OPTIMIZATION OF READY-MIXED CONCRETE TRUCK SCHEDULING USING METAHEURISTIC APPROACHES

Biman Darshana Hettiarachchi

(168031N)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

January 2019

OPTIMIZATION OF READY-MIXED CONCRETE TRUCK SCHEDULING USING METAHEURISTIC APPROACHES

Biman Darshana Hettiarachchi

(168031N)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

January 2019

DECLARATION, COPYRIGHT STATEMENT AND THE STATEMENT OF THE SUPERVISOR

"I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books)."

Signature: Date:

The above candidate has carried out research for the Masters thesis under our supervision.

Name of the supervisor: Dr. H.M.N. Dilum Bandara

Signature of the supervisor: Date:

Name of the supervisor: Eng. Nishal A. Samarasekera

Signature of the supervisor: Date:

ABSTRACT

Ready-Mixed Concrete (RMC) is a perishable product; hence, specifications such as ASTM C94 recommend the delivery of RMC under 1.5-hours to ensure the quality. It is known that certain scheduling practices and driving behaviors lead to operational inefficiencies and poor-quality RMC. We propose a model to schedule RMC trucks while maximizing both the profit and job coverage, as well as meeting constraints such as ASTM C94 and continuous casting. The proposed solution consists of a rule checker and a scheduler. Rule checker enforces constraints such as deadlines, working hours, and ASTM C94 specification for travel time. The scheduler uses simulated annealing to assign as many jobs as possible while maximizing the overall profit. We consider two scenarios where trucks are attached to a given RMC plant, as well as allowed to move across plants as per job requirements. Using a workload derived from an actual RMC delivery company, we demonstrate that the proposed solution has good coverage of jobs while maximizing the overall profit. For example, compared to the manual job allocation, proposed solution in the fixed-plant scenario increases the average job coverage and profit by 13% and 9%, respectively. Moreover, the solution could automatically adjust the first unload time by a few 10s of minutes to reduce job conflicts, and this further enhances average job coverage and profit to 21% and 13%, respectively. Further, free-to-move scenario enhances the average job coverage and profit by 16% and 14%, respectively indicating that the scheduling could be further optimized by allowing trucks to move across the plants as per the job requirements.

Keywords: Fleet Management; Ready-Mixed Concrete; Scheduling; Simulated Annealing

ACKNOWLEDGEMENT

Foremost, I would like to pay my sincere gratitude to my research supervisor Dr. H.M.N. Dilum Bandara for his knowledge, motivation and commitment guiding me throughout the Research Degree program to research and write this thesis. Moreover, my heartfelt gratitude is extended to my co-supervisor Eng. Nishal A. Samarasekera for his immense assistance and guidance in making this study an authentic product.

Further, I would like to thank Nimbus Venture (Pvt) Ltd for providing dataset for the research and I would also like to gratefully acknowledge the Senate Research Grant No. SRC/LT/2016/14 of the University of Moratuwa, Sri Lanka for funding this research.

Finally, I would like to extend my gratitude to my parents and my wife for providing me with unfailing support and continuous encouragement throughout these years.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ANN	Artificial Neural Network
ANS	Artificial Neural Systems
API	Application Programming Interface
DPSO	Discrete Particle Swarm Optimization
GA	Genetic Algorithm
GPS	Global Positioning System
HC	Hill Climbing
IBK	Instance Based Learner
ILP	Integer Linear Programming
J48	Decision Tree (Implementation of algorithm ID3)
ML	Machine Learning
NB	Naïve Bayes
PART	Rule based algorithm
RMC	Ready Mix Concrete
SA	Simulated Annealing
SMO	Sequential Minimal Optimization

1. INTRODUCTION

Fleet scheduling is a core strategic decision area of the freight industry to optimize the resource allocation of a fleet. Scheduling Ready-Mixed Concrete (RMC) delivery is a complex problem as RMC is a perishable product. For example, according to the ASTM C94 specification [1], quality of the concrete degrades with time; hence, needs to be delivered within 1.5-hours from the time water is added to the concrete mixture or 300 drum revolutions. Moreover, time bounds defined by the ASTM C94 specification for RMC also depend on the properties added into the concrete mix. Whereas job site managers usually want RMC trucks to wait in a queue at the construction site to avoid discontinuous casting which ultimately creates batch behavior of RMC trucks. Environmental factors such as time of the day, the impact of peak and off-peak traffic, maximum buffer time allowed by the site, and wash-down time of RMC trucks further affect the ability to meet the deadlines such as ASTM C94 [2]. Therefore, better scheduling patterns and driving behavior of fleets of RMC trucks are vital for enhancing the operational efficiency, reducing costs, and preventing fraudulent activities.

The RMC supply process consists of five main steps, namely production, loading, delivery, unloading, and vehicle return [3]. An RMC batching plant begins the production once a customer places an order, until then trucks are parked. Loading and delivery begin immediately after the production of concrete, as the production and dispatching of RMC are interrelated due to the perishable nature of the product. Time to unload at the job site is critical, as concrete should be thrown away if excessive delays are experienced between the production to unload. Moreover, to maintain the supply chain at an optimum level truck return also needs to be properly timed. These steps urge the fleet manager to schedule the trucks in an efficient manner, as an optimized schedule could reduce wastages while decreasing the overall operational costs [2].

With the introduction of Global Positioning System (GPS) based vehicle tracking devices and fuel sensors, RMC production has become more automated. Fleet managers could capture real-time data such as vehicle location, speed, travel time, and fuel consumption using the sensors [4]. Such data could be used to analyze the impact of various scheduling and driver-behavior practices, which could be considered in future scheduling decisions. However, it is difficult to analyze vast volumes of GPS data generated by multiple RMC trucks over a given period. Moreover, the problem gets harder as the locations of the RMC trucks vary depending on the job location, where some jobs require continuous casting while others tend to be one-off. Alternatively, the job-site manager wants to avoid discontinuous casting by requiring several RMC trucks to wait at the construction site. Consequently, the scheduling manager usually dispatches RMC trucks based on his/her experience while maintaining conservative time gaps between production to unload, which is known to be inefficient and lead to loss of potential profit [5]. Therefore, it is imperative to be able to optimally schedule RMC trucks and plant operation while satisfying the conflicting goals of job coverage, continuous casting, time bounds like ASTM C94, and profitability.

1.1 Motivation

Currently RMC truck scheduling is mostly manipulated manually by an experienced batching plant manager, who creates the next day's schedule at the end of the previous working day based on the orders received. He also needs to keep real-time track of the progress of jobs and make necessary adjustments due to dynamism as the day progresses as the continuous casting of the concrete is critical for any construction project to avoid cold joints in the concrete casting. However, with the increment of the number of jobs, scheduling becomes difficult task for the manager as he need to decide on the most appropriate truck and plant for a job such that both the customer and company goals are optimally satisfied. Therefore, RMC industry is in need for scalable and automated scheduling solutions that increases customer satisfaction, efficiency, company profits, and maximum utilization of resources of the company. Since the route and vehicle scheduling problems are known to be NP-hard, we cannot get an optimal solution within polynomial time [6], [7], [8], [9]. Therefore, it is essential to identify suitable heuristic-based solutions that can still maximize the customer satisfaction, efficiency, and company profit.

1.2 Problem Statement

We enforce the constraints of a RMC delivery company with a set of plants and trucks that are dispersed around a given geography. We assume the assignment of jobs is done by an experienced RMC truck scheduling manager at the end of the previous business day. Handling the last-minute jobs is left as future work. Therefore, the problem to be addressed can be formulated as:

Given a set of trucks T and jobs J, how to automatically schedule trucks and plants to jobs while maximizing customer satisfaction, efficiency, profit, and job coverage?

The related optimization problem is formally defined in Chapter 3.

1.3 Objectives

Following set of objectives are to be achieved to address the above problem statement:

- To identify parameters related to the trucks, plants, and construction sites and then formulate the truck scheduling problem as a constrained optimization problem with multiple constrains.
- To solve the constrained optimization problem using a suitable metaheuristic technique.
- To evaluate the performance of the proposed solution using a dataset from a real ready-mixed concrete company.

1.4 Outline

The rest of the thesis is organized as follows. Literature review is presented in Chapter 2 including an overview of truck scheduling patterns, RMC truck dispatching, machine-learning technique usage in scheduling and usage of genetic algorithm in RMC Scheduling. Problem formulation including truck, plant, and job constraints, characteristics of problem and optimization of problem are presented in Chapter 3.

Solution approach with the proposed rule checker and job scheduler are presented in Chapter 4. Performance analysis along with workload creation and simulation results is presented in Chapter 5. Summary, future work, and research limitations are presented in Chapter 6.

2. LITERATURE REVIEW

Concrete is one of the principal materials used in the industry and the development of the construction industry has greatly influenced the concrete industry in developing countries where the demand for concrete has grown at an increasing rate in recent years. As a result, concrete markets are facing both the opportunities of great profit and the risks of competition. Greater attention has been given to achieving higher efficiency for more benefits to suppliers of Ready-Mixed Concrete (RMC) [3]. Section 2.1 presents the background of RMC delivery process. Truck scheduling patterns including the usage of different algorithms is presented in Section 2.2.

2.1 Ready-Mixed Concrete Delivery

After the preparation of concrete, the RMC truck cycle begins as shown in Figure 2.1 by driving to the loading bay of the batching plant to load RMC. If there is no queue at the loading bay, truck can immediately load and leave the plant to head to the construction site. Else, it needs to wait till its turn. RMC truck may have to again be in a queue at the construction site till the previous trucks complete the unloading. Once the unloading is completed, truck goes through a quick wash to remove the residuals in the concrete holding drum. The truck will leave the construction site and reach the plant. If another job is already assigned it will drive to the loading bay. Else, truck will park till it gets another job.

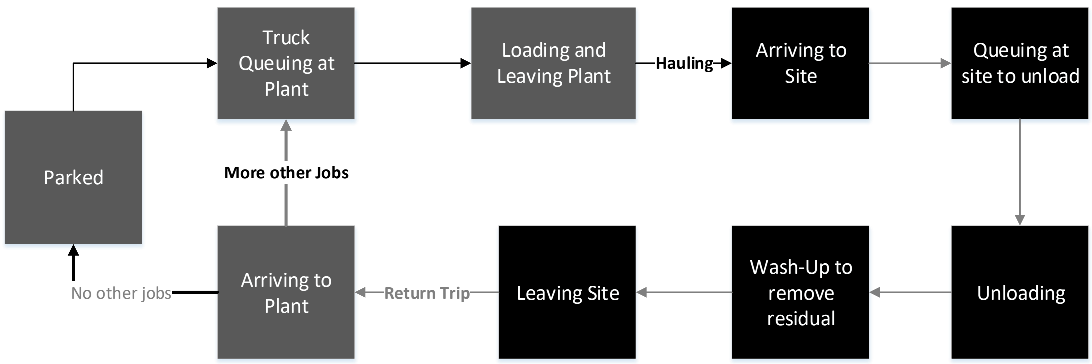

Figure 2.1. RMC truck cycle.

The time since mixing till placing should be no longer than concrete setting time in conditions as per ASTM C94 specifications for RMC. Therefore, route optimization and truck scheduling practices are needed to deliver the mix within this usability window. Unexpected delays and breakdowns may lead to wasting the whole batch and having too many trucks queuing for the mix is also not economical [10]. In the ideal scenario, arrival time of the next RMC truck need to be overlapped with the unload time of the previous truck which just finishes casting concrete [11].

Factors affecting RMC truck scheduling have been discussed in several related works [5], [10], [11], [12], [13], [14]. According to Bergthaler [12], the planning and scheduling of RMC delivery remain very difficult. As RMC is a perishable product, the time is considered the main constraint for RMC delivery. As per the ASTM C94 specification, this time window could be adjusted based on the special properties added to the concrete. A RMC truck scheduling solution that considers the travel, casting, mixing, and allowed buffer time, as well as the required number of RMC deliveries to make dispatching sequence decisions for each RMC truck from a single plant is presented in [5]. Biruk [10] analyzed the dispatching problem by considering the available number of trucks, batching plant operation time, the time between mixing and placing, and maximum breaking in the concreting process. According to [11], factors that affect truck routing and scheduling are starting time, traveling duration, casting duration, concrete mixing duration, number of trucks, the capacity of a truck, required volume of RMC, and buffer time. Moreover, Hill and Böse [13] emphasized that the arrival rate to the site and waiting time should be considered in scheduling.

2.2 Truck Scheduling Patterns

Bishop et al. [14] discussed three truck-scheduling patterns. They are hub-to-hub, trucks find each other on the road, and truck stop kiosks. In hub-to-hub scheduling, once the freight sorting is complete, trucks with similar dispatching activities can be paired at the terminal. In the second scheduling pattern, trucks can be driving on the road and automatically discover other linkable trucks. In the third pattern, private truck stops enable ad-hoc linking, where trucks that share similar routes are concentrated in these facilities. These three scheduling patterns have a certain level of platooning, e.g., in [14] two truck platooning strategies, namely catch-up and slow-down strategy are identified. In catch-up strategy, a follower truck increases its speed to catch up with its leading truck. In slow-down strategy, leading truck decreases its speed to allow the following truck to catch up. Furthermore, the speed variation of trucks in platoons, vary with time and distance between trucks. However, from a fuel efficiency point of view, neither platooning strategy is recommended, as fuel consumption unnecessarily increase during both slow down and speed up.

Fuel consumption rate, vehicle speed, and engine RPM were analyzed by Verwer et al. [15] to identify the driver behavior of a vehicle fleet. Such understanding of driver behavior can be used to regulate driving patterns to increase efficiency, reduce excessive idling, meet time constraints, and prevent fraudulent activities of RMC delivery. Moreover, Xiao and Konak [16] found that the average speed between origin and destination are time-dependent. Hence, time of day should also be a factor in determining a suitable schedule and while accepting a new job.

2.2.1 RMC Truck Dispatching

Currently, ready-mixed concrete truck dispatching is mainly handled manually by an experienced RMC batching plants staff [10]. Inefficient production scheduling and truck dispatching are crucial issues for a RMC plant and construction site management, as the batching plant manager needs to address both timeliness and flexibility while satisfying construction site operating constraints and environmental constraints such as traffic and breakdowns [17]. In practice, trucks must be cautiously dispatched to avoid cold joints in the casting of concrete. Subsequently, RMC production scheduling and truck dispatching affects the operating cost owing to the many complex factors and constraints [17]. Systematic optimization approaches to solve such an integrated problem have rarely been developed [17].

2.2.2 Machine-learning Based Techniques

M. Maghrebi et al. [18] illustrated that ready-mix concrete truck dispatching can be automated through Machine Learning (ML) techniques. Most construction related operational tasks are performed by humans due to the complexity of the tasks. Additionally, it is very difficult to accurately predict performance and unavoidable errors. Batching plant managers or senior engineers play a key role in ready-mixed concrete industry and their positions are replaced by an automated process in rare cases due to the complexity of the highly dependent on specific project constraints, environmental conditions, and must adapt quickly based on incomplete, as well as rapidly changing information. Feasibility of automation in RMC dispatching was studied using six ML techniques, namely decision tree J48 (implementation of algorithm ID3), PART, Artificial Neural Network (ANN), Sequential Minimal Optimization (SMO), Naive Bayes Classifier (NB), and Instance-Based Learner (IBK). Those techniques were selected and tested on data that was extracted from a developed simulation model. The results were compared by a human expert to ensure the accuracy of solutions. The simulation model consists of a single batch plant and three projects in a day.

A metropolitan area consisting seven suburbs including one batching plant were selected to simulate the model proposed. The 200 instances are prioritized by the dispatching manager in two stages with each time involving 100 instances. All six algorithms were evaluated using the same data set and the ten-folds cross-validation was selected for the evaluation. Therefore, the data set was divided into ten-folds with around 90% of each fold used for training and the remaining 10% of data being used for testing. ANN achieved the best performance while IBK had the worst accuracy which is the most important feature of an algorithm, as it reflects the ability to identify the correct decisions that are the main task of a classifier.

There is no significant difference between SMO, ANN, J48, NB, and PART apart from IBK which was outperformed by most of the other techniques. It reflects the strength of ML techniques in predicting human minds. In terms of solution building time J48 and NB gives better performance than PART, SMO, and ANN. In conclusion, authors demonstrated that the decision trees and *k*-nearest neighbor techniques give better results with lesser time than neural network and support vector machine-based solutions while automating the scheduling of RMC Trucks using ML techniques.

2.2.3 Genetic Algorithm for RMC Scheduling

Combined discrete-event simulation and genetic algorithms (GA) are applied in HKCONSIM to model and further optimize the one plant-multisite RMC plant operations in Hong Kong [19]. Further, Feng et al. [20] and Liuhenyuan et al. [3] introduced a solution where better optimization could be achieved by a Genetic Algorithm (GA) while focusing on scheduling RMC production across an environment with a single plant and single mixer where they further suggested to focus on multiple plant condition as a crucial research gap to be filled in future. Whereas, our work focusses on multiple plants, trucks, and construction sites.

2.2.4 Simulated Annealing for Scheduling

Simulated Annealing (SA) is a probabilistic, single-solution-based search method inspired by annealing process where a solid is slowly cooled until its structure reaches a minimum energy configuration [21]. SA is a type of local search algorithm that starts with an initial solution which is chosen at random. Then a neighbor of this solution is then generated and the change in cost is calculated. The current solution is replaced by the generated neighbor, if the reduction in cost is found during the comparison, otherwise the current solution will be retained. The process will be repeated until no further improvement can be found in the neighborhood of the current solution and so the descent algorithm terminates at a local minimum [22].

SA has its own technique to avoid converging to local optimums which is a commonly known disadvantage of other optimization algorithms. SA avoids local optimums by accepts worst solutions at higher temperatures by setting its acceptance probability to a higher value. Initial temperature, a rule for accepting a worse solution (i.e., lower profit solution), the cooling rate which is the rate of the temperature decrement, and a stop criterion are the key parameters of the algorithm [23].

Genetic Algorithm (GA) is a technique used for estimating computer models based on methods adapted from the field of genetics in biology. Possible model behaviors need to be encoded into "genes" to use this technique and current models are rated and allowed to mate and breed based on their fitness after each generation. Genes are exchanged in the process of mating to allow crossovers and mutation. Next generation is formed from the offspring of the current population and it is discarded after the formation. of modeling or optimization techniques are described by the Genetic Algorithm which mimics some aspect of biological modeling in choosing an optimum [24].

Adewole et al. [24] compared the performance of two algorithms. When the population size increases, GA can provide quality solutions. On the other hand, it increases the runtime significantly. It was revealed that both algorithms are very good solvers and can provide optimal solutions if the right set of parameters are set. Cooling rate closer to one should be set for simulated annealing to increase the quality of the solution which ultimately increases the no of iterations the algorithm will perform while affecting the runtime of the model [24]. Moreover, Wolpert and Macready [25] derived a number of “No Free Lunch” (NFL) theorems that demonstrate the danger of comparing algorithms by their performance on a small sample of problems. It also revealed the importance of embedding the specific knowledge into the behavior of the algorithm. Every sample was mapped to a unique new point because a search algorithm is deterministic.

2.2.5 Particle Swarm Optimization for RMC Scheduling

For use in real-number spaces, the particle swarm optimization algorithm has been introduced as an optimization technique where potential solution to a problem is represented as a particle having coordinates and rate of change in a D-dimensional space [26]. An improved Discrete Particle Swarm Optimization (DPSO is proposed to solve scheduling problem. A systematic approach including a mathematical model for the scheduling of dispatching RMC trucks was presented using an improved Discrete Particle Swarm Optimization (DPSO) by introducing swapping operators in [27]. Liu et al. [27] revealed that their approach is valuable in scheduling of dispatching RMC trucks and there is a promising future for the improved DPSO in the scheduling arena. Further, authors suggested that a more dynamic approach may be useful to deal with the uncertainty.

2.2.6 Ant Colony Optimization for RMC Scheduling

The Ant Colony Optimization (ACO) algorithm is one of the most recent meta-heuristic-based optimization technique that has been successfully used in complex routing problems [28]. Silva et al. [28] presented a new approach regarding job-to-truck assignment and the consequent truck routing as the routing problems are usually large combinatorial optimization problems. Therefore, it cannot be handled by the simple heuristics where the best known solutions for most of routing problems were obtained with meta-heuristics. ACO algorithms have demonstrated that they are competitive meta-heuristics for optimization problems which can be modeled in a graph environment.

2.3 Summary

During our literature review, we explored different truck scheduling patterns, current context of RMC truck dispatching and evaluation of the different machine-learning techniques used for RMC truck scheduling. Also, the literature revealed that the performance of simulated annealing algorithm outruns results of the genetic algorithm which also proved the efficiency of SA algorithm in solving travelling salesman problem. However, as the route and vehicle scheduling problems are known to be NP-hard, we cannot get an optimal solution within polynomial time [6], [7], [8], [9]. Therefore, it is essential to identify suitable heuristic-based solutions that can still maximize the customer satisfaction, efficiency, and company profit.

3. PROBLEM FORMULATION

Formulating the problem by identifying all parameters and constraints related to ready-mixed concrete truck scheduling process is presented in this chapter. Subsequently, constraint-based approach was used to filter out the possible search space as reducing the search space is important to achieve an acceptable solution to NP-hard problems [8]. Section 3.1 identifies the characteristics of the problem and Section 3.2 defines the constraints related to our problem while Section 3.3 summarizes the optimization problem of RMC truck scheduling.

3.1 Characteristics of Problem

Let **J** be the set of jobs, where each job $j \in \mathbf{J}$ has a delivery location and time of first unloading, volume, and buffer duration (list of symbols is given in Table 3.2). These jobs are to be processed by a set of plants **P** and set of trucks **T**, where each plant $p \in \mathbf{P}$ has a fixed location and load time, and each truck $t \in \mathbf{T}$ has a volume and fuel consumption rate with/without load. Moreover, industry specifications such as the ASTM C94 and regulatory requirements such as the maximum number of driving/working hours per driver per day need to be met. Let f_j be the fee for job j, which typically depends on the distance between pickup and delivery locations of the job. Cost per a volume of concrete is fixed regardless of the job; hence, not considered in our model. Our objective is to cover all jobs **J** with plants **P** and trucks **T**, such that profit is maximized across all the jobs. We use a constraint-based approach to filter out the possible search space as reducing the search space is important to achieve an acceptable solution to NP-hard problems [8].

Table 3.1 lists the characteristics of our problem which summarizes the nature of our problem. Table 3.2 lists the symbols related to job, plant, truck, and solution, respectively. e_{peak} and e_{off_peak} represent the traffic impact factor on fuel consumption and the peak hour starts in morning from 6:00am - 9:00am and evening from 4:00pm - 8:00pm. e_{speed_max} capture specific constraints such as speed regulations for special purpose vehicles. Maximum time between two consecutive truck loads which is defined to avoid cold joints in the concrete is represented as j_{buffer}. r_{time} is the time restriction on the delivery of RMC and it varies according to the special properties added to the RMC. Usually the time restriction is 90 minutes as per the ASTM C94 specification [29] for RMC if no special property is added. Fuel consumption of a truck is represented with and without load using $t_{fuel\_cons}^{load}$ and $t_{fuel\_cons}^{no\_load}$ and it is measured using liters per kilometer. The fuel consumption of a truck while idling at site or plant (with engine on), $t_{fuel\_cons}^{idling}$ is measured using liters per minute. $t_{maintenance}$ is the maintenance cost factor which captures the depreciation of the parts of a RMC truck occurs during each trip. Maximum travel time per truck on a given day is represented by $t_{time}^{max\_day}$ and it was calculated as per the road safety regulation published by the authorities which enforces adequate resting time for both driver and truck in continuous work. It supports in reducing the road accidents by providing the drive a fatigue free environment. t_{wash_down} is the time taken to remove the residuals in the concrete mixing barrel of the RMC truck and it occurs immediately after each unloading of RMC.

Table 3-1. Characteristics of the problem.

Attribute	Characteristic of Problem
Number of Plants	Multiple
Size of Available Fleet	Multiple
Type of Available Fleet	Homogeneous
Capacity of Available Fleet	Homogeneous
Nature of Demand	Pre-defined Delivery Time
Location of Demand	Known (Geographically Dispersed)
Costs	Vehicle Operating Cost, Waiting Cost

3.2 Constraints

To be eligible for a job, trucks and plants need to satisfy the following set of constraints:

Table 3-2. List of symbols related to job, plant, truck, and solution.

Symbol	Description
c_j	Cost for a job j
f_j	Fee for a delivery of job j
c_{liter}	Cost per liter
c_{travel}	Cost of travel for a job j
$c_{waiting}$	Cost of waiting for a job j
$c_{unit\_distance}$	Cost per one kilometer
e_{peak} / e_{off_peak}	Traffic impact factor on fuel consumption during peak / off-peak time
e_{speed_max}	Maximum speed allowed for a truck
j_{buffer}	Maximum time between 2 consecutive unloads of j
$j^{haul}_{distance}$	Distance to job site while hauling
$j^{return}_{distance}$	Distance to job site while returning
j_{first_unload}	Time to unload first load for job j at construction site
j_{id}	Job ID
$j_{location}$	Job location
$j_{total\_distance}$	Total distance traveled for job j
j_{volume}	Required volume of concrete in m^3 for job j
$j_{num\_truckloads}$	Required number of truckloads for job j
p_{id}	Plant ID
$p_{location}$	Batch plant location
p_{load_time}	Time to load mix to a RMC truck at plant p
p_{range}	Maximum serving range of plant p
r_{time}	Time restriction on delivery of RMC, e.g., as per ASTM C94 specification
$t^{load}_{fuel\_cons}$	Fuel consumption (l/km) of truck t with load
$t^{no\_load}_{fuel\_cons}$	Fuel consumption (l/km) of truck t without load
$t^{idling}_{fuel\_cons}$	Fuel consumption (l/minute) of t while idling
t_{haul_time}	Hauling time
$t^{j}_{haul\_time}$	Hauling time of j-th truck
t_{id}	Truck ID
t_{load_time} / t_{unload_time}	Loading/ Unloading Time
$t_{maintenance}$	Maintenance cost factor of truck t
$t^{j}_{return\_time}$	Return time of j-th truck
$t^{\max\_day}_{time}$	Maximum travel time per truck on a given day
t_{unload_wait}	Waiting time (unloading)
t_{volume}	Maximum volume of truck t in m^3
t_{wash_down}	Wash-down time of truck t

Time Constraint

Time restriction on delivery (r_{time}) as per ASTM C94 specification depends on the properties of the concrete, where it varies according to the special properties added to concrete during the mixing stage. Therefore, the time constraint for a truck t can be specified as:

$$\forall\ t_{id},\ p_{load\_time} + t_{haul\_time} + t_{unload\_time} \leq r_{time} \tag{3.1}$$

Range Constraint

The maximum speed of a truck (e_{speed_max}) should not exceed the speed limits set by regulators (this is an environmental parameter which is set by the business environment that RMC delivery operates). Therefore, a plant cannot serve jobs that are too far away to reach on time. Thus, *range constraint* for a plant *p* can be defined as follows:

$$p_{range} = e_{speed\_max} \times r_{time} \tag{3.2}$$

Job Area Constraint

Selecting a plant for a job depends on the maximum range of the plant. Therefore, job area constraint can be defined as follows:

$$j_{distance}^{haul} \leq p_{range} \tag{3.3}$$

Job Duration Constraint

If one job is allocated to a plant, another job with overlapping job duration cannot be allocated to the same plant. Hence, following job duration constraint needs to be satisfied:

$$j_{job\_duration}^{i} \neq\ j_{job\_duration}^{j}, \forall i \neq j \tag{3.4}$$

Travel Time Constraint per Truck

When assigning a new job j, the travel time constraint per truck can be defined as follows:

$$\left(\sum_{i=1}^{n-1}\left(t^{i}_{haul\_time} + t^{i}_{return\_time}\right)\right) + \left(t^{j}_{haul\_time} + t^{j}_{return\_time}\right) \leq t^{\max\_day}_{time} \quad (3.5)$$

3.3 Optimization Problem

Given a set of jobs **J**, trucks **T**, and batching plants **P**, our main objective is to cover as many jobs as possible. This is required to improve customer satisfaction as the majority of customers are engaged in a long-term business relationship. Further, there is a secondary objective to maximize RMC company's overall profit while minimizing the cost by allocating the most appropriate truck and batching plant to the job within the customer requested time frame. Dead runs and idling of the trucks and plants should be minimized to maximize the profit of the company. Therefore, the objective function can be formulated as follows:

$$\forall j_{id} \in \mathrm{J},\ \forall p_{id} \in \mathrm{P},\ \mathrm{Max}\ (|j \text{ with assigned } p|) \quad (3.6)$$

$$Max \sum_{\substack{j\in\mathbf{J} \\ p\in\mathbf{P}}} \left(f_j - c_j\right) \quad (3.7)$$

Constraint (3.6) maximizes the number of jobs with an assigned plant. $f_j - c_j$ is the profit for job j which we attempt to maximize in (3.7). Hauling and returning distances are the decision variables in our objective function. Because both f_j and c_j are depending on the hauling and returning distance, our objective function is bounded as both distances are limited by the range constraint and job area constraint which ultimately limits the search space for the job scheduler.

4. PROPOSED SOLUTION

We assume that every evening, the next day's schedule is determined based on the already confirmed jobs and available plants and trucks [9]. However, there is an ambiguity upon the job end time due to the delays at the job site and during travel, which depends on traffic and other environmental conditions such as weather and road closures that are hard to predict. Moreover, the route is planned to utilize the full truckload to avoid wastage and dead runs of the truck. Dynamic scheduling to overcome last-minute changes is left as future work. The solution consists of a rule checker that enforces the constraints mentioned in Section 3.1 and an optimization phase that attempts to cover as many jobs as possible while maximizing the overall profit. Section 4.1 presents the rule checker which enforces the constraints and the job scheduler which uses the simulated annealing to optimize the scheduling process is presented in Section 4.2.

4.1 Rule Checker

In NP-hard problems, search space reduction is essential to achieve an acceptable solution, which is achieved through the rule checker. Therefore, given a set of jobs, trucks, and plants, we developed a rule checker to first evaluates the constraints given in Section 3.2 and then to identify a potential list of plants and trucks for a given job. All the distances from the location of the job to available plant (within the range) are taken from Google Distance API [30] to achieve more reliable and accurate estimates. Timings were calculated using the distance taken from Google Distance API and maximum allowable speed for special-purpose vehicles.

As illustrated in Figure 4.1, when a new job is to be assigned, all plants and trucks are evaluated. Rule checker runs through the constraints from top to bottom and filters the eligible set of plants and trucks gradually. When it goes through a constraint, it simultaneously outputs the possible set of plants and trucks for all jobs. During the initial steps, rule checker proceeds with constraints such as time, range, and job area. It then checks whether the jobs are clashing to ensure job duration constraint. Jobs are randomly selected to enhance the search space to get all possible combinations and continuously checked to ensure start and end time of jobs do not overlap. If jobs are overlapping, it will eliminate the overlapping jobs and create a new matrix for the job scheduler. In the final step, rule checker runs through the travel time constraint to ensure that a truck is not exceeding the maximum allowed travel time per day as the driver of a truck should take adequate rest time to maintain road safety.

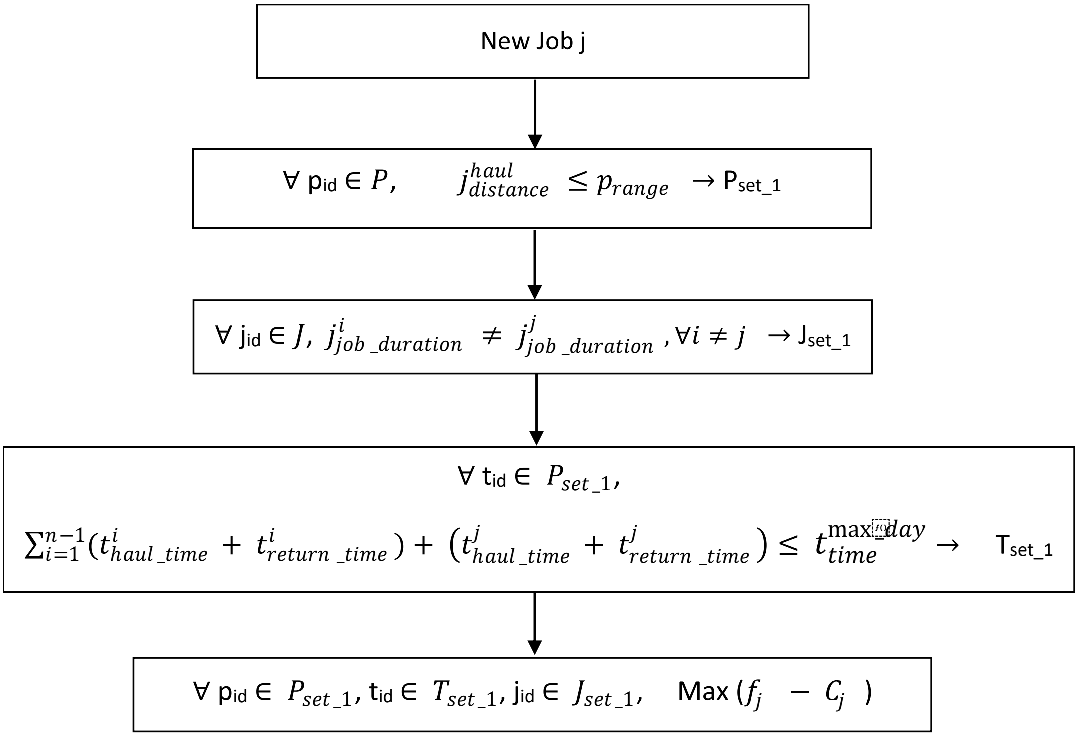

Figure 4.1. Solution model for rule checker.

4.2 Job Scheduler

Simulated Annealing (SA) algorithm is broadly used in global optimization problems including limousine renting [5] and vehicle delivery [31]. SA provides a reasonably optimized solution within a reasonable time and can be optimized according to the context [9]. Also, Maghrebi et al. [32] revealed that SA has the highest accuracy when comparing the solutions obtained from Bayesian network when both SA and Genetic Algorithm (GA) are set as search algorithms. Adewole et al. [24] compared the performance of SA and GA algorithms and concluded that as the population size increases, GA can provide a better quality solution. However, this comes at a significant increase in computational time compared to SA.

Recently, researchers have expanded on the original idea of PSO with alterations fluctuating from minor parameter adjustments to complete revamp of the algorithm and used PSO for comparison tests of other global optimization algorithms including GA [27]. Also, the intelligent optimization algorithms including GA, SA, and ant colony algorithms have been used to solve the question of process route optimization in recent researches.

Compared to SA, several shortages are experienced when the basic ant colony algorithm is used to solve practical problems. For example, higher search time is required with large search space and especially its tendency to converge early at local optimum while stopping the search process [33]. It has been shown that the process of convergence to the optimal solution is more express if the initial solution is taken by means of a good heuristic [34]. Whereas SA is a type of iterative improvement algorithm that provides a way to escape local optima. It is of non-deterministic type and is based upon the probability to obtain an optimal solution comparing the current situation of the objective function to the possible improvement bound to a *controlled movement* in the space of the feasible solutions [34].

Therefore, we chose SA for the optimization step in the job scheduler. Alternatively, other meta-heuristic techniques could be used for said optimization depending on how well the properties of the problem and technique matches. For example, we evaluate the suitability of PSO compared to SA in Chapter 5.

Once a candidate list of plants and trucks are identified for a job, we use the Simulated Annealing (SA) algorithm to schedule a plant and a truck for a job while satisfying the objective function in Eq. 3.6 and 3.7. Moreover, as the solution needs to be built to support many constraints and the data maybe chaotic and noisy, SA is a better fit for RMC scheduling because it is known to be a robust technique that can deal with such conditions [35]. SA algorithm was chosen for optimization as it is used in global optimization problems [36], provides a reasonably optimized solution within a reasonable time [24], and it can be optimized according to the context and application

[35]. Furthermore, SA is quite versatile, as it does not rely on any restrictive properties of the model.

The identified job, plant, and truck combinations (j_{id}, p_{id}, and t_{id}) are then fed to run through simulated annealing algorithm to find an optimal schedule while maximizing the job coverage and overall profit. SA algorithm picks a job randomly and then allocates a plant based on its availability. Because our goal is to maximize both job coverage and profit, SA will check different plant combinations for a single job as the availability of that plant will change for other jobs once it is assigned to a job which ultimately affects the overall profit and job coverage. For example, once a job is assigned, the plant is not eligible for another job with a job duration which overlaps with the job duration of the current job. Moreover, this will depend on the job picking sequence as it is randomized when inputting to the SA algorithm. Therefore, different job to plant combinations are compared again and again through rule checker, while assigning new jobs to plants using the SA algorithm. Cost for a job c_j is calculated based on the job distance, estimated total job time, fuel consumption, and environmental factors experienced by the job as follows:

$$c_j = (c_{travel} + c_{waiting}) \times c_{liter} \tag{4.1}$$

$$c_{travel} = (((j_{distance}^{haul} / t_{fuel\_cons}^{load}) \times e_{peak} / e_{off\_peak}) + ((j_{distance}^{return} / t_{fuel\_cons}^{no\_load}) \times e_{peak} / e_{off\_peak})) \times t_{maintenance} \tag{4.2}$$

$$c_{waiting} = ((t_{wait\_time}^{unload} + t_{wash\_down}) / t_{fuel\_cons}^{idling}) \times t_{maintenance} \tag{4.3}$$

All distances from job to available plants (within the range) are taken from Google Distance API [30] to achieve more reliable and accurate estimates. $j_{distance}^{haul}$ and $j_{distance}^{return}$ are the decision variables as the other factors affecting c_j and f_j are constant for a job/trip. Timings were calculated using the distance taken from Google Distance API and maximum allowable speed for special-purpose vehicles. Fee for a job f_j is calculated based on the estimated total job distance and is calculated as follows:

$$f_j = j_{total\_distance} \times c_{unit\_distance} \tag{4.4}$$

$$j_{total\_distance} = j_{num\_truckloads} \times (j_{distance}^{haul} + j_{distance}^{return}) \tag{4.5}$$

$$\lfloor j_{num\_truckloads} \rfloor = \frac{j_{volume}}{t_{volume}} \tag{4.6}$$

Initially, we consider that a job is fully served by a single plant and all the trucks will return to the respective plant after unloading. Also, we supported the option of either advancing or delaying a job by a certain time window to reduce conflicts between the overlapping job durations. In such cases, only one job can be covered even if the two jobs overlap by only a few tens of minutes. Rather than completely rejecting one of the jobs, it maybe possible to contact the client and renegotiate the delivery time. This is in fact practiced in the industry and advancing or delaying the j_{first_unload} by an hour is not uncommon. Therefore, we checked the possibility of adjusting j_{first_unload} by a given time window to check whether the job coverage and profit could further increase.

SA is a probabilistic, single-solution-based search method inspired by annealing process where a solid is slowly cooled until its structure reaches a minimum energy configuration [21]. SA has its own technique to avoid converging to local optimums which is a commonly known disadvantage of other optimization algorithms. SA avoids local optimums by accepting worst solutions at higher temperatures by setting its acceptance probability to a higher value. Initial temperature, a rule for accepting a worse solution (i.e., lower profit solution), the cooling rate which is the rate of the temperature decrement, and a stop criterion are the key parameters of the algorithm [23]. This nature of the algorithm avoids the local minima as our search space is large. When considering our solution, the primary objective is to maximize the job coverage and then the secondary objective is to optimize cost matrix to maximize the overall profit. In SA implementation, we prioritize to cover all possible jobs while maximizing the overall profit of the RMC company. As SA algorithm target global optimization, we even allow jobs with negative profit to gain maximum job coverage. However, in practice, it was noted that no jobs with negative profit were scheduled by the SA-based scheduler.

Table 4-1. Comparison of different solution scenarios.

Manual Solution	**Solution 1**	**Solution 2**	**Solution 3**
	Fixed Plant	**Fixed Plant with Time Window**	**Free to Move**
Job to plant/truck allocation basis	Job to plant/truck allocation basis	Job to plant/truck allocation basis	Trip to truck allocation basis
Sort jobs according to First Unload Time in ascending order	Enforce constraints and conditions	"Time of first unload" is adjusted with time windows to eliminate the job duration clashes	Enforce constraints and conditions with enhanced search space
Select plant which makes maximum profit for given job and assign to the plant	Randomly assign a job to plant/trucks	Same conditions and steps followed as same as "Fixed Plant" solution	Randomly assign a trip to truck
Eliminate the overlapping jobs	Job will be completely served by the assigned plant		Truck can move freely to another plant after completing a job
Repeat the process for all jobs while eliminating overlapping jobs at each step	Assigned plant will be available for another job only after completing the assigned job		Single job is served by multiple plants

Table 4.1 summarizes the solution approaches of four solutions including the manual solution and three SA based solutions. In comparison, "Manual Solution", "Solution 1" and "Solution 2" uses job to truck/plant allocation basis where the "Solution 3" uses trip to truck allocation. It differs the "Solution 3" from other three solutions (Manual Solution, Solution 1 and Solution 2) which allows a truck to move freely across the plants and a single job is served by multiple plants where the other three solutions are based on single plant while creating the job schedule for the RMC company.

5. PERFORMANCE ANALYSIS

We evaluate the performance of the rule engine and Simulated Annealing (SA) based solution using a set of workloads derived from a real RMC company. Section 5.1 presents the workload creation which was derived after a comprehensive descriptive analysis from real RMC company data set. Section 5.2 illustrates the results of the simulation.

5.1 Workload Creation

We did a descriptive analysis to a dataset given by a real RMC Company to understand the characteristics of the RMC truck scheduling. The data set we got from the company consisted the data of nine RMC trucks with the information of following parameters including latitude, longitude speed, engine status, real-time clock, bearing, device status, ignition status and fuel level. Truck were assigned to two plants and Figure 5.1. shows the areas served by two plants.

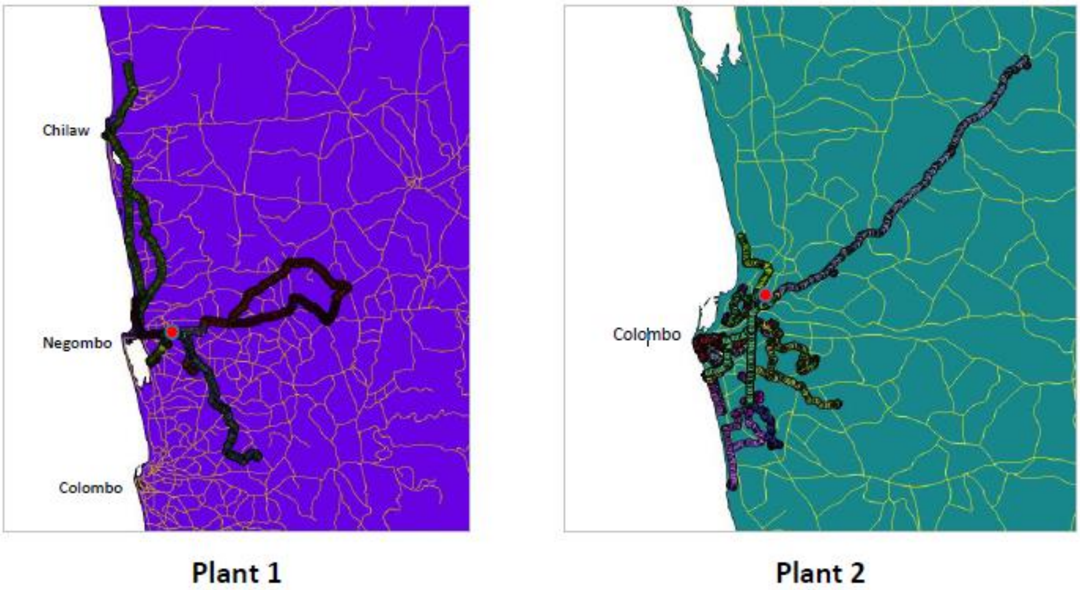

Figure 5.1. Areas served by the two plants.

A comprehensive data analysis was conducted to understand the total fuel usage, total travel distance, number of inactive days, and average fuel consumption of each truck. As shown in Figure 5.2, fuel consumption of each day was plotted, and it includes the consumption of idling at plant (after loading), trip (one way and return), and idling at construction site of a RMC truck.

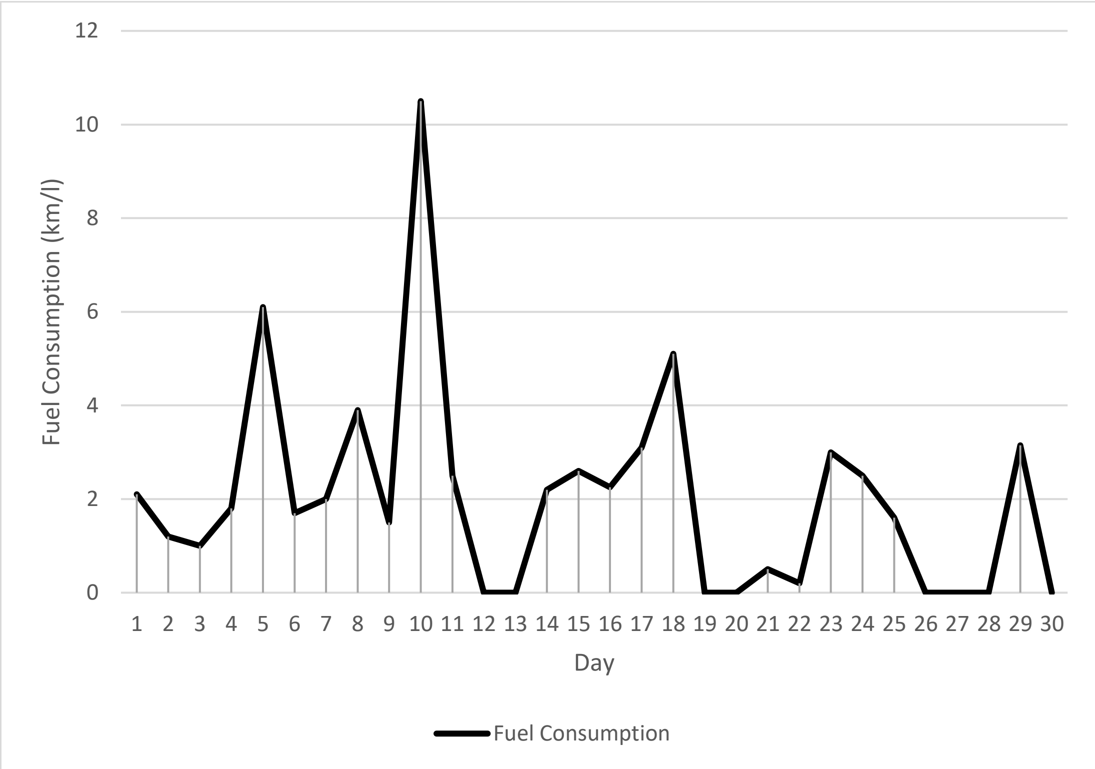

Figure 5.2. Daily fuel consumption plot of an RMC truck.

Moreover, we plotted each trip of the truck and it was found that many trips are violating the ASTM C94 regulation where 1.5 hour is the standard time to deliver RMC if no special property is added to the concrete. Figure 5.3 illustrates how different trips are violating the ASTM C94 regulation while delivering RMC.

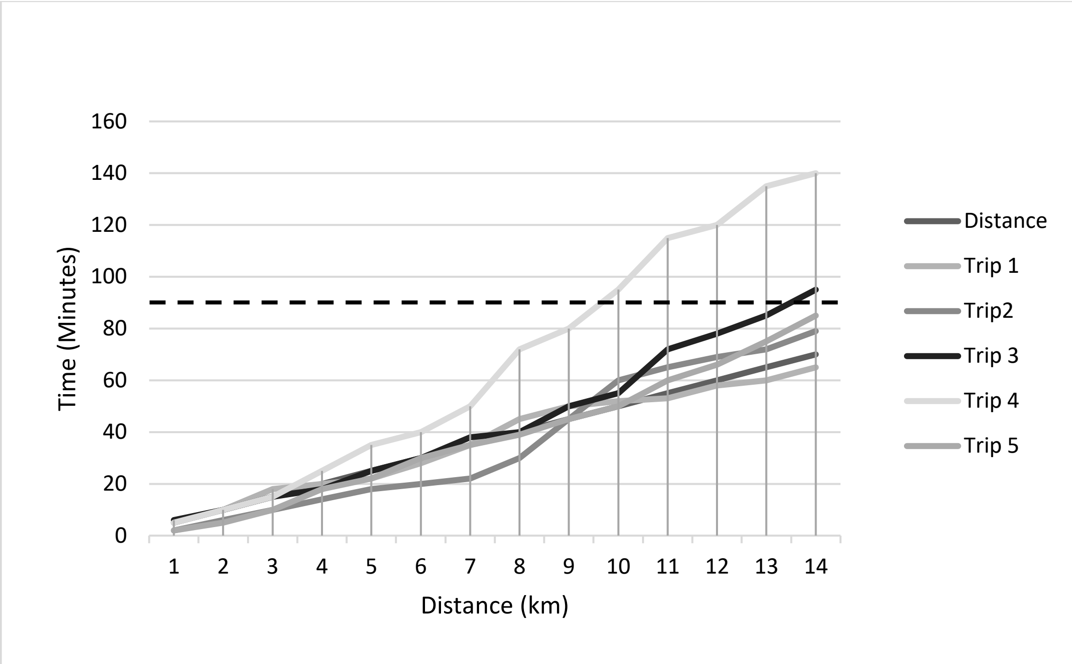

Figure 5.3. Travel time of a RMC truck (5 trips).

Based on our descriptive analysis, we used a dataset with 158 jobs with 735 truckloads during the week, against a set of 5 plants and 47 trucks as shown in Table 5.1. This dataset was created based on properties extracted from a dataset of a real RMC delivery company in Sri Lanka. This includes the distribution of job locations, delivery times (some jobs span across the week) and other constraints. Figure 5.4 illustrates the job and plant locations of our dataset. We considered 6:00am to 9:00am as peak time (e_{peak}) and 4:00pm to 8:00pm as off-peak time (e_{off_peak}). Truck maintenance cost ($t_{maintenance}$) is considered while calculating the cost of the job and we used the constant $t_{maintenance}$ = 1.1. After unloading the concrete, RMC truck needs to go through a quick wash to remove residual inside the drum of the truck. This is also a constant value and we denote t_{wash_down} = 8 minutes.

Table 5-1. Job distribution across the week.

Day of the Week	Number of Jobs	No of Available Trucks	No of Trips
Monday	22	47	104
Tuesday	20	47	91
Wednesday	21	47	104
Thursday	18	47	90
Friday	23	47	105
Saturday	26	47	116
Sunday	28	47	125
Total	**158**		**735**

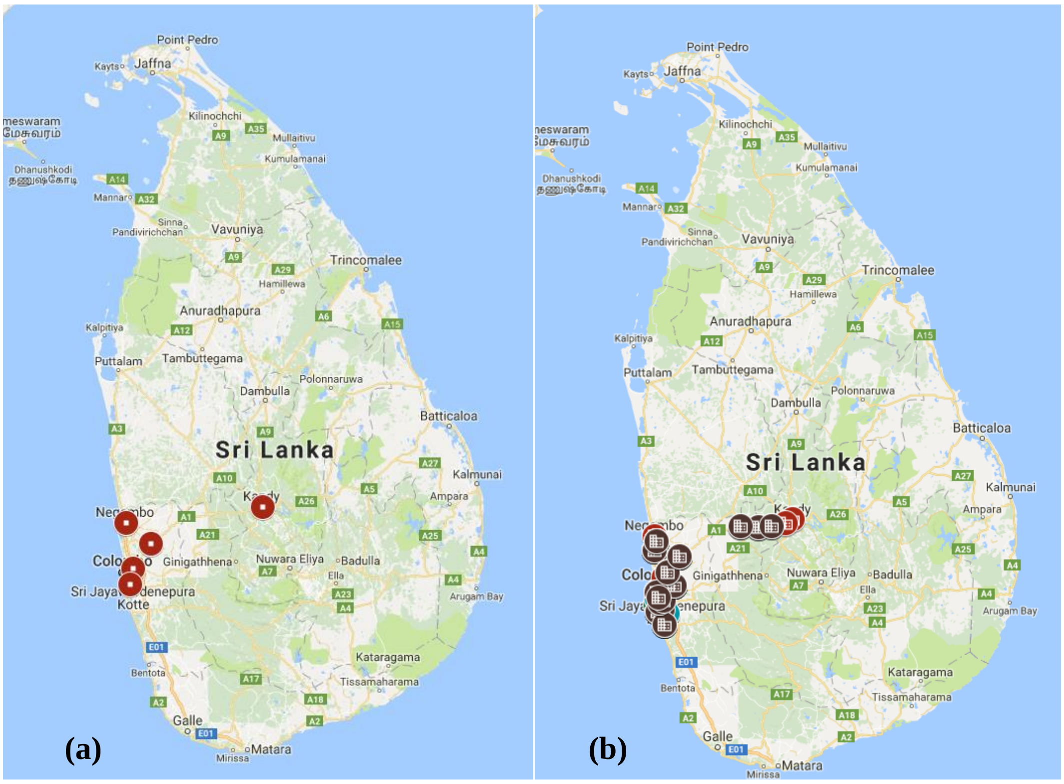

Figure 5.4. Distribution of (a) plant and (b) job locations.

The simulation ran on a machine which has Intel Core i7-6500U CPU, 2.60 GHz processor, 12 GB RAM, and 4 MB Cache. We implemented a R-based simulator which initiates from a R code for travel salesman problem. We used dplyr, gtools, lubridate, splitstackshape packages while creating the search space for the job scheduler. Customized SA algorithm was used in the optimization step as we need to create multiple data files including profit, optimized schedule including plant/truck allocation and job coverage. The alteration process of the SA algorithm depends on the cooling strategy such as linear or exponential, cooling rate, and the energy of the system [37]. Therefore, different combinations were tested on datasets resulting in the following set of parameters:

- initial temperature = 10^4
- cooling rate = 0.003
- terminating condition of temperature > 1.

In tuning process, we ran the simulation ten times with same dataset, and same configurations while varying the random seed. Moreover, we tried various initial temperatures against average job coverage. The results listed in Table 5.2 indicates the average job coverage against the initial temperature with the cooling rate of 0.003. We used an initial temperature of 10^4 which has a higher job coverage and can be processed within acceptable running time, and cross through relatively higher amount of transitions with cooling rate of 0.003.

Table 5-2. SA job coverage against initial temperatures with 0.003 cooling rate.

Initial Temperature	**Average Job Coverage (%)**
1,000	80
10,000	88
20,000	78

In addition to SA, PSO algorithm was used in the optimization step to compare with the results of SA job scheduler. We have used the discrete version of PSO (DPSO) where the representations of the position and velocity of the particle is extended from the real vector to integer vector. Kashan and Karimi [38] have accomplished the mapping between the job scheduling problem and the particle using the same DPSO algorithm. The alteration process of the DPSO algorithm depends on the few parameters. Therefore, different combinations were tested on datasets resulting in the following set of parameters:

- $\omega = 1$
- $C_1, C_2 = 2$
- Number of iterations = 60
- Population = 30

In the tuning process, we ran the simulation six times with same dataset and same configurations while varying the random seed. Moreover, we tried various number of iteration values against average job coverage. The results listed in Table 5.3 indicates the average job coverage against the number of iterations with the population of 30. We used 60 as the number of iterations which resulted in higher job coverage and can be processed within acceptable running time and iterate through relatively higher amount of transitions with population of 30.

Table 5-3. DPSO job coverage against number of iterations with 30 population.

Number of Iterations	Average Job Coverage (%)
25	80
50	78
60	87
75	82

5.2 Simulation Results

In the RMC business, time of unloading is crucial once the concreting process starts. Job buffer time allows trucks to arrive with a slight delay, but it is very limited as the concrete gets cold joints if the next truck is delayed more than the allowed buffer time. However, time of first unloading on most jobs are not strict and can be advanced or delayed up to an hour, as far as the company negotiates with the client. This enables flexibility in optimally assigning jobs to get the maximum job coverage while maximizing profit.

We compared the results with manual job scheduling. We considered a case of a human expert first sorting the jobs for each day according to the ascending order of j_{first_unload}. Then a job is allocated to the plant that gives the maximum profit for the job. For each day's workload, we plan the schedule on the previous evening.

Table 5.4 shows the results of the proposed solution (fixed plant scenario) and manual job allocation by a human expert. Proposed job scheduler enforces all the constraints and randomly assigns plant/truck to jobs and the selected job will be solely served by the same plant till it completes the job. On all days job coverage of the proposed automated scheduling solution is either the same or better, while profit is always higher. For example, job coverage on Thursday was increased from 83% to 100% with a 21% increase in profit.

Table 5-4. Job scheduling performance comparison (fixed plant scenario).

Day of the Week	Manual Job Scheduling		Proposed Job Scheduler	
	Traditional Scenario		Fixed Plant	
	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage
Monday	721	82%	791	87%
Tuesday	729	75%	754	90%
Wednesday	736	81%	736	91%
Thursday	567	83%	685	100%
Friday	881	78%	885	91%
Saturday	803	73%	924	88%
Sunday	910	75%	1,034	93%

Moreover, we compared the results of Manual Job Scheduling (Traditional Scenario) and Proposed Job Scheduler (Fixed Plant) with the best-case scenario where the time of first unload is decided by the RMC company. That way, the RMC company can have the best job coverage while trying to minimize cost. Table 5.5 reveals that the proposed job scheduler performs well as the profit difference between the Manual Job Scheduling (Best Case Scenario) varies between 1%-10%. In comparison, manual job scheduling (traditional scenario) has the profit variation up to 20% compared to the Manual Job Scheduling (Best Case Scenario). Also, the results of Table 5.5 reveal that the solutions of our Proposed Job Scheduler (Fixed Plant) are significantly closer to the Best Case Scenario of the Manual Job Scheduling method.

Table 5-5. Job scheduling performance comparison against the best case.

Day of the Week	Manual Job Scheduling		Manual Job Scheduling		Proposed Job Scheduler	
	Best Case Scenario		Traditional Scenario		Fixed Plant	
	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage
Monday	812	100%	721	82%	791	87%
Tuesday	795	100%	729	75%	754	90%
Wednesday	799	100%	736	81%	736	91%
Thursday	713	100%	567	83%	685	100%
Friday	915	100%	881	78%	885	91%
Saturday	975	100%	803	73%	924	88%
Sunday	1,150	100%	910	75%	1034	93%

Table 5.6 reveals the results of DPSO algorithm against the manual job scheduling and SA job scheduler. It clearly illustrates that the SA job scheduler outperforms the manual solution, as well as the results of the DPSO algorithm. However, when comparing the performance of the DPSO job scheduler with the Manual Job Scheduler (Traditional Scenario), DPSO-based Job Scheduler has significantly increased both profit and job coverage by 3% and 6%, respectively.

Table 5-6. Job scheduling performance comparison (Simulated Annealing vs. Discrete Particle Swarm Optimization).

Day of the Week	Manual Job Scheduling		Proposed Job Scheduler (SA)		Proposed Job Scheduler (DPSO)	
	Traditional Scenario		Fixed Plant		Fixed Plant	
	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage
Monday	721	82%	791	87%	735	83%
Tuesday	729	75%	754	90%	732	89%
Wednesday	736	81%	736	91%	732	83%
Thursday	567	83%	685	100%	615	87%
Friday	881	78%	885	91%	881	82%
Saturday	803	73%	924	88%	868	79%
Sunday	910	75%	1,034	93%	918	83%

In addition, we carried out another simulation by changing the job allocation basis. Compared to the earlier step, we did trip to truck allocation where trucks can move freely among plants. In this scenario, a job is served by multiple plants and it increases the utilization of plants and RMC trucks. Moreover, it clearly improved the profit, as well as the job coverage significantly compared to both manual scheduling and fixed plant scenario. As shown in Table 5.7 job coverage of fixed plant scenario ranges from 85% to 100% while free to move improved it to 90% to 100%. Additionally, there is a 5% increment of the profit on average compared to the fixed-plant scenario in this trip to truck allocation scenario. Moreover, the comparison results between the Best Case Scenario vs. Free to Move Scenario proves that the free to move scenario has optimized the job schedule as profit difference between the two methods varies only up to 2%. In summary, the proposed SA-based solution increased the average job coverage and profit of the company for each day of the considered week by 13% and 9%, respectively for the fixed plant scenario while 16% and 14% increment seen in free to move method where the home plant restriction is lifted.

Figure 5.5 summarizes the graphical representation of the three solutions and it also illustrates the increment of job coverage in three different scenarios. It clearly illustrates how the proposed SA Job Scheduler significantly increases the job coverage compared to the Manual Job Scheduling.

Table 5-7. Job scheduling performance comparison (fixed plant and free to move scenario).

Day of the Week	Manual Job Scheduling				Proposed SA Job Scheduler			
	Best Case Scenario		Traditional Scenario		Fixed Plant		Free to Move	
	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage	Profit (x10)	Job Coverage
Monday	812	100%	721	82%	791	86%	812	90%
Tuesday	795	100%	729	75%	754	90%	789	94%
Wednesday	799	100%	736	81%	736	90%	785	92%
Thursday	713	100%	567	83%	685	100%	710	100%
Friday	915	100%	881	78%	885	91%	903	94%
Saturday	975	100%	803	73%	924	88%	965	91%
Sunday	1,150	100%	910	75%	1034	93%	1,138	95%

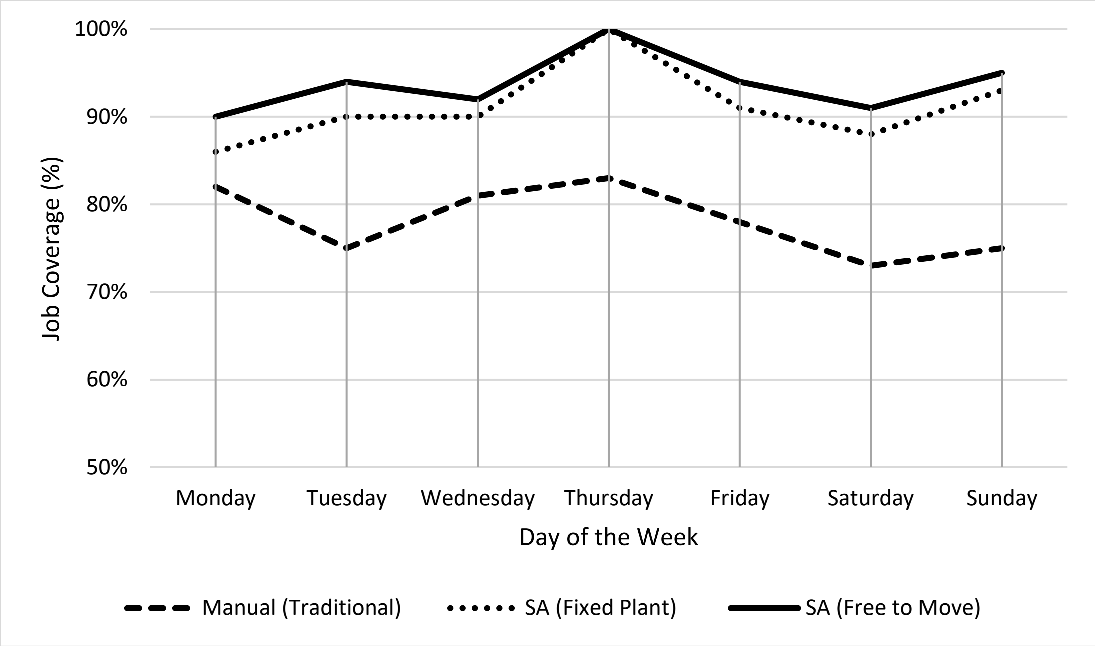

Figure 5.5. Graphical representation of the performance of three solutions.

In Table 5.8, we consider the case of allowing j_{first_unload} to be changed by a pre-defined time window in fixed plant scenario. We considered time windows of 15, 30, 45, and 60 minutes, where any job can be either advanced or delayed within the time window, as far as it leads to better job coverage. Table 5.8 shows that the introduction of a time window could further improve the job coverage and profit, in many cases. For example, on Tuesday having any sort of flexibility in changing j_{first_unload} enable higher job coverage. However, profit increases only when the time window is 60 min. Whereas on Sunday neither the job coverage or profit improves (except for 60 min window) with increasing time windows. Moreover, Sunday result also reveals that similar job coverage could produce different profits confirming that the profit can be maximized while creating a schedule with different job combinations. Profit can also be maximized with a lower job coverage, e.g., on Saturday. This situation is not ideal for an RMC company in the long-run as a few lost jobs today, even with better profit per job, could lead to long-term customer churn. Therefore, in cases where adjusting j_{first_unload} is useful, scheduling manager may negotiate with the client to adjust time of first unload of the relevant job(s).

However, sometimes by trying to accept one job by changing j_{first_unload} of the same job or another job could lead to a sub-optimal solution, as more profitable jobs could be missed in the future allocations as the plants and trucks may already be assigned within the SA algorithm. Consequently, this could reduce the overall job coverage and/or profit. For example, on Thursday ± 45 min time window reduces both the coverage and profit. This is an artifact of our design as it is willing to accept sub-optimal move with the hope for better coverage. Nevertheless, it is common for SA-based solutions to run multiple times and then pick the most preferred solution. Similarly, our solution can be executed multiple times per each time window, and the solution that maximizes both the coverage and profit can be chosen. For example, if we choose the best cases for each day and time window average job coverage and profit could be further increased to 21% and 13%, respectively compared to the manual scheduling. Furthermore, the comparison results of the Best Case Scenario vs Fixed Plant (With Time Window) Scenario shows that the addition of time window has increased proximity both profit and job coverage to the results of best case scenario as the variation of average job coverage and profit are 2% and 1% respectively. The execution time of SA algorithm increases as the number of jobs increase due to the increased search space. In our R-based implementation, execution time ranges between 30 to 50 min when a ± time window is introduced, as it significantly changes the search space. However, this is unlikely to be a practical hindrance, as the execution of each parameter combination can be parallelized, and SA algorithm implementation can be optimized. For example, our past experiences show an SA implementation based on more flexible languages such as Java and Python and could execute under 5 min with similar workloads [31].

The sensitivity analysis of Z. Liu et al. [3] shows that the cost rate can be reduced by 1.79% approximately with GA where our solution which is based on SA increases the profit by 9%, while increasing up to 13% by automatically adjusting the first unload time by a few 10s of minutes to reduce job conflicts.

Table 5-8. Proposed job allocation results with a varying time window (fixed plant scenario).

Time Window (Min)	**0**	**±15**	**±30**	**±45**	**±60**
Monday					
Job Coverage	86%	91%	86%	91%	97%
Profit (x10)	791	803	808	814	808
Tuesday					
Job Coverage	90%	100%	95%	100%	100%
Profit (x10)	754	721	744	717	795
Wednesday					
Job Coverage	90%	100%	90%	90%	90%
Profit (x10)	736	779	724	765	733
Thursday					
Job Coverage	100%	94%	100%	89%	94%
Profit (x10)	685	642	702	633	637
Friday					
Job Coverage	91%	87%	97%	96%	91%
Profit (x10)	885	852	895	852	858
Saturday					
Job Coverage	88%	88%	85%	92%	98%
Profit (x10)	924	889	891	897	972
Sunday					
Job Coverage	93%	93%	93%	86%	98%
Profit (x10)	1,034	990	1,027	970	1,060

Change of the cooling rate affects the number of transitions in SA under same initial temperature. We increased the cooling rate to 0.03 without changing the initial temperature. Figure 5.6 shows the profit against different cooling rates for the week while Figure 5.7 shows the jobs coverage against different cooling rates. However, in both cases, results of the 0.003 cooling rate are dominating, as it clearly shows that the 0.03 rate settles at a local optimum value rather than the global optimum value. Moreover, higher cooling rate tend to stop at local optimum values as it skips important combinations when cooling at a higher and faster rate than 0.003. Therefore, it does not support to achieve the ultimate research objective of our research by maximizing both profit and job coverage of an RMC company.

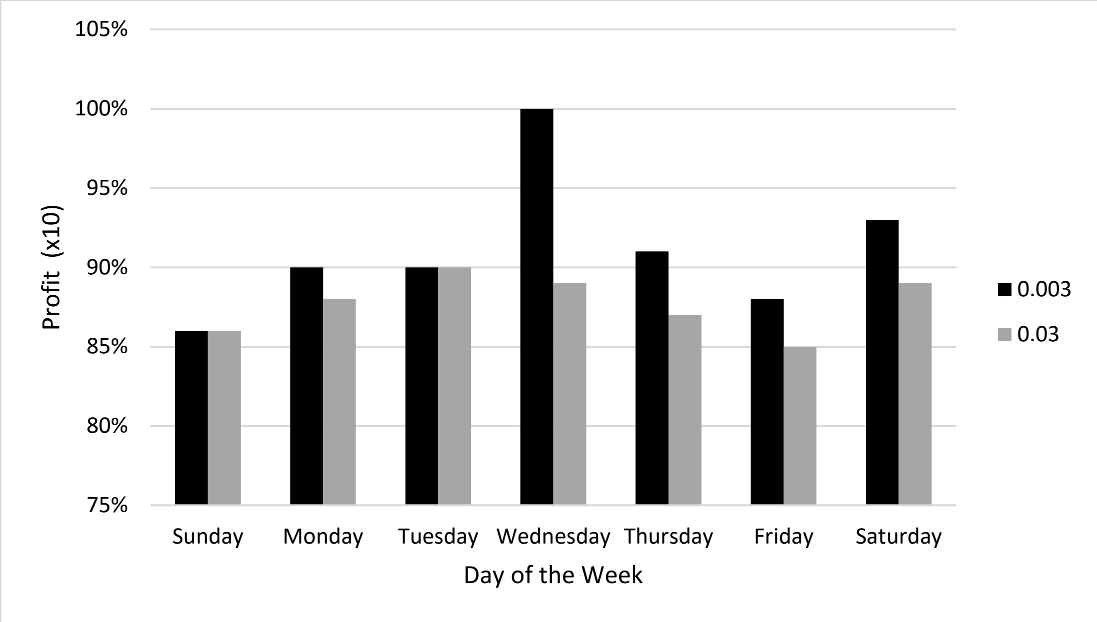

Figure 5.6. Profit comparison for the different cooling rates.

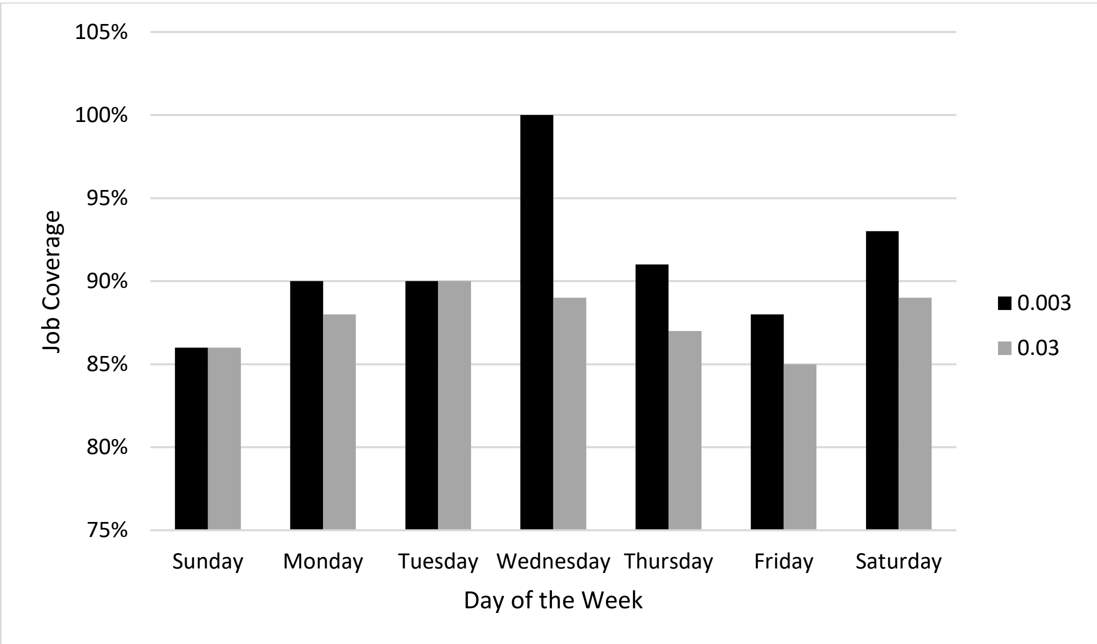

Figure 5.7. Job coverage comparison for the different cooling rates.

In summary, the proposed solution could assign jobs to plants and trucks while maximizing both the job coverage and profit. Moreover, by allowing the first unload time to of a few jobs be either advanced or delayed by a few 10s of minutes, the proposed solution could further improve job coverage and profit as shown in Table 5.9. We further demonstrated that by relaxing the constraints that a RMC truck must return to its loading plant, we could further improve both objectives by allowing trucks to feely move across plants based on job requirements.

Table 5-9. Comparison of proposed solution (increment of average job coverage and profit) vs. manual job allocation (traditional scenario).

	Simulated Annealing Job Scheduler			**Discrete Particle Swarm Optimization Job Scheduler (Fixed Plant)**
	Fixed Plant	**Fixed Plant (With Time Window)**	**Free to Move**	
Average Job Coverage	13%	21%	16%	6%
Profit	9%	13%	14%	3%

We also compared the results of the proposed solution with the Manual Job Allocation (Best Case Scenario). Table 5.10 summarizes the profit and the average job coverage variation compared to the Manual Job Allocation (Best Case Scenario) showing how close the results of SA and DPSO Job Schedulers to the results of the Manual Job Allocation (Best Case Scenario). In comparison, Fixed Plant (With Time Window) Scenario which uses the SA job scheduler gives the closest results to the Manual Job Allocation (Best Case Scenario) where the results of DPSO is the farthest to the Manual Job Allocation (Best Case Scenario) as the average job coverage and profit varies up to 16% and 11% respectively.

Table 5-10. Comparison of proposed solution (variation of average job coverage and profit) vs. manual job allocation (best case scenario).

	Simulated Annealing Job Scheduler			**Discrete Particle Swarm Optimization Job Scheduler (Fixed Plant)**
	Fixed Plant	**Fixed Plant (With Time Window)**	**Free to Move**	
Average Job Coverage	9%	1%	6%	16%
Profit	6%	2%	1%	11%

6. SUMMARY AND FUTURE WORK

Section 6.1 presents the conclusion of the research while Section 6.2 elaborates the limitations of the research. Future work including the suggestions to expand our work is presented in Section 6.3.

6.1 Conclusion

In this research, we proposed a rule engine and Simulated Annealing (SA) based automated solution to schedule RMC trucks. We considered an environment where jobs are scheduled in the previous evening based on a set of job, plant, truck, and environmental constraints.

Genetic algorithms are applied in combined discrete-event simulation in HKCONSIM to model and further optimize the one plant-multisite RMC plant operations in Hong Kong [19]. Further, Feng et al. [20] and Liuhenyuan et al. [3] introduced a solution where better optimization could be achieved by a Genetic Algorithm (GA) while focusing on scheduling RMC production across an environment with a single plant and single mixer where they further suggested to focus on multiple plant condition as a crucial research gap to be filled whereas our work focusses on multiple plants, trucks, and construction sites.

A systematic approach including a mathematic model for the scheduling of dispatching RMC trucks using an improved Discrete Particle Swarm Optimization (DPSO) was discussed in [27]. Authors suggested that a more dynamic approach may be useful to deal with the uncertainty if some emergencies appear due to the environmental factors to improve the effectiveness of the scheduling.

Since the route and vehicle scheduling problems are known to be NP-hard, we cannot get an optimal solution within polynomial time [6], [7], [8], [9]. Therefore, it is crucial to identify suitable heuristic-based solutions that can still maximize the customer satisfaction, efficiency, and company profit.

Performance analysis based on a workload derived from a real RMC delivery company with 158 jobs and 735 truckloads shows that the proposed solution could assign jobs to plants and trucks while maximizing both the job coverage and profit. Moreover, by allowing the first unload time to of a few jobs be either advanced or delayed by a few 10s of minutes, the proposed solution could further improve job coverage and profit. We further demonstrated that by allowing trucks to feely move across plants we could further improve both objectives.

Our solution consists of a rule checker that enforces the constraints and a job scheduler based on Simulated Annealing (SA). SA algorithm schedules RMC trucks by considering the trends and patterns observed using the past and present data. Our study emphasizes on focusing on multiple plants, trucks and construction sites whereas related work focuses only on a single plant condition. Moreover, we support the case of allowing RMC trucks to feely move across plants based on job requirements (without being restricted to a home plant), as far as it leads to better job coverage and profit. Our analysis based on a workload derived from a real RMC company revealed that both the job coverage and profit can be maximized, compared to typical manual scheduling. Also, we compared the performance of the SA job scheduler with a DPSO-based job scheduler. For example, compared to manual job scheduling, the proposed solution increases the average job coverage and profitability by 13% and 9%, respectively. Results also shown that the DPSO can improve both job coverage and profit by 6% and 3% compared to the typical manual scheduling. Furthermore, both job coverage and profit further improve by 16% and 14%, respectively compared to the typical manual scheduling when the trucks can freely move access plants as per job requirements. Moreover, it eliminates the error-prone and labor-intensive resource allocation [4] and enables experts in RMC dispatching rooms to focus more on the exceptions such as truck breakdowns or last-minute changes by construction sites.

6.2 Research Limitations

The solution was limited by the errors of the fuel sensors fixed in RMC trucks. Therefore, the analysis could not have extended towards fuel usage prediction aspects while scheduling the RMC trucks. Our analysis was restricted by the limited data provided by the RMC company. Also, we had to remove three trucks from our descriptive analysis due to the errors occurred in the GPS and fuel sensors. Furthermore, significant number of trips were removed from the descriptive analysis due to the sensor malfunction. Moreover, our analysis was completely based on the dataset as we did not have access to see the actual order book of RMC delivery jobs.

Moreover, we had only two plants to conduct our descriptive analysis and it affected us in understanding the empty truck movement from plant to plant, as we have proposed a new scheduling scenario where truck can move freely among plants. In conclusion, our data set was limited to one month and we could have reach a better conclusion in our descriptive analysis and final solution if more recent and accurate data was provided to conduct the research.

Our solution did not capture vehicle breakdowns, weather and traffic conditions of the route which may also increase the number of constraints for the rule checker. Setting up constraints for weather and traffic condition may also lead the scheduling process to a complex scenario as both conditions are hard to predict with the available data.

We assume that every evening, the next day's schedule is determined based on the already confirmed jobs and available plants and trucks as there is an ambiguity upon the job end time due to the delays at the job site and during travel, which depends on traffic and other environmental conditions such as weather and road closures that are hard to predict. Moreover, the route is planned to utilize the full truckload to avoid wastage and dead runs of the truck. Therefore, dynamic scheduling to overcome last-minute changes is left as future work as it needs more sophisticated rule checker and job scheduler to capture the last-minute changes in the job schedule.

6.3 Future Work

We proposed solution to schedule RMC trucks and considered an environment where jobs are scheduled in the previous evening. Therefore, the solution can be extended towards real-time scheduling where the last-minute job is accommodated the truck/plant schedule. Moreover, the proposed solution can be further extended to tolerate unexpected delays in the process, capture last-minute delivery requests arriving within the day, as well as to tolerate unexpected events such as accidents, breakdowns, and traffic. Furthermore, accommodating last-minute changes will increase the number of constraints for the rule checker and leads the job scheduler to run the SA algorithm while unchanging the ongoing jobs and rescheduling the other jobs to accommodate the last-minute changes in the delivery schedule.

Additionally, the analysis can be extended towards fuel prediction by accommodating fuel sensor data to reduce the vehicle operating cost significantly. Beside the above suggestions, the solution can be used in predicting the additional truck requirement for the company to achieve 100% job coverage which ultimately affects the profit of the company. Moreover, it would be useful to introduce a priority to long-term customers to ensure that their jobs are covered with high certainty as retention of key customers are essential for long-term sustainability.

REFERENCES

[1] M. Lu and H. C. Lam, "Simulation-optimization integrated approach to planning ready mixed concrete production and delivery: Validation and applications," in *Winter Simulation Conf., Dec.* 2009, pp. 2593–2604, 2009.

[2] B. D. Hettiarachchi, H. M. N. D. Bandara, and N. A. Samarasekera, "Automated Multi-Plant Scheduling of Ready-Mixed Concrete Trucks," in *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Aug. 2018*, 2018, pp. 43–48.

[3] Z. Liu, Y. Zhang, and M. Li, "Integrated scheduling of ready-mixed concrete production and delivery," *Automation in Construction*, vol. 48, pp. 31–43, 2014.

[4] S. Wickramanayake and H. M. N. Dilum Bandara, "Fuel consumption prediction of fleet vehicles using Machine Learning: A comparative study," in *2016 Moratuwa Engineering Research Conference (MERCon), Apr. 2016*, 2016, pp. 90–95.

[5] N. Mayteekrieangkrai and W. Wongthatsanekorn, "Optimized ready mixed concrete truck scheduling for uncertain factors using bee algorithm," *Songklanakarin Journal of Science Technology*, vol. 37, no. 2, pp. 221–230, 2015.

[6] A. Wren, S. Fores, A. Kwan, R. Kwan, M. Parker, and L. Proll, "A flexible system for scheduling drivers," *Journal of Scheduling*, vol. 6, no. 5, pp. 437–455, 2003.

[7] S. Fores, L. Proll, and A. Wren, "An improved ILP system for driver scheduling," *Computer-Aided Transit Scheduling*, no. 1988, pp. 43–61, 1999.

[8] M. Maghrebi and C. Sammut, "Using Column Generation for Solving Large Scale Concrete Dispatching Problems," Sydney, 2013.

[9] B. Laurent and J. K. Hao, "Simultaneous vehicle and driver scheduling: A case study in a limousine rental company," *Computers & Industrial Engineering*, vol. 53, no. 3, pp. 542–558, 2007.

[10] S. Biruk, "Dispatching concrete trucks using simulation method," *Budownictwo i Architektura*, vol. 14, no. 2, pp. 5–10, 2015.

[11] Autthapon Sirisuwan; Ladda Tanwanichku, "Optimization of Scheduling and Dispatching RMC Truck using Genetic Algorithms for One plant Multi sites Genetic Algorithms for One plant – Multi sites," in *6th Regional Symposium on Infrastructure Development*, 2014, no. March, pp. 1–7.

[12] S. Bergthaler, "Delivery of Ready-Mixed Concrete in a Dynamic Environment," University of Vienna, 2010.

[13] A. Hill and J. W. Böse, "A decision support system for improved resource planning and truck routing at logistic nodes," *Information Technology Management*, vol. 18, no. 3, pp. 241–251, 2017.

[14] R. Bishop, D. Bevly, J. Switkes, and L. Park, "Results of Initial Test and Evaluation of a Driver - Assistive Truck Platooning Prototype," in *2014 IEEE Intelligent Vehicles Symposium, June 2014*, p. 100.

[15] S. Verwer, M. De Weerdt, and C. Witteveen, "Learning driving behavior by timed syntactic pattern recognition," in *IJCAI International Joint Conference on Artificial Intelligence*, *July 2011*, pp. 1529–1534.

[16] Y. Xiao and A. Konak, "A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness," *Applied Soft Computing Journal*, vol. 34, pp. 372–388, 2015.

[17] S. Yan, W. Lai, and M. Chen, "Production scheduling and truck dispatching of ready mixed concrete," *Transportation Research Part E Logistics and Transportation Review*, vol. 44, no. 1, pp. 164–179, 2008.

[18] Maghrebi Mojtaba, Claude Sammut, and S. Travis Waller, "Feasibility Study of Automatically Performing the Concrete Delivery Dispatching Through Machine Learning Techniques," *Eng. Constr. Archit. Manag.*, vol. 22, no. 5, pp. 573–590, 2015.

[19] M. Lu and H.-C. Lam, "Optimized Concrete Delivery Scheduling Using Combined Simulation and Genetic Algorithms," in *Winter Simulation Conference, Dec. 2005*, no. 1, pp. 202–208.

[20] C. W. Feng, T. M. Cheng, and H. T. Wu, "Optimizing the schedule of dispatching RMC trucks through genetic algorithms," *Automation in Construction*, vol. 13, no. 3, pp. 327–340, 2004.

[21] E. H. L. Aarts, J. H. M. Korst, and P. J. M. Van Laarhoven, "Simulated Annealing," in *Local Search in Combinatorial Optimization*, E. Aarts and J. K. Lenstra, Eds. John Wiley & Sons Ltd., 1997, pp. 91–120.

[22] R. W. Eglese, "Simulated annealing: A tool for operational research," *European Journal of Operations Research*, vol. 46, no. 3, pp. 271–281, 1990.

[23] S. Anily and A. Federgruen, "Simulated Annealing Methods With General Acceptance Probabilities.," *Journal of Applied Probability*, vol. 24, no. 3, pp. 657–667, 1987.

[24] A. P. Adewole, "A Comparative Study of Simulated Annealing and Genetic Algorithm for Solving the Travelling Salesman Problem," *International Journal of Applied Information Systems*, vol. 4, no. 4, pp. 6–12, 2012.

[25] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[26] J. Kennedy and R. C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," pp. 4–8, 1997.

[27] P. Liu, L. Wang, X. Ding, and X. Gao, “Scheduling of Dispatching Ready Mixed Concrete Trucks Trough Discrete Particle Swarm Optimization Pan,” in *IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Oct. 2010*, pp. 4086–4090.

[28] C. A. Silva, J. M. Faria, P. Abrantes, J. M. C. Sousa, M. Surico, and D. Naso, “Concrete Delivery using a combination of GA and ACO,” in *44th IEEE Conference on Decision and Control, and the European Control Conference Dec. 2005*, pp. 7633–7638.

[29] D. G. Daniel and C. L. Lobo, *User's Guide to ASTM Specification C 94 on Ready-Mixed Concrete*. 2005.

[30] “Google Maps Distance Matrix API | Google Developers.” [Online]. Available: https://developers.google.com/maps/documentation/distance-matrix/. [Accessed: 10-Feb-2018].

[31] S.R. Muramudalige and H.M.N.D. Bandara, “Automated Driver Scheduling for Vehicle Delivery,” in *Kováčiková T., Buzna Ľ., Pourhashem G., Lugano G., Cornet Y., Lugano N. (eds) Intelligent Transport Systems – From Research and Development to the Market Uptake (INTSYS 2017), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 222, pp. 215-225, Nov. 2017, Springer, Cham.

[32] M. Maghrebi and S. Travis Waller, “Exploring Experts Decisions in Concrete Delivery Dispatching Systems Using Bayesian Network Learning Techniques,” in *2nd International Conference on Artificial Intelligence, Modelling, and Simulation, Nov. 2014*, 2014, pp. 103–108.

[33] D. Zhai, F. Zhang, B. Gao, W. Han, T. Zhang, and J. Zhang, “Ant colony algorithm and simulated annealing algorithm based process route optimization,” in *2nd International Conference on Enterprise Systems, Oct. 2014*, pp. 102–107.

[34] O. Catoni, “Solving Scheduling Problems by Simulated Annealing,” *SIAM Journal on Control and Optimization*, vol. 36, no. 5, pp. 1539–1575, 1998.

[35] F. Busetti, “Simulated annealing overview,” 2003.

[36] C. Tsallis and D. A. Stariolo, “Generalized Simulated Annealing,” in *Computational Optimization in Engineering - Paradigms and Applications*, 1995.

[37] Y. Nourani and B. Andresen, “A comparison of simulated annealing cooling strategies,” *Journal of Physics A: Mathematical and General*, vol. 31, no. 41, pp. 8373–8385, 1998.

[38] A. H. Kashan and B. Karimi, “Computers & Industrial Engineering A discrete particle swarm optimization algorithm for scheduling parallel machines,” *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 216–223, 2009.