Using Column Generation for Solving Large Scale Concrete Dispatching Problems

Mojtaba Maghrebi[1] Vivek Periaraj[2] S. Travis Waller[3] Claude Sammut[1]

[1] University of New South Wales, Australia
{mojtabam,claude}@cse.unsw.edu.au
[2] University of Arizona, USA
vivek.periaraj@gmail.com
[3] University of New South Wales and NICTA, Australia
s.waller@unsw.edu.au

Technical Report
UNSW-CSE-TR-201334
November 2013

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Ready Mix Concrete (RMC) dispatching forms a critical component of the construction supply chain. However, optimization approaches within the RMC dispatching continue to evolve due to the specific size, constraints and objectives required of the application domain. In this paper, we develop a column generation algorithm for Vehicle Routing Problems with time window constraints as applied to RMC dispatching problems and examine the performance of the approach for this specific application domain. The objective of the problem is to find the minimum cost routes for a fleet of capacitated vehicles serving concrete to customers with known demand from depots within the allowable time window. The VRP is specified to cover the concrete delivery problem by adding additional constraints that reflect real situations. The introduced model is amenable to the Dantzig-Wolfe reformulation for solving pricing problems using a two-staged methodology as proposed in this paper. Further, under the mild assumption of homogeneity of the vehicles, the pricing sub-problem can be viewed as a minimum-cost multi-commodity flow problem (MMCF) and solved in polynomial time using efficient network simplex method implementations. A large-scale field collect dataset is used for evaluating the model and the proposed solution method, with and without time window constraints. In addition, the method is compared with the exact solution found via enumeration. The results show that on average the proposed methodology attains near optimal solutions for many of the large sized models but is 10 times faster than branch-and-cut.

1 Introduction

Although Ready Mixed Concrete (RMC) dispatching is a common practical need within the construction industry, optimization methods continue to evolve within the application domain. Often, the previously proposed optimization approaches have pursued either:(i) Integer Programming (IP)and Mixed-Integer Programming (MIP) approaches, which have difficulty with large problem sizes or (ii) meta-heuristic approaches which can solve larger problems but tend to lack optimality or, at the least, bounding properties. This paper examines the specific Vehicle Routing Problem (VRP) variation defined by the RMC dispatching industry, develops a tailored solution method via Column Generation (CG) with bounding properties, and examines the performance of this method within the application domain on field data. Specifically, the RMC dispatching problem consists of delivering a specified amount of concrete to customers from depots using capacitated trucks. At each location in the transportation network, the trucks are expected to start and leave within the specified time window that is required to load and unload the concrete. Further, a penalty is incurred for not delivering concrete to a customer. In addition, the model presents the added constraints of ensuring that at most one of the customers is served by the trucks from at most one of the depots. Different trucks incur the same travelling time between depots and customers. However, there could be different travel times between the start and depot locations and between customer and finish locations. Moreover, the time window constraints are a function of the location rather than the vehicles, with each vehicle incurring the same service times at a given location. Assuming all the vehicles can fulfill the demand at a customer location, the fleet of vehicles can be considered homogenous. A tour of a truck is the sequence of locations it visits from the start to the finish. A minimally traversing tour consists of a start, a depot, a customer and a finish location. A single truck could also serve multiple customers from multiple depots. A set of tours of the trucks in the network is feasible, however only when the following conditions are met: (i) all locations visited by a truck are in sequence, (ii) at most one of the customers is served, (iii) at most one of the depots is used in the delivery and (iv) the time window requirements at all locations are satisfied. Acquiring a near optimum solution for RMC dispatching problems is a challenging supply chain issue. In large scale metropolitan areas, the RMC dispatching problems cannot be solved optimally due to the intractability of the Vehicle Routing Problem (VRP) given the aforementioned constraints and considerations. In other words, the optimum solution of the problem in a polynomial time is computationally intractable. To overcome this issue within the domain of RMC, this paper employs the column generation mathematical technique. Column generation creates solution iteratively, and then forms convex combinations to achieve feasibility. The proposed method facilitates the examination of RMC dispatching problems in an optimization setting which has not previously been possible for this particular domain. This paper consists of four sections. In first section, the relevant literature in this area is reviewed. Section two covers column generation and reformulating steps. In section three, the results with the field data set are presented and the proposed method is compared in practice with the results from branch-and-cut; and in the last section the achieved results are discussed and conclusions drawn.

2 Literature Survey

Several attempts have been made to model the dispatching and particularly RMC dispatching effectively, such as [1],[8],[19],[22]. It has been proved that an RMC optimization problem is an NP-hard problem [1],[27], [18] . Therefore, to deal with this problem, heuristic methods have been widely used in the literature such as [3],[5],[9], [10], [17]. Despite developments in this area, the solution structure among most introduced methods is quite similar, especially in the Genetic Algorithm (GA) based method where the chromosome structure consists of two merged parts: the first part defines the sources of deliveries; the second part expresses the priorities of customers. In the literature, in addition to GA other approaches have also been studied that will be discussed briefly in the text that follows. [27] introduced a numerical method for solving the RMC optimization problem by cutting the solution space and incorporating the branch and bound technique and the linear programming method. [26] used decomposition and relaxation techniques coupled with a mathematical solver to solve the problem, and [20] applied Variable Neighborhood Search (VNS) to deal with RMC optimization problems. [1] made the mathematical modelling much simpler by dividing the depots and customers into sub-depots and sub-customers. However, Column Generation techniques have not been used particularly when it is coupled with Dantzig-Wolfe. Since the time Dantzig-Wolfe [6] proposed the principles of the decomposition of linear programs, the method has been applied to a variety of combinatorial integer programs with great success. Many of the models found in various applications are amenable to the Dantzig-Wolfe reformulation. In particular, column generation has been successfully applied to different types of vehicle routing problems. [14] applies column generation and branch and price algorithms for VRP problems in the presence of soft time window constraints. The pricing problem in their model is a resource-constrained shortest path problem which is an NP hard problem and a bi-directional dynamic programming algorithm was used to solve it optimally. [11] presents a column generation heuristic for general heterogeneous VRP problems with time windows. Their model consists of vehicles with different capacities and incurs different travel times between locations. Several authors discuss methods for obtaining reduced costs in the context of the Dantzig-Wolfe reformulation of the master problem. [12] proposed a method to derive the reduced cost of the arcs from a path based reformulation of the Dantzig-Wolfe master problem. In this method, the reduced cost of an arc is computed as the minimum reduced cost of the path the arc uses. The path's reduced cost can be computed efficiently using a bi-directional search technique. [25] proposed a method where the dual variables for the linear relaxation of the compact formulation can be derived starting from the duals corresponding to the last simplex iteration of the master problem and the pricing sub-problems that are solved subsequently in the same iteration. The dual variables thus obtained are feasible and optimal to the linear relaxation of compact formulation as well. Dantzig-Wolfe reformulation also has been used in transportation particularly for dynamic assessment of traffic such as [16],[15],[4] but in this paper we only focus on VRP based problems.

3 Methodology

Column Generation (CG) is a common method for solving large-scale integer programs. First, it must be established that CG is applicable to RMC dispatching problems specifically. To examine the applicability of CG to RMC dispatching, we can consider two principles of column generation. First, a major proportion of the variables are non-basic at the optimal solution, hence it is required to generate only those columns whose reduced costs are negative. In the other words, CG deals only with those columns that are associated with providing the best improvement of the objective. Second, by applying branch-and-cut to the reduced problem, CG will lead to achieving improvements on the computing performance compared to applying branch-and-cut to the original problem. In column generation a sequence of master and pricing problems are solved. The master problems are the continuous relaxation of the original problem and consist of only a subset of columns to start with. They are also called restricted master problems. The pricing problem is the minimization of the reduced costs. The RMC problem can be viewed as a set of tours made by each truck. In each iteration, the tours that have the most negative reduced costs are selected and added to the restricted master problems. This process is repeated until no more columns can be generated or until any of the termination criteria is met. Then the branch-and-cut is applied to the original problem with only the generated columns. In this section the RMC dispatching problem is reformulated via the column generation technique and introduces a method for formulating RMC dispatching problems. The terminology used in this paper for modelling the original RMC formulation is similar to that of [1] The original RMC formulation assumes the dispatching problem is a graph in which depots and customers are nodes and a delivery is depicted by an arc between a depot and a customer. To retain the unity throughout the formulation and the algorithm, all required parameters are defined in follow:

C Set of customers
C_K Set of customers visited by a truck k
D Set of depots
D_k Set of depots visited by truck k
K Set of vehicles
U_S Set of starting points
V_f Set of ending points
S_u Service time at the depot u
t_{uvk} Travel time between u and v with vehicle k
q_k Maximum capacity of vehicle k
q_c Demand of customer c
w_u Time at node u
β_c Penalty of unsatisfying the customer c
M A large constant
γ Maximum time to haul the concrete
x_{uvk} 1 if route between u and v with vehicle k is selected, 0 otherwise
y_c 1 if total demand of customer c is supplied, 0 otherwise
z_{uvk} Cost of travel between u and v with vehicle k

$$minmize \sum_{u} \sum_{v} \sum_{k} z_{uvk} x_{uvk} - \sum_{c} \beta_c y_c \tag{3.1}$$

Subject to:

$$\sum_{u \in u_s} \sum_{v} \sum_{k} x_{uvk} = 1 \quad \vee k \in K \tag{3.2}$$

$$\sum_{u}\sum_{v\in v_f}\sum_{k} x_{uvk} = 1 \quad \vee\, k \in K \tag{3.3}$$

$$\sum_{u}\sum_{v} x_{uvk} - \sum_{v}\sum_{w} x_{vwk} = 0 \quad \vee\, k \in K, v \in C \cup D, \tag{3.4}$$

$$\sum_{u\in D}\sum_{k} x_{uvk} \leq 1 \quad \vee\, v \in C \tag{3.5}$$

$$\sum_{v\in C}\sum_{k} x_{uvk} \leq 1 \quad \vee\, u \in D \tag{3.6}$$

$$\sum_{u\in D}\sum_{k} q_k x_{uvk} \geq q_c y_c \quad \vee\, c, vv \in C \tag{3.7}$$

$$-M(1 - x_{uvk}) + s_u + t_{uvk} \leq w_v - w_u \quad \vee\, (u, v, k) \in E \tag{3.8}$$

$$M(1 - x_{uvk}) + \gamma + s_u + \geq w_v - w_u \quad \vee\, (u, v, k) \in E \tag{3.9}$$

$$x_{uvk} \in \{0, 1\} \;\; and \;\; y_c \in \{0, 1\} \tag{3.10}$$

3.1 Master Problem

The Dantzig-Wolfe decomposition as applied to integer programs is generally known to provide strong dual bounds as the feasible region of the master problem is a tighter formulation compared to that of linear relaxation. It's a well-known result in network flow theory that an extreme point $x_{uvk} \in x$ is also a path pP in the network. The natural choice for network flow problems is to consider a path based reformulation of the Dantzig-Wolfe master problem. [23] considers a reformulation of the master problem for VRP in which multiple vehicles are aggregated into a single problem with an extreme point representing a feasible route any vehicle could cover. In our reformulation framework, we retain the routes covered by individual vehicles. An extreme point in our model consists of truck tours and is a unique traversal in the network as governed by the constraints 3.5 and 3.6 which ensure that at most one of the depots and customers is served in the path. Thus, the compact formulation is decomposable by truck tours. Constraints 3.2, 3.3, 3.4, 3.8, and 3.9have block diagonal structures with respect to trucks while constraints 3.53.7 are the coupling constraints with variables associated with all the trucks. Each truck tour can be equivalently expressed as follows:

$$\lambda_k \leq \sum_{u}\sum_{v} x_{uvk} \tag{3.11}$$

The cost coefficient of each truck tour is defined as the duration of the truck's tour in the network path and is expressed as:

$$z_k = \sum_{u}\sum_{v} z_{uvk} x_{uvk} - \sum_{c\in C_k} \beta_c y_c \tag{3.12}$$

To achieve the Dantzig-Wolfe restricted master formulation, the compact formulation can then be reformulated in terms of truck tours as:

$$minimize \sum_k \sum_p z_p^k \lambda_p^k \tag{3.13}$$

Subject to:

$$\sum_{k \in C_k} \sum_p \lambda_p^k \leq 1 \quad \vee\, C_k \in C \tag{3.14}$$

$$\sum_{k \in D_k} \sum_p \lambda_p^k \leq 1 \quad \vee\, D_k \in D \tag{3.15}$$

$$\sum_{k \in C_k} \sum_p q_k \lambda_p^k \geq q_c \quad \vee\, C_k \in C \tag{3.16}$$

$$\sum_p \lambda_p^k = 1 \quad \vee\, k \in K \tag{3.17}$$

$$\lambda_p^k \geq 0 \quad \vee\, k \in K, p \in P \tag{3.18}$$

The above formulation is also called the extensive formulation. Each truck tour $x_{uvk} \in P$ can be represented as the convex combination of truck tours through the convexity constraints (17). In the presence of the linking constraints 3.11 between λ_k^p and x_{uvk}, the optimal solution of the Dantzig-Wolfe restricted master problem λ_k^{p*} can be used to recover the solution to the compact formulation when $\{0, 1\}$. However, when the linking constraints are removed and λ_k^p is relaxed, the optimal solution of the Dantzig-Wolfe restricted master problem forms the primal bound for the compact formulation. From each solution of the pricing problem, an extreme point is added to the extensive formulation which is indexed as p. We let the duals corresponding to the constraints 3.14, 3.15 and 3.16 to be π and the dual corresponding to the convexity constraint of truck k to be σ_k.

3.2 Computation of Reduced Costs

As discussed in the literature review, there have been a few studies related to computing the reduced costs of the variables in the compact formulation when the Dantzig-Wolfe decomposition is applied. For instance, [7] proposes a method to formulate an explicit Dantzig-Wolfe master called Explicit Master that retains the linking constraints 3.11 between the λ_k^p and x_{uvk}. From each solution of Explicit Master to optimality, the reduced costs for the variables in the compact formulation can be directly obtained from the optimal duals corresponding to the constraints 3.14,3.15 and 3.16. In our column generation methodology, the pricing problem is solved in two stages with the stage 1 formulation being a linear program at a reduced dimension relative to the compact formulation and the stage 2 formulation being a mixed integer program. We obtain a dual vector of the compact formulation from the optimal dual solutions of the Dantzig-Wolfe restricted master problem and from the linear relaxation of a newly formulated problem called the auxiliary restricted master problem. The auxiliary restricted master problem formulation is identical to that of the compact formulation but consists of only the generated variables until that point and thus forms the dual bound to the compact formulation. The duals corresponding to the constraints 3.2, 3.3, 3.4, 3.8, 3.9 and 3.10 obtained from the auxiliary restricted master problem are denoted by μ. If A_1 is the

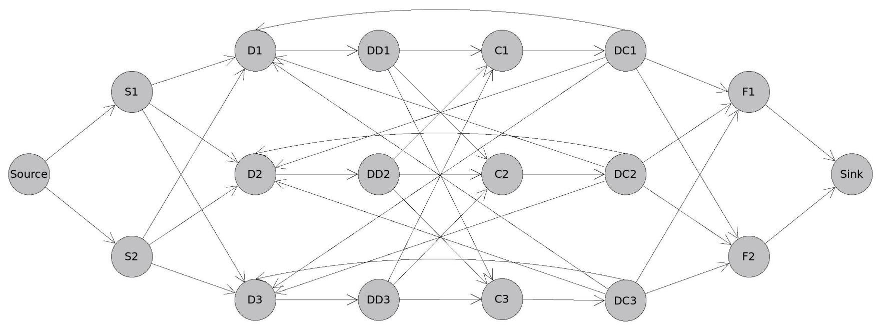

Figure 3.1: Network Graph

constraint coefficient matrix of the auxiliary restricted master problem and A_2 is the constraint coefficient matrix of the Dantzig-Wolfe restricted master problem, then the reduced cost of a variable of the compact formulation is computed as follows:

$$
\begin{aligned}
rc_{uvk} &= z_{uvk} - A_1\mu \quad && \forall\, u \in u_s, v \in D, k \in K \\
rc_{uvk} &= z_{uvk} - A_1\mu \quad && \forall\, u \in C, v \in v_f, k \in K \\
rc_{uvk} &= z_{uvk} - A_1\mu \quad && \forall\, u \in C, v \in D, k \in K \\
rc_{uvk} &= z_{uvk} - A_1\mu - A_2\pi \quad && \forall\, u \in D, v \in C, k \in K \\
rc_{uvk} &= \beta_c - A_2\pi \quad && \forall\, u \in u_s, v \in D, k \in K
\end{aligned}
\tag{3.19}
$$

3.3 Pricing Problem

The pricing problem is solved in two stages. In stage 1, the sum of the reduced costs of a transformed problem is minimized and in stage 2, the optimal assignments corresponding to the original problem are obtained. A small RMC network is depicted in Figure 1. The RMC model can be considered homogenous with all of the trucks incurring the same time to travel between depots and customers. The stage 1 network consists of start nodes, depot nodes, customer nodes and finish nodes. The network is constructed with the source node connecting to all start nodes, the start nodes connecting to all depot nodes, the depot nodes connecting to all customer nodes, the customer nodes connecting to all depot nodes and the depot nodes connecting to all finish nodes. Finally, all finish nodes are connected to a sink (Figure 3.1).

The dummy nodes at the depots (DD) ensure that at most one of the depots is assigned and the dummy nodes at the customers (DC) ensure that at most one of the customers is assigned, satisfying constraints 3.5 and 3.6(5) of the compact formulation. The supply at the source and demand at the sink is set to the number of trucks in the network. The lower bound and upper bounds on the arcs connecting the nodes are set to 0 and 1 respectively. The time feasibility at various nodes is maintained by changing the capacity on the arcs connecting the nodes. If any of the time constraints are not satisfied on an arc, then the upper bound on the arc's capacity is set to 0. The cost on the arc is set to the minimum of the reduced costs of different trucks that use the arc. Thus, the stage 1 pricing problem can be viewed as a minimum-cost multi-commodity flow problem (MMCF) where the objective is to find the optimal routes for identical trucks in the network that satisfy the flow and demand requirements such that the sum of the minimum of reduced costs is minimized.

Stage 1 Formulation

The MMCF pricing problem can be formally stated as follows. Given a flow network $G(V, E)$, where edge $(u, v) \in E$ has capacity C_{uv}, there are k identical commodities, defined by $K = (s, t, d)$ where s and t are the source and sink of commodity and d is the demand. The flow of a commodity along edge (u, v) is f_{uv} .

$$minimize \sum a_{uv} f_{uv} \tag{3.20}$$

Subject to:

$$f_{uv} \leq c_{uv} \quad \vee\, c_{uv} \tag{3.21}$$

$$f_{uv} - \sum_{w \in V} f_{wu} = 0 \quad \vee\, u \in V, v \in V, u, v \notin s, t \tag{3.22}$$

$$\sum_{w \in V} f_{uv} = \sum w \in V f_{wt} = d \tag{3.23}$$

$$c_{uv} = 1 \quad \vee\, u \in V, v \in V, u \notin s, \notin t \tag{3.24}$$

$$c_{uv} = 0 \quad \vee\, u \in V, v \in V, and\, if\; E(u, v)\; is\; not\; feasible \tag{3.25}$$

$$a_{uv} = \underbrace{MIN}_{k \in K} rc_{uvk} \tag{3.26}$$

$$a_{sw} = a_{wt} = 0 \quad \vee\, w \in V \tag{3.27}$$

In network flow problems, the basic solutions are computed without any multiplication or division and the following theorem arises from this property. The theorem states that for flow problems with integer supplies and demand, every basic feasible solution and every basic optimal solution assigns integer flow to every arc [24]. If the objective function of a minimum cost flow problem is bounded from below on the feasible region, the problem has a feasible solution, and if the vectors b, l and u are integers, then the problem has at least one integer optimum solution.

$$Minimize \{cx : Ax = b, l \leq x \leq u\} \tag{3.28}$$

Since the demand and the lower and upper bound on the capacity of the arcs in the MMCF network are integers, the solution from the MMCF pricing problem is also integer. The MMCF pricing problem is solved using the primal network simplex method. Efficient implementation of the network simplex method is known to have polynomial time complexity. If m is the number of arcs in the network, n is the number of nodes in the network, C is the maximum cost on the arcs in the network and U is the maximum capacity on the arcs in the network, then the time complexity of a generic implementation of the network simplex method [13] is given by $((m + n)mnC^2U)$. The time complexity of the MMCF problem as applied to the RMC network is given by $((m + n)mnC^2)$. Table 4.2 lists average solution times of the pricing problem across different instances.

Stage 2 Formulation

The solution obtained from the MMCF pricing problem (C^*, D^*) is transformed to the original problem dimension by solving a mixed integer program that optimizes the assignments across different trucks. While the MMCF pricing problem may result in tours that are infeasible with respect to the demand constraints 3.7 in particular, the stage 2 pricing formulation ensures that the final tours are feasible with respect to all the constraints of the compact formulation. Each feasible solution thus obtained from the stage 2 pricing problem forms an extreme point to the compact formulation.

$$minmize \sum_{u} \sum_{v} \sum_{k} rc_{uvk} x_{uvk} + \sum_{c \in C} rc_c y_c + \sum_{k \in K} \sigma_k \tag{3.29}$$

Subject to:

$$\sum_{u \in u_s} \sum_{v} \sum_{k} x_{uvk} = 1 \quad \vee\, k \in K \tag{3.30}$$

$$\sum_{u} \sum_{v \in v_f} \sum_{k} x_{uvk} = 1 \quad \vee\, k \in K \tag{3.31}$$

$$\sum_{u} \sum_{v} x_{uvk} - \sum_{v} \sum_{w} x_{vwk} = 0 \quad \vee\, k \in K, v \in C \cup D, \tag{3.32}$$

$$\sum_{u \in D} \sum_{k} x_{uvk} \leq 1 \quad \vee\, v \in C \tag{3.33}$$

$$\sum_{v \in C} \sum_{k} x_{uvk} \leq 1 \quad \vee\, u \in D \tag{3.34}$$

$$\sum_{u \in D} \sum_{k} q_k x_{uvk} \geq q_c y_c \quad \vee\, c, vv \in C \tag{3.35}$$

$$-M(1 - x_{uvk}) + s_u + t_{uvk} \leq w_v - w_u \quad \vee\, (u, v, k) \in E \tag{3.36}$$

$$M(1 - x_{uvk}) + \gamma + s_u + \geq w_v - w_u \quad \vee\, (u, v, k) \in E \tag{3.37}$$

$$x_{uvk} \in \{0, 1\} \;\; and \;\; y_c \in \{0, 1\} \tag{3.38}$$

$$0 \leq x_{uvk} \leq 1, \quad if \;\vee\, u \in D^*, v \in C^*, k \in K \tag{3.39}$$

$$0 \leq x_{uvk} \leq 1, \quad else\, if \;\vee\, u \in C^*, v \in D^*, k \in K \tag{3.40}$$

$$0 \leq x_{uvk} \leq 1, \quad else\, if \;\vee\, u \in u_s, v \in D^*, k \in K \tag{3.41}$$

$$0 \leq x_{uvk} \leq 1, \quad else\, if \;\vee\, u \in C^*, v \in v_f, k \in K \tag{3.42}$$

$$0 \leq x_{uvk} \leq 0, \quad otherwise \tag{3.43}$$

Multiple Column Generation

The column generation scheme we adopt generates many columns in each solution of the master and pricing problems. Traditionally, the approach has been to generate and add a single column with the most negative reduced cost to the restricted master problems. [7] discuss a scenario of multiple column generation when the constraints of the master problem are nicely structured. The constraints 3.14,3.15 and 3.16 of our master problem are consistent with the multiple column generation approach. Our motivation behind this scheme is also to ensure that the columns that are generated form feasible truck tours. This can be viewed as generating the best cost improving truck tours out of many possible ones. From each pricing problem solution, truck tours with a column that satisfies the minimum reduced cost threshold are generated in addition to including an extreme point to the Dantzig-Wolfe restricted master problem. This scheme also has the added advantage of exploiting many of the solution improving heuristics that are available with most of the modern branch-and-cut solutions. Some of these heuristics employ methods which make minor changes to the solution vector in order to attain vastly improved solutions in a short time. This is especially effective in routing problems where a swap of nodes between the routes could result in a better solution. The master problems are again solved to optimality whose duals are used in the next pricing problem solution. This process is continued until no more negative reduced cost tours can be generated or when any of the termination criteria is met. Due to the potentially large time required to reach the zero reduced cost threshold for larger models, the column generation phase is terminated within the specified number of iterations. The column generation is also terminated when the optimal solution of the Dantzig-Wolfe restricted master problem (the primal bound) is within the specified tolerance of the optimal solution of the auxiliary restricted master problem (the dual bound). In the final phase, branch-and-cut is applied to the original problem with only the generated columns from the column generation phase.

4 Results

The proposed column generation algorithm was tested on actual field instances of wide ranging transportation networks delivering to up to 197 customers per day. Note, smaller networks were used to test the theoretical convergence properties. The field data that was used here belongs to an active RMC network in Adelaide (Australia). 9 instances were selected randomly from the available database which characterizes the selected instances as given in (Table. 4.1). The algorithm was developed in C++ and tested on a RedHat(R) CentOS(R)5.9 Linux server with 8 3.60GHz Intel(R) Xeon(R) CPUs and a 188 GB physical memory. The IBM CPLEX version 12.5.0.0 with parallel optimizers using up to 8 threads was used in the study. We found the solution polishing heuristics [21] available with the CPLEX mixed integer optimizer to be particularly effective in finding improved solutions for larger sized models with time window constraints. The heuristic was applied to the best solution attained from branch-and-cut which was terminated when the EP gap of 1% was achieved or when the time limit was reached. The EP gap was calculated according to 4.1.

$$EP = \frac{|Best\ Integer\ Solution - Best\ Dual\ Bound|}{10^{-10} + |Best\ Integer\ Solution|} \tag{4.1}$$

'Barrier/Dual' [2] is selected to solve the Dantzig-Wolfe restricted master prob-

Table 4.1: Problem data attributes

Instance ID	Variables	Constraints	# K	# U_s	# D	# C	# V_f	Start to Depot	Depot to Customer	Customer to Depot	Customer to Finish
Ade_30	138,486	199,108	17	9	114	30	9	17,442	58,140	58,140	4,590
Ade_40	166,684	241,280	17	9	106	40	9	16,218	72,080	72,080	6,120
Ade_47	367,168	535,188	20	9	174	47	9	31,320	163,560	163,560	8,460
Ade_53	480,924	702,564	24	9	170	53	9	36,720	216,240	216,240	11,448
Ade_63	917,249	1,346,014	29	9	230	63	9	60,030	420,210	420,210	16,443
Ade_93	1,499,846	2,212,774	32	9	236	93	9	67,968	702,336	702,336	26,784
Ade_112	2,343,152	3,467,646	31	9	320	112	9	89,280	1,111,040	1,111,040	31,248
Ade_153	3,299,695	4,895,476	33	9	313	153	9	92,961	1,580,337	1,580,337	45,441
Ade_197	5,790,391	8,607,378	41	9	346	197	9	127,674	2,794,642	2,794,642	72,693
Average	1,667,066	2,467,492	27	9	223	88	9	59,957	790,954	790,954	24,803

lem and auxiliary master problem for models with and without a time window. 'Barrier/Dual' is the hybrid optimizer with barrier as the primary LP solver with dual simplex used for crossover. 'Barrier' is the LP solver without crossover. 'Primal' is the primal simplex LP solver. Column Generation is terminated when: (i) no more tours with negative reduced cost column are found or (ii) the difference between the primal and dual bound is within the tolerance or (iii) the maximum number of iterations is reached. The termination criteria for B&C of the compact formulation with generated columns and IP/MIP is E-06. In addition, the starting criteria for polishing in B&C of the compact formulation with generated columns and MIP is 1.00E-2 which is applied to instances with more than 100 deliveries. (Table 4.2) compares the solution times of the stage 1 and stage 2 pricing sub-problems for models with and without time window constraints. In (Table 4.3) the achieved results from the proposed Column Generation model are compared with MIP when the time window is allowed. Similarly (Table 4.4) shows this when the time window is not permitted. Ade_197, which is the largest instance with MIP, is not solvable with the given computational resources; therefore the relevant cells in (Table. 4.3) are filled by NA (Not Applicable). From the evaluation data, it was found that the compact formulation consists of demand constraints and there were situations where the tours that were generated from the stage 2 pricing problem were infeasible with respect to demand constraints. A reformulation of the master problems that eliminated variables Y_c was found to be effective in pricing tours that are feasible with respect to demand constraints. In addition, out of all the instances evaluated, the assumption of homogeneity of vehicles held well for the majority of them. To cite one particular test case (Ade_53), the model consisted of two customer locations (c11 and c12) with a demand of 11 tonnes each and could only be served by one truck (t23) of capacity 11 tonnes. The algorithm was successful in pricing a tour that served both these customers using the same truck, thus leading towards the optimal solution.

The performance of the algorithm is evaluated according to the solution times and the quality of the final solution attained in the branch-and-cut phase. The metrics compared the branch-and-cut on the compact formulation with the generated columns and branch-and-cut. Both of these were run with identical parameter settings to the solver.

Table 4.2: Pricing problem solution times

Input			Models with Time Window Constraints				Models without Time Window Constraints			
Instance ID	Nodes	Arcs	Avg Stage 1 Time	Avg Stage 2 Time	Avg Sub Time	Avg CG Time	Avg Stage 1 Time	Avg Stage 2 Time	Avg Sub Time	Avg CG Time
Ade_30	308	8298	0.0031	0.3618	0.3903	0.7802	0.0020	0.3476	0.3814	0.6794
Ade_40	312	9958	0.0035	0.4892	0.5294	1.1178	0.0032	0.4630	0.4993	0.8961
Ade_47	462	18584	0.0098	1.1017	1.1956	2.4127	0.0072	1.0166	1.1164	2.0816
Ade_53	466	20268	0.0114	1.4766	1.6044	3.1379	0.0085	1.3554	1.4912	2.8073
Ade_63	606	31928	0.0143	2.8369	3.0992	6.0958	0.0214	2.6540	2.9642	5.5371
Ade_93	678	47204	0.0229	4.8751	5.4913	10.5457	0.0241	4.4460	5.1397	9.5846
Ade_112	884	76018	0.0345	7.8112	8.8719	16.4542	0.0253	7.0829	8.5075	15.0253
Ade_153	952	100456	0.0792	12.0451	14.1816	25.7412	0.0561	10.7590	14.7229	23.3018
Ade_197	1106	141772	0.1436	26.1087	32.7445	59.6305	0.1021	17.7377	21.8480	42.4430
Average	641	50,498	0.0358	6.3451	7.5676	13.9907	0.0278	5.0958	6.2967	11.3729

Nodes – Number of nodes in the MMCF network
Arcs – Number of arcs in the MMCF network
Avg Stage 1 Time – Average time in seconds for a network simplex method solve of stage 1
Avg Stage 2 Time – Average time in seconds for a MIP solve of stage 2
Avg Stage Sub Time – Average time in seconds for a sub-problem solve including stage 1, stage 2 and data processing
Avg CG Time – Average time in seconds for a column generation iteration including master problems solve, pricing problems solve and data processing

In addition to comparing the final primal solution attained between the runs, the final dual bounds (inclusive of the cutting planes generated on the linear relaxation of the branch-and-cut tree) were compared between the runs. The bound attained from IP/MIP (B&C dual bound, B&C solution) and Column Generation (B&C dual bound, B&C solution) are almost the same with minor variations. These dissimilarities are embedded in (Table 4.5). The summary of results is in (Table 4.6). This table shows the time and cost (distance) improvements when the proposed column generation method is used by comparing the CG model against the IP and MIP model. The proposed algorithm attains a true optimum for many of the smaller sized networks. For the models with time window constraints, the primal bound (the objective of the Dantzig-Wolfe master problem) was within 0.94 % of the dual bound (the objective of the auxiliary restricted master problem). In a few instances, the algorithm terminated when the primal bound was within a tolerance of E-05 of the dual bound, where the optimal solution to the model was equal to the primal and the dual bounds. Through empirical experiments that were based on the tailing-off effect of the duals, an iteration limit of 250 was found to be effective in pricing a sufficient number of columns and was chosen for many of the instances. On average, for models with time window constraints, the algorithm generated about 1.77% of columns and achieved solutions within 0.03% of those of the branch-and-cut solvers. With respect to solves times, the algorithm achieved up to 15.15 times improvement over the branch-and-cut solvers. On average, for models without time window constraints, the algorithm generated about 1.12% of columns and achieved solutions within 0.00% of those of the branch-and-cut solvers. With respect to solve times, the algorithm achieved up to 2.21 times improvement over the branch-and-cut solvers. (Figure. 4.1) and (Figure. 4.2) reflect a deeper investigation into the behaviour of the proposed model over iterations. (Figure. 2) plots the primal bound and dual bound for all instances when a time window is not permitted; similarly, (Figure. 4.2) does the exact same job as (Figure. 4.1) but for a model with a time window. In (Figure 4.1), and for models 30, 40, 47 and 63, the primal bound at termination was within 0.00% of the dual bound; this value for models 53, 93, 112 and 153 is within 1.17% of the dual bound, and for model 197 the primal bound at termination was within 1.74% of the dual bound. Moreover, for models 53 and 197, dual stabilization techniques were employed to counter the heading-in effect of duals commonly observed in column generation. We can perceive the terminations of models with a time window from (Figure. 4.2) where for models 30, 40, 47, 53 and 63, the primal bound at termination was within 0.85% of the dual bound, and for models 93, 112 and 153 where it was within 1.44% of the dual bound, and for the largest instance (Ade_197) where the primal bound at termination was within 4.55% of the dual bound.

Table 4.3: Results of the algorithm for models with time window constraints

Instance ID	Column Generation				B&C with Generated Columns							B&C with Original Columns				
	CG Iterations	CG Columns Generated	CG Column Generated Pct	CG Solve Time	Avg Sub Solve Time	B&C Unassigned Customers	B&C Distance	B&C Dual Bound	B&C Solve Time	B&C Polish Time	CG and B&C Solve Time	MIP Unassigned Customers	MIP Distance	MIP Dual Bound	MIP Solve Time	MIP Polish Time
Ade_30	145	3122	2.25%	113.13	0.39	0	176.62	176.62	0.60	0	129.57	0	176.62	176.62	7.92	0
Ade_40	215	6224	3.73%	240.33	0.53	0	270.18	270.12	16.98	0	277.77	0	270.18	270.12	200.38	0
Ade_47	250	8370	2.28%	603.17	1.20	0	407.34	407.34	4.84	0	651.15	0	407.34	407.34	493.70	0
Ade_53	250	9199	1.91%	784.47	1.60	1	623.99	623.23	17.19	0	858.94	1	623.99	623.09	1609.40	0
Ade_63	250	9555	1.04%	1523.95	3.10	0	367.82	367.82	9.12	0	1642.55	0	367.82	367.82	2975.21	0
Ade_93	250	25834	1.72%	2636.44	5.49	0	1196.22	1192.96	96.57	1001.15	3924.90	0	1195.67	1191.94	30375.72	8210.92
Ade_112	250	22512	0.96%	4113.56	8.87	0	624.37	623.26	87.48	1001.39	5503.71	0	623.46	623.46	33526.22	10373.93
Ade_153	250	34871	1.06%	6435.30	14.18	0	901.07	899.50	740.86	3602.04	11207.19	0	900.81	899.50	169763.58	68406.51
Ade_197	250	53549	0.92%	14907.61	32.74	0	1678.72	1624.87	18002.13	10823.31	44521.63	NA	NA	NA	NA	NA
Average	234	19248	1.76%	3484.22	7.57	0	570.95	687.30	2108.42	1825.32	3024.47	0	570.736543	569.9865	29869.0168	10874

CG Iterations – Number of iterations in CG
CG Columns Generated – Total number of columns generated from tours with at least one negative reduced cost column, from all the iterations.
CG Column Generated Pct – Percentage of columns generated compared against the total number of columns in the compact formulation
CG Solve Time – Total time in seconds for the column generation phase to terminate
Avg Sub Solve Time – Average time in seconds for solving a pricing problem (inclusive of solving both stage 1/stage 2 formulations and data processing)
B&C Unassigned Customers – Total number of unassigned customers of CG B&C
B&C Distance – Final solution attained from B&C inclusive of polishing, terminated either by time or tolerance
B&C Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with generated columns
B&C Solve Time – Total time spent in B&C phase until either the starting criteria for polishing or time limit is reached
B&C Polish Time – Total time spent in polishing the best solution of CG B&C
CG and B&C Solve Time – Total elapsed time in seconds for Column Generation including CG phase, B&C phase and data processing
MIP Unassigned Customers – Total number of unassigned customers of MIP B&C
MIP Distance – Final solution attained from B&C inclusive of polishing, terminated either by time or tolerance
MIP Dual Bound–- Final best dual bound inclusive of cuts from B&C of the compact formulation with original columns
MIP Polish Time – Total time spent on polishing the best solution of MIP B&C
MIP Solve Time – Total elapsed time in seconds for MIP B&C inclusive of polishing and data processing

Table 4.4: Results of the algorithm for models without time window constraints

Instance ID	Column Generation				B&C with Generated Columns						B&C with Original Columns			
	CG Iterations	CG Columns Generated	CG Column Generated Pct	CG Solve Time	Avg Sub Solve Time	Unassigned Customers	B&C Distance	B&C Dual Bound	B&C Solve Time	Total Time (CG and B&C)	Unassigned Customers	IP Distance	IP Dual Bound	IP Solve Time
Ade_30	36	1704	1.23%	24.46	0.38	0	182.46	182.46	0.34	42.10	0	182.46	182.46	1.21
Ade_40	44	2883	1.73%	39.43	0.50	0	271.37	271.37	0.67	59.73	0	271.37	271.37	3.20
Ade_47	70	4571	1.24%	145.72	1.12	0	409.43	409.43	1.63	190.13	0	409.43	409.43	7.29
Ade_53	103	6187	1.29%	289.15	1.49	2	625.78	625.78	1.49	344.35	2	625.78	625.78	10.80
Ade_63	74	6415	0.70%	409.75	2.96	0	371.57	371.57	2.88	517.92	0	371.57	371.57	33.37
Ade_93	150	19345	1.29%	1437.69	5.14	0	1199.10	1199.10	18.19	1631.19	0	1199.10	1199.10	127.40
Ade_112	150	18675	0.80%	2253.80	8.51	0	628.30	628.30	24.56	2559.08	0	628.30	628.30	638.95
Ade_153	150	30953	0.94%	3495.27	14.72	0	906.36	906.36	143.65	4035.65	0	906.36	906.36	3184.20
Ade_197	250	47641	0.82%	10610.76	21.85	0	1695.31	1695.31	611.15	11951.08	0	1695.31	1695.31	26423.12
Average	114	15375	1.12%	2078.45	6.30	~ 0	698.85	698.85	89.40	2370.14	~ 0	698.85	698.85	3381.06

CG Iterations – Number of iterations in CG
CG Columns Generated –Total number of columns generated from tours with at least one negative reduced cost column, from all the iterations
CG Column Generated Pct – Percentage of columns generated compared against the total number of columns in the compact formulation
CG Solve Time – Total time in seconds for the column generation phase to terminate
Avg Sub Solve Time – Average time in seconds for solving a pricing problem (inclusive of solving both stage 1/stage 2 formulations and data processing)
B&C Unassigned Customers – Total number of unassigned customers of CG B&C(For model 53, unassigned customers are 2 for CG and B&C for non-TW model)
B&C Distance – Final solution attained from B&C inclusive of polishing, terminated either by time or tolerance
B&C Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with generated columns
B&C Solve Time – Total time spent in B&C phase until either the starting criteria for polishing or time limit is reached
CG and B&C Solve Time – Total elapsed time in seconds for Column Generation including CG phase, B&C phase and data processing
IP Unassigned Customers – Total number of unassigned customers of IP B&C
IP Distance – Final solution attained from B&C inclusive of polishing, terminated either by time or tolerance
IP Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with original columns
IP Solve Time – Total elapsed time in seconds for MIP B&C inclusive of polishing and data processing

Table 4.5: Various bounds attained at different phases of the algorithm

Instance ID	With Time Window Constraints						Without Time Window Constraints					
	Primal Bound	Dual Bound	CG B&C Dual Bound	CG B&C Solution	MIP B&C Dual Bound	MIP B&C Solution	Primal Bound	Dual Bound	CG B&C Dual Bound	CG B&C Solution	IP B&C Dual Bound	IP B&C Solution
Ade_30	178.130234	176.619900	176.619900	176.619900	176.620000	176.619900	182.461656	182.461656	182.461656	182.461656	182.461656	182.461656
Ade_40	270.235565	270.120261	270.120000	270.180752	270.120000	270.180750	271.373370	271.373370	271.373370	271.373370	271.373370	271.373370
Ade_47	407.397646	407.340044	407.340044	407.340044	407.340044	407.340044	409.426508	409.426508	409.426508	409.426508	409.426508	409.426508
Ade_53	624.346680	622.694567	623.228000	623.993460	623.094000	623.993460	626.176284	625.777408	625.777408	625.777408	625.777408	625.777408
Ade_63	368.697225	367.816687	367.816687	367.816687	367.816687	367.816687	371.569752	371.569752	371.569752	371.569752	371.569752	371.569752
Ade_93	1197.501233	1192.700003	1192.964000	1196.217850	1191.943000	1195.756900	1199.269804	1199.102536	1199.102536	1199.102536	1199.102536	1199.102536
Ade_112	627.579871	623.257573	623.257573	624.368394	623.458000	623.458350	628.380634	628.297246	628.297246	628.297246	628.297246	628.297246
Ade_153	912.683063	899.499455	899.499000	901.074815	899.499000	900.810680	907.945008	906.362747	906.362747	906.362747	906.362747	906.362747
Ade_197	1699.612827	1622.318884	1624.866000	1678.724354	NA	NA	1725.371426	1695.307693	1695.307693	1695.307693	1695.307693	1695.307693
Average	698.464927	686.929708	687.301245	694.037362	569.986341	570.747096	702.441605	698.853213	698.853213	698.853213	698.853213	698.853213

Primal Bound – Optimal solution to the Dantzig-Wolfe restricted master problem
Dual Bound – Optimal solution to the auxiliary restricted master problem.
CG B&C Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with generated columns
CG B&C Solution – Optimal solution from B&C within the specified tolerance of the compact formulation with generated columns
MIP B&C Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with original columns for models with time window constraints
MIP B&C Solution – Optimal solution from B&C within the specified tolerance of the compact formulation with original columns for models with time window constraints
IP B&C Dual Bound – Final best dual bound inclusive of cuts from B&C of the compact formulation with original columns for models without time window constraints
IP B&C Solution – Optimal solution
from B&C within the specified tolerance of the compact formulation with original columns for models without time window constraints

Table 4.6: Comparing the CG model with IP and MIP models

Instance ID	Without Time Window		With Time Window	
	Column Generation Versus IP		Column Generation Versus MIP	
	Cost Improvement	Time Improvement	Cost Improvement	Time Improvement
Ade_30	0.0000%	3380%	0.0000%	1536%
Ade_40	0.0000%	1767%	0.0000%	39%
Ade_47	0.0000%	2508%	0.0000%	32%
Ade_53	0.0000%	3088%	0.0000%	-47%
Ade_63	0.0000%	1452%	0.0000%	-45%
Ade_93	0.0000%	1180%	-0.0456%	-87%
Ade_112	0.0000%	301%	-0.1460%	-84%
Ade_153	0.0000%	27%	-0.0293%	-93%
Ade_197	0.0000%	-55%	NA	NA

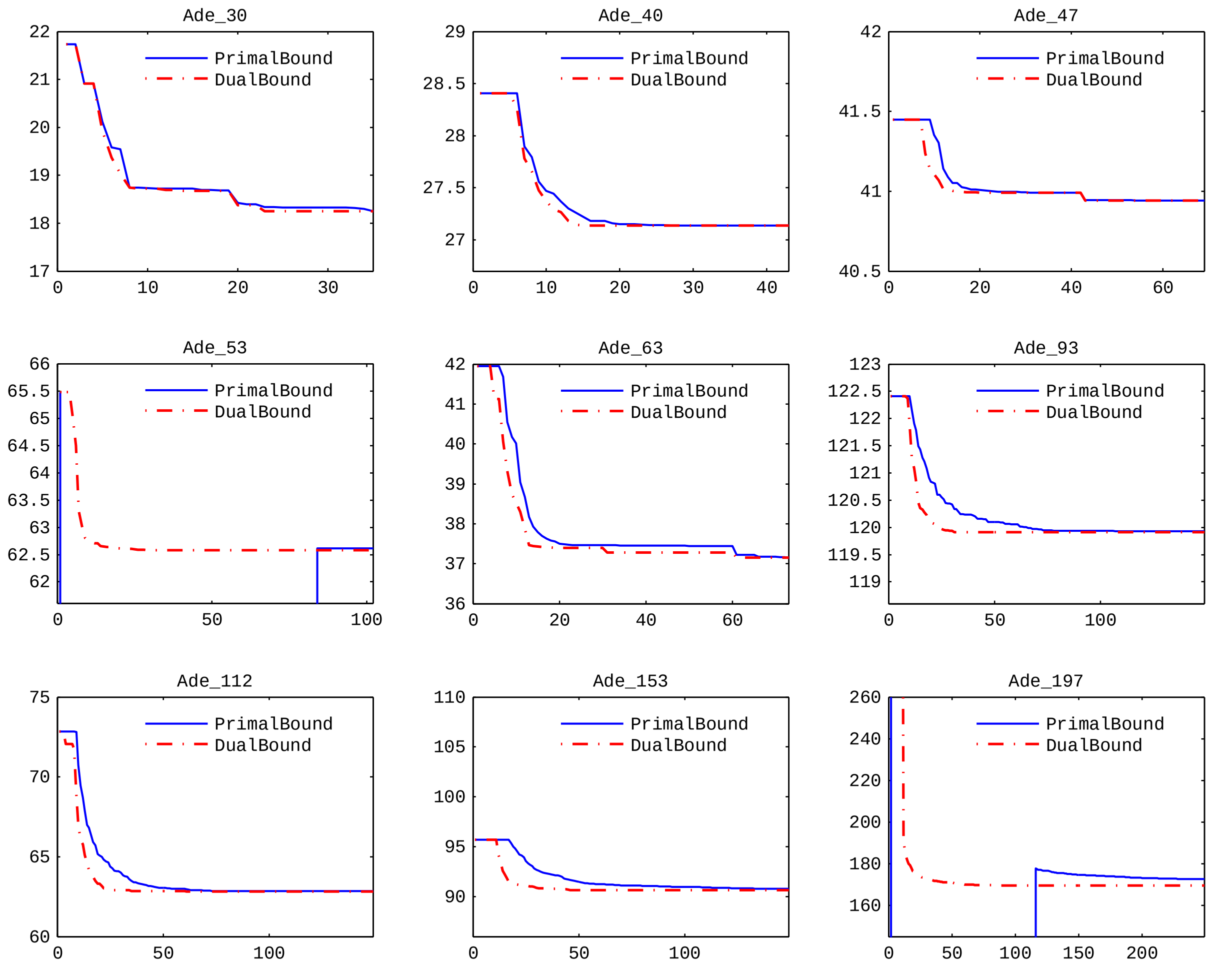

Figure 4.1: Behaviour of the proposed column generation method when a time window is not allowed. The horizonal axis is number of iterations and the vertical axis is cost.

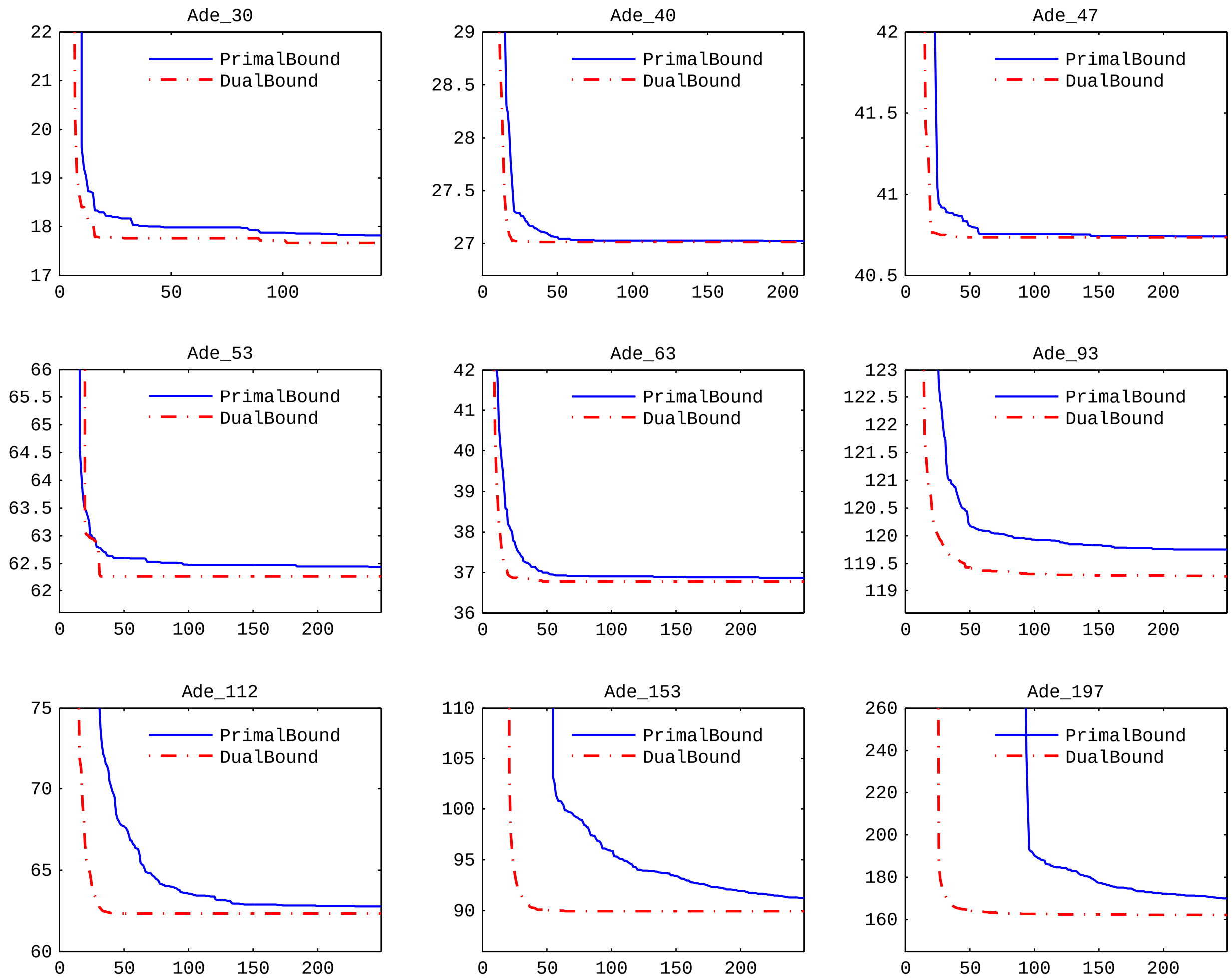

Figure 4.2: The behaviour of the proposed column generation method when a time window is allowed. The horizonal axis is number of iterations and the vertical axis is cost.

5 Conclusion

This paper proposed a mathematical model based on the Column Generation technique to solve Ready Mixed Concrete (RMC) dispatching problem with and without a time window. The Dantzig-Wolfe method was used for reformulating the problem and then to provide solutions within a two-stage procedure. The proposed method was compared with integer programming and mixed-integer programming. For evaluation, a real database belonging to an active Ready Mixed Concrete (RMC) was used, and from the available data nine instances of different sizes were chosen randomly. The number of unassigned customers by the proposed method in situations both with and without time window is zero. Moreover, when a time window is not allowed, the distances acquired by the proposed method and IP are exactly the same; however, on average, column generation converges 30% more quickly than IP. The MIP solution for large scale instances (such as Ade-197) is intractable when the proposed method converges. Despite this issue, among the instances in which the MIP solution exists, on average the column generation method attained results around 10 times faster than MIP with around 1% increase in distance.

Acknowledgement

NICTA is funded by the Australian Department of Communications and the Australian Research Council through the ICT Centre of Excellence program.

Note

This paper has been submitted to Computer-Aided Civil and Infrastructure Engineering and the reason of temporary publishing as CSE Technical report is to provide an opportunity to cite it in the parallel research is being conducted by the authors.

Bibliography

[1] Lasse Asbach, Ulrich Dorndorf, and Erwin Pesch. Analysis, modeling and solution of the concrete delivery problem. *European Journal of Operational Research*, 193(3):820–835, 2009.

[2] Robert E Bixby and Matthew J Saltzman. Recovering an optimal lp basis from an interior point solution. *Operations Research Letters*, 15(4):169–178, 1994.

[3] Ming Cao, Ming Lu, and Jian-ping Zhang. Concrete plant operations optimization using combined simulation and genetic algorithms. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 7, pages 4204–4209. IEEE.

[4] Elaine Chang, Athanasios Ziliaskopoulos, David Boyce, and S Travis Waller. Solution algorithm for combined interregional commodity flow and transportation network model with link capacity constraints. *Transportation Research Record: Journal of the Transportation Research Board*, 1771(1):114–123, 2001.

[5] M. Y. Cheng, H. C. Tsai, and C. L. Liu. Artificial intelligence approaches to achieve strategic control over project cash flows. *Automation in Construction*, 18(4):386–393.

[6] George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.

[7] M Poggi de Aragao and Eduardo Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. *Techical Report, Departamento de Informatica, PUC-Rio*, 2003.

[8] Chung-Wei Feng, Tao-Ming Cheng, and Hsien-Tang Wu. Optimizing the schedule of dispatching rmc trucks through genetic algorithms. *Automation in Construction*, 13(3):327–340, 2004.

[9] Chung-Wei Feng and Hsien-Tang Wu. Integrating fmga and cyclone to optimize the schedule of dispatching rmc trucks. *Automation in Construction*, 15(2):186–199, 2006.

[10] JM Garcia, S Lozano, K Smith, T Kwok, and G Villa. Coordinated scheduling of production and delivery from multiple plants and with time windows using genetic algorithms. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on*, volume 3, pages 1153–1158. IEEE.

[11] Asvin Goel and Volker Gruhn. A general vehicle routing problem. *European Journal of Operational Research*, 191(3):650–660, 2008.

[12] Stefan Irnich, Guy Desaulniers, Jacques Desrosiers, and Ahmed Hadjar. Path-reduced costs for eliminating arcs in routing and scheduling. *INFORMS Journal on Computing*, 22(2):297–313, 2010.

[13] Damian J Kelly and Garrett M ONeill. *The minimum cost flow problem and the network simplex solution method*. PhD thesis, University College Dublin, Graduate School of Business, 1991.

[14] Federico Liberatore, Giovanni Righini, and Matteo Salani. A column generation algorithm for the vehicle routing problem with soft time windows. *4OR*, 9(1):49–82, 2011.

[15] Dung-Ying Lin, Ampol Karoonsoontawong, and S Travis Waller. A dantzig-wolfe decomposition based heuristic scheme for bi-level dynamic network design problem. *Networks and Spatial Economics*, 11(1):101–126, 2011.

[16] Dung-Ying Lin, Varunraj Valsaraj, and S Travis Waller. A dantzig-wolfe decomposition-based heuristic for off-line capacity calibration of dynamic traffic assignment. *Computer-Aided Civil and Infrastructure Engineering*, 26(1):1–15, 2011.

[17] M Lu. Hkconsim: a simulation platform for planning and optimizing concrete plant operations in hong kong. In *Proceedings of International Conference on Innovation and Sustainable Development of Civil Engineering in the 21st Century*, pages 278–283, 2002.

[18] Mojtaba Maghrebi, S. Travis Waller, and Claude Sammut. Scheduling concrete delivery problems by a robust meta heuristic method. In *Computer Modeling and Simulation (EMS), 2013 Seventh UKSim/AMSS European Symposium on*. IEEE.

[19] David Naso, Michele Surico, Biagio Turchiano, and Uzay Kaymak. Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177(3):2069–2099, 2007.

[20] Liu Pan, Wang Liya, Ding Xihai, and Gao Xiang. Scheduling of dispatching ready mixed concrete trucks trough discrete particle swarm optimization. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 4086–4090.

[21] Edward Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS Journal on Computing*, 19(4):534–541, 2007.

[22] V Schmid. Trucks in movement: hybridization of exact approaches and variable neighborhood search for the delivery of ready-mixed concrete. *Unpublished doctoral dissertation, Universitat Wien, Austria*, 2007.

[23] Ilaria Vacca and Matteo Salani. The vehicle routing problem with discrete split delivery and time windows. In *Proceedings of the 9th STRC Swiss Transport Research Conference*, 2009.

[24] Robert Vanderbei. Linear programming: Foundations and extensions. 1996.

[25] Warren E Walker. A method for obtaining the optimal dual solution to a linear program using the dantzig-wolfe decomposition. *Operations Research*, pages 368–370, 1969.

[26] S Yan, HC Lin, and XY Jiang. A planning model with a solution algorithm for ready mixed concrete production and truck dispatching under stochastic travel times. *Engineering Optimization*, 44(4):427–447, 2012.

[27] Shangyao Yan, Weishen Lai, and Maonan Chen. Production scheduling and truck dispatching of ready mixed concrete. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):164–179, 2008.