 & \text{if } \gamma^* < 0 \\ \gamma_{\max} & \text{if } \gamma^* \geq \gamma_{\max} \\ \gamma^* & \text{otherwise} \end{cases}$

25: Update $S_\tau = \gamma\Phi_\tau + (1-\gamma)S_\tau$

26: The Lagrangian multipliers are adjusted as

$$\lambda_\tau^{t(k+1)} = \max\left\{0, \lambda_\tau^{t(k)} - \rho(UB^* - LB^*)\frac{S_\tau^t}{||S_\tau||^2}\right\}$$

27: Use Wedelin based algorithm 2 to update the multipliers.

28: $gap = (UB^* - LB^*)/UB^*$

29: $k = k + 1$

0.65 implies that atmost 65% weightage is given to the violations in the current iteration and remaining weightage 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 original article describing the volume algorithm [6] does not provide clear direction for the selection of this parameter. The value $\gamma_{\max} = 0.65$ is suggested in [29].

Steps 16-25 of algorithm 1 performs various steps from 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 current iteration. Once $\gamma_{\max}$ is increased to 0.9, the same value is used for next 4 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:

Wedelin [35] proposed an algorithm to accelerate the convergence of Lagrangian relaxation methods. There are many interpretations available for this algorithm. 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. [7] proposed generalisation and other improvements to it. This algorithm explores the dual-primal relationship and adds minor perturbation to the multipliers. It favours the variables with smaller reduced cost to come to feasible region. At the same time, other variables are discouraged to balance this perturbation.

We present an algorithm inspired by the Wedelin algorithm to update the Lagrangian multipliers. Originally Lagrangian multiplier are defined for each train class τ at time t. To apply the customisation for each mine, we define $\lambda_{i\tau}^{t}$ as the Lagrangian multiplier used for mine i's sub-problem. The output, $\lambda_{i\tau}^{t}$, of following algorithm 2 is used instead of λ_{τ}^{t} in constraint (5.2).

Resetting Multipliers:

It is possible for the Lagrangian algorithm to get stuck in a particular direction without any improvement in 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, step size may be very small to escape from the influence of local optima. Therefore we reset the step size to its maximum value if there is atleast 300 secs of cpu time remaining for the algorithm 1(see steps 14 and 15).

5.2 Upper bound computation

Among other factors, the convergence of above Lagrangian scheme (1) depends on the quality of the upper bound found in various iterations. The sub-problems from each of the iterations will suggest a feasible production schedule and the expected train combinations for individual mines. It is unlikely that simply amalgamating "as-is" the individual mines solutions will produce a globally feasible schedule. In this section, we therefore propose a MILP based procedure to compute an upper bound using the information on production and train combinations for individual mines. Note that the solution from mines' sub-problem is already feasible with respect to all but the resource constraint (4.11). Therefore, the aim of the MILP presented

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

Input: $\tau, t, \Phi^t_\tau, \lambda^t_\tau, R_\tau, C_l,$
$\{\hat{q}^0_{i\tau}, \ldots, \hat{q}^T_{i\tau}\}\ \forall\, i.$ */* The optimal solution from the sub-problems at that iteration */*
Output: $\lambda^t_{i\tau}$

Compute $\chi^t_{i\tau} = (\hat{q}^{t+S_\tau}_{i\tau} - \hat{q}^{t-E_\tau}_{i\tau}) \quad \forall\, i, t.$
/ $\chi^t_{i\tau}$ tells whether a train of class τ is servicing for mine i at time t */*
Compute the perturbation interval $[a, b]$ based on the lower bound improvement counter, C_l. Where

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

Initialise $\lambda^t_{i\tau} = \lambda^t_\tau$ for all mine i
if $(\Phi^t_\tau \ge 0)\ \vee\ (\Phi^{t-1}_\tau < 0)\ \vee\ (a = b = 0)$ **then**
    return */* No modifications to the Lagrangian multipliers. */*
Compute the perturbation $\epsilon = 0.01(a - (b - a)\ \Phi^t_\tau / R_\tau)$
for (each mine i) **do**
    if $(\chi^{t-1}_{i\tau} > 0)$ **then**
        $\beta = (1 - \epsilon)$ */* Favour mine i */*
    else
        $\beta = (1 + \epsilon)$ */* Discourage mine i */*
    $t' = t$
    while $(\chi^{t'}_{i\tau} > 0)$ **do**
        $\lambda^t_{i\tau} = \beta \lambda^t_\tau$
        $t' = t' + 1$

below is to find an optimal schedule which satisfies the resource constraint (4.11) with values of some of the decision variables from individual mines' sub-problems as fixed. Clearly, this gives an upper bound for the integrated model described in section 4. Specifically, the MILP formulation will consider following values of decision variables from each mine as input,

- *Production Schedule*, $\hat{\nu} = \{\hat{\nu}_i^0, \hat{\nu}_i^1, \ldots, \hat{\nu}_i^T\}$
- *Train Class Combinations*, $\hat{q} = \{\hat{q}_{i\tau_1}^T, \hat{q}_{i\tau_2}^T, \ldots, \hat{q}_{i\tau_N}^T\}$
- *Order Demurrage*, $\hat{y} = \{\hat{y}_i^0, \hat{y}_i^1, \ldots, \hat{y}_i^T\}$

As the number of trips with trains in a class for a mine are fixed, we can *a priori* calculate number of round-trips to this mine 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 mine and T_j is the train 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 train 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 mine i. We refer this job based model as **UBM** in remaining text. MILP formulation of UBM is presented below:

Define,
$X_i^t = \sum_{t'=0}^{t} \hat{\nu}_i^{t'}$, cumulative production at the mine i at time t.
$z_j^t = 1$ if the train for job j arrives at the terminal by time t. 0, otherwise.

Objective

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 mines. The objective of UBM and IM (4.1) are identical. The objective function is

$$[\mathbf{UBM}] \qquad \min \sum_t \sum_i \left(\theta_i^t H_i^t + \omega_i^t F_i^t + y_i^t C_i^t + \sum_{j \in \mathcal{J}^i} (z_j^t - z_j^{t-1}) A_i^t \right) \tag{5.4}$$

Constraints

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

$$\theta_i^t = X_i^t - \sum_{j \in \mathcal{J}^i} z_j^{t+E_j} V_j \quad \forall\, i, t. \tag{5.5}$$

2. Cumulative supply cannot be more than the production.

$$\sum_{j \in \mathcal{J}^i} z_j^{t+E_j} V_j \le X_i^t \qquad \forall\, i, t \tag{5.6}$$

3. An additional tightening integer cut can be derived from the previous constraint (5.6).

$$\sum_{j \in \mathcal{J}^i} z_j^{t+E_j} \left\lceil \frac{V_j}{V_{\min}} \right\rceil \le \left\lceil \frac{X_i^t}{V_{\min}} \right\rceil \qquad \forall\, i, t \tag{5.7}$$

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

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

$$\omega_i^t \geq \sum_{j \in \mathcal{J}^i} z_j^t V_j - D_i^t \qquad \forall\, i, t \tag{5.8}$$

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

$$y_i^t \geq 1 - \sum_{j \in \mathcal{J}^i} z_j^t V_j / D_i^t \qquad \forall\, i, t \tag{5.9}$$

6. Similar to (4.17), additional tightening integer cuts are derived from the constraint (5.9).

$$(1 - y_i^t) \left\lceil \frac{D_i^t}{V_{\min}} \right\rceil \leq \sum_{j \in \mathcal{J}^i} z_j^t \left\lceil \frac{V_j}{V_{\min}} \right\rceil \quad \forall\, i, k, \Delta_{ik} \leq t < \Delta_{ik+1} \tag{5.10}$$

7. Constraints (4.15) and (4.18) related with the demurrage variables are also valid for this formulation.
8. An order must be satisfied before next order's due-date.

$$\sum_{j \in \mathcal{J}^i} z_j^{\Delta_{ik}} V_j \geq D_i^{\Delta_{ik-1}} \qquad \forall\, i, k \tag{5.11}$$

9. Once job is completed it stays as completed.

$$z_j^t \geq z_j^{t-1} \qquad \forall\, j, t \tag{5.12}$$

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

$$z_j^T = 1 \qquad \forall\, j \tag{5.13}$$

11. Not more than one train can load at any mine.

$$\sum_{j \in \mathcal{J}^i} (z_j^{t+E_j} - z_j^{t+\Psi_j}) \leq 1 \qquad \forall\, i, t \tag{5.14}$$

12. Total number of trains of a class τ running at any time t cannot exceed its maximum availability.

$$\sum_{j | T_j = \tau} (z_j^{t+G_j} - z_j^t) \leq R_\tau \quad \forall\, \tau, t \tag{5.15}$$

13. 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 following constraint to break this symmetry.

$$z_{j_2}^t \leq z_{j_1}^{t+\Psi_{j_1} - E_{j_2}} \qquad \forall\, t, j_1 < j_2 \tag{5.16}$$

Note that the above constraint will not effect the optimal solution. Indeed, if in the optimal solution for some t, $z_{j_2}^t > z_{j_1}^{t+\Psi_{j_1} - E_{j_2}}$, then given the two jobs have the same train class and mine we can easily relabel j_1 as j_2 and vice versa, without effecting the solution.

14. 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 $\mathcal{J}^i \setminus \mathcal{J}'$. This gives following constraint

$$\sum_{j \in \mathcal{J}^i \setminus \mathcal{J}'} z_j^t + y_i^t \geq 1 \quad \forall\, i, t \tag{5.17}$$

Due to symmetry breaking constraint (5.16), constraint (5.17) 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 same train class in $\mathcal{J}^i \setminus \mathcal{J}'$, $\mathcal{J}''$ contains the job with smallest index. More formally, $\mathcal{J}''$ must satisfy following two conditions

- For any two jobs $j_1, j_2 \in \mathcal{J}''$, $T_{j_1} \neq T_{j_2}$ whenever $j_1 \neq j_2$
- 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 can be tightened as

$$\sum_{j \in \mathcal{J}''} z_j^t + y_i^t \geq 1 \quad \forall\, i, t \tag{5.18}$$

For example, consider $\mathcal{J}^i = \{1, 2, 3, 4, 5\}$ where $T_1 = T_2 = \tau_1$, $T_3 = T_4 = T_5 = \tau_2$, $V_{\tau_1} = 3000, V_{\tau_2} = 5000$ and $\mathcal{J}' = \{3\}$ defined for a demand of 6000 at t. Then the constraint (5.17) gives

$$z_1 + z_2 + z_4 + z_5 + y_i^t \geq 1. \tag{5.19}$$

Symmetry breaking constraint (5.16) enforces 1 to complete before 2 and 4 to complete before 5 and $\mathcal{J}'' = \{1, 4\}$. Hence, as per constraint (5.18), the constraint (5.19) can be tightened as

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

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

$$y_i^t = 1 \qquad \forall\, i, t \ \text{ if } \ \hat{y}_i^t = 1 \tag{5.20}$$

16. Boundary conditions and scope,

$$\theta_i^0 = 0, \omega_i^0 = 0, y_i^T = 0, z_j^0 = 0, z_j^t, y_i^t \in \{0, 1\}, \theta_i^t, \omega_i^t \geq 0 \tag{5.21}$$

The algorithm 3 computes the upper bound for the integrated model in algorithm 1. As the production schedule for every mine in UBM is fixed, it is possible that in the optimal solution of UBM, mines' production can be delayed to further reduce any inventory holding cost at the mine. 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, $\mathcal{P}$, 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.

Algorithm 3 Upper bound computation algorithm

Input: $k, \mathcal{P},\ (\hat{\nu}, \hat{q}, \hat{y})$ */* Solution set */*
Output: $UB^{(k)}$
if $(\hat{\nu}, \hat{q}, \hat{y}) \in \mathcal{P}$ **then**
    return */* This solution was previously evaluated */*
else
    $\mathcal{P} = \{(\hat{\nu}, \hat{q}, \hat{y})\} \cup \mathcal{P}$
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 5.2 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.

6 Computational Experiments

The distributed optimisation model (LR) was compared with the integrated model (IM) via computational experiments on 240 randomly generated instances. These experiments were done with CPLEX 12.1 on a 64-bit server machine[4]. In IM, CPLEX was terminated with either at 0.1% of relative gap or with a cpu time limit of one hour, whichever came first. For a fair comparison, same termination criteria was also used for the LR method. In LR, UBM was terminated after 120 secs or with zero percentage relative gap.

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. These series have following common properties

- Production capacity (P_i) of all mines is taken as 400 tonnes per period, where one period is equivalent to an hour.
- Inventory capacity (B_i) at the mine is 20000 tonnes.
- Average order per mine is 2.5
- Average trips per train is 5
- Objective coefficients: $H_i^t = 1, F_i^t = 3, C_i^t = 50000, A_{i\tau}^t = 100$ for all mine i and train class τ
- For series with 5 and 6 mines, train classes used are 3000, 5400 and 7200 tonnes. For every other series train class of 8400 tonnes is also used.
- Loading time (L_τ) for a train class τ is

τ	3000	5400	7200	8400
L_τ	1	2	3	4

All 30 instances in a particular series have following properties:

- Any two instances have a constant number of mines (I), trains and train classes (N). Moreover, the forward (S_τ) and return travel times (Ψ_τ) for trains in a class τ are also constant.

[4] Configuration: 64-bit, 16 core, 2.93 GHz Intel Xeon(R) X7350 processor OS: SUSE Linux Enterprise Server 10

- In each instance, the number of orders in a mine (K_i), the order quantity (Q_{ik}) and order due-dates (Δ_{ik}) are generated randomly. Specifically,
  $K_i \sim U(1,4); \quad Q_{ik} \sim 5000 + 100U(0,100); \quad \Delta_{ik} \sim \Delta_{ik-1} + 10 + U(0,T/K_i).$
- The average demand for each mine is 25000 tonnes.

The following table 6.1 summarises the properties of each data series. Since over all the instances the average order per mine and the average trips per train are more or less the same, results in this section will show that increase in the number of mines alone makes this problem challenging. Indeed, once the number of mines increase the system will have more overlapping due-dates and will be forced to do more number of trips in the same planning horizon. This makes increasingly difficult for IM to optimally allocate train resources. The complexity explosion is also reflected in the results as in more than 40% of instances, IM failed to find even a feasible solution.

Table 6.1: Data series summary and properties

I	5	6	7	8	9	10	12	15
N	3	3	4	4	4	4	4	4
# Trains	4	4	7	6	7	8	8	10
§ Trips	20	23	26	33	44	37	30	55
§ Orders	10	12	14	16	18	20	24	30
T	150	150	150	200	200	200	200	200
R_τ	[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]
S_τ, Ψ_τ	[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]

§= Average number of, # = number of

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, relative performance of the presented solution approaches is compared via Student's t-test. For the lower bound comparison, relative performance ratio is defined as $(LB_{LR} - LB_{IM})/\max(LB_{LR}, LB_{IM})$. A positive ratio means that LR is better as it has a higher lower bound. Similarly for the upper bound, the ratio is defined as $(UB_{LR} - UB_{IM})/\min(UB_{LR}, UB_{IM})$. Here, a positive ratio means that IM is better. Estimated mean and 95% confidence interval from the Student's t-test for the ratios are presented in figures 6.1 and 6.2. Interval boundaries and the mean are marked with thick red cross lines.

Figures 6.1 and 6.2 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 larger number of mines. For example, in instances with 10, 12 and 15 mines quality of lower bounds found by LR was at least 20% better than that of IM.
2. the upper bound obtained by IM is only slightly better than the upper bound obtained from LR for the series with 5 mines. In this case, the difference is only marginal as 95% confidence interval for the ratio is $[0.0007, 0.0293]$.
3. for instances with 6 or more mines, LR always found better upper bounds than IM. This difference was much more significant as the number of mines increased.

In every series, lower bound ratio is computed from 30 observations. But less than 11 obser-

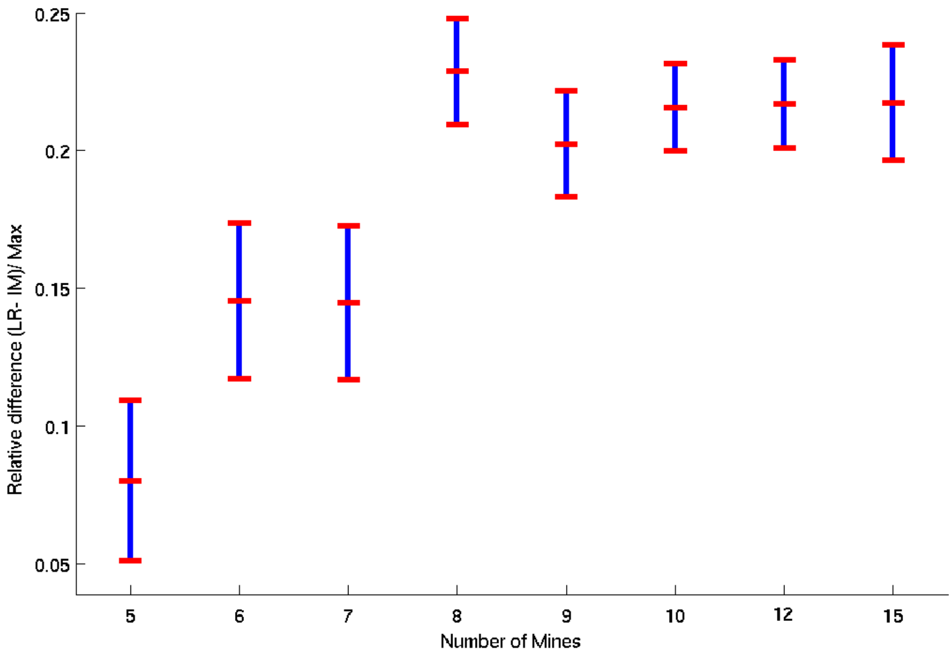


Figure 6.1: 95% confidence interval for the relative difference in lower bound

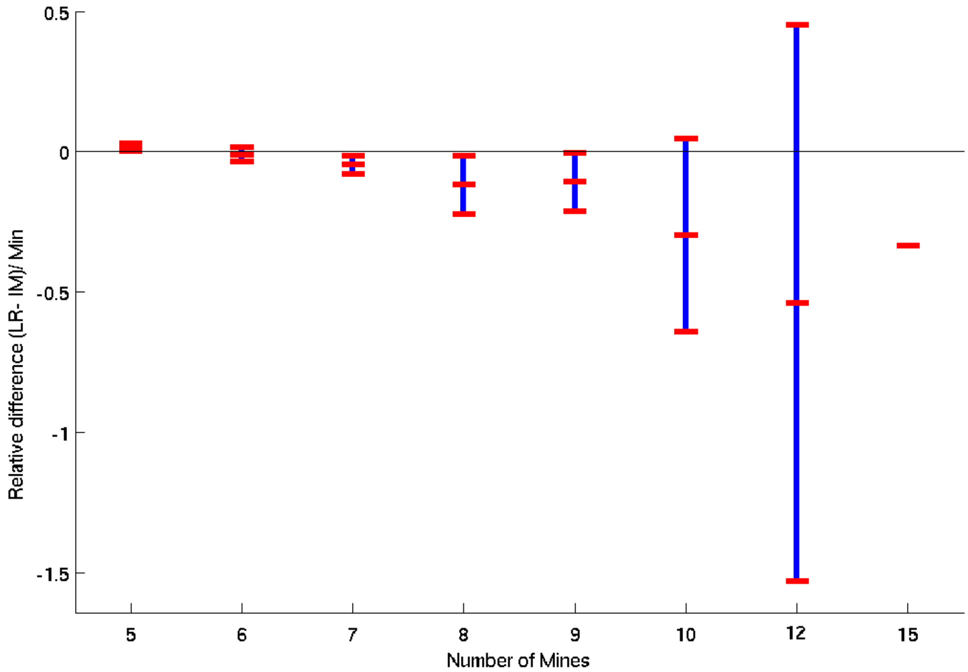


Figure 6.2: 95% confidence interval for the relative difference in upper bound

vations were available for the upper bound ratio in the series with 9, 10, 12 and 15 mines. Therefore, based on IM's performance and to undertake a more detailed analysis of the results, we further classify the data sets into four groups.

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.

The following table 6.2 summarise distribution of the instances and their properties in each group. The table clearly indicates that as the number of mines increase it becomes increasingly difficult for IM to find any feasible solution or to converge to a solution within reasonable gap.

Table 6.2: Data group summary and properties

# Mines (I)	5	6	7	8	9	10	12	15			
Group	Number of data sets								Total	§Orders	§Trips
Easy	14	5	5	0	0	0	0	0	24	10	18
Medium	7	8	6	5	2	1	1	0	30	14	26
Hard	9	17	17	13	8	7	3	1	75	16	30
Very Hard	0	0	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 figures 6.3 and 6.4.

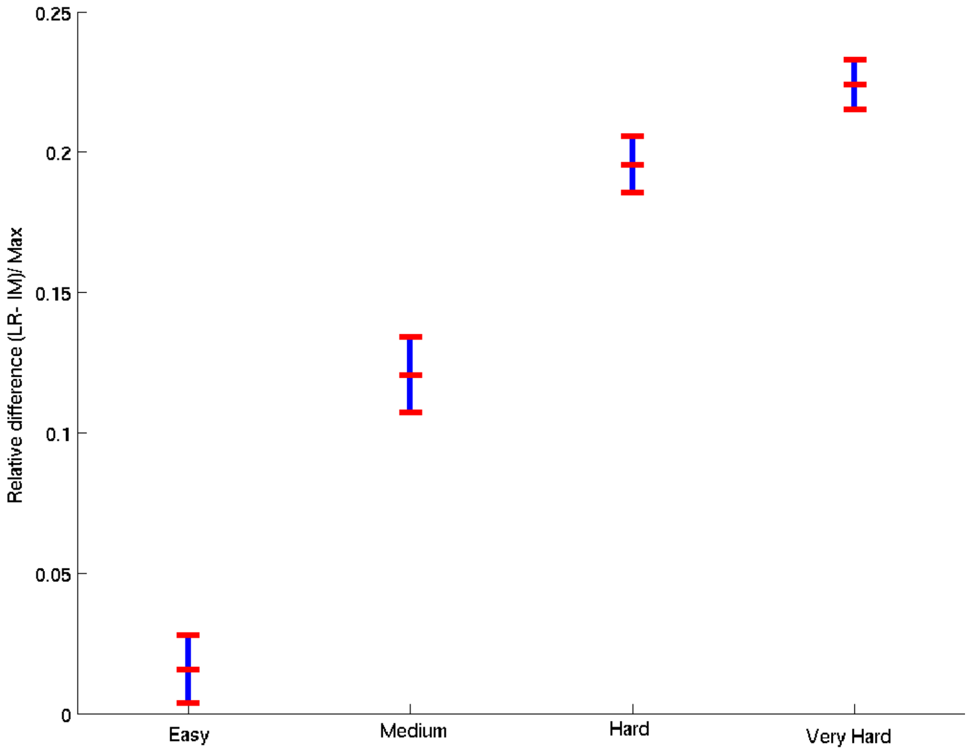


Figure 6.3: 95% confidence interval for the relative difference in lower bound for different groups

Once again, the gap between IM and LR is visibly increasing and demonstrates the advantage of distributed optimisation scheme proposed in this paper. The results show that

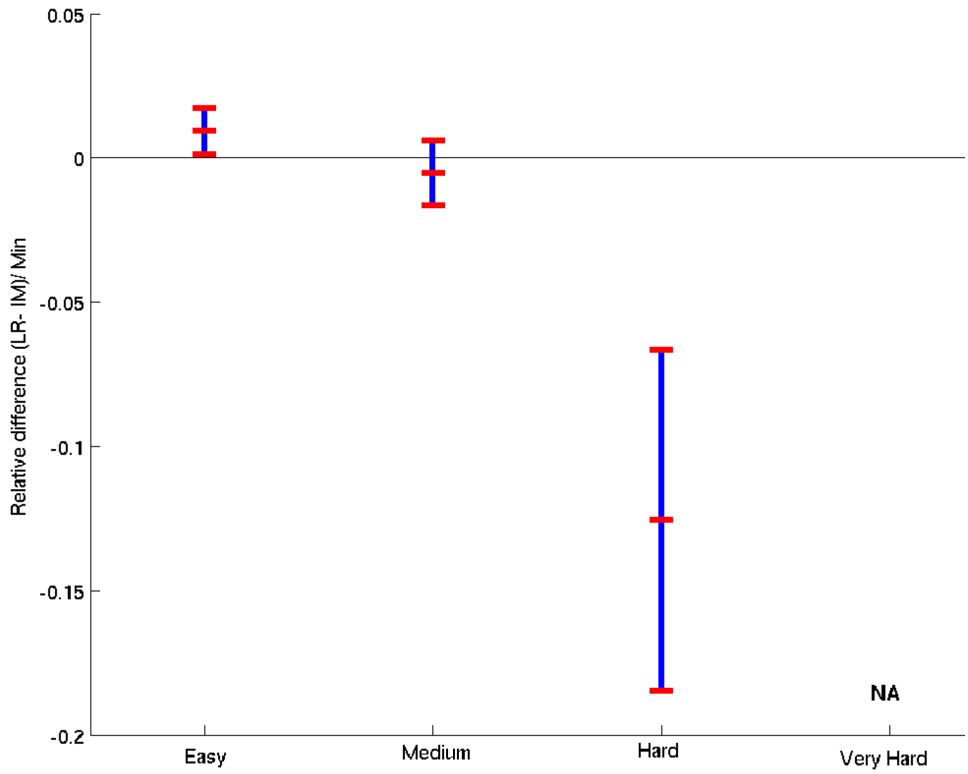


Figure 6.4: 95% confidence interval for the relative difference in upper bound for different groups

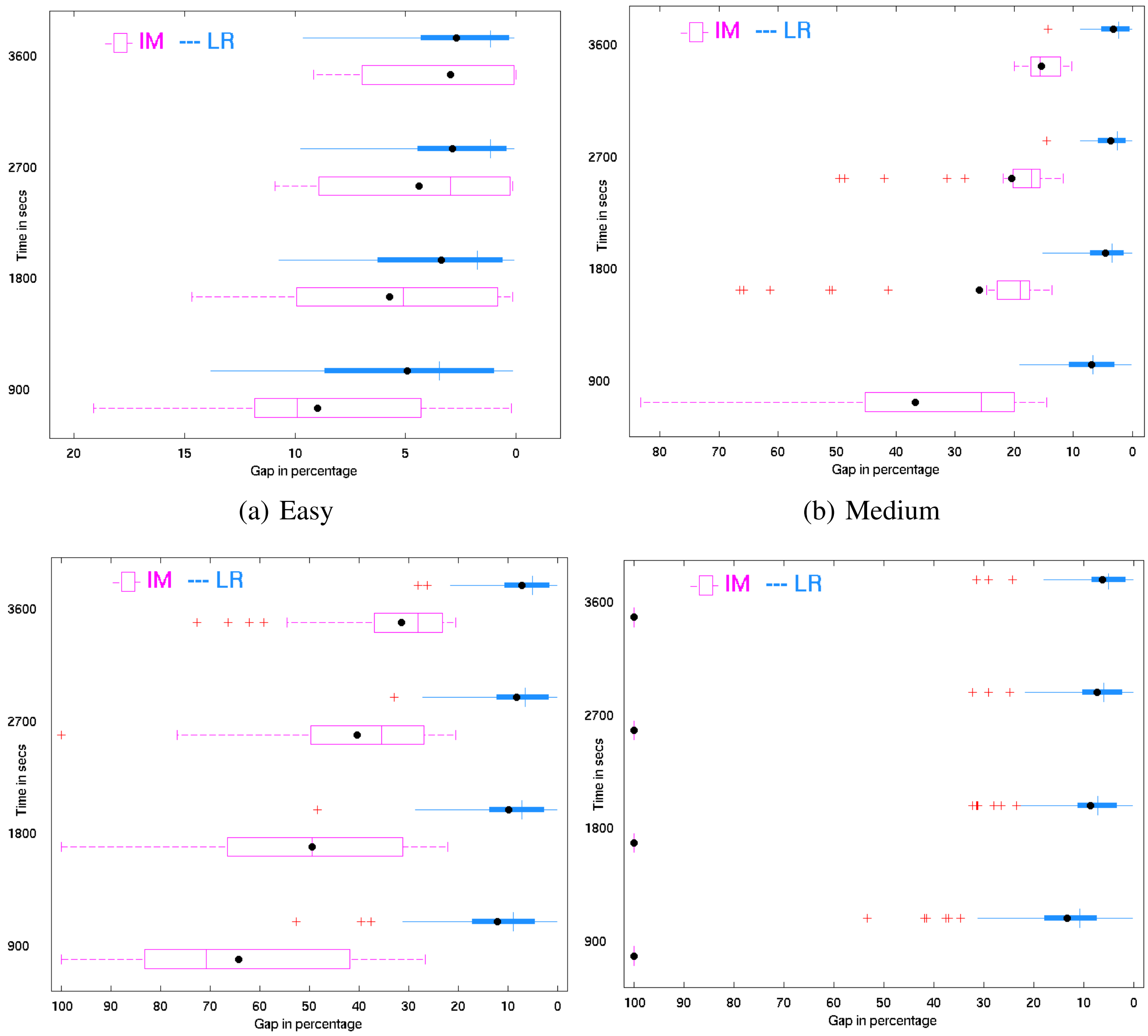
1. The average improvement of LR over IM in lower bound is 2, 12, 19 and 22% for the group easy, medium, hard and very hard respectively.
2. Only in 'easy' instances IM could find upper bounds better than the LR method, but the difference was only marginal.
3. The upper bound found by LR is on average better by 12.5% in the 'hard' group, which includes almost 1/3 of instances.
4. In 'very hard' group, IM could not find any feasible solution. Therefore upper bound ratio is not available. However LR scheme could find feasible solutions with a median gap less than 10% in 1800 seconds (figure 6.6(d)).

Just comparing the final results at the end of the termination criteria do not completely convey the performance of both methods. Therefore, we also compare the *gap* obtained from LR and IM at 4 different time periods. The gap is defined as $(UB - LB)/UB$. Figure 6.5 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 red colour 'plus' (+) symbol.

Table 6.3 presents the median gap computed for each series. In figure 6.6(d) and table 6.3, the gap of IM is taken as 1 (or 100%) in absence of any feasible solution.

Following observations can be made from figure 6.5 and table 6.3.

1. Gap of LR is consistently better in all groups and series at all time periods. LR has less



(a) Easy (b) Medium

(c) Hard (d) Very Hard

Figure 6.5: Gap at different time points for different groups

Table 6.3: Median of gap in percentage

	Integrated Model (IM)				Distributed Model (LR)			
	time in seconds				time in seconds			
# Mines	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

than 17% gap in 229 out of 240 instances.

2. For the series with more than 9 mines, IM is intractable as the median gap is 100%.
3. For the series with 9, 10, 12 and 15 mines, LR's median gap is 3.44%, 1.83%, 7.98% and 3.03% respectively. It shows the strength of LR in instances with large number of mines.
4. LR gets feasible solutions very quickly. In 900 seconds, LR gap's median is close to 5% gap in easy and medium groups and 10% in hard and very hard groups.
5. In table 6.3, as the number of mines 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 distributed algorithm over the integrated models.
6. During the time period of 1800 to 3600 seconds, the reduction in gap of LR is very minimal (on average 1% per 900 seconds). As part of ongoing research, stronger constraints which will act as cuts will be explored to improve the convergence of LR.

As a summary, 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. But 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%. Briefly, the above computational experiment confirms the effectiveness of the LR scheme to give solutions (with gap $< 10\%$) very quickly even if the number of mines increases.

7 Conclusion and Future Work

An integrated planning-scheduling problem motivated from the coal supply chain in Australia is considered in the paper. For this problem we propose a distributed optimisation algorithm that incorporates the Volume and Wedelin Algorithms. The strength of the algorithm is demonstrated by comparing its performance with an MILP, via extensive computational experiments. The results show that the distributed algorithm found significantly better lower and upper bounds than MILP.

At present two major decisions, production planing and train scheduling, are considered in this model. Ongoing research extends this study by including the decisions related with terminal operations and track maintenance. The convergence of the distributed scheme will be further strengthened by improvements to Lagrangian relaxation, Volume algorithm and Wedelin algorithm. The authors also plan to develop new distributed algorithms for the case of partial information sharing and multi-objective.

The integrated model and the distributed scheme assume complete information sharing between the partners, which requires additional resources and investment. The mines owned by different enterprises, are competing for the same resource. Therefore they will not be willing to share their competitive information. At the same time, mines will have minimal control and information on train operations. The rail operator will not announce all train schedules to all the mines. Only the train class, not individual trains, information is available to the mines. In this partial information sharing environment, more decentralised and distributed decision making models will be more relevant and advisable. This direction is explored in our ongoing research.

References

[1] ABARES. *Australian mineral statistics 2011 - March Quarter*, pages 18–19. Australian Bureau of Agricultural and Resource Economics and Sciences, 2011.

[2] Arshinder, Arun Kanda, and S.G. Deshmukh. Supply chain coordination: Perspectives, empirical studies and research directions. *Int. J. of Production Economics*, 115(2):316–335, October 2008.

[3] Alper Atamturk. Cover and pack inequalities for (mixed) integer programming. *Annals of Operations Research*, 139:21–38, 2005.

[4] Imam Baihaqi, Nicholas Beaumont, and Amrik Sohal. Information sharing in supply chains: A survey of australian manufacturing. *Int. Review of Business Research Papers*, 4(2):1–12, 2008.

[5] Kenneth R. Baker and Gary D. Scudder. Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38(1):22–36, 1990.

[6] F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

[7] Oliver Bastert, Benjamin Hummel, and Sven de Vries. A generalized wedelin heuristic for integer programming. *INFORMS J. on Computing*, 22:93–107, January 2010.

[8] Julian Benjamin. An analysis of inventory and transportation costs in a constrained network. *Transportation Science*, 23(3):177–183, 1989.

[9] Peter Brucker, Andreas Drexl, Rolf Mhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European J. of Operational Research*, 112(1):3 – 41, 1999.

[10] Peter Brucker and Sigrid Knust. *Complex Scheduling*. Springer-Verlag Berlin and Heidelberg GmbH and Co. K, February 2006.

[11] P. Chandra and M.L. Fisher. Coordination of production and distribution planning. *European J. of Operational Research*, 72:503–517, 1994.

[12] Zhi-Long Chen and George L. Vairaktarakis. Integrated scheduling of production and distribution operations. *Management Science*, 51(4), 2005.

[13] M. Chiu and G. Lin. Collaborative supply chain planning using the artificial neural network approach. *J. of Manufacturing Technology Management*, 15(8):787–796, 2004.

[14] Marshall L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 50(12):pp. 1861–1871, 2004.

[15] I.E. Grossmann. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE Journal*, 51:1846–1857, 2005.

[16] I.E. Grossmann and L.T. Biegler. Part ii. future perspective on optimization. *Computers and Chemical Engineering*, 28(8):1193–1218, 2004.

[17] T. J. R. Johnson. *An algorithm for the resource-constrained project scheduling problem*. PhD thesis, MIT, 1967.

[18] J.S.K. Lau, G.Q. Huang, K.L. Mak, and L. Liang. Agent-based modeling of supply chains for distributed scheduling. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(5):847 –861, sept. 2006.

[19] H.L. Lee and C. Billington. Managing supply chain inventory: pitfalls and opportunities. *Sloan Management Review*, 33(3):65–73, 1992.

[20] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.

[21] Joseph Y. T. Leung. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman and Hall/CRC, April 2004.

[22] Chiung Moon, Young Hae Lee, Chan Seok Jeong, and YoungSu Yun. Integrated process planning and scheduling in a supply chain. *Computers and Industrial Engineering*, 54(4):1048 – 1061, 2008.

[23] J. Mula, R. Poler, J.P. Garcia-Sabater, and F.C. Lario. Models for production planning under uncertainty: A review. *Int. J. of Production Economics*, 103(1):271 – 285, 2006.

[24] Josefa Mula, David Peidro, Manuel Diaz-Madronero, and Eduardo Vicens. Mathematical programming models for supply chain production and transport planning. *European J. of Operational Research*, 204(3):377 – 390, 2010.

[25] Nikolaos Papadakos. Integrated airline scheduling. *Computers and Operations Research*, 36(1):176 – 195, 2009.

[26] Jun-Hyung Ryu, Vivek Dua, and Efstratios N. Pistikopoulos. A bilevel programming framework for enterprise-wide process networks under uncertainty. *Computers and Chemical Engineering*, 28(6-7):1121 – 1129, 2004.

[27] Ana Maria Sarmiento and Rakesh Nagi. A review of integrated analysis of production-distribution systems. *IIE Transactions*, 31:1061–1074, 1999.

[28] Nikisha Shah, Georgios K.D. Saharidis, Zhenya Jia, and Marianthi G. Ierapetritou. Centralized-decentralized optimization for refinery scheduling. *Computers and Chemical Engineering*, 33(12):2091 – 2105, 2009.

[29] Gaurav Singh and Andreas T. Ernst. Resource constraint scheduling with a fractional shared resource. *Operations Research Letters*, 39(5):363 – 368, 2011.

[30] Gaurav Singh and Rene Weiskircher. A multi-agent system for decentralised fractional shared resource constraint scheduling. *Web Intelli. and Agent Sys.*, 9:99–108, April 2011.

[31] Anu Thomas, Mohan Krishnamoorthy, Gaurav Singh, and Jayendran Venkateswaran. Collaborative services provision in production-logistics-scheduling: A coal industry example. In *Proceed. of 2nd Int. Conference on Services in Emerging Markets ICSEM 2011*. Service Research and Innovation Institute (SRII), 29th Sept – 01 Oct 2011.

[32] Douglas J. Thomas and Paul M. Griffin. Coordinated supply chain management. *European J. of Operational Research*, 94(1):1 – 15, 1996.

[33] 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.

[34] Carlos J. Vidal and Marc Goetschalckx. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European J. of Operational Research*, 98(1):1 – 18, 1997.

[35] Dag Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57:283–301, 1995.

[36] Laurence A. Wolsey and George L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley-Interscience, November 1999.