COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **31** (2016) 145–159

Column Generation-Based Approach for Solving Large-Scale Ready Mixed Concrete Delivery Dispatching Problems

Mojtaba Maghrebi

School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia, Department Civil Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Iran

Vivek Periaraj

Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA

S. Travis Waller*

School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia, National Information and Communications Technology Australia (NICTA), Sydney, Australia

&

Claude Sammut

School of Computer Science Engineering, University of New South Wales (UNSW), Sydney, 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 article, we develop a column generation algorithm for vehicle routing problems (VRPs) 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 article. 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 and solved in polynomial time using efficient network simplex method implementations. A large-scale field collect data set 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.*

*To whom correspondence should be addressed. E-mail: *s.waller@unsw.edu.au*.

DOI: 10.1111/mice.12182

1 INTRODUCTION

Although ready mix 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 article 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 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 article employs the CG mathematical technique. CG 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 article consists of four sections. In the first section, the relevant literature in this area is reviewed. Section 2 covers CG and reformulating steps. In section 3, 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. Feng and Wu (2000) introduced a single depot model by focusing on minimizing idle times. The developed model was solved heuristically. Feng and Wu introduced a more advanced model in 2006 (Feng and Wu, 2006). Naso et al. (2007) introduced one of the most advanced RMC models so far. It can cover a multi-depots RMC but with a homogeneous fleet by considering multi-objectives. Their model can take into account the hired trucks as well as the out-sourced deliveries. This model deals with deliveries to a customer and the assigned truck/s and depot/s for each delivery. The only drawback of this model is the large number of decision variables as well as the number of side constraints. All the decision variables are binary; therefore, computing time is a challenge in this model when the optimum solution is desirable. Yan et al. (2008) presented a new formulation for a single depot with a homogeneous fleet; similar to Naso et al. (2007), it splits a customer depending on the number of required deliveries. A wide variant of RMC formulation was introduced by Yan and his colleagues; such as when the overtime is considered (Yan and Lai, 2007), or covering the incidents (Yan et al., 2012) and also associating stochastic travel times (Yan et al., 2012). Lin et al. (2010) presented a new model by focusing on minimizing the waiting time when there is uncertainty in demand. They assumed RMC dispatching as a job shop problem when the construction site represents a job and trucks correspond with a workstation. This model can be used for a single depot with a heterogeneous fleet. Another model in this context was presented by Schmid et al. (2009) for a single depot with a heterogeneous fleet. Their model forces MIP to avoid unsupplied customers by penalizing the unsatisfied customers in the objective function; later on a new version of this model was introduced in 2010 (Schmid et al., 2010). Asbach et al. (2009) introduced a novel model whose structure is much simpler than that of other introduced models and which can be used for modeling multi-depots and a heterogeneous fleet. It has been proved that an RMC optimization problem is an NP-hard problem (Yan et al., 2008; Asbach et al., 2009). Therefore, to deal with this problem, heuristic methods have been widely used in the literature such as Cao et al. (2004), Feng and Wu (2006), Garcia et al. (2002), Maghrebi et al. (2013b), Maghrebi et al. (2014d), and Srichandum and Rujirayanyong (2010). 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. Yan et al. (2008) 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. Yan et al. (2012) used decomposition and relaxation techniques coupled with a mathematical solver to solve the problem, and Payr and Schmid (2009) applied variable neighborhood search to deal with RMC optimization problems. Asbach et al. (2009) made the mathematical modeling much simpler by dividing the depots and customers into sub-depots and sub-customers and most recently Benders Decomposition (Maghrebi et al., 2014a), Machine learning approach (Maghrebi and Waller, 2014; Maghrebi et al., 2013a; Maghrebi et al., 2015a,c), assessing experts' decisions in RMC dispatching room (Maghrebi et al., 2014c), new formulation (Maghrebi et al., 2014b), and Lagrangian approach were applied (Narayanan et al., 2015) to achieve solutions with a slight optimality gap but within a practical time. Most recently, Maghrebi et al. (2015b) assessed the optimality gap of experts' decision in RMC by comparing their decisions with IP/MIP and two heuristic methods.

However, CG techniques have not been used particularly when it is coupled with Dantzig–Wolfe. Since the time Dantzig and Wolfe (Dantzig and Wolfe, 1960) 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, CG has been successfully applied to different types of VRPs. Liberatore et al. (2011) apply CG 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. Goel and Gruhn (2008) present a CG 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. Irnich et al. (2010) 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. Walker (1969) 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 in Lin et al. (2011a), Lin et al. (2011b), and Chang et al. (2001), but in this article we only focus on VRP-based problems.

3 METHODOLOGY

Column generation is a common method for solving large-scale integer problems. 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 CG. First, a major proportion of the variables are nonbasic at the optimal solution, hence it is required to generate only those columns whose reduced costs are negative. In 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 CG, 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

Table 1
List of symbols

Symbol	*Description*
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
Υ	Maximum time to haul the concrete
λ_k	Tour of truck k
λ_k^p	Path p in tour of truck k
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
z_k	Cost of truck k in λ_k
z_k^p	Cost of truck k in λ_k^p
f_{uv}	Flow of a commodity along edge (u, v)

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 CG technique and introduces a method for formulating RMC dispatching problems. The terminology used in this article for modeling the original RMC formulation is similar to that of Asbach et al. (2009).

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 Table 1.

$$Minimize \sum_u \sum_v \sum_k z_{uvk} x_{uvk} - \sum_c \beta_c y_c \quad (1)$$

Subject to :

$$\sum_{u \in u_s} \sum_v x_{uvk} = 1 \; \forall \, k \in K \quad (2)$$

$$\sum_u \sum_{v \in v_f} x_{uvk} = 1 \; \forall \, k \in K \quad (3)$$

$$\sum_u x_{uvk} - \sum_j x_{vjk} = 0 \; \forall \, k \in K, \; v \in C \cup D \quad (4)$$

$$\sum_{u \in D} \sum_k x_{uvk} \leq 1 \; \forall v \in C \quad (5)$$

$$\sum_{v \in C} \sum_k x_{uvk} \leq 1 \; \forall u \in D \quad (6)$$

$$\sum_{u \in D} \sum_k q_k x_{uvk} \geq q_c y_c \; \forall c, v \in C \quad (7)$$

$$-M \; (1 - x_{uvk}) + s_u + t_{uvk} \leq \; w_v - w_u \quad \forall \, (u, v, k) \in E \quad (8)$$

$$M \; (1 - x_{uvk}) + \gamma + s_u \geq w_v - w_u \quad \forall \, (u, v, k) \in E \quad (9)$$

$$x_{uvk} \in \{0, 1\} \; and \; y_c \in \{0, 1\} \quad (10)$$

The objective function (Equation (1)) forces optimization to find feasible solutions for all customers and penalizes if a feasible solution for customer c cannot be found by applying zero to y_c. Therefore, due to the large value of β_c, optimization attempts to avoid unsupplied customers. Equation (2) ensures that a truck at the start of the day leaves once from its base and similarly Equation (3) necessitates return of a truck to the depot/its home at the end of the day. In reality, a truck arrives at either a depot or a customer then leaves that node after loading/unloading. This concept is called conservation of flow and Equation (4) ensures this issue if u ∈ C then (v ∈ D and j ∈ C) but if u ∈ D then (v ∈ C and j ∈ Dv_f). In this formulation, a depot is divided into a set of sub-depots based on the number of possible loadings. Similarly, a customer is divided into a set of sub-customers according to the number of required deliveries. To simplify the text, from here on a depot means a sub-depot which can load a truck only at a specific time and similarly a customer means a sub-customer that requires a delivery only at a specific time. Therefore, Equations (5) and (6), respectively, certify sending a truck only to a customer and a depot only supplies a customer. Equation (7) checks demand satisfaction for customers. Equations (8) and (9) are designed to control timing issues. Equation (8) ensures that concrete will be supplied to customers within the specified time, and similarly Equation (9) ensures that the travel time for each customer will not exceed the permitted time for delivery (γ) because fresh concrete is a perishable material and it is not possible to haul it more than (γ) which varies according to the type of concrete.

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 is a well-known result in network flow theory that an extreme point $x_{uvk} \in X$ is also a path $p \in P$ in the network. The natural choice for network flow problems is to consider a path-based reformulation of the Dantzig–Wolfe master problem. Vacca and Salani (2009) consider 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 5 and 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 2, 3, 4, 8, and 9 have block diagonal structures with respect to trucks whereas constraints 5, 6, and 7 are the coupling constraints with variables associated with all the trucks. Each truck tour can be equivalently expressed as follows:

$$\lambda_k \leq \sum_{u} \sum_{v} x_{uvk} \tag{11}$$

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:

$$z_k = \sum_{u} \sum_{v} z_{uvk} x_{uvk} - \sum_{c \in C_k} \beta_c \tag{12}$$

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

$$\text{Minimize} \sum_{k} \sum_{p} z_k^p \lambda_k^p \tag{13}$$

Subject to:

$$\sum_{u \in u_s} \sum_{v} x_{uvk} = 1 \ \forall \ k \in K \tag{14}$$

$$\sum_{k \in D_k} \sum_{p} \lambda_k^p \leq 1 \ \forall \ D_k \in D \tag{15}$$

$$\sum_{k \in C_k} \sum_{p} q_k \, \lambda_k^p \geq q_c \ \forall \ C_k \in C \tag{16}$$

$$\sum_{p} \lambda_k^p = 1 \ \forall \ k \in K \tag{17}$$

$$\lambda_k^P \geq 0 \ \forall \ k \in K, \ p \in P \tag{18}$$

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 11 between λ_k^p and x_{uvk}, the optimal solution of the Dantzig–Wolfe restricted master problem $\lambda_k^{p^*}$ can be used to recover the solution to the compact formulation when $\lambda_k^p \in \{0, 1\}$. 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 14, 15, and 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, de Aragao and Uchoa (2003) propose a method to formulate an explicit Dantzig–Wolfe master called *Explicit Master* that retains the linking constraints 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 14, 15, and 16.

In our CG 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 2, 3, 4, 8, 9, and 10 obtained from the auxiliary restricted master problem are denoted by μ. If A_1 is the 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:

$$\left\{\begin{array}{l} rc_{uvk} = z_{uvk} - A_1\mu \ \forall\ u \in u_s,\ v \in D,\ k \in K \\ rc_{uvk} = z_{uvk} - A_1\mu \ \forall\ u \in C,\ v \in v_f,\ k \in K \\ rc_{uvk} = z_{uvk} - A_1\mu \ \forall\ u \in C,\ v \in D,\ k \in K \\ rc_{uvk} = z_{uvk} - A_1\mu - A_2\pi \ \forall\ u \in D,\ v \in C,\ k \in K \\ rc_c = \beta_c - A_2\pi \ \forall\ c \in C \end{array}\right\} \quad (19)$$

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 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 5 and 6 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.

3.4 Stage 1 formulation

The MMCF pricing problem can be formally stated as follows. Given a flow network G(V, E), where edge (u, v) ∈ 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}.

$$\text{Minimize} \sum\nolimits^{a_{uv}} f_{uv} \quad (20)$$

$$\text{Subject to}: \quad f_{uv} \le c_{uv\ \forall u,v} \quad (21)$$

$$f_{uv} - \sum_{w \in V} f_{wu} = 0\ \forall u \in V, v \in V,\ (u,v) \notin s,t \quad (22)$$

$$\sum_{w \in V} f_{uv} = \sum_{w \in V} f_{wt} = d \quad (23)$$

$$c_{uv} = 1\ \forall\ u \in V, v \in V,\ u \notin s,\ v \notin t \quad (24)$$

$$c_{uv} = 0 \quad \forall\ u \in V,\ v \in V \ and\ if\ E(u,v)\ is\ not\ feasible \quad (25)$$

$$a_{uv} = \underbrace{MIN}_{k \in K}\ rc_{uvk} \quad (26)$$

$$a_{sw} = a_{wt} = 0\ \forall w \in V \quad (27)$$

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 (Vanderbei, 2008). 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.

$$Minimize \{cx : Ax = b,\ l \le x \le u\} \quad (28)$$

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

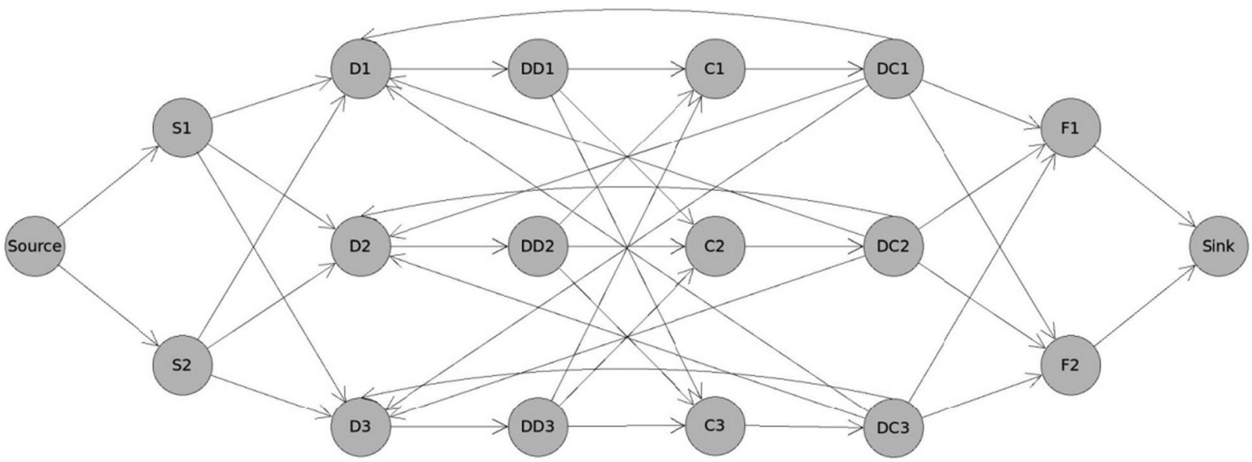

Fig. 1. Network graph.

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 (Kelly and ONeill, 1991) 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 3 lists average solution times of the pricing problem across different instances.

3.5 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. Although the MMCF pricing problem may result in tours that are infeasible with respect to the demand constraints 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.

$$Minimize \sum_u \sum_v \sum_k rc_{uvk} x_{uvk} + \sum_{c \in C} rc_c y_c + \sum_{k \in K} \sigma_k \quad (29)$$

Subject to :

$$\sum_{u \in u_s} \sum_v x_{uvk} = 1 \; \forall k \in K \quad (30)$$

$$\sum_u \sum_{v \in v_f} x_{uvk} = 1 \; \forall k \in K \quad (31)$$

$$\sum_u x_{uvk} - \sum_j x_{vjk} = 0 \; \forall k \in K, \; v \in C \cup D \quad (32)$$

$$\sum_{u \in D} \sum_k x_{uvk} \leq 1 \; \forall v \in C \quad (33)$$

$$\sum_{v \in C} \sum_k x_{uvk} \leq 1 \; \forall u \in D \quad (34)$$

$$\sum_{u \in D} \sum_k q_k x_{uvk} \geq q_c y_c \; \forall c, v \in C \quad (35)$$

$$-M \; (1 - x_{uvk}) + s_u + t_{uvk} \leq w_v - w_u \;\; \forall (u, v, k) \in E \quad (36)$$

$$M \; (1 - x_{uvk}) + \gamma + s_u \geq w_v - w_u \;\; \forall (u, v, k) \in E \quad (37)$$

$$x_{uvk} \in \{0, 1\} \; and \; y_c \in \{0, 1\} \quad (38)$$

$$0 \leq x_{uvk} \leq 1, \; \forall u \in D^*, v \in C^*, k \in K \quad (39)$$

$$0 \leq x_{uvk} \leq 1, \; \forall u \in C^*, v \in D^*, k \in K \quad (40)$$

$$0 \leq x_{uvk} \leq 1, \; \forall u \in u_s, v \in D^*, k \in K \quad (41)$$

$$0 \leq x_{uvk} \leq 1, \; \forall u \in C^*, v \in v_f, k \in K \quad (42)$$

$$0 \leq x_{uvk} \leq 0, \; otherwise \quad (43)$$

3.6 Multiple column generation

The CG 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. de Aragao and Uchoa (2003) discuss a scenario of multiple CGs when the constraints of the master problem are nicely structured. The constraints 14, 15, and 16 of our master problem are consistent with the multiple CG 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 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 long time required to reach the zero reduced cost threshold for larger models, the CG phase is terminated within the specified number of iterations. The CG 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 CG phase.

4 RESULTS

The proposed CG 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 were used here belong to an active RMC network in Adelaide (Australia). Nine instances were selected randomly from the available database which characterizes the selected instances as given in Table 2. The algorithm was developed in C++ and tested on a RedHat® CentOS®5.9 Linux server with 8 3.60GHz Intel® Xeon® 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 (Rothberg, 2007) 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 Equation (44) where ε is empirically defined 10^{-10} 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 long time required to reach the zero reduced cost threshold for larger models, the CG phase is terminated within the specified number of iterations. The CG 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 CG phase.

$$EP\ Gap = \frac{|Best\ Integer\ Solution - Best\ Dual\ Bound|}{\varepsilon + |Best\ Integer\ Solution|} \tag{44}$$

"Barrier/Dual" (Bixby and Saltzman, 1994) is selected to solve the Dantzig–Wolfe restricted master problem 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. CG 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 branch-and-cut (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

Table 2
Problem data attributes

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

Table 3
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	8,298	0.0031	0.3618	0.3903	0.7802	0.0020	0.3476	0.3814	0.6794
Ade_40	312	9,958	0.0035	0.4892	0.5294	1.1178	0.0032	0.4630	0.4993	0.8961
Ade_47	462	18,584	0.0098	1.1017	1.1956	2.4127	0.0072	1.0166	1.1164	2.0816
Ade_53	466	20,268	0.0114	1.4766	1.6044	3.1379	0.0085	1.3554	1.4912	2.8073
Ade_63	606	31,928	0.0143	2.8369	3.0992	6.0958	0.0214	2.6540	2.9642	5.5371
Ade_93	678	47,204	0.0229	4.8751	5.4913	10.5457	0.0241	4.4460	5.1397	9.5846
Ade_112	884	76,018	0.0345	7.8112	8.8719	16.4542	0.0253	7.0829	8.5075	15.0253
Ade_153	952	100,456	0.0792	12.0451	14.1816	25.7412	0.0561	10.7590	14.7229	23.3018
Ade_197	1,106	141,772	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

Notes: 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.

100 deliveries. Table 3 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 the achieved results from the proposed CG model are compared with MIP when the time window is allowed. Similarly Table 5 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 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 toward 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

Table 4
Results of the algorithm for models with time window constraints

	Column generation				*B&C with generated columns*							*B&C with original columns*				
Instance ID	*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	3,122	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	6,224	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	8,370	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	9,199	1.91%	784.47	1.60	1	623.99	623.23	17.19	0	858.94	1	623.99	623.09	1,609.40	0
Ade_63	250	9,555	1.04%	1,523.95	3.10	0	367.82	367.82	9.12	0	1,642.55	0	367.82	367.82	2,975.21	0
Ade_93	250	25,834	1.72%	2,636.44	5.49	0	1,196.22	1,192.96	96.57	1,001.15	3,924.90	0	1,195.67	1,191.94	30,375.72	8,210.92
Ade_112	250	22,512	0.96%	4,113.56	8.87	0	624.37	623.26	87.48	1,001.39	5,503.71	0	623.46	623.46	33,526.22	10,373.93
Ade_153	250	34,871	1.06%	6,435.30	14.18	0	901.07	899.50	740.86	3,602.04	11,207.19	0	900.81	899.50	169,763.58	68,406.51
Ade_197	250	53,549	0.92%	14,907.61	32.74	0	1,678.72	1,624.87	18,002.13	10,823.31	44,521.63	NA	NA	NA	NA	NA
Average	234	19,248	1.76%	3,484.22	7.57	0	570.95	687.30	2,108.42	1,825.32	3,024.47	0	570.736543	569.9865	29,869.0168	10,874

Notes: 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 solve time – Total elapsed time in seconds for MIP B&C inclusive of polishing and data processing; MIP polish time—Total time spent on polishing the best solution of MIP B&C.

Table 5
Results of the algorithm for models without time window constraints

	Column generation				*B&C with generated columns*						*B&C with original columns*			
Instance ID	*CG iterations*	*CG columns generated*	*CG column generated pct*	*CG solve time*	*Avg sub solve time*	*Unassigned cus-tomers*	*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	1,704	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	2,883	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	4,571	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	6,187	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	6,415	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	19,345	1.29%	1,437.69	5.14	0	1,199.10	1,199.10	18.19	1,631.19	0	1,199.10	1,199.10	127.40
Ade_112	150	18,675	0.80%	2,253.80	8.51	0	628.30	628.30	24.56	2,559.08	0	628.30	628.30	638.95
Ade_153	150	30,953	0.94%	3,495.27	14.72	0	906.36	906.36	143.65	4,035.65	0	906.36	906.36	3,184.20
Ade_197	250	47,641	0.82%	10,610.76	21.85	0	1,695.31	1,695.31	611.15	11,951.08	0	1,695.31	1,695.31	26,423.12
Average	114	15,375	1.12%	2,078.45	6.30	~ 0	698.85	698.85	89.40	2,370.14	~ 0	698.85	698.85	3,381.06

Notes: 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); 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; Total time (CG and B&C) – Total elapsed time in seconds for column generation including CG phase, B&C phase, and data processing; 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 6
Various bounds attained at different phases of the algorithm

	With time window constraints						*Without time window constraints*					
Instance ID	*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	17.8130	17.6620	17.6620	17.6620	17.6620	17.6620	18.2462	18.2462	18.2462	18.2462	18.2462	18.2462
Ade_40	27.0236	27.0120	27.0120	27.0181	27.0120	27.0181	27.1373	27.1373	27.1373	27.1373	27.1373	27.1373
Ade_47	40.7398	40.7340	40.7340	40.7340	40.7340	40.7340	40.9427	40.9427	40.9427	40.9427	40.9427	40.9427
Ade_53	62.4347	62.2695	62.3228	62.3993	62.3094	62.3993	62.6176	62.5777	62.5777	62.5777	62.5777	62.5777
Ade_63	36.8697	36.7817	36.7817	36.7817	36.7817	36.7817	37.1570	37.1570	37.1570	37.1570	37.1570	37.1570
Ade_93	119.7501	119.2700	119.2964	119.6218	119.1943	119.5757	119.9270	119.9103	119.9103	119.9103	119.9103	119.9103
Ade_112	62.7580	62.3258	62.3258	62.4368	62.3458	62.3458	62.8381	62.8297	62.8297	62.8297	62.8297	62.8297
Ade_153	91.2683	89.9499	89.9499	90.1075	90.0811	90.1275	90.7945	90.6363	90.6363	90.6363	90.6363	90.6363
Ade_197	169.9613	162.2319	162.4866	167.8724	NA	NA	172.5371	169.5308	169.5308	169.5308	169.5308	169.5308
Average	69.8465	68.6930	68.7301	69.4037	57.0150	57.0805	70.2442	69.8853	69.8853	69.8853	69.8853	69.8853

Notes: 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 7
Comparing the CG model with IP and MIP models

	Without time window		*With time window*	
	Column generation versus IP		*Column generation versus MIP*	
Instance ID	*Cost improvement*	*Time improvement*	*Cost improvement*	*Time improvement*
Ade_30	0.0000%	–3,380%	0.0000%	–1,536%
Ade_40	0.0000%	–1,767%	0.0000%	–39%
Ade_47	0.0000%	–2,508%	0.0000%	–32%
Ade_53	0.0000%	–3,088%	0.0000%	+47%
Ade_63	0.0000%	–1,452%	0.0000%	+45%
Ade_93	0.0000%	–1,180%	–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

compact formulation with the generated columns and branch-and-cut. Both of these were run with identical parameter settings to the solver. 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 CG (B&C dual bound, B&C solution) are almost the same with minor variations. These dissimilarities are embedded in Table 6. The summary of results is in Table 7. This table shows the time and cost (distance) improvements when the proposed CG 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,

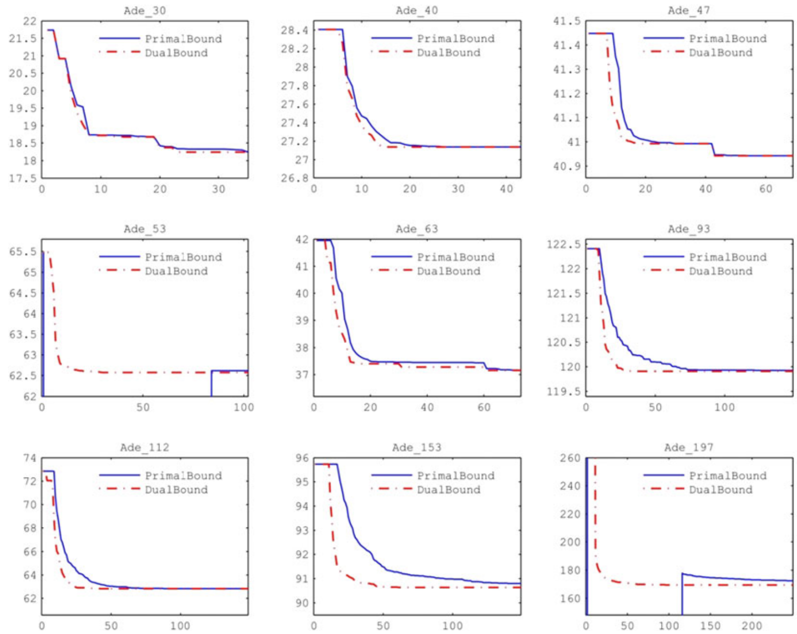

Fig. 2. Behavior of the proposed column generation method when a time window is not allowed. The horizontal axis is number of iterations and the vertical axis is cost.

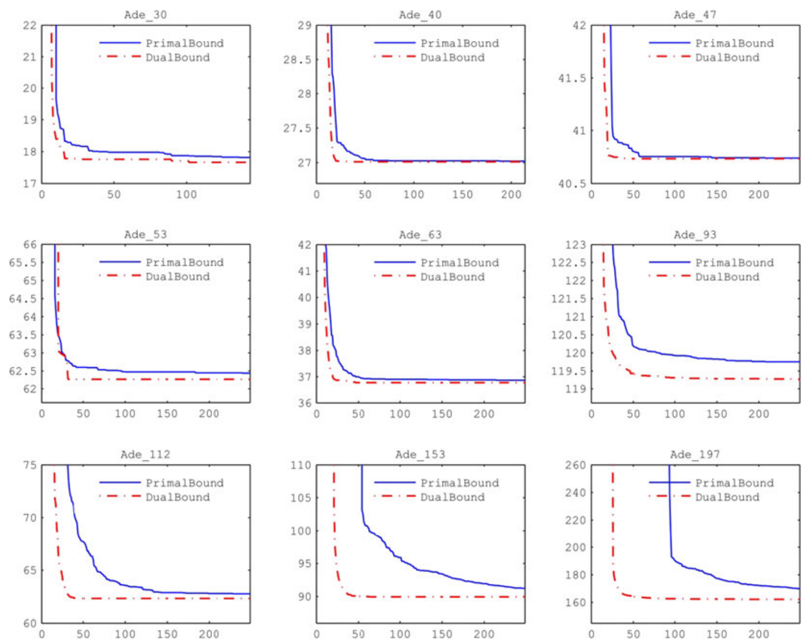

Fig. 3. The behavior of the proposed column generation method when a time window is allowed. The horizontal axis is number of iterations and the vertical axis is cost.

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.

Figures 2 and 3 reflect a deeper investigation into the behavior 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 3 does the exact same job as Figure 2 but for a model with a time window. In Figure 2, and for models 30, 40, 47, and 63, the primal bound at termination was within 0.0% 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 CG.

The terminations of models with a time window can be perceived from Figure 3 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.

5 CONCLUSIONS

This article proposed a mathematical model based on the CG technique to solve 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 IP and MIP. For evaluation, a real database belonging to an active 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, CG 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 CG method attained results around 10 times faster than MIP with around 1% increase in distance.

ACKNOWLEDGMENT

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

REFERENCES

Asbach, L., Dorndorf, U. & Pesch, E. (2009), Analysis, modeling and solution of the concrete delivery problem, *European Journal of Operational Research*, **193**, 820–35.

Bixby, R. E. & Saltzman, M. J. (1994), Recovering an optimal LP basis from an interior point solution, *Operations Research Letters*, **15**, 169–78.

Cao, M., Lu, M. & Zhang, J.-P. (2004), Concrete plant operations optimization using combined simulation and genetic algorithms. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, IEEE, 4204–9.

Chang, E., Ziliaskopoulos, A., Boyce, D. & Waller, S. T. (2001), 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**, 114–23.

Dantzig, G. B. & Wolfe, P. (1960), Decomposition principle for linear programs, *Operations Research*, **8**, 101–11.

De Aragao, M. P. & Uchoa, E. (2003), *Integer Program Reformulation for Robust Branch-and-Cut-and-Price Algorithms*. Techical Report, Departamento de Informatica, PUC-Rio.

Feng, C.-W. & Wu, H.-T. 2000, Using genetic algorithms to optimize the dispatching schedule of RMC cars. *Proceedings of the 17th International Symposium on Automation and Robotics in Construction*, Taipei, Taiwan, 927–32.

Feng, C.-W. & Wu, H.-T. (2006), Integrating fmGA and CYCLONE to optimize the schedule of dispatching RMC trucks, *Automation in Construction*, **15**, 186–99.

Garcia, J., Lozano, S., Smith, K., Kwok, T. & Villa, G. (2002), Coordinated scheduling of production and delivery from multiple plants and with time windows using genetic algorithms. ICONIP'02. *Proceedings of the 9th International Conference on Neural Information Processing*, IEEE, 1153–8.

Goel, A. & Gruhn, V. (2008), A general vehicle routing problem, *European Journal of Operational Research*, **191**, 650–60.

Irnich, S., Desaulniers, G., Desrosiers, J. & Hadjar, A. (2010), Path-reduced costs for eliminating arcs in routing and scheduling, *INFORMS Journal on Computing*, **22**, 297–313.

Kelly, D. J. & Oneill, G. M. (1991), *The Minimum Cost Flow Problem and the Network Simplex Solution Method*, University College Dublin, Graduate School of Business.

Liberatore, F., Righini, G. & Salani, M. (2011), A column generation algorithm for the vehicle routing problem with soft time windows, *4OR*, **9**, 49–82.

Lin, D.-Y., Karoonsoontawong, A. & Waller, S. T. (2011a), A Dantzig-Wolfe decomposition based heuristic scheme for bi-level dynamic network design problem, *Networks and Spatial Economics*, **11**, 101–26.

Lin, D. Y., Valsaraj, V. & Waller, S. T. (2011b), A Dantzig-Wolfe decomposition-based heuristic for off-line capacity calibration of dynamic traffic assignment, *Computer-Aided Civil and Infrastructure Engineering*, **26**, 1–15.

Lin, P.-C., Wang, J., Huang, S.-H. & Wang, Y.-T. (2010), Dispatching ready mixed concrete trucks under demand postponement and weight limit regulation, *Automation in Construction*, **19**, 798–807.

Maghrebi, M., Periaraj, V., Waller, S. T. & Sammut, C. (2014a), Using Benders Decomposition for solving ready mixed concrete dispatching problems, *The 31st International Symposium on Automation and Robotics in Construction and Mining*, UTS, Sydney.

Maghrebi, M., Rey, D., Waller, S. T. & Sammut, C. (2014b), Reducing the number of decision variables in ready mixed concrete for optimally solving small instances in a practical time, *CSCE 2014 General Conference*, Halifax.

Maghrebi, M., Sammut, C. & Waller, S. T. (2015a), Feasibility study of automatically performing the concrete delivery dispatching through machine learning techniques, *Engineering, Construction and Architectural Management*, **22**(5), 573–90.

Maghrebi, M. & Waller, S. T. (2014), Exploring experts decisions in concrete delivery dispatching systems using Bayesian network learning techniques, *2nd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, IEEE, 103–8.

Maghrebi, M., Waller, S. T. & Sammut, C. (2013a), Reconstruction of an expert's decision making expertise in concrete dispatching by machine learning, *Journal of Civil Engineering and Architecture*, **7**, 1540–7.

Maghrebi, M., Waller, S. T. & Sammut, C. (2013b), Scheduling concrete delivery problems by a robust meta heuristic method, *Seventh UKSim/AMSS European Symposium on Computer Modeling and Simulation (EMS)*, 20–22 Nov. 2013, 150–6.

Maghrebi, M., Waller, S. T. & Sammut, C. (2014c), Assessing the accuracy of expert-based decisions in dispatching ready mixed concrete, *Journal of Construction Engineering and Management*, 140(**6**):06014004-1-7.

Maghrebi, M., Waller, S. T. & Sammut, C. (2014d), Sequential meta-heuristic approach for solving large-scale ready-mixed concrete dispatching problems, *Journal of Computing in Civil Engineering*, 04014117-1-11.

Maghrebi, M., Waller, S. T. & Sammut, C. (2015b), Optimality gap of experts' decisions in concrete delivery dispatching, *Journal of Building Engineering*, **2**, 17–23.

Maghrebi, M., Waller, S. T. & Sammut, C. (2015b), *Text Mining Approach for Reviewing the Ready Mixed Concrete Literature*. Tokyo, Japan.

Narayanan, P. K., Rey, D., Maghrebi, M. & Waller, S. T. (2015), Using Lagrangian relaxation to solve ready mixed concrete dispatching problems, *Transportation Research Record: Journal of the Transportation Research Board*, **2498**:84–90.

Naso, D., Surico, M., Turchiano, B. & Kaymak, U. (2007), Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete, *European Journal of Operational Research*, **177**, 2069–99.

Payr, F. & Schmid, V. (2009), Optimizing deliveries of ready-mixed concrete. Logistics and Industrial Informatics, 2009, LINDI 2009. 2nd International, Linz, September 10-12, 2009, 1–6, DOI: 10.1109/LINDI.2009.5258569.

Rothberg, E. (2007), An evolutionary algorithm for polishing mixed integer programming solutions, *INFORMS Journal on Computing*, **19**, 534–41.

Schmid, V., Doerner, K. F., Hartl, R. F. & Salazar-González, J.-J. (2010), Hybridization of very large neighborhood search for ready-mixed concrete delivery problems, *Computers & Operations Research*, **37**, 559–74.

Schmid, V., Doerner, K. F., Hartl, R. F., Savelsbergh, M. W. & Stoecher, W. (2009), A hybrid solution approach for ready-mixed concrete delivery, *Transportation Science*, **43**, 70–85.

Srichandum, S. & Rujirayanyong, T. (2010), Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization, *American Journal of Engineering & Applied Sciences*, **3**, 823–30.

Vacca, I. & Salani, M. (2009), The vehicle routing problem with discrete split delivery and time windows, *Proceedings of the 9th STRC Swiss Transport Research Conference*, Ascona, Switzerland.

Vanderbei, R. J. (2008), *Linear Programming; Foundations and Extensions*, Springer, New York, NY.

Walker, W. E. (1969), A method for obtaining the optimal dual solution to a linear program using the Dantzig-Wolfe decomposition, *Operations Research*, **17**(2), 368–70.

Yan, S. & Lai, W. (2007), An optimal scheduling model for ready mixed concrete supply with overtime considerations, *Automation in Construction*, **16**, 734–44.

Yan, S., Lai, W. & Chen, M. (2008), Production scheduling and truck dispatching of ready mixed concrete, *Transportation Research Part E: Logistics and Transportation Review*, **44**, 164–79.

Yan, S., Lin, H. & Jiang, X. (2012), A planning model with a solution algorithm for ready mixed concrete production and truck dispatching under stochastic travel times, *Engineering Optimization*, **44**, 427–47.