



# Carrier assignment models in transportation procurement

Y Guo<sup>1</sup>, A Lim<sup>2</sup>, B Rodrigues<sup>3\*</sup> and Y Zhu<sup>2</sup>

<sup>1</sup>National University of Singapore, Singapore; <sup>2</sup>Hong Kong University of Science and Technology, Hong Kong; and <sup>3</sup>Singapore Management University, Singapore

This paper extends carrier assignment models used in winner determination auctions for transportation procurement to include shipper non-price objectives and carrier transit point costs. The models are unlike traditional carrier assignment models which incorporate only carrier lane bids, and different from combinatorial auction models which focus on packets of lanes without considering transit point costs. We develop solutions, including metaheuristics, for the new models and through computational experimentation show that the algorithms work well and can be easily implemented. *Journal of the Operational Research Society* (2006) 57, 1472–1481. doi:10.1057/palgrave.jors.2602131  
Published online 28 December 2005

**Keywords:** transportation; procurement; bid; metaheuristics

## 1. Introduction

In the huge trucking market, valued at \$600 billion in the United States alone (American Trucking Association, 2002; Caplice and Sheffi, 2003) procuring transportation services is crucial to shippers who must secure high-quality carrier services while controlling supply chain costs. In buying transportation services, a shipper typically tenders a request for quotes (RFQ) for a network of lanes following a bid preparation stage (Rhinehart, 1989; Foster and Strasser, 1991; Gibson *et al.*, 1993; Caplice and Sheffi, 2003; Sheffi, 2004), where a lane is a one-way movement from an origin to a destination with an associated set of shipments for the period covered by the RFQ (Caplice and Sheffi, 2003). Once bids are received, a bid-analysis exercise is used to allocate lanes to carriers.

Foster and Strasser (1991) studied RFQ auctions where the shipper provides a list of lanes to carriers to bid for, and decides the winners using a single price criterion. RFQ auctions continue to be used in transportation procurement as reported by Sheffi (2004). In combinatorial auction (Sheffi, 2004) mechanisms, used by many shippers and third-party-logistics providers to encourage more aggressive bidding, shippers request bids for groups of lanes, in addition to individual lanes. This allows carriers to form bid packages based on their economics—existing client base, driver domiciles, maintenance networks, etc—and therefore cut costs and pass on part of the savings to the shipper. In deciding the winning bids for lanes, the shipper uses an optimization carrier assignment model (CAM) to minimize

transportation costs while ensuring that all lanes are covered. Set covering models, including those with combinatorial input, have been well studied in the literature (Ledyard *et al.*, 2002; Song and Regan, 2002; Caplice and Sheffi, 2003; Elmaghraby and Keskinocak, 2003; Vohra and de Vries, 2003; Sheffi, 2004).

Recently, Caplice and Sheffi (2003) and Sheffi (2004) studied non-price and level-of-service factors in transportation procurement. These include shipper restrictions on the number of lanes a carrier can win, favouring incumbents, keeping specific carriers off certain lanes, restricting carriers from serving parts of the network, and incorporating performance as a factor in carrier selection. This has led to specialized RFQ auctions where winner determination CAMs are required to deal with these factors, other than price alone. Sheffi (2004) reports that many leading companies, including Colgate-Palmolive, Ford Motors, Lucent Technologies, Proctor and Gamble, and Wal-Mart Stores, have used combinatorial auctions successfully to obtain low transportation costs and high levels of service.

Here, a level-of-service performance requirement can be announced by the shipper in his RFQ as a non-price attribute, and evaluation criteria given for it. The carrier responds with a description of his service quality level and a bid price. The shipper can then set a value for this service level and can, for example, choose to select the carrier with the largest surplus margin between service level and bid price. This illustrates the fact that the RFQ process is a sealed bid auction with independent private values. As determining the winner is a combinatorial problem with many objectives, a CAM must be used. In the example of service level input, a penalty cost can be modelled in the

\*Correspondence: B Rodrigues, School of Business, Singapore Management University, 50 Stamford Road, Singapore 178899, Singapore.  
E-mail: br@smu.edu.sg

CAM, inverse to the value set by the shipper for the carrier's level of service or the shipper can adjust the fees charged by the carrier to reflect the service level for the carrier on each lane (Caplice and Sheffi, 2003). Transportation and logistics solution providers such as Manugistics Inc. ([http://www.manu.com/solutions/transportation\\_logistic.aspx](http://www.manu.com/solutions/transportation_logistic.aspx)) offer electronically distributed e-RFQs with which shippers can 'utilize configurable algorithms to further analyse and optimize carrier bids across a number of business constraints.' Using online bidding platforms, carriers are able to adjust to the shipper's requirements. For example, if the shipper imposes a cap on the number of lanes the carrier can win, then the carrier will most likely respond by strategically increasing its bids over the network using many combinatorial bids (Sheffi, 2004). If, for example, the carrier is aware that his level of service is low in view of the shipper's criteria, it might choose to reduce its bid price. In the case of shippers insisting on low package prices while awarding only a single lane or partial packages, a carrier will submit bids for single lanes and partial packages to protect itself (Sheffi, 2004). Depending on specifications of the RFQ provided by the shipper, carriers respond by adjusting their bid strategy. As most shippers use only a single round in the process (Sheffi, 2004), combinatorial bidding is important to carriers who cannot use multiple rounds to signal each other as to which lanes they want and must hedge against uncertainty in response to the bidding method used by the shipper (Caplice and Sheffi, 2003). In all cases, carriers have to make assumptions whether they will win lanes which interact with lanes in their network and estimate the probability of achieving economies of scope arising from line interdependencies etc, where the cost of hauling on one lane is affected by other serviced lanes.

In this work, we focus on optimization models and extend CAMs to address shipper's non-price business objectives mentioned above. Further to this, we develop a model that incorporates carrier transit point costs, in addition to lane costs. This provision originates from the authors' study with Royal Philips Electronics (a shipper) which used carriers that incurred varying costs at transit points. These costs included stopover, parking/berthing, warehousing, taxes and accommodation costs. Combinatorial auction models available do not address these shipper and carrier considerations since combinations studied are concerned with packets of lanes and not with the points lanes transit. In reality, however, carrier quotes and shipper supply chain planning are connected to *both* transportation routes and transit/terminal locations, and interdependencies of costs resulting from economies of scope (Caplice and Sheffi, 2003; Sheffi, 2004) are not only derived from connecting served lanes but also on the locations that connect lanes.

The work is organized as follows: In the next section, the bid analysis process is described. In section 3, two new models that incorporate shipper's objectives and transit point costs are given. Solutions for the models are then

provided. In section 4, computational experiments to compare the solution approaches are described. The work is concluded in section 5.

## 2. Background

### 2.1. CAMs in bid analysis

In the bid analysis stage in the transportation procurement process, a CAM helps the shipper minimize total costs while ensuring that each lane is served and its required capacity satisfied. Generalized CAMs specify that allocated lanes and volumes are feasible for both shippers and carriers.

In CAM models, decision variables are binary, whereas more general forms are mixed integer programs (MIP) for which solution approaches are available (Nemhauser and Wolsey, 1999; Caplice and Sheffi, 2003) using MIP solvers. There has been much interest in combinatorial auctions used in bid analysis in transportation procurement. In an early work, Moore *et al.* (1991) employed MIP for carrier selection without conditional bids, while, more recently, Ledyard *et al.* (2002) allowed for conditional bids without consideration of capacity limitations and performance factors.

Shippers often use optimization models in 'What-if' sensitivity analysis (Gibson *et al.*, 1993): 'What if I assign only incumbents and do not allow new carriers'; 'What if I reduce the number of carriers servicing city X', etc. It is preferable therefore that shippers can analyze contingencies directly in CAMs.

### 2.2. Shipper considerations

As pointed out, shipper input is usually absent from traditional CAMs, including those of a combinatorial nature. The inclusion of shipper non-price considerations has been found to be 'one of the strongest added values to the whole process' as reported recently by Caplice and Sheffi (2003). The authors list the following as some of the considerations that enhance the practical value of CAMs:

- Setting minimum/maximum carrier numbers: Shippers would require that no more and no less than a certain number of carriers can win lanes to determine optimal sizing of a carrier group.
- Favouring incumbents: Because of the additional costs of new carriers, the shipper can apply a penalty to non-incumbents (or reward incumbents). In strategic supply management, shippers want to ensure they have the right set of suppliers.
- Setting maximum/minimum coverage: Shippers may wish to restrict the amount of traffic a carrier can win on a lane or in the system.
- Restricting carriers: Shippers may wish to restrict carriers or groups of carriers from serving part of the network. For example, a shipper may want to restrict a carrier from serving certain nodes in the network in an international

network for political reasons or may want to penalize a shipper at a city where its staff are unfamiliar with the carrier's operations. In international transportation, carriers foreign to countries through which lanes transit may be less preferred.

- Factoring in performance factors: Level of service provided by carriers can be a factor leading to awarding lanes. One way to do this, suggested by the authors, is to modify cost coefficients.

### 2.3. Carrier considerations

Carriers traditionally submit quotes only for lanes. It is common, however, that carriers have strengths (or weaknesses) in certain cities, regions or groups of cities and incur varying costs by operating through different points. For example, carriers can have hubs at certain cities through and from which they can operate with lower costs. In other situations, carriers may have to invest in start-up costs at cities new to their network. Yet in other situations, carriers may be liable for operating costs, including *ad hoc* ones, for example, taxes and levies in international carriage at transit points.

Take the case where a carrier wins a lane connecting A and B, and a lane connecting A and C but will incur separate costs at A, B and C. It will be preferable to both the carrier and the shipper if the quote given for lanes AB and AC did not include operating costs due to A, B and C, since, otherwise, they would accrue twice from A, resulting in a higher total bid for lanes AB and AC. In designing CAM's, the use of explicit transit point costs can therefore only benefit both carriers and shippers. Carriers can better identify optimal lane packages and provide more realistic bids when lane costs are clearly separated from transit point costs, rather than subsumed into one set of lane costs.

## 3. Extended carrier assignment models

Although it is impractical to include every shipper non-price and level-of-service factor in one model, we address some of these factors here. To achieve this, we provide two models. The first addresses the issue of restricting the number of lanes a carrier can win.

Although this is determined by the shipper, carriers can suggest the maximum lane coverage threshold they wish to impose on themselves. The model is an integer program, for which a network flow solution is provided. The second model addresses other factors which include favoring incumbents, restricting carriers to lanes and service performance factors. From the carrier's point of view, the model allows for explicit costs at transit points to be managed separately from lane costs. We call these separate costs, carrier transit point costs. This model is shown to be NP-complete for which metaheuristic solutions are developed.

### 3.1. Model 1: a CAM with shipper's business constraints

*Shipper's perspective.* As we have seen, shippers may wish to have a maximum (minimum) number of carriers on each lane or wish to restrict the number of carriers in parts or all of its network. One way this can be achieved is by controlling the number of lanes awarded to each carrier. From the shipper's point of view, the number of lanes a carrier can ultimately win is dependent on several factors; typically, these include the shipper's perception of the carrier capability, track records, synergies with the shipper and spread requirements. This can be addressed in various ways. Enforcing a cap on the total number of winnable lanes can ensure a better spread and larger carrier participation resulting in better choices for the shipper. Conversely, the shipper can wish to award a minimum number of lanes to a preferred carrier.

*Carriers perspective.* In competition for contracts, small transportation companies often submit lower bids compared to their larger counterparts. However, these carriers often have smaller capabilities and, as a result, usually service only a limited number of lanes. Because of this, they are restricted to bid for smaller numbers of lanes although they would be better off bidding for many packets of lanes to increase their chances of winning routes that maximize profits. A carrier with a five-lane capability will want to bid for a number of five-lane packages hoping to win one with the best profit. The possibility, however, of winning more lanes than it can handle is a consequence that the carrier may not be able to bear. Carriers therefore will want to limit the number of winnable contracts to be within their capabilities, but otherwise attempt to bid for as many combinations of lanes or packages as possible.

The following shipper's integer programming model addresses both these shipper and carrier objectives. As pointed out, lane caps can be shipper determined, or provided to the shipper by the carrier.

*Parameters:*

- $L =$  the number of lanes,
- $S =$  the number of carriers,
- $M =$  a sufficiently large number,
- $b_{kj} =$  0, 1, 2, 3, ... is the bid value carrier  $k$  places on lane  $j$ ;  
 $b_{kj} = M$  if carrier  $k$  does not bid for lane  $j$  ( $1 \leq k \leq S$ ,  $1 \leq j \leq L$ ),
- $k_{\min} =$  the minimum total number of lanes assignable to carrier  $k$  ( $1 \leq k \leq S$ ) determined by the shipper (or the carrier if desired),
- $k_{\max} =$  the maximum total number of lanes assignable to carrier  $k$  ( $1 \leq k \leq S$ ) determined by the shipper (or the carrier if desired).

*Decision variables:*

- $x_{kj} = 1$  if lane  $j$  is assigned to carrier  $k$  ( $1 \leq k \leq S$ ,  $1 \leq j \leq L$ );  
 0 otherwise

Objective:

$$\text{minimize } C = \sum_{k=1}^S \sum_{j=1}^L b_{kj} x_{kj} \quad (1)$$

subject to

$$\sum_{k=1}^S x_{kj} = 1, \quad 1 \leq j \leq L \quad (2)$$

$$\sum_{k=1}^S b_{kj} x_{kj} \leq M - 1, \quad 1 \leq j \leq L \quad (3)$$

$$k_{\min} \leq \sum_{j=1}^L x_{kj} \leq k_{\max}, \quad 1 \leq k \leq S \quad (4)$$

Constraint (2) ensures that each lane is assigned to exactly one carrier and constraint (3) ensures that when lane  $j$  is assigned to carrier  $k$ , the bid  $b_{kj}$  value cannot be  $M$ . In (4), we note that when the maximum values  $k_{\max}$  are set to  $L$  (with  $k_{\min}$  set to 0), the model reduces to a basic CAM where there is no cap on the number of lanes. In the model, greater spread of lanes among carriers can be achieved by reducing  $k_{\max}$ , where in the extreme case  $k_{\max} = 1$  for all  $k$ .

In the case that carriers wish to restrict the number of lanes that can be won by any carrier in a particular region, constraint (4) can be modified to have only the set of lanes present in the region used and  $k$  restricted to that particular carrier. Here constraints (4) would be transformed to:  $k_{\min} \leq \sum_{j \in R} x_{kj} \leq k_{\max}$ , where  $R$  denotes the region or subset of lanes in question.

**3.1.1. A network maximum flow solution.** A minimum cost maximum flow solution is given for this problem. Without loss of generality, take  $k_{\min} = 0$  and apply a transformation to the graph which represents the problem. In the graph, edges are created for every lane bid a carrier submits.

To illustrate the network, we use a simple example with two carriers and three lanes, and where  $B$  in Figure 1 denotes the corresponding carrier-lane bid matrix. Nodes  $a$  and  $b$  represent carrier 1 and carrier 2, respectively, whereas nodes  $c$ ,  $d$  and  $e$  represent the three lanes. The pair  $i/j$  on each edge represents the cost  $i$  of a unit flow and the capacity  $j$  of that edge. Carriers 1 and 2 are allowed to serve at most 2 and 1 lanes, respectively.

In constructing the network in Figure 1, two additional nodes are added: a source and a sink. Phase 1 consists of the capacity constraints of each carrier, where edges are constructed from the source node to the carrier nodes. Here, 0/2 indicates that carrier 1 can serve a maximum of two lanes. In phase 2, edges between carrier nodes  $\{a, b\}$  and lane nodes  $\{c, d, e\}$  are constructed. Here, 3/1 indicates the cost to carrier  $a$  to cover route  $c$  is 3 and flow capacity 1 for consistency between phases 1 and 3. In phase 3, edges

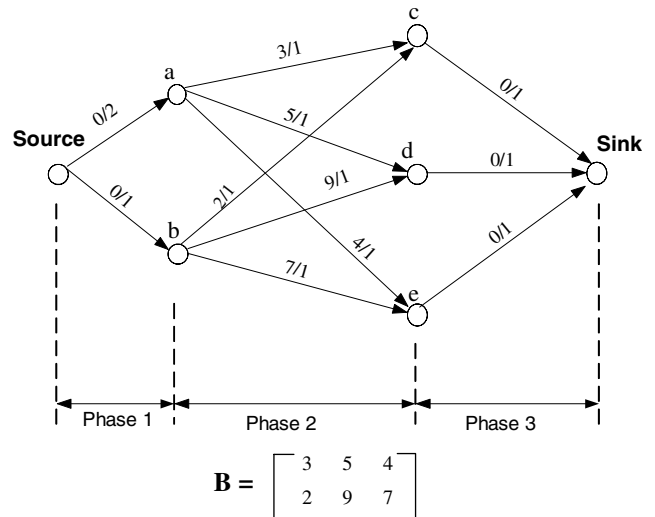


Figure 1 Network flow example.

connecting lane nodes to the sink are constructed with cost 0 and capacity 1 to ensure each edge is served by a carrier only once.

It is now easy to see that the minimum cost maximum flow in the network solves the problem. First, edge capacities in phase 1 ensure that no carrier is assigned more than the number lanes allowed and flow in phase 3 ensures every edge is served by a carrier once. Next, any solution with cost less than the minimum cost maximum flow, can be transformed to a network flow as described implying a cost lower than the minimum cost maximum flow, which is a contradiction. Hence, the minimum cost maximum flow must solve the problem.

Algorithms for finding the minimum cost maximum flow in a network have been well-studied (Oldham, 2001). By applying the transformation above, the problem can be solved in  $O((L + S)^3 * L)$  time.

**3.2. Model 2: a CAM with penalty and transit point costs (CAMPC)**

*Shipper's perspective.* Shipper non-price business considerations include favouring incumbents, restricting carriers to lanes, excluding or penalizing carriers at transit points and including carrier performance factors, such as level of service. To deal with these objectives, we propose a CAM that includes penalty costs. In the case where an attribute is desirable, as with level of service, 'penalty cost' can be taken as the inverse value of the level-of-service price assigned to the carrier by the shipper. The model can easily be extended to have more than one penalty cost. Alternatively, the model can be used to achieve each of these objectives separately. For example, if the shipper wished to exclude a carrier from a lane, then high penalty costs can be assigned to the carrier

at locations adjacent to the lane. If the shipper wished to restrict or exclude a carrier at a transit location, he would assign a high penalty to the carrier at that point. By adjusting penalty costs, the shipper can determine the carrier allocation which is best suited to his business objectives.

*Carrier's perspective.* Carriers and shippers will benefit if lane bids can be separated from transit point costs which provides for more realistic bidding. Carriers can adjust bids according to varying costs incurred at different transit locations.

A model (CAMPC) which addresses these shipper and carrier objectives is given in the integer program:

*Parameters:*

- $n =$  the number of nodes
- $L =$  the number of lanes
- $S =$  the number of carriers
- $M =$  a sufficiently large number
- $a_{ij} =$  1 if and only if node  $i$  is adjacent to lane  $j$  ( $1 \leq i \leq n, 1 \leq j \leq L$ ); 0 otherwise
- $b_{kj} =$  0, 1, 2, ... is the value carrier  $k$  bids for lane  $j$ ;  $b_{kj} = M$  if carrier  $k$  does not bid for lane  $j$  ( $1 \leq k \leq S, 1 \leq j \leq L$ )
- $b_{ki}^{\bullet} =$  0, 1, 2, ... is the carrier  $k$ 's bid cost at transit node  $i$  ( $1 \leq k \leq S, 1 \leq i \leq n$ )
- $p_{ki} =$  0, 1, 2, ... is the shipper's penalty cost assigned to carrier  $k$  at node  $i$  ( $1 \leq k \leq S, 1 \leq i \leq n$ )

*Decision variables:*

- $x_{kj} =$  1, if lane  $j$  is assigned to carrier  $k$  ( $1 \leq k \leq S, 1 \leq j \leq L$ ); 0 otherwise,
- $y_{ki} =$  1, if carrier  $k$  wins a lane adjacent to node  $i$  ( $1 \leq k \leq S, 1 \leq i \leq n$ ); 0 otherwise

*Objective:*

$$\begin{aligned} \text{minimize } C = & \sum_{k=1}^S \sum_{j=1}^L b_{kj} x_{kj} + \sum_{k=1}^S \sum_{i=1}^n b_{ki}^{\bullet} y_{ki} \\ & + \sum_{k=1}^S \sum_{i=1}^n p_{ki} y_{ki} \end{aligned} \tag{5}$$

subject to

$$\sum_{k=1}^S x_{kj} = 1, \quad 1 \leq j \leq L \tag{6}$$

$$\sum_{k=1}^S b_{kj} x_{kj} \leq M - 1, \quad 1 \leq j \leq L \tag{7}$$

$$\sum_{j=1}^L a_{ij} x_{kj} \geq y_{ki}, \quad 1 \leq i \leq n, \quad 1 \leq k \leq S \tag{8}$$

$$\sum_{j=1}^L a_{ij} x_{kj} \leq M y_{ki}, \quad 1 \leq i \leq n, \quad 1 \leq k \leq S \tag{9}$$

Constraint (6) ensures that each lane is assigned to exactly one carrier and constraint (7) ensures that when lane  $j$  is assigned to carrier  $k$ , the bid  $b_{kj}$  is not  $M$ . Constraint (8) ensures that when  $y_{ki}$  is 1, there is at least one edge  $j$  connecting node  $i$  to carrier  $k$  and (9) ensures that when  $y_{ki}$  is 0, no edge connecting node  $i$  is assigned to carrier  $k$ .

The CAMPC is NP-complete and a proof of this can be found in the Appendix.

*3.2.1. Benchmarking the CAPMPC using branch-and-bound solutions.* A branch-and-bound (B&B) complete search can be used to examine all possible assignments of the lanes to carriers for small test sizes, where the time performance of the algorithm depends largely on the bounding function used (Viswanathkumar and Srinivasan, 2002).

The B&B algorithm begins with an empty solution set and divides the problem into  $L$  recursive stages, where a lane is assigned to a carrier in each stage. At stage  $k$ , when lanes 1 to  $k-1$  have been assigned, carriers who bid for lane  $k$  are examined. Given a carrier  $s'$  from this set, lane  $k$  is assigned to  $s'$  if the current best solution is larger than the lower bound (see below); otherwise, the search branch is discarded and the search moves to consider the next carrier. The process moves recursively to stage  $k+1$  to assign a carrier to lane  $k+1$ .

To calculate the lower bound: Let  $s(i)$  ( $1 \leq i \leq L$ ) be the carrier that edge  $i$  is assigned to. If  $s(i)$  is determined for  $1 \leq i \leq k$  and undetermined for  $k+1 \leq i \leq L$ , the lower bound for the total cost is the sum of three components:  $L_1$ ,  $L_2$  and  $L_3$ , where  $L_1$  is the bid cost for edges 1 to  $k$ , already assigned to some carrier,  $L_2$  is the bid lower bound for edges  $k+1$  to  $L$ , yet to be assigned to any carriers, and  $L_3$  is the least possible total penalty cost. Letting  $\mathbf{B}$  denote the  $S \times L$  carrier bid matrix, we have

$$L_1 = \sum_{i=1}^k \mathbf{B}_{s(i),i}$$

$$L_2 = \sum_{i=k+1}^L \min\{\mathbf{B}_{j,i} | 1 \leq j \leq S\}$$

$$L_3 = \sum_{i=1}^n p_i$$

where  $p_i$  is the penalty cost incurred at node  $i$ . If there is an edge  $j$  with  $1 \leq j \leq k$  which is connected to node  $i$ ,  $p_i$  is the sum of penalty costs assigned to carriers who serve some edge between 1 and  $k$ ; if there is no edge  $j$  with  $1 \leq j \leq k$  connected to node  $i$ , then  $p_i$  is the minimum penalty assigned to any carrier that can cover any edge connected to node  $i$ . Thus, at any point in the B&B process, the bounding function used is  $L_1 + L_2 + L_3$ .

**3.2.2. Using metaheuristics to solve the CAMPC.** Heuristics have been used for combinatorial auctions problems (Sandholm et al, 2002; Sandholm, 2002; Vohra and de Vries, 2003) and for other difficult combinatorial optimization problems (eg Foster and Strasser, 1991; Lim et al, 2004). As the CAMPC is an NP-complete problem, metaheuristic solutions are developed based on widely used genetic algorithm and tabu search techniques, which have been successful in other comparable applications. A hybrid of these is then constructed which provides a third heuristic approach to the problem.

*A genetic algorithm* Genetic algorithms have been widely used for combinatorial optimization problems (Dowsland, 1996); for example, they have been applied to task allocation problems (Song and Regan, 2002; Wen and David, 2001). Here, a genetic algorithm (GA) is used for the CAMPC which is described as follows:

*Outline.* For each distinct chromosome pair in a subset of a randomly generated initial population with size  $pop\_size$ , perform crossover operations and mutate newly generated chromosomes according to a mutation probability. By evaluating the objective function value of the new and old chromosomes, retain  $pop\_size$  best chromosomes. In the implementation, termination is effected when the best solution does not improve in a given number of iterations or if a maximum number of iterations is reached.

*Chromosome representation.* Solutions are encoded as chromosome strings  $s=(s(1), s(2), \dots, s(L))$  containing  $L$  integers, where  $L$  is the number of edges to be covered. Each  $s(i)$  ( $1 \leq i \leq L$ ) is an integer which represents the carrier index that is assigned to edge  $i$ , so that  $1 \leq s(i) \leq S$ . Initial solutions are generated so that  $b_{s(i),i} < \infty$  is ensured.

*Crossover operation.* For chromosomes  $s$  and  $s'$ , a random number  $k$  is generated between 1 and  $L-1$  and the genes are cut at position  $k$ . The subsequence  $(s(k+1), s(k+2), \dots, s(L))$  is placed after  $(s'(1), s'(2), \dots, s'(k))$  and  $(s'(k+1), s'(k+2), \dots, s(L))$  is placed after  $(s(1), s(2), \dots, s(k))$  to generate two new chromosomes. With this operation, newly formed chromosomes are always feasible, given their parents are feasible since the positions of carrier assignments are not shifted when performing the crossover.

*Mutation operation.* For each chromosome in the offspring generation, a mutation probability  $q$  is used. Generating a random number in  $r \in [0, 1]$  for each  $i$ , if  $r < q$ ,  $s(i)$  is changed randomly to another value thus changing the carrier assigned to edge  $i$ .

In implementation, the following parameters were used:  $pop\_size = 1000$ ,  $q = 0.5$ . The convergence criteria were 50 generations reached or more than 20 generations with no improvement to current best solution.

*A tabu search.* Tabu search is a search strategy which moves iteratively from one solution to another in a neighbourhood search space using an adaptive memory. The method declares *tabu*, solutions with attribute changes

recorded in the short-term memory from being reused, where the time a restriction is in effect depends on a *tabu tenure* parameter (Glover and Laguna, 1997).

Tabu search is applied to the CAMPC, with the solution representation used in *GA*, that is, a string of integers. A neighbourhood move is defined as a change in carrier-edge assignment in the solution (similar to the mutation operator in *GA*) and, to avoid recycling, tabu lists consist of the recent *tabu tenure* solutions.

The heuristic (TS) is outlined as follows: Find an initial solution  $x_{now}$ . Set  $x_{best} = x_{now}$ . If the termination condition is satisfied, quit with  $x_{best}$ . The termination condition is satisfied if either more than 200 iterations have been executed or there is no improvement in the current best solution in the most recent 100 iterations. Otherwise, generate the neighbourhood  $N(x_{now})$  of  $x_{now}$  with 1000 new neighbourhood solutions by randomly selecting three lanes among the  $L$  lanes in  $x_{now}$  and reassigning them to other feasible carriers randomly. Evaluate the cost of  $x_{trial}$  for each new candidate trial solution  $x_{trial}$ . Select  $x_{next} = \min_{x_{trial} \in N(x_{now})} Cost(x_{trial})$  for  $x_{trial}$  not in the tabu list. Update the tabu memory by adding the new current solution  $x_{next}$  into the tabu list. Set  $x_{now} = x_{next}$ . If  $cost(x_{now}) < cost(x_{best})$ , set  $cost(x_{best}) = cost(x_{now})$ . In implementation, 1000 recent solutions were maintained in the tabu list.

*A genetic algorithm with tabu search (GA + TS)* As initial solutions can contribute to the quality of solutions provided by the metaheuristics, we developed a third heuristic using initial solutions from *GA* in *TS*.

#### 4. Computational experiments for the CAMPC

In order to gauge the performance of the solution approaches, a series of experiments were conducted. The *B&B* method and three heuristics (*GA*, *TS*, *GA + TS*) were coded using C++ and run on a Pentium IV 1.4G PC with 256 Mb of memory.

The experiments consisted of two parts. The first set of experiments used small-size test instances as optimal solutions could be found with the exact enumerative *B&B* method for these sizes. Solutions obtained by the heuristic approaches were compared with the optimal solutions found. The second set of experiments compared the heuristics for larger-size problems against the best solutions obtained from the heuristics.

Test instances were generated using the following steps:

*Step 1:* Given the input data:  $n$  nodes,  $S$  carriers and  $L$  edges, construct a graph from an initial graph with  $n$  nodes and no edges by connecting randomly selected nodes with no edge between them. Repeat this until all  $L$  edges are added to the graph.

*Step 2:* Assign carrier costs to nodes: For each carrier and each node, pick a number from  $\{1, 2, 3\}$  randomly

to indicate the range the carrier cost will be in—1 for a low range, 2 for a middle range, and 3 for a high range. Take each cost range to be given by an interval. With this, assign a cost value randomly from the range.

*Step 3:* Assign carrier costs to edges: Assign costs as in Step 2.

*4.1. Comparisons with branch-and-bound*

In total, 300 small size instances were generated with the number of edges ranging from 3 to 21. The exact optimal cost value was found by *B&B*, and the running time (in seconds) calculated for each algorithm, for each instance.

From the experiments, we found that the heuristics performed well when instances were small, which is expected of a good heuristic. All three methods provided optimal solutions for most of the 300 instances. The mean values of the percentage difference between each heuristic solution and the exact solution found by *B&B* were very close to 0 for all the heuristics. The mean values and standard deviations are provided in Table 1. Mean values were 0.082 for *GA*, and 0.004 for *TS* and *GA + TS*, and the standard deviations were 24.482 for *GA*, 1.056 for *TS* and 1.056 for *GA + TS*. The time required for *B&B* varied with size and among the heuristics, the *GA* algorithm consumed more time than the other two heuristics for most instances since it depends on the population size of each generation. The times required

for *B&B* were large since its performance depended on input size and graph structure. In contrast, the heuristics were more likely to be independent of these inputs and had more stable running times than *B&B*.

Running time statistics are provided in Table 2. In the table,  $\mu'_{B\&B}$ ,  $\mu'_{GA}$ ,  $\mu'_{TS}$ ,  $\mu'_{GA+TS}$  is the mean running time for *B&B*, *GA*, *TS* and *GA + TS* respectively, and  $\sigma'$  is the time standard deviation in each group.

From the table, *B&B* required considerably more time than the other heuristics when the number of edges was 13 and higher. Moreover, *B&B* had a large deviation for running times.

Although the heuristics performed well for small cases, this is not sufficient to guarantee good performance for larger-size cases. In the next section, we provide a best case analysis for the heuristics for larger cases.

*4.2. Comparisons between the heuristics*

To determine the performance of the heuristics for large test sizes, a total of 250 instances were tested in five groups which were generated with sizes of up to 500 edges. Statistics for the experiments are provided in Tables 3 and 4.

In Table 3, the instance size is in the form: number of edges\_number of nodes\_number of carriers, and  $\mu^C_{GA+TS}$ ,  $\mu^C_{TS}$  and  $\mu^C_{GA}$  denote the solution objective function mean value obtained using *GA + TS*, *TS* and *GA*, respectively, whereas  $\mu^t_{GA}$ ,  $\mu^t_{TS}$  and  $\mu^t_{GA+TS}$  is the mean time required by

**Table 1** Performance of heuristics for small test sizes

No. of edges	No. of instances	$\mu_{B\&B}$	$\sigma_{B\&B}$	$\mu_{GA}$	$\sigma_{GA}$	$\mu_{TS}$	$\sigma_{TS}$	$\mu_{GA+TS}$	$\sigma_{GA+TS}$
3–21	300	0.0	0.0	0.082	24.282	0.004	1.056	0.004	1.056

**Table 2** Performance of heuristics for small test sizes—time statistics

No. of edges	No. of instances	$\mu'_{B\&B}$	$\sigma'_{B\&B}$	$\mu'_{GA}$	$\sigma'_{GA}$	$\mu'_{TS}$	$\sigma'_{TS}$	$\mu'_{GA+TS}$	$\sigma'_{GA+TS}$
1–8	60	0.0	0.0	127.3	4.2	7.2	6.7	7.2	6.7
9–12	110	1.8	2.8	133.4	43.7	17.5	4.2	17.1	0.6
13–17	90	432.7	905.4	130.5	3.9	17.9	1.6	19.4	3.0
18–21	40	1675.4	3260.8	131.2	3.0	18.6	2.2	17.9	0.1

**Table 3** Lane experiments

Size	$\mu^C_{GA+TS}$	$\mu^t_{GA+TS}$	$\mu^C_{TS}$	$\mu^t_{TS}$	$\delta_1$	$\mu^C_{GA}$	$\mu^t_{GA}$	$\delta_2$
100_50_10	5022	64.23	5028	68.17	0.13	5238	506.11	4.31
200_70_10	9118	69.44	9118	71.76	0	10564	565.43	15.86
300_80_15	13299	74.65	13334	75.22	0.27	16821	639.21	26.49
400_90_20	18001	80.27	18076	80.93	0.42	24283	771.74	34.90
500_100_20	20812	85.19	20886	87.96	0.35	29762	860.76	43.00

the algorithms to find solutions. The values  $\delta_1$  and  $\delta_2$  denote the ratio of difference between  $\mu_{TS}^C$  and  $\mu_{GA}^C$  with  $\mu_{GA+TS}^C$  as a percentage. From this table,  $GA + TS$  and  $TS$  can be seen to perform equally well, with results differing by no more than 1%, all of which were found within 90 s. On the other hand, the  $GA$  algorithm did not perform well when compared to its performance for smaller test cases. The relative performance difference between  $GA$  and  $GA + TS$  increased from about 4% to 43% as size increased. In addition,  $GA$  required 8–10 times the running time required by  $GA + TS$ .

Table 4 provides additional statistical information on these experiments. In the table,  $\beta_{GA+TS}$ ,  $\beta_{TS}$ , and  $\beta_{GA}$ , denote the number of best solutions obtained in each group by the algorithms  $GA + TS$ ,  $TS$  and  $GA$ , respectively, and  $\sigma_{GA+TS}^C$  and  $\sigma'_{GA+TS}$ , etc, is the standard deviation of the solutions and running times, respectively. Again, there was a marginal advantage of  $GA + TS$  over  $TS$  in the number of best solutions obtained, while  $GA$  failed to provide results that are competitive with the other heuristics. Both  $GA + TS$  and  $TS$  solutions had a small standard deviation in the objective function values and running times. The  $GA$

algorithm, on the other hand, had standard deviations significantly higher than  $GA + TS$  and  $TS$ .

The heuristics were further tested using instances with different lane densities. Fixing the number of lanes to be 200 and number of carriers to be 15, 300 instances were generated in six groups according to the number of nodes. Statistics from the experiments are given in Table 5 and Table 6. The attributes used in the tables are similar to those used for Tables 3 and 4. In Table 5,  $\lambda$  is the lane density calculated by  $L/((n/2)(n-1))$ , where the denominator is the maximum possible number of lanes in a graph with  $n$  nodes.

From Table 5, it can be seen that the performance of the heuristics methods was not sensitive to the lane density, as the solution quality and time requirements of each method were similar.  $TS$  results were very close to  $GA + TS$  with no more than 1% difference on average, whereas the  $GA$  algorithm solutions deviated from 17 to 22% from the other algorithms.

In Table 6, additional statistics on the lane density experiments is provided, similar to those given in Table 4.

**Table 4** Lane experiments—statistics

Size	$\beta_{GA+TS}$	$\beta_{TS}$	$\beta_{GA}$	$\sigma_{GA+TS}^C$	$\sigma_{TS}^C$	$\sigma_{GA}^C$	$\sigma'_{GA+TS}$	$\sigma'_{TS}$	$\sigma'_{GA}$
100_50_10	29	21	0	162.01	158.10	178.84	2.44	1.42	14.70
200_70_10	24	26	0	195.67	192.08	250.67	3.01	1.74	13.15
300_80_15	29	21	0	177.22	167.98	406.65	1.87	2.41	22.80
400_90_20	33	17	0	167.81	181.82	380.08	1.96	1.18	59.93
500_100_20	32	18	0	252.31	273.04	650.52	2.62	0.77	14.16

**Table 5** Lane density experiments

$n$	$\lambda$	$\mu_{GA+TS}^C$	$\mu'_{GA+TS}$	$\mu_{TS}^C$	$\mu'_{TS}$	$\delta_1$	$\mu_{GA}^C$	$\mu'_{GA}$	$\delta_2$
25	0.67	7576	69.71	7604	70.03	0.38	9186	553.07	21.26
30	0.46	7886	70.45	7898	69.90	0.15	9480	557.03	20.22
40	0.26	8323	71.35	8337	71.49	0.17	9845	566.35	18.30
50	0.16	8670	72.33	8674	72.17	0.05	10284	600.10	18.62
60	0.11	8907	72.80	8925	72.96	0.20	10570	577.88	18.68
70	0.08	9166	93.03	9177	73.86	0.12	10759	612.57	17.37

**Table 6** Lane density experiments—statistics

$n$	$\beta_{GA+TS}$	$\beta_{TS}$	$\beta_{GA}$	$\sigma_{GA+TS}^C$	$\sigma_{TS}^C$	$\sigma_{GA}^C$	$\sigma'_{GA+TS}$	$\sigma'_{TS}$	$\sigma'_{TS}$
25	34	16	0	136.08	139.79	263.99	2.15	2.32	15.16
30	30	20	0	143.57	140.72	285.93	2.09	1.82	12.60
40	25	25	0	154.34	167.94	256.37	1.74	1.69	14.40
50	29	21	0	154.44	152.05	234.50	1.86	1.73	30.61
60	29	21	0	176.30	158.98	269.53	1.83	1.57	11.04
70	26	24	0	121.41	139.78	268.80	8.64	1.50	37.02



## 5. Conclusions

In this paper, optimization models used in winner determination processes in transportation procurement were introduced. Although traditional combinatorial auction models focus on carrier input using lane bids, these models deal with shipper non-price business considerations such as carrier spread, and allowed for the added benefit of including transit point costs. In one model, the shipper is able to cap the number of lanes any one carrier can win. This integer program was solved as a network flow problem and a polynomial-time algorithm provided. In a second model, a CAM with penalty and transit point costs was given to encourage more realistic bidding by carriers. The problem was shown to be NP-complete, and branch-and-bound and heuristics were developed to find solutions. Computational experiments were conducted to evaluate the algorithms on a range of test instances. It was found that among the heuristics, a hybrid genetic algorithm with tabu search provided the best solutions.

This work provides a basis for the design and development of models that address shipper non-price attributes and system constraints in CAMs, and which include carrier cost input. The use of metaheuristics, particularly a genetic algorithm with tabu search, has been shown to be effective for these problems and could be useful in other similar optimization models, especially when mixed integer programming commercial solvers cannot be applied.

*Acknowledgements*—We thank the anonymous referees for suggestions which have helped improve this work.

## References

- American Trucking Associations (2002). *American Trucking Trends 2002*. American Trucking Associations: Alexandria, Virginia.
- Caplice C and Sheffi Y (2003). Optimization based procurement for transportation services. *J Bus Logis* **24**: 109–128.
- Cormen Y, Leiserson CE, Rivest RL and Stein C (2001). *Introduction to Algorithms*, 2nd edn. MIT Press: Cambridge, MA, USA.
- Dowland KA (1996). Genetic algorithms—a tool for OR. *J Opnl Res Soc* **47**: 550–561.
- Elmaghraby W and Keskinocak P (2003). Combinatorial auctions in procurement. In: Billington C, Harrison T, Lee H, Neale J (eds). *The Practice of Supply Chain Management*. Kluwer Academic Publishers, Norwell, MA, USA, pp 245–258.
- Foster JR and Strasser S (1991). Carrier/modal selection factors: the shipper/carrier paradox. *Transportation Research Forum* **31**: 206–212.
- Gibson BJ, Sink H and Mundy R (1993). Shipper-Carrier relationship and carrier selection criteria. *Logist Transport Rev* **29**: 371–390.
- Glover F and Laguna M (1997). *Tabu Search*. Kluwer Academic Publishers: Norwell, MA, USA.
- Ledyard J *et al* (2002). The first use of a combined value auction for transportation services. *Interfaces* **32**(5): 4–12.
- Lim A, Rodrigues B and Song L (2004). Manpower allocation with time windows. *J Opnl Res Soc* **55**: 1178–1186.
- Moore EW, Warmke JM and Gorban L (1991). The indispensable role of management science in centralizing freight operations at Reynolds Metals Company. *Interfaces* **21**(1): 107–129.
- Nemhauser GL and Wolsey LA (1999). *Integer and Combinatorial Optimization*. Wiley: New York.
- Oldham J (2001). Combinatorial approximation algorithms for generalized flow problems. *J Algorithms* **38**: 135–169.
- Rhinehart LM (1989). Organizational and personal factors influencing the negotiation of motor carrier contracts: a survey of shippers and motor carriers. *Transport J* **29**: 4–14.
- Sandholm T, Suri S, Gilpin A and Levine D (2002). Winner determination in combinatorial auction generalizations. In: Castelfranchi C and Johnson WL (eds). *The Proceedings of The First International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, Vol. 1*. ACM Press, New York, USA, pp 69–76.
- Sandholm T (2002). Algorithms for optimal winner determination in combinatorial auctions. *Artificial Intelligence* **135**: 1–54.
- Sheffi Y (2004). Combinatorial auctions in the procurement of transportation services. *Interfaces* **34**(4): 245–252.
- Song J and Regan A (2002). Combinatorial auctions for transportation service procurement: the carrier perspective. Working paper UCI-ITS-LI-WP-02-9, Institute of Transportation Studies, University of California, Irvine, USA.
- Viswanathkumar G and Srinivasan G (2002). A branch and bound algorithm to minimize completion time variance on a single processor. *Comput Opns Res* **30**: 1135–1150.
- Vohra R and de Vries S (2003). Combinatorial auctions: a survey. *INFORMS J Comput* **15**: 284–309.
- Wen R and David AK (2001). A genetic algorithm based method for bidding strategy coordination in energy and spinning reserve markets. *Artificial Intelligence in Engineering* **15**: 71–79.

## Appendix

**Theorem** *The CAMPC is NP-complete.*

**Proof** Transit point costs can be neglected and taken with penalty costs in the objective function. In order to show that the CAMPC is NP-complete, we show that the decision form of the problem is NP-complete. The decision form can be stated as: *Given  $S$  carriers, a bid cost matrix  $\mathbf{B}$ , a penalty cost matrix  $\mathbf{C}$ , and an integer  $k$ , can we find a carrier-edge assignment in the representative graph with total cost  $k$ ?*

In order to prove this problem is NP-complete, it suffices to prove the problem is in NP and it is NP-hard. Obviously, given a carrier-edge assignment, it is possible to determine feasibility in polynomial time, so the problems in NP.

*Proof that the CAMPC is NP-hard:* To show the problem is NP-hard, we reduce the vertex-cover problem (VCP)—a well-known NP-complete problem—to the CAMPC. A *vertex cover* of an undirected graph  $G=(V, E)$  is a subset  $V' \subseteq V$  such that if  $(u, v)$  is an edge of  $G$ , then either  $u \in V'$  or  $v \in V'$  (or both). The VCP is to find a subset  $V'$  with the minimal cardinality (Cormen *et al*, 2001). Let  $G(V, E)$  be an instance of the VCP. We construct an instance of CAMPC in polynomial time. For the CAMPC, an instance consists of a graph  $G'(V', E')$  of node-edge relationships, a matrix for the carrier-edge bids costs, and a matrix of penalty costs.

We construct the input from  $G(V, E)$  as follows: Let  $G'(V', E') \cong G(V, E)$  (ie  $V=V'$  and  $E=E'$ ), and  $n$  the number of nodes,  $L$  the number of edges and  $S$  the number of carriers. We have  $n = |V|$  and  $L = |E|$ . Letting  $S = n$ , construct a  $S \times L$  carrier-edge bid cost matrix  $\mathbf{B}$  with  $b_{ij} = 0$  if node  $i$  is adjacent to edge  $j$ , and  $b_{ij} = \infty$  otherwise ( $1 \leq i \leq S$ ,  $1 \leq j \leq L$ ). Construct a  $S \times n$  carrier-node penalty cost matrix  $\mathbf{C} = [p_{ij}]$  with  $p_{ij} = 0$  if  $i \neq j$ , and  $p_{ij} = 1$  otherwise ( $1 \leq i \leq S$ ,  $1 \leq j \leq n$ ). As  $S = n$ , the matrix is the unit square matrix. An example is given in Figure 2:

These can be completed within polynomial time. Next we show that the VCP has a solution with  $k$  vertices if and only if the CAMPC has a solution with cost  $k$ . First, we prove that, if the vertex-cover problem has a feasible solution of  $k$  vertices, then the CAMPC has a feasible solution with cost  $k$ . Let the set of chosen vertices for the vertex-cover problem be  $V_1$ , so that  $|V_1| = k$  and let  $h(i)$  ( $1 \leq i \leq k$ ) be the index of the  $i$ th node in  $V_1$ . In the CAMPC, we choose  $k$  carriers  $s(1) \dots s(k)$  to be the edges, where  $s(i) = h(i)$  ( $1 \leq i \leq k$ ) which is possible as  $S = n$ . By the definition of the VCP, for any edge  $(u, v)$  in  $G(V, E)$ , there is a node  $h_i$  in  $V_1$  which is connected to  $(u, v)$ ; similarly, in the CAMPC, an edge  $(u, v)$  in  $G'(V', E')$  can be assigned to the carrier with index  $s(i)$ . This is a feasible solution if we set the bid cost to be 0 for carrier  $s(i)$  and edge  $(u, v)$  when node  $h(i)$  is adjacent to  $(u, v)$ . Furthermore, the penalty cost assigned to carrier  $s(i)$  at node  $h(i)$  in matrix  $\mathbf{C}$  is 1 since  $s(i) = h(i)$ . Thus, this feasible solution has cost  $k$  since each of the  $k$  carriers can only incur the cost of 1 and no other cost is involved.

Conversely, we prove that if the CAMPC has a feasible solution of cost  $k$ , the VCP also has a feasible solution with  $k$  vertices. From the definition of the CAMPC, the only way to obtain the feasible solution with cost  $k$  is to choose  $k$  pairs of

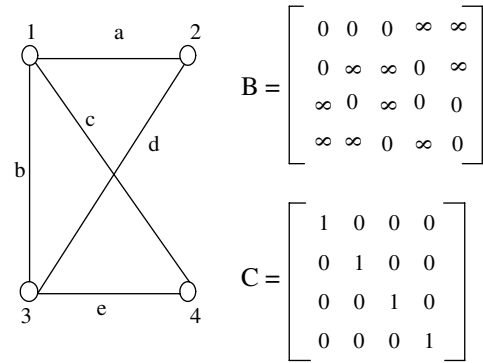


Figure 2 Reduction example.

elements  $(h(i), h(i))$  ( $1 \leq i \leq k$ ) in the diagonal of  $P$ . In the VCP, choose  $k$  vertices to be  $h(i)$  for  $1 \leq i \leq k$  as in the CAMPC with  $s(i) = h(i)$ . In the CAMPC, each edge is assigned to one of the  $k$  carriers  $s(1) \dots s(k)$ , and if edge  $(u, v)$  is assigned to carrier  $s(j)$ , then edge  $(u, v)$  is connected to node  $h(j)$  where  $h(j) = s(j)$  because the bid matrix  $\mathbf{B}$  must be 0; otherwise the feasible solution for the penalty cost problem with cost  $k$  is not possible. Thus, if the set of carriers  $s(1) \dots s(k)$  can serve the edges with a cost  $k$  in the CAMPC then the set of nodes  $h(1) \dots h(k)$  form a vertex cover with  $k$  vertices in the VCP.

Hence, we have shown that the VCP can be reduced to the CAMPC by a polynomial-time transformation so that the CAMPC is NP-hard. □

*Received June 2004;  
accepted September 2005 after four revisions*