Les Cahiers du GERAD ISSN: 0711–2440

An exact solution approach for bid construction in truckload transportation procurement auctions with side constraints

M. Rekik, G. Desaulniers,
M. Saddoune, I. Elhallaoui

G–2017–51

June 2017

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

An exact solution approach for bid construction in truckload transportation procurement auctions with side constraints

Monia Rekik [a,c,d]
Guy Desaulniers [b,c]
Mohammed Saddoune [e]
Issmail Elhallaoui [b,c]

[a] *Laval University, Departement of Operations and Decision Systems, (Québec) Canada, G1V 0A6*

[b] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[c] *Group for Research in Decision Analysis (GERAD), (Québec), Canada, H3T 2A7*

[d] *Interuniversity Research Centre on Entreprise Networks, Logistics and Transportation (CIRRELT), (Québec), Canada, H3T 1J4*

[e] *Faculté des Sciences et Techniques Mohammadia, Department of Computer Science, Mohammadia 20650, Morocco*

monia.rekik@fsa.ulaval.ca
guy.desaulniers@gerad.ca
Mohammed.Saddoune@gerad.ca
issmail.elhallaoui@gerad.ca

June 2017
Les Cahiers du GERAD

G–2017–51

Abstract: The bid construction problem (BCP) for combinatorial total truckload transportation service procurement auctions consists of determining one or several bids and the ask price for each bid, where a bid is defined by a subset of shipping contracts. In this paper, we consider novel variants of the BCP which include business side constraints, namely, upper limits on the total number of contracts covered by the generated bids and on the number of contracts included in each bid. To solve the BCP variants, we propose a branch-price-and-cut algorithm which is based on techniques that have proven their efficiency on classic vehicle routing problems, and test it on generated BCP instances. Our computational results show that the proposed algorithm identifies optimal solutions within reasonable computing times for the different variants of the BCP, with and without business side constraints. Regarding bids quality, we observe that adding these side constraints may be more or less beneficial for a carrier, depending on its size, and on the values given to the parameters defining these constraints.

Keywords: Bid construction, total truckload transportation, combinatorial auctions, column generation, branch-price-and-cut, business constraints

1 Introduction

Transport operations can be of two types : total truckload (TL) and less-than-truckload (LTL). In TL operations, shipments must be driven directly from pick-up to delivery locations without any intermediate stop. At the opposite, LTL shipments are consolidated in terminals and hubs. Caplice and Sheffi (2006) reported that the TL segment represents more than 78% of the total trucking transportation market in the USA.

In practice, transportation services can be procured either on the spot or the long-term contract markets. Long-term contracts assume a long-term commitment (one to three years) between a shipper and a carrier: the core carriers selection is performed at the strategic level of the planning process. The spot market is rather considered at the operational level, generally when the transportation capacity brought under long-term contracts is insufficient. In this case, the shipper must contract additional capacity on the spot market with relatively high prices. Garrido (2007) reports that most freight transportation services are assigned to core carriers using long-term contracts. Caplice (2007) estimates that the spot market comprises less than 10% of the total TL market.

This paper deals with long-term TL transportation markets. More precisely, we consider an auction-based trading mechanism where carriers compete by submitting combinatorial bids on a set of shipping contracts requested by one or multiple shippers. Auctions can be of different types (Abrache et al., 2007). In auction jargon, this paper considers one-sided reverse combinatorial auctions. In other words, there is only one buyer, the auctioneer, which wants to acquire a number of items and multiple sellers which compete by submitting bids on the buyer requests. In a transportation auction, the auctioneer is a shipper that decided to outsource its transportation activities and bidders are the external carriers invited to participate to the auction. Combinatorial bidding enables bidders to submit a bid on a package of auctioned items rather than on each item separately. We refer the reader to de Vries and Vohra (2003) for a detailed analysis on combinatorial auctions and their merits.

Combinatorial auctions have been proved to be of great interest for transportation service procurement markets. Caplice and Sheffi (2006) report that five software tools proper to transportation markets and incorporating package bids were developed from 1997 to 2003 and used by more than one hundred companies resulting in an average saving of 13% in transportation costs. The success of combinatorial bidding for transportation markets is mostly due to the economy of scope characterizing transport operations. That is, the value of a lane (i.e., an origin-destination pair) to a carrier depends on whether other lanes are won. Carriers are thus interested in lanes forming optimized routes that minimize repositioning costs in their networks. Caplice (2007) reports that a significant portion of the trucking industry costs is due to these repositioning moves.

In general, the auction process requires solving two main decisional problems: a bid construction problem (BCP) and a winner determination problem (WDP). For the BCP, decision makers are the bidders (the carriers) which must decide on the items to bid on (the shipping contracts) and the associated ask price. A WDP aims at determining the winning bids and is solved by the shipper(s) or a third-party acting as the shipper(s) representative. To the best of our knowledge, the majority of the published papers emphasize the WDP, assuming known results for the BCP (Nandiraju, 2006). The few papers on the BCP in combinatorial auctions for TL transportation markets either consider only spot markets (Triki et al., 2014; Chen et al., 2009) or propose exact solution methods yielding large computing times (Lee et al., 2007).

Our paper makes a number of contributions in the field of the BCP for transportation procurement auctions. First, it proposes an exact algorithm to solve a standard BCP for long-term transportation contracts procurement by adapting a number of novel techniques that have proven their efficiency for classic vehicle routing problems (VRPs). Our results prove that the proposed algorithm can solve large instances including up to 173 contracts in relatively short computing times (172 seconds, on average). Second, the proposed approach offers the carrier the possibility to submit multiple bids under different languages without re-solving again the BCP. To the best of our knowledge, the majority of the models proposed in the literature for the BCP generate a single bid or propose to iterate the same decisional process by excluding some contracts. Finally, and most importantly, our paper considers and solves to optimality new variants of BCP where two types of business side constraints are handled: an LTC constraint which Limits the Total number of Contracts covered by the generated bids, and an LBC constraint which Limits the number of Contracts submitted in each Bid. These constraints may derive from explicit rules imposed by the shipper running the auction or from the carrier's strategy or its desire to build a pool of varied bids. Our paper discusses the impact of adding such business side constraints on computing times, on the carrier's profit, and on bids quality while taking into account the carrier size (large versus small carriers). To the best of our knowledge, this paper is again the first to address and analyse these variants.

The remainder of the paper is organized as follows. The next section briefly reviews the literature on bid construction problems for transportation auctions. The standard BCP and the proposed new variants are defined in Section 3. Section 4 describes the proposed solution approach. Section 5 reports our computational results. Section 6 concludes the paper and opens on future research avenues.

2 Literature review

Different approaches have been proposed in the literature to model and solve the BCP in combinatorial transportation auctions. This problem has been modeled as a linear set partitioning/set covering problem (Song and Regan, 2005), a minimum-cost flow problem (Cohn et al., 2008; Chen et al., 2009), or a non-linear integer program (Lee et al., 2007; Triki et al., 2014).

Song and Regan (2005) propose optimization-based approximation algorithms for the BCP in the cases where existing commitments must be managed or not. They assume that the carrier's capacity is unlimited, trucks are available at any location and are not required to return to any central depot. An exhaustive search algorithm is proposed to enumerate all admissible routes. The resulting VRP is then modeled as a set partitioning problem in which each route represents a candidate bid and the objective function minimizes the total empty movement cost. A classic branch-and-bound procedure is used to solve this model.

Wang and Xia (2005) develop two heuristics to solve a BCP where a pick-up time window is associated with each committed and each auctioned lane. The objective is to minimize the total expected empty travel distance given that not all the combinations of lanes are guaranteed to be won. The first heuristic is based on a fleet assignment model where a binary decision variable is assigned to each arc of the carrier network to indicate whether an assignment exists on that arc. The second heuristic is based on the nearest insertion method: lanes are gradually added to vehicle routes based on the extra empty travel distance they generate. These algorithms are tested on a set of instances including up to 10 vehicles and 20 auctioned lanes. The authors report that, on average, the fleet assignment based heuristic generates slightly better bids than the nearest insertion method.

Lee et al. (2007) introduce a non-linear integer program with a quadratic objective function and a quadratic constraint that integrates the generation and the selection of routes including both existing and new contracts. The objective is to maximize the carrier net revenue by adding new lanes to its existing network. A single package of new contracts is generated. A decomposition approach based on column generation (CG) and Lagrangian relaxation is proposed. The results obtained show that the resulting algorithm yields relatively short computing times (less than 100 seconds) for relatively small instances with less than 35 new contracts and 18 existing ones. However, larger instances involving more than 48 new contracts and 25 existing ones require large computing times (more than 20 hours, for example, when the number of new contracts is 131 and the number of existing ones is 42).

In Chang (2009), a so-called bidding advisor is proposed to help the carrier generates desirable bids in one-shot combinatorial auctions for TL spot markets. The spot market assumes a short planning horizon (one week) with known and estimated information on the number of available vehicles, the current locations of all vehicles, the booked, forecasted and auctioned loads, their pick-up and delivery times, their revenues, etc. The problem is modeled as a synergetic minimum-cost flow problem in the sense that the potential synergy effect between consecutive loads on the same vehicle tour is taken into account. A path-based formulation is proposed and solved by CG in which a shortest path algorithm with synergy considerations is designed to generate all optimal paths.

Recently, Triki et al. (2014) address the BCP in a single-round sealed-bid combinatorial auction for a TL spot market. They propose a probabilistic mixed integer programming model that incorporates the package bid price as a decision variable while defining the auction clearing prices as random variables. The clearing price associated with a package bid represents the lowest price offered by the competitors for that package. The probabilistic constraints enable to guide the selection of the package whose price is guaranteed to be less than the corresponding clearing price by a certain probability threshold. The BCP is formulated as a fleet management problem based on a time-space network. The authors report that the CPLEX branch-and-cut algorithm required unreasonable computing times to solve the proposed model even for problems with 20 auctioned loads. Thus, they develop two greedy heuristics: a sequential ascending method and a sequential descending method. The ascending method starts with an empty set of contracts and computes the incremental profit induced by adding each contract separately. The contract with the largest positive incremental profit is then added to the set of contracts and the process is iterated until either all the contracts are added or no contracts with a positive incremental profit exist. The descending method is symmetric as it starts with a set that includes all contracts and removes from this set the contract that increases the most the profits when removed. The computational results obtained show that the ascending heuristic clearly outperforms the other for all tested instances.

From this literature review, it clearly appears that more elaborated exact algorithms need to be developed to solve large BCPs to optimality in relatively short computing times. Moreover, almost all published papers propose a solution approach that generates a single package bid and consider standard contexts where no side constraints related to the auction rules or the shipper restrictions are handled. Our paper attempts to fill these gaps by: (1) proposing an efficient exact solution algorithm based on a number of novel techniques that have proven their efficiency for classic VRPs, (2) considering new variants of the BCP where two types of business side constraints are handled: the LTC and LBC constraints. The carrier is offered the possibility to submit a single package bid or multiple OR package bids in all cases.

3 Problem definition and formulation

In a BCP, a carrier participating to a combinatorial transportation auction needs to determine the set of profitable shipping contracts to bid on. The carrier existing network should be taken into account, i.e., there is a set of contracts to which the carrier has already committed itself and, thus, that must be ensured. The question is: which new contracts the carrier should bid on and incorporate in its current network?

In this section, we first describe the problem context and assumptions for a standard BCP as classically defined in the literature. A route-based mathematical formulation is then proposed for the standard BCP. Second, we define new variants of the BCP that incorporate side constraints related to the auction process. We discuss the relevance of such constraints and show how the model proposed for a standard BCP can be adapted to handle them.

3.1 Context and assumptions

A shipping contract k is described by a three-uple (o_k, d_k, v_k), where o_k and d_k represent the distinct origin and destination locations, respectively, and v_k is the volume to be carried between o_k and d_k. The set of TL contracts K is partitioned into two subsets K^e and K^n, where K^e includes all the pre-existing contracts that the carrier must satisfy and K^n represents the set of the new contracts traded within the auction. Let $H = \bigcup_{k \in K} \{o_k, d_k\}$ be the set of all locations excluding the depot. Note that a location appears only once in H if it corresponds to the origin or the destination location of more than one contract. Furthermore, let $H^o \subseteq H$ (resp. $H^d \subseteq H$) be the subset of locations which correspond to the origin (resp. destination) of at least one contract $k \in K$.

Given the TL context, each contract k must be served by a single vehicle which picks up the whole volume v_k at the origin location o_k and goes directly to the destination location d_k to deliver it. The carrier possesses a homogeneous fleet composed of M vehicles. Each vehicle is characterized by a capacity which is considered sufficiently large to service any contract, a maximum route length T and a fixed price f. All vehicles are located in the depot s at the beginning of their route and must return to it at the end of the route.

Let $G = (N, A)$ be a directed multi-graph, where N is the node set and A the arc set. Set N contains two nodes representing the depot at the start (node s) and at the end (node e) of a route, and all locations in H, i.e., $N = \{s, e\} \cup H$. Set A comprises the following arcs: (1) for each contract $k \in K$, there is a *contract* arc from node o_k to node d_k that represents performing contract k; (2) for each location $h \in H^o$, there is a *start* arc linking node s to h which represents an empty movement (i.e., without load) at the beginning of a vehicle route; (3) similarly, for each location $h \in H^d$, there is an *end* arc linking h to node e and representing an empty return to the depot at the end of a route; (4) for each pair of locations $(h, h') \in H^d \times H^o$ with $h \neq h'$, there is a *deadhead* arc linking h to h' which represents an empty movement between these two locations. We denote by $A^C = \{(o_k, d_k) : k \in K\}$, $A^S = \{(s, h) : h \in H^o\}$, $A^E = \{(h, e) : h \in H^d\}$, and $A^D = \{(h, h') : h \in H^d, h' \in H^o, h \neq h'\}$ the sets of contract, start, end, and deadhead arcs, respectively. Thus, $A = A^C \cup A^S \cup A^E \cup A^D$.

With each arc $a \in A$, we associate a traveling cost c_a as well as a traveling time t_a required to traverse it. If $a \in A^C$ represents a contract $k \in K$, we also define a price $p_a = p_k$ that is the price the carrier would receive for serving the corresponding contract k. These prices are assumed to be known in advance. They are given for the existing contracts. When iterative auctions are used, exact or approximate prices for the new contracts can be deduced after each round through the auction pricing process (for more details, see, e.g., Kwasnica et al. (2005)). In single-round (also called one-shot) auctions, new contract prices can be estimated from the spot market or the carrier's experience.

Let R be the set of feasible routes. A route is feasible if: (i) it starts and ends at the depot, (ii) its total duration does not exceed T, and (iii) it does not cover the same contract more than once. Formally, a feasible route $r \in R$ corresponds to a path in G which is defined by a sequence of W arcs $(a_1, a_2, \ldots, a_w)$ such that (i) $a_1 \in A^S$, $a_W \in A^E$, (ii) $\sum_{w=1}^{W} t_{a_w} \leq T$, and (iii) for all $w \in \{2, \ldots, W-1\}$ such that $a_w \in A^C$, there exists no $w' \in \{2, \ldots, W-1\}$ such that $w' \neq w$ and $a_{w'} = a_w$. Its cost is given by: $c_r = f + \sum_{w=1}^{W} c_{a_w}$ and its net revenue by: $g_r = \sum_{w=1}^{W} p_{a_w} - c_r$. Condition (iii) above is termed the elementarity requirements and a path that satisfies these requirements is said to be *elementary* (although it may visit the same node more than once).

3.2 Mathematical model for a standard BCP

For each feasible route $r \in R$, we define a binary variable x_r that equals 1 if route r is chosen and 0, otherwise. For each route $r \in R$ and each contract $k \in K$, we also define a binary parameter δ_{rk} equal to 1 if route r serves contract k and 0 otherwise.

The BCP can be formulated as the following integer program:

$$(IP) \qquad \max \sum_{r \in R} g_r x_r \tag{1}$$

$$\text{s.t.} \qquad \sum_{r \in R} \delta_{rk} x_r = 1, \qquad \forall k \in K^e \tag{2}$$

$$\sum_{r \in R} \delta_{rk} x_r \leq 1, \qquad \forall k \in K^n \tag{3}$$

$$\sum_{r \in R} x_r \leq M, \tag{4}$$

$$x_r \in \{0, 1\}, \qquad \forall r \in R. \tag{5}$$

The objective function (1) maximizes the carrier net revenue. Constraints (2) ensure that each existing contract is served exactly once. Constraints (3) ensure that each new contract is served at most once. Constraint (4) limits the number of vehicles according to the carrier fleet size. Finally, constraints (5) are integrality constraints.

3.3 Bids structure

Using the route-based model (IP) enables a carrier to submit either a single bid or multiple OR bids. Let $X^* = (x_r^*)$ denote an optimal solution to (IP).

1. A single bid b^s includes all the profitable new contracts covered by the optimal solution X^*, that is, the subset of contracts $K_{b^s} = \{k \in K^n : \sum_{r\in R} \delta_{rk} x_r^* = 1\}$.
2. A maximum of M OR bids b^r are generated. Each OR bid includes all the profitable new contracts covered by a route r in X^*, that is, the subset of contracts $K_{b^r} = \{k \in K^n : \delta_{rk} x_r^* = 1\}$. Observe that $K_{b^r} \cap K_{b^{r'}} = \emptyset$ for all $r, r' \in R$ such that $r \neq r'$ and $x_r^* = x_{r'}^* = 1$.

Mainly, a single bid b^s, if won, ensures the carrier to win all the contracts submitted in it. So, the net profit as computed by the BCP will be realized if the bid wins. On the opposite, OR bids offer a carrier the possibility to win all or some of these bids. So, there is no guarantee that all the submitted contracts are won. Despite this risk, OR bids increase the carrier's chances to win contracts. Indeed, to determine the winning bids, the shipper solves the WDP on a set of multiple bids submitted by the competing carriers. A WDP generally incorporates set partitioning constraints to ensure that a contract must be allocated to exactly one bid. With a single bid, the carrier has a unique chance to win contracts and could be easily discarded. On a counterpart, multiple OR bids are more likely to match with the other competing bids. It is noteworthy that more complex bids combining single bid b^s with all or a part of OR bids b^r are possible. For example, the carrier could submit a XOR-of-OR bid of the form: b^s XOR (b^{r_1} OR b^{r_2} OR …). For a detailed discussion on OR, XOR and other bid languages, see, e.g., Nisan (2000).

Regarding the ask price, we propose a pricing approach that gives the carrier a price interval $[\underline{P}_b, \overline{P}_b]$ for a bid b rather than a unique fixed price P_b. Depending on the context and on its experience, the carrier will decide which price to submit. The way the minimum and maximum bid prices are determined depends on the bid structure.

For a single bid b^s, determining the maximum bid price $\overline{P}_b$ is quite simple: to have a chance to win the desired contracts, the carrier should not ask for a price that is greater than the sum of the prices associated with the new contracts, that is, $\overline{P}_{b^s} = \sum_{k\in K_{b^s}} p_k$. The minimum bid price $\underline{P}_{b^s}$ is computed as the incremental cost resulting from adding the new contracts in K_{b^s} to the carrier existing network. Formally, $\underline{P}_{b^s} = \sum_{r\in R} c_r x_r^* - C(K^e)$, where $C(K^e)$ is the optimal transportation cost incurred by the carrier for serving only its existing contracts. Note that this optimal cost can be computed by considering model (IP) with $K^n = \emptyset$. Obviously, a bid b^s will be submitted by a carrier if it generates a positive profit. Since it covers all the profitable new contracts as determined by model (IP), one will always have $\sum_{r\in R} c_r x_r^* - C(K^e) \leq \sum_{k\in K_{b^s}} p_k$.

When multiple OR bids are submitted, determining the minimum and maximum bid prices is more challenging and depends on the carrier risk behaviour. The main reason is that with OR bids, the carrier may win only a subset of the new contracts identified by model (IP). So, there is no guarantee that winning only a subset of new contacts would be still profitable for the carrier. In this case, we propose to compute for each OR bid b^r the optimal incremental cost obtained by adding to the carrier existing network only the new contracts covered by b^r. This is done by computing the difference between $C(K^e \cup K_{b^r})$ and $C(K^e)$. Here again, $C(K^e \cup K_{b^r})$ is obtained through solving model (IP) with an empty set of new contracts and a modified set of exiting contracts that includes in addition to the contracts in K^e, the new contracts in K_{b^r}. Here, it may happen that the incremental cost associated with bid b^r is not compensated by the additional revenue obtained from serving the new contracts covered by b^r, that is, $C(K^e \cup K_{b^r}) - C(K^e) > \sum_{k\in K_{b^r}} p_k$. A risk-averse carrier would chose in this case, a minimum bid price equal to the incremental cost $C(K^e \cup K_{b^r}) - C(K^e)$ and a maximum bid price slightly larger, being conscious that such bid is likely to lose the auction. A risk-seeker carrier accepts to submit a minimum bid price equal to $\sum_{k\in K_{b^r}} p_k$, knowing that, if only this bid wins, it would incur monetary losses. Observe that more elaborated techniques for managing OR bids could be proposed. We leave them for a future work.

3.4 BCP with side constraints

Caplice and Sheffi (2006) report that, in practice, shippers generally impose a variety of business constraints when determining winning bids. These business constraints depend on the shipper strategy and risk behavior. For example, a shipper may limit the number of winning carriers in an auction to alleviate managing its transport operations, or to become a significant customer to winning carriers. Caplice and Sheffi (2006) also report that selecting a too small number of carriers at the strategic level may result in large unexpected costs at the operational level if some of these carriers go out of business. To increase their chances of winning, participating carriers should be aware of such business constraints and implicitly or explicitly incorporate them while solving the BCP.

In what follows, we study two types of side constraints: the LTC and the LBC constraints. These constraints may derive from explicit rules imposed by the shipper running the auction. They may be also due to the carrier's strategy or to its desire to build a pool of varied bids.

3.4.1 LTC constraint

The LTC constraint limits the total number of contracts covered by the carrier bids. Such a constraint is relevant in case the shipper explicitly imposes a limit on the number of contracts allocated to carriers to avoid relying too heavily on a single partner. Besides, the carrier may add such a constraint of its own to avoid bidding on a large number of contracts if it does not know enough about the shipper (e.g., first experience) or looks for diversifying its shippers base.
To handle this restriction, model (IP) is modified by adding the following constraint:

$$\sum_{r \in R} \sum_{k \in K^n} \delta_{rk} x_r \leq \beta |K^n|, \tag{6}$$

where β denotes the maximum percentage of new contracts the carrier can bid on.

3.4.2 LBC constraint

The LBC constraint limits the number of contracts submitted in a bid to a pre-specified value, denoted α, in the following. This constraint is relevant in case OR bids are used. As already mentioned, OR bids offer the carrier the possibility to submit diversified and multiple bids that are more likely to match with other bids from the other competing carriers. Imposing a limit on the number of contracts covered in an OR bid offers the carrier the opportunity to submit and test the relevance of "smaller bids". To handle this restriction, only routes including up to α new contracts are considered in set R.

4 Solution algorithm

The set R of all feasible routes is generally very large and the resulting model (IP) cannot be tackled directly by a commercial mixed integer programming solver. To solve it, we propose a branch-price-and-cut algorithm (see Barnhart et al. (1998); Desaulniers et al. (2005); Lübbecke and Desrosiers (2005)), that is, a branch-and-bound method in which upper bounds in the search tree (for a maximization problem) are computed by CG and cutting planes are generated to tighten the linear relaxations.

The proposed branch-price-and-cut algorithm is composed of state-of-the-art tools that have proven to be successful for solving various VRPs (see Toth and Vigo (2014)). In this respect, our methodological contribution is rather limited to gathering these tools into an efficient algorithm for the BCP. Therefore, in this section, we do not provide all technical details but refer the reader to appropriate references whenever needed.

4.1 Column generation

At each node of the branch-and-bound search tree, an upper bound is computed by solving the linear relaxation of model (IP) (including the LTC constraint (6)), augmented by the applicable cutting planes and branching decisions. In a CG context, this linear relaxation is called the *master problem (MP)* and can be formulated as follows:

$$(MP) \qquad \max \sum_{r \in R} g_r x_r \tag{7}$$

$$\text{s.t.} \qquad \sum_{r \in R} \delta_{rk} x_r = 1, \qquad \forall k \in K^e \qquad (\lambda_k) \tag{8}$$

$$\sum_{r \in R} \delta_{rk} x_r \leq 1, \qquad \forall k \in K^n \qquad (\lambda_k) \tag{9}$$

$$\sum_{r \in R} x_r \leq M, \qquad (\pi) \tag{10}$$

$$\sum_{r \in R} \sum_{k \in K^n} \delta_{rk} x_r \leq \beta |K^n|, \qquad (\gamma) \tag{11}$$

$$\sum_{r \in R} q_{rl} x_r \leq u_l, \qquad \forall l \in L \qquad (\sigma_l) \tag{12}$$

$$x_r \geq 0, \qquad \forall r \in R, \tag{13}$$

where $\left(\lambda_k\right)_{k \in K^e}$, $\left(\lambda_k\right)_{k \in K^n}$, π, γ, and $\left(\sigma_l\right)_{l \in L}$ denote the dual variables associated with constraints (8)–(12), respectively, and the set of constraints (12) corresponds to the applicable cutting planes and branching decisions. We denote by L the index set of these constraints, by q_{rl} the contribution of route r to constraint $l \in L$ and by u_l its right-hand side.

To solve the MP, a CG method is applied. Such a method is iterative and solves at each iteration a restricted master problem (RMP) and one or several subproblems. The RMP is the master problem restricted to a subset of its x_r variables. Solving it provides a primal and a dual solution. Given this dual solution, the role of the subproblems is to determine whether or not there exist positive reduced cost variables (columns) x_r that are not considered in the current RMP. If none exists, the solution of the current RMP is optimal for the MP and the CG algorithm stops. Otherwise, positive reduced cost columns identified by the subproblems are added to the RMP before starting a new iteration.

For the BCP, there is a single subproblem which aims at finding a route in R with the largest reduced cost. Given a dual solution to the current RMP defined by the dual values $\left(\lambda_k^*\right)_{k \in K}$, π^*, γ^*, and $\left(\sigma_l^*\right)_{l \in L}$, the reduced cost associated with a route $r \in R$ is given by:

$$\bar{g}_r = g_r - \sum_{k \in K} \delta_{rk} \lambda_k^* - \pi^* - \sum_{k \in K^n} \delta_{rk} \gamma^* - \sum_{l \in L} q_{rl} \sigma_l^*.$$

The subproblem is then defined as:

$$(SP) \qquad \max_{r \in R} \ \bar{g}_r \,. \tag{14}$$

This subproblem is an elementary longest path problem with resource constraints which is equivalent to an elementary shortest path problem with resource constraints (ESPPRC, see Feillet et al. (2004); Irnich and Desaulniers (2005)) if we minimize the negative of the reduced cost instead of maximizing the reduced cost. This ESPPRC is defined on the multi-graph G. Every feasible route $r \in R$ corresponds to a path from node s to node e in G. However, not all $s - e$ paths in G corresponds to a feasible route. Resource constraints are thus considered to enforce feasibility. A resource is a quantity that varies along a path and whose value is restricted to fall within a given interval, called a resource window, at each node of the network. In our case, there is one resource to compute the duration of a route (with resource window $[0, T]$ on all nodes), one resource for each contract $k \in K$ to compute the number of times that the corresponding contract arc is traversed (with resource window $[0, 1]$ on all nodes), and another resource to compute the total number of new contracts covered (with resource window $[0, \alpha]$ on all nodes) when the LBC constraint is applicable.

To ensure that the cost of a path in G representing a route $r \in R$ is equal to g_r, we must define the appropriate arc reduced cost $\bar{g}_a$ for each arc $a \in A$. We will assume here that parameters q_{rl} in constraints (12) can all be written as: $q_{rl} = \sum_{a \in A(r)} q_{al}$, where $A(r)$ is the set of arcs traversed by the path representing route r and q_{al} is the contribution of arc a to constraint l. In this case, the (reduced) cost $\bar{g}_a$ of arc $a \in A$ is given by:

$$\bar{g}_a = \begin{cases} -f - c_a - \pi^* - \sum_{l \in L} q_{al}\sigma_l^* & \text{if } a \in A^S \\ -c_a - \sum_{l \in L} q_{al}\sigma_l^* & \text{if } a \in A^E \cup A^D \\ p_a - c_a - \lambda_{k(a)}^* - \sum_{l \in L} q_{al}\sigma_l^* & \text{if } a \in A^{CE} \\ p_a - c_a - \lambda_{k(a)}^* - \gamma^* - \sum_{l \in L} q_{al}\sigma_l^* & \text{if } a \in A^{CN}, \end{cases} \tag{15}$$

where A^{CE} and A^{CN} denote the subsets of existing and new contract arcs, respectively, and $k(a) \in K$ indicates the contract associated with arc a. It is easy to verify that, with these arc costs, $\bar{g}_r = \sum_{a \in A(r)} \bar{g}_a$.

The ESPPRC can be solved by a labeling algorithm (see Irnich and Desaulniers (2005)). In this algorithm, a path from node s to a node $i \in N$ is represented by a label $E = (E^{cost}, E^{dur}, (E^{cont_k})_{k \in K}, E^{new})$, where E^{cost} is the reduced cost of this path, E^{dur} its duration, E^{cont_k} indicates whether or not contract k has been covered by this path, and E^{new} is the number of new contracts covered (if the LBC constraint is considered). Starting with an initial label $E = (0, 0, (0)_{k \in K}, 0)$ at node s, labels are extended through network G using label extension functions. Labels associated with infeasible paths (i.e., for which a label component exceeds the corresponding resource window upper bound) are discarded. To avoid enumerating all feasible paths, a dominance rule that discards all non-Pareto-optimal labels is applied. See Irnich and Desaulniers (2005) for more details.

To speed up the ESPPRC solution process, a bidirectional labeling algorithm (Righini and Salani, 2006) can be used. In this three-step algorithm, labels are first extended forwardly in G starting from node s until reaching half of the maximum duration T, i.e., a label E is extended if $E^{dur} < T/2$. Second, labels are extended backwardly from node e using backward label extension functions until reaching half of the maximum duration, i.e., a label E is extended if $E^{dur} < T/2$ assuming that $E^{dur} = 0$ in the initial label at node e. In the final step, forward and backward labels associated with the same node are merged to yield complete feasible $s - e$ paths.

The ESPPRC is known to be NP-hard in the strong sense (Dror, 1994). Consequently, solving it at each iteration can be highly time-consuming. Several works have shown that it was much more efficient to use a relaxed subproblem and generate routes that contain cycles (i.e., cover the same contract more than once). In this case, the coefficient δ_{rk} is redefined as the number of times that route r covers contract k. As such a route cannot be part of a feasible integer solution, it will be excluded through branching and cutting if it is involved in a fractional-valued solution. We propose to use a very efficient subproblem relaxation, called the ng-route relaxation (Baldacci et al., 2011), which is defined as follows for the BCP. For each contract $k \in K$, let NG_k be the neighborhood of k which contains a subset of the contracts in K including k itself. A route r is an ng-route if, for any sequence $k_1 - k_2 - \cdots - k_m$ of consecutive contracts covered by r, either $k_1 \neq k_m$ or there exists $j \in \{2, \ldots, m-1\}$ such that $k_1 \notin NG_{k_j}$. If $NG_k = K$ for all $k \in K$, then any ng-route is elementary. On the other hand, if $NG_k = \{k\}$, then all cycles are allowed in an ng-route. For our tests, all neighborhoods NG_k are of size 10 and contain k and the contracts that can be reached as soon as possible after completing contract k. To handle ng-routes, the labeling algorithm must be slightly modified. For details, see, e.g., Desaulniers et al. (2014).

4.2 Cutting planes

We consider only one family of valid inequalities, namely, the subset row inequalities introduced in Jepsen et al. (2008). They are rank-1 Chvátal-Gomory inequalities derived from constraints (2) and (3). As done in several other papers including Jepsen et al. (2008), we restrict the separation (which is performed by enumeration) to subsets of three constraints and equal Chvátal-Gomory multipliers of value 0.5. Under these assumptions, the subset row inequalities can be expressed as follows:

$$\sum_{r \in R} \Big(\big(\sum_{k \in U} \delta_{rk} \big) \bmod 2 \Big) x_r \leq 1, \qquad \forall U \subseteq K : |U| = 3. \tag{16}$$

If R contains only elementary routes, such an inequality stipulates that any feasible solution contains at most one route covering two or three contracts in subset U. If R contains ng-routes, these inequalities are still valid.

Cutting planes (16) take the form of constraints (12). However, for such a constraint l (defined for a subset U) and a route r, the coefficient q_{rl} cannot be computed as a sum of the contributions of the arcs in route r ($q_{rl} = \sum_{a\in A(r)} q_{al}$). Hence, its dual variable cannot be considered in the arc reduced costs $\bar{g}_a$, $a \in A$, defined by (15). To subtract it from a path reduced cost in the labeling algorithm, an additional resource is required for each generated subset row cut (16) to compute the number of times that the path traverses an arc representing a contract in U. The label extension functions and the dominance rule must be modified to take these cuts into account. See Jepsen et al. (2008) and Desaulniers et al. (2011) for further details.

4.3 Branching

As it is well known, it is difficult to branch directly on the MP variables x_r, $r \in R$, because imposing the decision $x_r = 0$ on one branch requires to forbid the generation of route r by the subproblem. Nevertheless, numerous types of branching decisions are applicable. Based on some preliminary test results, we decided to branch on the following entities that are listed in order of priority.

1. Total number of vehicles used;
2. Total number of new contracts covered;
3. Whether or not a new contract should be covered;
4. The total flow into a node of N;
5. The total flow between two consecutive contracts;
6. The flow traversing an arc of A.

All decision types 1, 2, 3, 4, and 6 are imposed by modifying an existing constraint, adding a constraint (12), or removing a node or an arc in G. Decisions of type 5, which are also called inter-task or follow-on constraints in the literature, require a modification of the labeling algorithm as discussed in Irnich and Desaulniers (2005). Note that, among these types, types 5 and 6 are necessary to ensure a complete enumeration process.

The search tree is explored using a depth-first search strategy.

5 Experimental study

The objectives of our experimental study are threefold. First, we want to assess the computational performance of the proposed solution algorithm for both standard BCPs and BCPs with side constraints. Second, we aim to study the impact of adding these side constraints on the solution quality in terms of the potential profit that could be realized by the carrier and its degree of competitiveness. The degree of competitiveness is measured through the percentage of auctioned contracts covered by the generated bids and the number of alternative OR bids. A sensitivity analysis is conducted in which we vary the values of the β and α parameters for the LTC and LBC constraints, respectively. Our third objective is to measure the benefits/drawbacks of adding these side constraints with regard to the carrier size. In other words, is the LTC constraint more/less beneficial for small carriers than large ones? For which values of β? What about the LBC constraint?

5.1 BCP test instances

Our experimental study considers a set of 20 basic instances defined by fixing the number of cities (a city may act as an origin or a destination location of an existing or new contract), the number of new and existing contracts, the carrier size, and the fixed cost. The number of cities, denoted n, is first selected from existing locations in Canada and the USA. Then, different pick-up and delivery locations are randomly chosen from this set of n cities and assigned to the required number of existing and new contracts. Five combinations of the number of cities, number of new contracts and number of existing contracts are considered. They are generated so that the larger the number of cities is, the wider the territory covered by the contracts is.

Google Map is used to compute real traveling times and distances for each arc in the generated network. The traveling cost c_a associated with arc $a \in A$ is computed as $c_a = 0.75 \times D_{i(a),j(a)}$, where $i(a)$ (resp. $j(a)$) is the origin (resp. destination) location associated with arc a and $D_{i,j}$ is the distance between locations i and j. Finally, for each contract k, either existing or new, the net profit is computed as $p_k = v_k \times 0.75 \times D_{o_k,d_k}$, where v_k is randomly generated within the interval $[5, 60]$.

Two types of carriers are considered for each of these five combinations: small and large. A small carrier possesses a fleet size large enough to serve at least 20% of the new contracts. The fleet size of a large carrier enables satisfying 100% of new contracts for the small networks (i.e., when $n \leq 10$) and at least 50% of new contracts for large ones (i.e., when $n = 12$ or 15). Finally, for each combination and each carrier type, two fixed costs are considered ($f = 500$ or 1000). The limit on the total tour duration is fixed to 1500 minutes for all settings.

In the following, an instance is denoted as n-CS-f, where CS refers to the carrier size ('L' for large carriers, and 'S' for small ones). Observe that a same value for n implies the same sets of existing and new contracts. Table 1 reports the parameter values for each instance.

All tests were performed on a linux PC machine equipped with an Intel Core i7-4770 clocked at 3.4 GHz, using a single processor. The implementations are coded in C and use the GENCOL CG library (version 4.5) and the linear programming solver of CPLEX 12.6 for solving the RMPs.

Table 1: BCP test instances

Instance	Nb. cities (n)	# contracts New	# contracts Existing	Fleet size	Fixed cost (f)
6-L-500	6	24	12	4	
8-L-500	8	35	18	7	
10-L-500	10	48	25	10	500
12-L-500	12	76	28	15	
15-L-500	15	131	42	26	
6-L-1000	6	24	12	4	
8-L-1000	8	35	18	7	
10-L-1000	10	48	25	10	1000
12-L-1000	12	76	28	15	
15-L-1000	15	131	42	26	
6-S-500	6	24	12	2	
8-S-500	8	35	18	4	
10-S-500	10	48	25	7	500
12-S-500	12	76	28	10	
15-S-500	15	131	42	18	
6-S-1000	6	24	12	2	
8-S-1000	8	35	18	4	
10-S-1000	10	48	25	7	1000
12-S-1000	12	76	28	10	
15-S-1000	15	131	42	18	

5.2 Results for the standard BCP

Table 2 reports the results obtained for a standard BCP. The second column in Table 2 displays the computing time, in seconds, required by the proposed branch-price-and-cut algorithm to solve the instance to optimality. The third column gives the optimal net profit that could be realized by the carrier if all the generated bids are submitted and allocated to the carrier after solving the WDP. The column 'Cov.' reports the percentage of new contracts that are covered by the generated bids. This information is reported so that one can see the output of a standard BCP without the LTC constraint. The column '# bids' gives the number of OR bids generated which corresponds to the number of routes that include at least one new contract. Finally, we report under the column 'Max cont per bid' the maximum number of new contracts covered by an OR bid that is used to analyze the output of a standard BCP without the LBC constraint.

Table 2: Results for the standard BCP

Instance	Time (s)	Profit	Cov.(%)	# bids	Max cont per bid
6-L-500	13.7	22,283	100.00	4	9
8-L-500	8.7	33,734	94.29	7	7
10-L-500	14.3	42,634	77.08	10	6
12-L-500	202.8	61,022	73.68	15	6
15-L-500	730.2	97,750	64.89	26	5
6-L-1000	5.2	20,283	100.00	4	7
8-L-1000	8.0	30,234	94.29	7	7
10-L-1000	17.8	37,634	77.08	10	5
12-L-1000	134.0	53,522	73.68	15	6
15-L-1000	706.4	84,750	64.89	26	5
6-S-500	24.4	13,374	41.67	2	5
8-S-500	257.5	21,700	45.71	4	6
10-S-500	61.8	32,993	50.00	7	7
12-S-500	115.6	45,363	50.00	10	6
15-S-500	282.3	71,611	41.22	18	6
6-S-1000	20.4	12,374	41.67	2	7
8-S-1000	354.3	19,700	45.71	4	6
10-S-1000	55.0	29,493	50.00	7	6
12-S-1000	144.1	40,363	50.00	10	7
15-S-1000	283.9	62,611	41.22	18	7

As one can see from Table 2, almost all the instances are solved to optimality in relatively small computing times. The average time is 172 seconds and the largest one, 730.2 seconds, is required to solve instance 15-L-500 corresponding to a network with 131 new contracts, 42 existing ones, and 26 vehicles. For large carriers, the computing times clearly increase with the network size (the number of existing and new contracts) and are less sensitive to the value of the fixed cost. For small carriers, the same behavior is observed, except for instances 8-S-500 and 8-S-1000 (with 35 new and 18 existing contracts). Finally, we observe that instances corresponding to small networks ($n = 6, 8, 10$) are easier to solve for large carriers than for small ones. The trend is reversed for large networks ($n = 12, 15$).

Regarding the solution quality, the net profit decreases when the fixed cost increases, independently of the carrier size. This was predictable since all the available vehicles are used in all the instances. So, the difference in the net profit essentially comes from the difference in the fixed costs. We observe also that a large carrier realizes higher profits than a small carrier. This is also predictable since it has the fleet capacity to cover more new contracts (as depicted in the column 'Cov').

The results of Table 2 also show that when the LTC constraint is not considered, large carriers may win all the new contracts or at least 64.89% of them. Small carriers cannot win more than 50% of the auctioned contracts because of their limited fleet capacity. Hence, from the shipper (the auctioneer) side, imposing an LTC constraint as an auction rule helps avoiding that a single carrier wins the lion's share. From the carrier perspective, especially small ones, the presence of an LTC constraint as an auction rule would increase their chance to win contracts even if large carriers are in the landscape.

Finally, from Tables 1 and 2, we observe that the number of alternative OR bids is always equal to the number of available vehicles. So, large carriers have the opportunity to submit more OR bids than small ones. At first glance, submitting more bids definitely gives the carrier more chances to win new contracts. However, this could not be the case all the time, especially when the number of contracts in the bid is too large, making it less matchable with other competing bids. This is the case for instance 6-L-500, for example, where bids including up to nine contracts are generated where as the total number of auctioned contracts is 24. Adding the LBC constraint may thus be beneficial in such contexts.

In the following sections, we study the impact of considering LTC and LBC constraints on computing times and solution quality.

5.3 Impact of side constraints on computing times

Table 3 reports the computing times in seconds required by the proposed solution algorithm under three bidding contexts. The first context corresponds to the standard BCP. The second context, denoted BCP-β, refers to a BCP with the LTC constraint. Two values of the β parameter are tested: $\beta = 20\%$ and $\beta = 50\%$. The third context, denoted BCP-α, corresponds to a BCP with the LBC constraint. Five values are tested for the α parameter: $\alpha = 2, 3, 4, 5, 6$. The last line in Table 3 gives the average computing time over the 20 instances for each context and each value of the corresponding parameter, when applicable.

Table 3: Impact of side constraints on computing times

	Standard	BCP-β		BCP-α				
Instance	BCP	20%	50%	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$	$\alpha = 6$
6-L-500	13.7	3.0	2.4	0.1	0.1	0.2	1.4	5.0
8-L-500	8.7	0.1	0.5	0.1	0.1	0.4	21.5	5.5
10-L-500	14.3	0.1	1.1	0.2	12.7	30.7	16.3	25.4
12-L-500	202.8	0.1	6.8	0.5	0.8	714.2	534.3	223.0
15-L-500	730.2	0.2	0.8	0.4	785	9733.8	710.7	771.6
6-L-1000	5.2	1.0	3.6	0.1	0.1	0.3	0.5	51.7
8-L-1000	8	0.3	5.2	0.1	0.2	0.3	20.4	7.6
10-L-1000	17.8	0.2	1.4	0.2	16.2	34.7	23.9	27.6
12-L-1000	134	12.7	6.6	0.5	0.8	800.9	770.2	174.8
15-L-1000	706.4	0.1	0.9	0.5	790	9317.1	787.5	816.5
6-S-500	24.4	2.0	24.6	2.8	4.0	201.8	1.0	0.7
8-S-500	257.5	0.2	413.4	0.4	8.0	305.1	479.8	525.9
10-S-500	61.8	0.2	61.4	4.8	81.0	52.5	83.6	104.3
12-S-500	115.6	0.1	74.2	4.4	205.7	178.7	268.4	230.8
15-S-500	282.3	0.1	307.9	1.5	1416.9	81.2	298.3	341.3
6-S-1000	20.4	5.7	20.4	3.4	2.6	248.6	1.1	1.6
8-S-1000	354.3	0.3	347.7	0.4	8.0	303.3	467.3	525.8
10-S-1000	55	0.3	58.0	6.5	90.6	65.8	85.9	73.7
12-S-1000	144.1	10.6	147.2	4.2	201.3	195.6	69.5	362.6
15-S-1000	283.9	0.1	299.7	1.5	1453.2	81.1	297.8	344.9
Avg.	**172.0**	**1.9**	**89.2**	**1.6**	**253.9**	**1117.3**	**247.0**	**231.0**

The results of Table 3 show that computing times remain reasonable for the majority of the instances in the presence of either the LTC or the LBC constraint. Incorporating the LTC constraint drastically reduces computing times for large carriers, when compared to a standard BCP. For small carriers, computing times considerably decrease when $\beta = 20\%$ but tend to either increase or decrease when $\beta = 50\%$. Indeed, when $\beta = 20\%$, identifying the optimal solution becomes easy since the number of admissible solutions is relatively small. For small carriers, a value of $\beta = 50\%$ has no impact on the size of the feasible region when compared to the standard BCP since, as depicted in Table 2, small carriers are not able to cover more than 50% of the new contracts even in the absence of the LTC constraint due to their limited fleet size.

The results of Table 3 also show that adding the LBC constraint results in an increase of the average computing time when $\alpha \geq 3$. This increase is more important for large carriers and large networks ($n = 12, 15$) when $\alpha \geq 4$. In fact, a small value of α limits the number of feasible routes and makes the problem easier to solve. Increasing the value of α results in an increase in the number of feasible routes. Even if the number of feasible routes is smaller than in a standard BCP, adding a resource constraint to the subproblem to handle the LBC constraint makes the subproblem harder to solve, yielding an increase in the total computing time.

5.4 Impact of side constraints on solution quality

Table 4 reports the relative monetary loss, in percentage, resulting from adding the LTC or the LBC constraint to the standard BCP in case all the generated bids are submitted and won by the carrier. The lines Avg^L and Avg^S compute the average losses for large and small carriers, respectively.

Table 4: Impact of side constraints on profits

	BCP-β		BCP-α				
Instance	20%	50%	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$	$\alpha = 6$
6-L-500	-55.55	-30.98	-46.10	-33.30	-21.43	-10.22	0.00
8-L-500	-49.64	-28.83	-37.13	-22.03	-8.12	0.00	0.00
10-L-500	-42.97	-16.05	-23.49	-7.74	-0.02	0.00	0.00
12-L-500	-47.58	-16.86	-26.89	-9.90	-0.21	-0.01	0.00
15-L-500	-44.45	-11.10	-21.42	-3.81	-0.07	0.00	0.00
6-L-1000	-56.10	-31.57	-50.65	-56.10	-23.55	-11.23	0.00
8-L-1000	-50.43	-30.51	-41.43	-50.43	-9.06	0.00	0.00
10-L-1000	-44.69	-18.18	-26.61	-44.69	-0.02	0.00	0.00
12-L-1000	-49.46	-19.22	-30.66	-49.46	-0.24	-0.01	0.00
15-L-1000	-46.36	-12.80	-24.70	-46.36	-0.08	0.00	0.00
Avg^L	-48.72	-21.61	-32.91	-32.38	-6.28	-2.15	0.00
6-S-500	-25.94	0.00	-25.94	-15.75	-7.76	0.00	0.00
8-S-500	-21.71	0.00	-19.16	-9.38	-0.17	0.00	0.00
10-S-500	-26.30	0.00	-16.75	-4.67	0.00	0.00	0.00
12-S-500	-29.49	0.00	-21.61	-8.00	-0.17	0.00	0.00
15-S-500	-24.17	0.00	-14.15	-0.84	0.00	0.00	0.00
6-S-1000	-28.03	0.00	-28.03	-17.03	-8.39	0.00	0.00
8-S-1000	-23.92	0.00	-21.11	-10.34	-0.18	0.00	0.00
10-S-1000	-29.42	0.00	-18.74	-5.23	0.00	0.00	0.00
12-S-1000	-32.98	0.00	-24.29	-9.00	-0.19	0.00	0.00
15-S-1000	-27.40	0.00	-16.19	-0.96	0.00	0.00	0.00
Avg^S	-26.94	0.00	-20.60	-8.12	-1.69	0.00	0.00

Under the BCP-β context, the larger the value of β is, the less important is the monetary loss with respect to a standard BCP. For small carriers, the same profit as for a standard BCP is obtained when $\beta = 50\%$. This was predictable since in the standard BCP, at most 50% of the new contracts could be covered by small carriers. Adding the LTC constraint is much more detrimental for large carriers with an average loss of 48.72% when $\beta = 20\%$ and 21.61% when $\beta = 50\%$.

Considering the LBC constraint has less impact on the generated profit for small carriers when $\alpha \geq 4$. Here, the results are more nuanced since in the standard BCP, there are bids for which the number of covered contracts exceeds five for almost all the instances (see the last column of Table 2). Hence, there exist multiple optimal solutions to the BCP for which the number of new contracts per bid is equal to or less than five with no impact on the profit, for small carriers. For large carriers, the profit remains the same as in a standard BCP for $\alpha = 6$. However, important losses are observed when $\alpha \leq 3$. This can be explained by the fact that the fleet size, even large, is not able to cover enough new contracts with such restrictions.

Table 5 reports the percentage of new contracts covered by the generated bids under the different contexts. The lines Avg^L and Avg^S compute the average coverage for large and small carriers, respectively. These results show that the LTC constraint is almost tight for all the instances. For the LBC constraint, even if the main objective of this constraint is to limit the number of contracts per bid, this has an impact on the total number of new covered contracts. When compared to the standard BCP, the percentage of new covered contracts considerably decreases for large carriers for $\alpha \leq 4$. For small carriers, almost the same percentage of covered contracts is obtained for $\alpha \geq 4$.

Besides, adding the LTC constraint (for both values of β) gives to each participating carrier, independently of its size, the chance to cover the same number of contracts, if not close numbers. The difference in contract coverage between large and small carriers remains reasonable with the LBC constraint for $\alpha = 2$. It increases with the value of α becoming much more significant for $\alpha = 5, 6$.

Table 6 reports the number of alternative OR bids that could be submitted to the auction under the different contexts. These results indicate that the LBC constraint has no impact on the number of OR bids. This is because, in the standard BCP, both carrier types already use all their available vehicles to cover new contracts. However, a too restrictive LTC constraint ($\beta = 20\%$) considerably reduces the number of

Table 5: Impact of side constraints on contract coverage

	Standard	BCP-β		BCP-α				
Instance	BCP	20%	50%	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$	$\alpha = 6$
6-L-500	100.00	16.67	50.00	33.33	50.00	66.67	83.33	100.00
8-L-500	94.29	20.00	48.57	40.00	60.00	80.00	94.29	94.29
10-L-500	77.08	18.75	50.00	41.67	62.50	77.08	77.08	77.08
12-L-500	73.68	19.74	50.00	39.47	59.21	73.68	73.68	73.68
15-L-500	64.89	19.85	49.62	39.69	59.54	64.89	64.89	64.89
6-L-1000	100.00	16.67	50.00	33.33	50.00	66.67	83.33	100.00
8-L-1000	94.29	20.00	48.57	40.00	60.00	80.00	94.29	94.29
10-L-1000	77.08	18.75	50.00	41.67	62.50	77.08	77.08	77.08
12-L-1000	73.68	19.74	50.00	39.47	59.21	73.68	73.68	73.68
15-L-1000	64.89	19.85	49.62	39.69	59.54	64.89	64.89	64.89
Avg^L	81.99	19.00	49.64	38.83	58.25	72.46	78.65	81.99
6-S-500	41.67	16.67	41.67	16.67	25.00	33.33	41.67	41.67
8-S-500	45.71	20.00	45.71	22.86	34.29	45.71	45.71	45.71
10-S-500	50.00	18.75	50.00	29.17	43.75	50.00	50.00	50.00
12-S-500	50.00	19.74	50.00	26.32	39.47	48.68	50.00	50.00
15-S-500	41.22	19.85	41.22	27.48	41.22	41.22	41.22	41.22
6-S-1000	41.67	16.67	41.67	16.67	25.00	33.33	41.67	41.67
8-S-1000	45.71	20.00	45.71	22.86	34.29	45.71	45.71	45.71
10-S-1000	50.00	18.75	50.00	29.17	43.75	50.00	50.00	50.00
12-S-1000	50.00	19.74	50.00	26.32	39.47	48.68	50.00	50.00
15-S-1000	41.22	19.85	41.22	27.48	41.22	41.22	41.22	41.22
Avg^S	45.72	19.00	45.72	24.50	36.75	43.79	45.72	45.72

alternative OR bids for large carriers and large networks, tightening thus the gap between both carrier types. This was expectable since, in these cases, the number of covered contracts is limited (as depicted in Table 5) making it more profitable for the carrier to use less vehicles. This also explains the slight decrease in the number of bids for large carriers when $\beta = 50\%$ and $n = 6, 8$.

Table 6: Impact of side constraints on the number of OR bids

	Standard	BCP-β		BCP-α				
Instance	BCP	20%	50%	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$	$\alpha = 6$
6-L-500	4	2	3	4	4	4	4	4
8-L-500	7	4	6	7	7	7	7	7
10-L-500	10	6	10	10	10	10	10	10
12-L-500	15	9	15	15	15	15	15	15
15-L-500	26	16	26	26	26	26	26	26
6-L-1000	4	2	3	4	3	4	4	4
8-L-1000	7	4	6	7	3	7	7	7
10-L-1000	10	6	10	10	3	10	10	9
12-L-1000	15	9	15	15	3	15	15	15
15-L-1000	26	16	26	26	3	26	26	26
6-S-500	2	2	2	2	3	2	2	2
8-S-500	4	4	4	4	3	4	4	4
10-S-500	7	6	7	7	3	7	7	7
12-S-500	10	10	10	10	3	10	10	10
15-S-500	18	17	18	18	3	18	18	18
6-S-1000	2	2	2	2	3	2	2	2
8-S-1000	4	4	4	4	3	4	4	4
10-S-1000	7	6	7	7	3	7	7	7
12-S-1000	10	9	10	10	3	10	10	10
15-S-1000	18	16	18	18	3	18	18	18

In conclusion, our experimental study highlights the efficiency of the proposed branch-price-and-cut algorithm. The optimization tool we propose can thus be used as a decision support system that feeds up the carrier with different alternative bids, under different auction rules. These rules may be either imposed by the shipper running the auction or activated by the carrier voluntarily.

6 Conclusion

This paper proposes a route-based formulation to model a standard BCP for combinatorial TL transportation service procurement auctions. The carrier is offered the possibility to submit multiple OR bids. The model is extended to incorporate side constraints that may derive from explicit rules imposed by the auctioneer or from the carrier's strategy. Two side constraints are considered: an LTC constraint which limits the total number of contracts covered by the generated bids, and an LBC constraint which limits the number of contracts submitted in each bid. A branch-price-and-cut algorithm is proposed to solve efficiently the BCP with side constraints. It gathers state-of-the-art tools that have proven to be successful for solving various VRPs. Our computational results highlight the computational efficiency of the proposed algorithm. They also show that considering side constraints may be more or less beneficial for a carrier, depending on its size, and on the values given to the parameters defining these constraints.

The way OR bids are currently generated maximizes the carrier profit given that all the OR bids are submitted and won. However, it may happen that only a subset of OR bids are retained by the WDP with no guarantee that the carrier still makes profits. A research avenue that we are currently exploring, is to propose a bid generation algorithm which ensures the carrier to make a positive profit even if only a subset of the generated OR bids are allocated. Such an approach should take into account uncertainty related to other competing bids and the shipper 'non-revealed' business constraints.

References

Abrache, J., Crainic, T., Gendreau, M., Rekik, M., 2007. Combinatorial auctions. Annals of Operations Research 153, 131–164.

Baldacci, R., Mingozzi, A., Roberti, R., 2011. New route relaxation and pricing strategies for the vehicle routing problem. Operations Research 59 (5), 1269–1283.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., Vance, P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46 (3), 316–329.

Caplice, C., 2007. Electronic markets for truckload transportation. Production and Operations Management 16 (4), 423–436.

Caplice, C., Sheffi, Y., 2006. Combinatorial auctions for truckload transportation. In: P. Cramton, Y. Shoham, R. S. (Ed.), Combinatorial Auctions. MIT Press, Ch. 21, pp. 539–571.

Chang, T.-S., 2009. Decision support for truckload carriers in one-shot combinatorial auctions. Transportation Research Part B: Methodological 43 (5), 522–541.

Chen, R. L., AhmadBeygi, S., Cohn, A., R.Beil, D., Sinha, A., 2009. Solving Truckload Procurement Auctions Over an Exponential Number of Bundles. Transportation Science 43 (4), 493–510.

Cohn, A., Beil, D., Sinha, A., AhmadBeygi, S., Chen, R. L., 2008. Using implicit bidding to solve truckload procurement auctions. In: Proceedings of 2008 NSF Engineering Research and Innovation Conference, Knoxville, USA.

de Vries, S., Vohra, R., 2003. Combinatorial auctions: a survey. INFORMS Journal on Computing 15, 284–309.

Desaulniers, G., Desrosiers, J., Solomon, M., 2005. Column generation. Springer, New York, NY.

Desaulniers, G., Desrosiers, J., Spoorendonk, S., 2011. Cutting planes for branch-and-price algorithms. Networks 58 (4), 301–310.

Desaulniers, G., Madsen, O., Ropke, S., 2014. The vehicle routing problem with time windows. In: Toth, P., Vigo, D. (Eds.), Vehicle routing: Problems, methods, and applications, 2nd Edition. MOS-SIAM Series on Optimization. SIAM, Ch. 5, pp. 119–159.

Dror, M., 1994. Note on the complexity of the shortest path models for column generation in vrptw. Operations Research 42, 977–978.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks 44, 216–229.

Garrido, R. A., 2007. Procurement of transportation services in spot markets under a double-auction scheme with elastic demand. Transportation Research Part B: Methodological 41 (9), 1067–1078.

Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M. (Eds.), Column Generation. Springer, Ch. 2, pp. 33–65.

Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D., 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research 56 (2), 497–511.

Kwasnica, A. M., Ledyard, J. O., Dave Porter, a. C. D., 2005. A new and improved design for multiobject iterative auctions. Management Science 51 (3), 419–434.

Lee, C.-G., Kwon, R. H., Ma, Z., 2007. A carriers optimal bid generation problem in combinatorial auctions for transportation procurement. Transportation Research Part E: Logistics and Transportation Review 43 (2), 173–191.

Lübbecke, M., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53 (6), 1007–1023.

Nandiraju, S., 2006. Strategic freight transportation contract procurement. Ph.D. thesis, University of California, Irvine.

Nisan, N., 2000. Bidding and allocation in combinatorial auctions. In: Proceedings of the 2nd ACM conference on Electronic commerce. pp. 1–12.

Righini, G., Salani, M., 2006. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization 3 (3), 255–273.

Song, J., Regan, A., 2005. Approximation algorithms for the bid construction problem in combinatorial auctions for the procurement of freight transportation contracts. Transportation Research Part B: Methodological 39 (10), 914–933.

Toth, P., Vigo, D. (Eds.), 2014. Vehicle Routing: Problems, Methods, and Applications, 2nd Edition. MOS-SIAM Series on Optimization. SIAM, Philadelphia, PA.

Triki, C., Oprea, S., Beraldi, P., Crainic, T. G., 2014. The stochastic bid generation problem in combinatorial transportation auctions. European Journal of Operational Research 236 (3), 991 – 999.

Wang, X., Xia, M., 2005. Combinatorial bid generation problem for transportation service procurement. Business 1923 (1), 189–198.