# Transportation Bid Analysis Optimization with Shipper Input


Yunsong Guo
*School of Computing*
*National University of Singapore*
*Science Drive 2*
*Singapore 117543*
*guoyunso@comp.nus.edu.sg*

Andrew Lim, Yi Zhu
*Dept. of IEEM*
*Hong Kong Univ of Sci. and Tech.*
*Clear Water Bay*
*Kowloon, Hong Kong*
*{iealim,zhuyi}@ust.hk*

Brian Rodrigues
*School of Business*
*Singapore Management Univ*
*469 Bukit Timah Road*
*Singapore 259756*
*br@smu.edu.sg*



## Abstract

*This paper extends carrier assignment models used in bid analysis for transportation procurement to incorporate shipper business considerations. These include restricting carrier numbers, favoring incumbents and performance considerations. We provide representative models and develop solutions for these which include the use of meta-heuristics. Experimentation shows that our algorithms work well.*


## 1. Introduction

In the huge trucking market - $600 billion in the US alone [2], [1], - transportation services procurement is central to shippers whose priorities are to secure high-quality carrier service while controlling costs and organizing supply chains. In seeking transportation services from vendors, a shipper would typically tender a request for quotes (RFQ) for service for which the carrier would submit a bid where prices are determined by competition between sellers. In a bid preparation stage shippers can determine which carriers to invite for bidding and on opportunities for shipper-carrier relationships. In executing the bid process, the shipper provides bid information to carriers who respond with bids where the process can vary from one shipper to another.

The focus of this work is to study the bid analysis stage of the procurement process where decisions are made to allocate lanes to shippers. At this point shippers decide on the optimal assignment of carriers to lanes by optimization using carrier assignment models (CAM's) where the objective is to minimize total costs to the shipper and constraints include ensuring each lane is served (with enough capacity) among others. Such models, including those with combinatorial input, have been well studied (see, for example, [5], [6],[3], [8]). In this work, we extend CAM's to include a wider class of models which are enriched by the inclusion of other business considerations. Some of these considerations, garnered from extensive practical interactions, have been documented recently in [2]. These include shippers restrictions on the number of carriers that can win freight; favoring incumbents; keeping specific carriers off certain lanes; restricting carriers from serving portions of the network and incorporating performance as a factor for carrier selection. As far as the authors are aware, there has been no bid analysis study which incorporates such shipper considerations explicitly in a CAM. Further, we propose meta-heuristic techniques for CAM. These are well suited for the models we study and could be used in future work when dealing with more complex models where, for example, cost functions can be non-linear [2].

## 2. A CAM with Penalty Costs (CAMPC)

Shipper business considerations discussed here include favoring incumbents, restricting carriers to lanes, excluding or penalizing carriers from transit points and performance factors. In order to address these, we propose a basic CAM with a component of "penalty costs" in its objective. Such penalty costs can be used by the shipper as a composite proxy measure for such business considerations. This measure can be found through assessment of carriers and weighting the various business considerations. Alternatively, the model can be run specifically to analyze each of these business considerations separately. For example, if the shipper wished only to restrict carriers to certain lanes then high penalty costs can be assigned to that carrier at an appropriate node adjacent to the lane. By adjusting penalty costs the shipper can determine the best carrier assignment allocation which will meet his requirements of having all lanes serviced while using carriers which best suit his business needs (i.e. those with lowest or no penalty costs). We model this as an Integer Programming as follows:

*Parameters:*
$n$: the number of nodes
$L$: the number of lanes
$S$: the number of carriers
$M$: a sufficiently large number

$a_{ij} = 0, 1$: 1 if and only if node $i$ is adjacent to lane $j$ $(1 \leq i \leq n, 1 \leq j \leq L)$; 0 otherwise

$b_{kj} = 0, 1, 2, 3, ...$ is the bid value if edge $j$ is assigned to carrier $k$; $b_{kj} = M$ if lane $j$ is not bid for by carrier $k$ $(1 \leq k \leq S, 1 \leq j \leq L)$

$p_{ki} = 1, 2, 3, ...$is the penalty cost assigned to carrier $k$ at node $i$ $(1 \leq k \leq S, 1 \leq i \leq n)$

*Decision Variables:*

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

$y_{ki} = 0, 1$: 1 if carrier $k$ wins a lane adjacent to node $i$ $(1 \leq k \leq S, 1 \leq i \leq n)$; 0 otherwise

*Objective:*

$$\text{minimize} \sum_{k=1}^{S} \sum_{j=1}^{L} b_{kj} x_{kj} + \sum_{k=1}^{S} \sum_{i=1}^{n} p_{ki} y_{ki} \quad (1)$$

*Subject to:*

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

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

$$\sum_{j=1}^{L} a_{ij} x_{kj} \geq y_{ki}, \ 1 \leq i \leq n, 1 \leq k \leq S \quad (4)$$

$$\sum_{j=1}^{L} a_{ij} x_{kj} \leq M y_{ki}, \ 1 \leq i \leq n, 1 \leq k \leq S \quad (5)$$

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 $b_{kj}$ is not $M$. The last two constraints ensure the penalty cost constraint: Constraint (4) ensures that when $y_{ki}$ is 1, there is at least one edge $j$ connecting node $i$ to carrier $k$. Constraint (5) ensures that when the decision variable $y_{ki}$ is 0, no edge connecting node $i$ is assigned to carrier $k$. In deciding to restrict or disallow a carrier from a lane, the shipper can assign high penalty costs to the carrier at the nodes adjacent to this lane. If the shipper only wishes to restrict or disallow a carrier from a transit point, he would assign a high penalty to that carrier at that point (node).

### 2.1. Proof of NP-completeness

In order to show that the problem is **NP**-*complete*, we show that the decision problem is **NP**-*complete*. The decision problem can be stated as: *Given S carriers, a bid cost matrix B, a penalty cost matrix P, and an integer K, can we find a carrier-edge assignment in the representative graph with total cost K.*

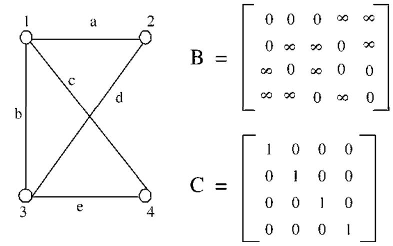


**Figure 1. Reduction Example**

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 [7]. 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 $G'(V', E') \cong G(V, E)$ (i.e. $V = V'$ 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 $B$ with $b_{ij} = 0$ if node $i$ is adjacent to edge $j$, and $b_{ij} = \infty$ otherwise $(1 \leq i \leq p, 1 \leq j \leq L)$. Construct a $S \times n$ carrier-node penalty cost matrix $P$ with $p_{ij} = 0$ if $i \neq j$, and $p_{ij} = 1$ otherwise $(1 \leq i \leq S, 1 \leq j \leq n)$. Since $S = n$, the matrix is the unit square matrix $I_{n \times n}$.

These steps can be completed within polynomial time. Next we prove: The VCP has a solution of $K$ vertices if and only if the CAMPC has a solution of 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 $s(1)...s(K)$ to serve the edges, where $s(i) = h(i) (1 \leq i \leq K)$ which is possible since $S = n$. 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, for 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 over carrier $s(i)$ and edge $(u, v)$ when node $h(i)$ is adjacent to $(u, v)$. Furthermore, the penalty cost between carrier $s(i)$ and node $h(i)$ in matrix $P$ is 1 since $s(i) = h(i)$ where the pair $(s(i), h(i))$ lies in the main diagonal. 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 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 $s(i)...s(K)$, 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 $B$ must be 0; otherwise the feasible solution for the penalty cost problem with cost $K$ is not possible. Thus, if the set of carriers $s(1)...s(K)$ can serve the edges with a cost $K$ in the CAMPC then the set of nodes $h(1)...h(K)$ 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* ■

## 2.2. Benchmarking with Branch-and-Bound

A branch-and-bound (B&B) algorithm can be used to examine all possible assignments of the edges to carriers. Performance of such an algorithm depends largely on the bounding function adopted.

Let $g_i$ $(1 \leq i \leq L)$ be the index of carrier that edge $i$ is assigned to. When $g_i$ is known for $1 \leq i \leq k$ and $g_i$ is unknown for $k+1 \leq i \leq n$, the lower bound for the total cost which the bounding function gives consists of three components: $T_1$, $T_2$ and $N$, where $T_1$ is the bid cost for edges 1 to $k$, already assigned to some carrier, $T_2$ is the bid cost lower bound for edges $k+1$ to $L$, not as yet assigned to any carriers and $N$ is the total least penalty cost possible. With $B$ as the bid cost matrix and $P$ as the penalty cost matrix, the three values could be calculated as: $T_1 = \sum_{i=1}^{k} B_{g_i,i}$, $T_2 = \sum_{i=k+1}^{L} \text{Min}\{B_{j,i}, 1 \leq j \leq p\}$, and $N = \sum_{i=1}^{n} p_i$ , where $p_i$ is the penalty cost incurred at node $i$. If there is any edge $j$ with $1 \leq j \leq k$ connected to node $i$, $p_i$ is the sum of penalty costs incurred on carriers who serve some edge $e$ where $1 \leq e \leq k$; if there is no edge $j$ with $1 \leq j \leq k$ connected to node $i$, then $p_i$ is then the minimum penalty cost incurred with any carrier that can cover any edge $j$ that is connected to node $i$. Thus, at any point in the B&B process, the bounding function used is $(T_1 + T_2 + N)$. In the experiments, we see that this bounding function is efficient for test cases with large penalty costs compared to bid costs.

## 2.3. Metaheuristics Applied to the CAMPC

Metaheuristics can provide effective solutions to complex problems and have increasingly been used in sophisticated logistics DSS's. Because the CAMPC is **NP**-*complete*, we used metaheuristics and, in particular, developed genetic algorithm and tabu search approaches for it. We then constructed a hybrid of these which suited the problem better. Heuristics have previously been developed for combinatorial auctions. Here, we provide brief outlines of the use of these metaheuristics.

**2.3.1. Using a Genetic Algorithm** Genetic algorithms (GA) is a well-known metaheuristic approach used of combinatorial optimization problems which has been applied in task allocation problems (see, for example, [9]). Here we use a GA procedure to solve the CAMPC. In the following, the algorithm is outlined and chromosome representation, crossover and mutation operations which are used are discussed.

*Genetic Algorithm Outline.* The GA approach used for the CAMPC is outlined as follows: For every distinct chromosome pair $c_1$ and $c_2$, in a subset of the a randomly generated initial population *pop_size*, perform the crossover operation and mutate these newly- generated chromosomes with probability $q$. Evaluate the objective function values of new and old chromosomes and retain *pop_size* best chromosomes. In our implementation, the termination condition is when the best solution has not improved within a fixed number of iterations or if a maximum number of iterations is reached.

*Chromosome Representation.* Solutions were encoded as strings $s$ containing $L$ integers, where $L$ is the number of edges to be covered. Each integer $s(i)$ $(1 \leq i \leq m)$ represents the index of the carrier that is assigned to edge $i$, so $1 \leq s(i) \leq p$. Initial solutions are generated so that $b_{s(i),i} < \infty$ is ensured.

*Crossover Operation.* The crossover operation is executed as follows: For chromosomes $s$ and $s'$, we first generate a random number $p$ between 1 and $m-1$ and cut these at position $k$. The subsequence $(s(k+1), s(k+2), \ldots, s(m))$ is put after $(s'(1), s'(2), \ldots, s'(k))$ and $(s'(k+1), s'(k+2), \ldots, s(m))$ is put after $(s(1), s(2), \ldots, s(k))$ to produce the two new chromosomes. Following this operation, newly formed chromosomes are always feasible solutions, given their parents are feasible. This is since we do not shift the positions of the carriers assignment when performing crossover.

*Mutation Operation.* For each chromosome in the offspring generation, we used a probability $q$ to perform the mutation operation. The mutation operation selects an integer $s(i)$ $(1 \leq i \leq m)$ and changes its value, changing the carrier for edge $i$. The mutation ensures the new carrier can be assigned to edge $i$.

**2.3.2. Using Tabu Search** Tabu search (TS) is an aggressive metaheuristic search procedure which moves iteratively from one solution to another by moves in a neighborhood space with the assistance of adaptive memory[4].

TS was applied to the CAMPC, where the same solution representation is used as GA, i.e., a string with $m$ integers. A neighborhood move is defined as changing a carrier-edge assignment in the solution (similar to the mutation operator in GA) and the tabu list consists of the recent *tabu tenure* (the value identifies the number of iterations a particular restriction remains in force) solutions, which will be forbidden in the next move.

We describe our TS heuristic framework as follows: Get an initial solution $x_{now}$. Set $x_{best} = x_{now}$. If the termination condition is satisfied, quit with $x_{best}$. Otherwise, generate the neighborhood set $N(x_{now})$. Evaluate each candidate solution $x_{trial}$ as $f(x_{trial})$. Select $x_{next} = \min_{x_{trial} \in N(x_{now})} f(x_{trial})$, and $x_{trial}$ is not in the tabu list. Update the tabu memory. Set $x_{now} = x_{next}$. If $f(x_{now}) < f(x_{best})$, set $f(x_{best}) = f(x_{now})$.

**2.3.3. Using Genetic Algorithm and Tabu Search (GA + TS)** The initial solution selected for TS method would affect the result quality. So we use solutions obtained by GA method as initial solutions for TS, in a hybrid denoted by GA+TS. The results turn out to be best in the experiments.

## 3. Experimental Results

To analyze the performance of our solution methods, we conducted series of experiments. 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 256M memory.

The experiment consisted of two parts. The first part uses small sizes instances to compare the results of the heuristic methods with the optimal results obtained by our B&B method. The second part compares heuristic methods for large scale problems.

### 3.1. Optimality Comparisons with Branch-and-Bound

We generated 30 small-scale instances. The first 15 instances have bid costs which are about 10 times larger than the penalty costs. The next 15 instances have the penalty costs which are about 10 times larger than bid costs. In the first 15 instances, the number of nodes, number of edges and number of carriers are 10, 16 and 6 respectively; these numbers are 10, 31 and 6 in the other half of the problems tested. The results are shown in Table 1. (T1, T2, T3 and T4 depict running time for B&B, GA, TS and GA+TS respectively and OPT is the optimal value found by B&B)

From the table, we see that the performance of the heuristics is good when the instances are small. This would be a basic expectation of a good heuristic. All three methods provide solutions very close (or exactly equal) to the optimal solutions. Although it seems that GA requires much more time than B&B in some cases, this is because GA requires a number of iterations and the time consumed is not entirely dependent on the number of edges or number of nodes and carriers, but rather on the population for each generation.

TS and GA+TS methods efficiently provide results of good quality. The two methods both consume very little time ($\leq 2$ seconds) compared with B&B and GA for all instances tested. This is because tabu search is effective at moving out of local optima and can converge quickly.

In addition, we noticed that the time requirements for B&B are much larger than GA, TS and GA+TS. This is because the B&B method depends largely on the input size and the actual structure of the graph and cost matrices. In contrast, the heuristic methods are more likely to be independent of input size and graph structure.

### 3.2. Comparisons between Heuristics

We examine the performance of different metaheuristic methods for large scale instances. We conducted experiments using two categories of test data: one category classifies instances using number of edges and the other classifies them using edge densities.

80 instances with various number of edges are generated to test the metaheuristics. They are grouped into 8 groups with 10 instances in each group. The results are shown in Table 2. (N is the number of nodes; T1, T2, and T3 are the running times for GA, TS and GA+TS; C1, C2 and C3 are the results found by GA, TS and GA+TS respectively; $\delta$ measures the cost difference from GA+TS).

It is clear that the GA+TS hybrid performs better when compared with GA and TS. It is also interesting to note that GA consumes much less time than TS and GA+TS. This is since GA is relatively more independent of input size. On the other hand, the running time for TS and GA+TS methods increase with the input size.

We tested the heuristic methods using instances with different edge densities. There were 100 instances which are classified by the edge density into 10 groups (The number of edges and carriers are 1200 and 100 resp. for all test instances). Edge density is defined as the ratio between the number of edges and the number of edges in a complete graph with the same number of nodes. The results are shown in Table 3 (E denotes the edge density).

| Case | OPT | BB | GA | | TS | | GA+TS | | Case | OPT | B&B | GA | | TS | | GA+TS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | T1[1] | T2 | $\delta^2$ | T3 | $\delta$ | T4 | $\delta$ | | | T1 | T2 | $\delta$ | T3 | $\delta$ | T4 | $\delta$ |
| 1 | 17727 | 3 | 72 | 0.0 | 1 | 0.0 | 0 | 0.0 | 16 | 24328 | 144 | 63 | 0.0 | 2 | 0.0 | 1 | 0.0 |
| 2 | 17034 | 7 | 76 | 0.0 | 1 | 0.0 | 0 | 0.0 | 17 | 21741 | 18 | 68 | 0.0 | 2 | 0.0 | 1 | 0.0 |
| 3 | 17137 | 275 | 68 | 0.0 | 1 | 0.1 | 0 | 0.0 | 18 | 24836 | 40 | 62 | 4.0 | 2 | 0.3 | 1 | 3.7 |
| 4 | 17471 | 3 | 72 | 0.0 | 1 | 0.0 | 0 | 0.0 | 19 | 26780 | 134 | 66 | 1.1 | 2 | 1.1 | 1 | 0.2 |
| 5 | 17148 | 23 | 76 | 0.0 | 1 | 0.2 | 0 | 0.0 | 20 | 23761 | 100 | 78 | 0.8 | 1 | 0.0 | 1 | 0.0 |
| 6 | 17044 | 56 | 71 | 0.0 | 2 | 0.0 | 0 | 0.0 | 21 | 24152 | 41 | 67 | 3.7 | 1 | 0.0 | 1 | 0.6 |
| 7 | 16967 | 43 | 67 | 0.0 | 1 | 0.0 | 0 | 0.0 | 22 | 23686 | 87 | 66 | 0.0 | 2 | 0.2 | 1 | 0.0 |
| 8 | 17163 | 20 | 69 | 0.0 | 1 | 0.0 | 0 | 0.0 | 23 | 24989 | 42 | 57 | 0.2 | 1 | 0.0 | 1 | 0.2 |
| 9 | 17648 | 1 | 73 | 0.0 | 1 | 0.0 | 0 | 0.0 | 24 | 23641 | 172 | 56 | 0.1 | 2 | 0.2 | 1 | 0.0 |
| 10 | 17173 | 17 | 67 | 0.0 | 1 | 0.0 | 0 | 0.0 | 25 | 24181 | 42 | 54 | 0.4 | 1 | 0.0 | 1 | 0.4 |
| 11 | 17629 | 5 | 68 | 0.0 | 1 | 0.0 | 0 | 0.0 | 26 | 25691 | 96 | 64 | 0.3 | 2 | 0.3 | 1 | 0.3 |
| 12 | 17268 | 16 | 71 | 0.0 | 1 | 0.7 | 0 | 0.0 | 27 | 27119 | 191 | 64 | 0.8 | 2 | 0.0 | 1 | 0.8 |
| 13 | 16941 | 66 | 71 | 0.0 | 2 | 0.0 | 0 | 0.0 | 28 | 23764 | 80 | 59 | 0.3 | 1 | 0.0 | 1 | 0.0 |
| 14 | 17027 | 21 | 71 | 0.0 | 1 | 0.0 | 0 | 0.0 | 29 | 24747 | 102 | 59 | 0.4 | 1 | 0.0 | 1 | 0.4 |
| 15 | 17562 | 71 | 123 | 0.0 | 2 | 0.0 | 0 | 0.0 | 30 | 22886 | 17 | 62 | 0.0 | 3 | 0.0 | 1 | 0.0 |

[1] All time measurements are in seconds

[2] $\delta$ is the difference from optimal as a percentage

| N | T1 | C1 | $\delta_1$ | T2 | C2 | $\delta_2$ | T3 | C3 |
|---|---|---|---|---|---|---|---|---|
| 300 | 23 | 786426 | 10.3 | 86 | 720376 | 1.0 | 83 | 713183 |
| 500 | 21 | 1324770 | 12.3 | 100 | 1196558 | 1.4 | 98 | 1179796 |
| 800 | 17 | 2134452 | 13.6 | 131 | 1928014 | 2.6 | 121 | 1879432 |
| 1000 | 26 | 2674481 | 13.9 | 150 | 2417912 | 3.0 | 142 | 2347893 |
| 1200 | 22 | 3211872 | 13.8 | 164 | 2921399 | 3.5 | 143 | 2822622 |
| 1500 | 29 | 4008714 | 13.1 | 165 | 3681958 | 3.9 | 160 | 3544463 |
| 1800 | 21 | 4802923 | 12.7 | 182 | 4435753 | 4.1 | 180 | 4262138 |
| 2000 | 18 | 5319605 | 12.2 | 189 | 4943677 | 4.3 | 189 | 4741080 |

**Table 2. Experimental Results for Lanes**

| N | E | T1 | C1 | $\delta_1$ | T2 | C2 | $\delta_2$ | T3 | C3 |
|---|---|---|---|---|---|---|---|---|---|
| 200 | 0.06 | 48 | 2757260 | 18.1 | 77 | 2430293 | 4.1 | 80 | 2334522 |
| 180 | 0.07 | 47 | 2827862 | 17.8 | 79 | 2498818 | 4.1 | 88 | 2399984 |
| 160 | 0.09 | 43 | 2884790 | 17.4 | 85 | 2557879 | 4.1 | 95 | 2457049 |
| 140 | 0.12 | 51 | 2930276 | 17.1 | 90 | 2598392 | 3.9 | 90 | 2501516 |
| 120 | 0.17 | 45 | 2997426 | 16.8 | 102 | 2671001 | 4.1 | 98 | 2565863 |
| 100 | 0.24 | 53 | 3059023 | 16.4 | 107 | 2727848 | 3.8 | 109 | 2628952 |
| 80 | 0.38 | 64 | 3094491 | 15.8 | 115 | 2767766 | 3.6 | 114 | 2672500 |
| 70 | 0.50 | 49 | 3122390 | 15.2 | 124 | 2801255 | 3.4 | 123 | 2710441 |
| 60 | 0.68 | 42 | 3151600 | 15.2 | 131 | 2828566 | 3.4 | 129 | 2736611 |
| 50 | 0.98 | 44 | 3169459 | 14.7 | 137 | 2857410 | 3.4 | 141 | 2763061 |

**Table 3. Experimental Results for Lane Densities**

We find that the GA+TS method is again the best performing heuristic for various edge densities achieved within comparable running times.

## 4. Conclusion

In this paper, we discussed one possible representation for CAM used in bid analysis for transportation procurement. Although traditional models address only carrier input via bids, the model extends CAM to address shipper business considerations such as carrier maximum numbers and coverage, shipper-carrier synergies and carrier performance. As an extension we proposed a CAM with cost penalties which would address the needs of shippers other business considerations. We show this problem to be **NP**-*complete* and propose a branch-and-bound method together with several heuristics for it. Several groups of test instances are generated for experiments to test these solution approaches. We found that the GA+TS approach is superior to other methods in various sizes and densities.

## References


[1] American Trucking Associations, (2002), American Trucking Trends 2002, Alexandria, Virginia

[2] Caplice, C and Sheffi, Y. (2003), Optimization based procurement for transportation services, to appear in Journal of Business Logistics

[3] Elmaghraby, W. and Keskinocak, P., (2003), Combinatorial Auctions in Procurement, to appear in The Practice of Supply Chain Management, eds. C. Billington, T. Harrison, H. Lee, J. Neale, Kluwer Academic Publishers

[4] Glover, F. and Laguna, M., (1997), Tabu Search, Kluwer Academic Publishers, 1997.

[5] Klemperer, P., (1999), Auction theory: A guide to the literature, to appear in Journal of Economic Surveys

[6] Ledyard, J., Olsen, M., Porter, D., Swanson, J., Torma, D. (2002), the first use of a combined value auction for transportation services, Interfaces, Vol. 32 (5), pp. 4-12

[7] Thomas, H. Cormen, C.E., Leiserson, Rivest, R.L. and Stein, C., (2001), Introduction to Algorithms, 2nd Edition, MIT Press

[8] Vohra, R. and de Vries, S., Combinatorial Auctions: A survey, (2003), to appear in INFORMS Journal of Computing

[9] Wen, F., and David, A. K., (2001), A genetic algorithm based method for bidding strategy coordination in energy and spinning reserve markets, Artificial Intelligence in Engineering, 15(1), pp. 71–79