Asia-Pacific Journal of Operational Research
Vol. 21, No. 2 (2004) 241–257

AN INTEGER L-SHAPED ALGORITHM FOR TIME-CONSTRAINED TRAVELING SALESMAN PROBLEM WITH STOCHASTIC TRAVEL AND SERVICE TIMES

S. Y. TENG, H. L. ONG and H. C. HUANG

Industrial and Systems Engineering Department
National University of Singapore, 10 Kent Ridge Crescent
Singapore 119260

Received August 2003
Revised January 2004

The time-constrained traveling salesman problem (TCTSP) is a variant of the classical traveling salesman problem, where only a subset of the customers can be visited due to the time limit constraint. In this paper, we consider the TCTSP with stochastic travel and service times. Given the normal working hours T and a tolerance time ΔT, the total travel and service times of a route can exceed T as long as it is within $T + \Delta T$, though a penalty proportional to the amount in excess of T will be imposed. The problem consists of optimally selecting and sequencing a subset of customers to visit in the presence of random travel and service times to maximize the expected profit while satisfying the time limit constraint. We formulate the problem as a two-stage stochastic program with recourse, and propose an integer L-shaped solution method for solving it. Computational results show that the algorithm can solve problems with moderate size to optimality within reasonable time.

Keywords: Integer L-shaped method; time-constrained traveling salesman problem; stochastic program with recourse; Benders' decomposition.

1. Introduction

Given a set of cities, the classical traveling salesman problem (TSP) tries to determine a minimal cost cycle that passes through each node exactly once and starts and ends at the same city. The problem has many applications, such as sequencing of jobs on a single machine (Gilmore and Gomory, 1964), material handling in a warehouse (Ratliff and Rosenthal, 1981), large-scale integration (VLSI) chip fabrication (Korte, 1989), and X-ray crystallography (Bland and Shallcross, 1989). The TSP is an NP-Hard problem (Aarts and Lenstra, 1997) and has been well studied in the literature (see Lawler *et al.*, 1985; Jünger *et al.*, 1994; Reinelt, 1994; Aarts and Lenstra, 1997; and Korte and Vygen, 2000).

The time-constrained traveling salesman problem (TCTSP), which was first introduced and discussed by Cloonan (1966) is a variant of the classical TSP. The objective of TCTSP is to maximize the profit realized from serving a subset of customers, subject to the time constraint imposed on the problem. Gensch (1978) proposed a solution method based on tree search for an industrial application of this problem; and Golden *et al.* (1981) developed a simple iterative procedure for the problem. Some researchers also call TCTSP the selective traveling salesman problem (STSP) where they consider a preset constant route length as the constraint. Exact and heuristic algorithms for STSP have been developed by Laporte and Martello (1990) and Gendreau *et al.* (1998a,b).

In the TCTSP, due to the effective working time limit constraint, one factor that directly affects the total profit generated from the TCTSP tour is the travel and service times required for visiting the customers, which is usually assumed to be deterministic. However, in practical situations, both travel time and service time may not be known with certainty in advance. The weather conditions (rain or snow) and the traffic conditions (road repair or traffic accidents) may affect the travel time between the customers; while the service time may depend on the kind of service a customer requires. Therefore, it is important to consider the stochastic nature of the travel and service times of this problem. However, for the TCTSP, or even in a much broader context, the vehicle routing problem, studies focused on the stochastic travel and service times are quite few. In Laporte *et al.* (1992), they addressed the vehicle routing problem with stochastic travel times. Given a time limit on the total travel and service times of a route over which a penalty would occur, the problem is to form the vehicle routes to minimize the total fixed cost, routing cost, and the expected penalty incurred.

The problem studied in this paper can be described as follows: Given a time limit T and a tolerance time ΔT, the total travel and service times of a route can exceed T as long as it is within $T + \Delta T$, though a penalty proportional to the amount in excess of T will be imposed. The problem consists of optimally selecting and sequencing a subset of customers to visit in the presence of random travel and service times to maximize the expected profit while satisfying the time limit constraint. Here, the randomness of the travel and service time variables would not only affect the visiting order of the customers, but also the selection of a set of customers to be included in the tour. Another feature of the problem is that a penalty incurs whenever the total time of the route exceeds T; furthermore, a solution becomes infeasible once the total time of the route in excess of T exceeds time tolerance ΔT.

We formulate the problem as a two-stage stochastic problem with recourse. A subset of the customers has to be optimally selected and sequenced before the random travel and service times are known, these are called first-stage decisions. Once the customers are selected and their visiting order is fixed, it is possible to calculate the total travel and service times associated with the tour. Thus, in the second-stage, recourse actions can be taken to impose an expected penalty on the objective function. In practice, drivers usually are paid overtime for work done after normal hours; it is therefore reasonable to set the penalty to be proportional to the total travel and service times of the route in excess of a preset constant T.

The L-shaped method of Van Slyke and Wet (1969) is a cutting plane or Benders' decomposition (Benders, 1962) technique for solving the two-stage stochastic linear problem when the random variables have finite support. The name "L-shaped method" is due to the special block structure of the two-stage stochastic problem in its extensive form. Birge and Louveaux (1988) extended the L-shaped algorithm with single optimality cut to a multicut algorithm. They showed that the effectiveness of the multi-cut algorithm is conditional: it is more effective when the number of realizations of the random variables is not significantly larger than the number of first-stage constraints. In Laporte and Louveaux (1993), they presented an integer L-shaped method for the stochastic integer program with complete recourse, in which a branching procedure is incorporated into the L-shaped method to resolve the integrality of the variables. New optimality cuts were derived for the case when first-stage variables are binary. These cuts are more efficient when random variables have many states or have continuous distributions, or when the second-stage problem cannot be formulated in terms of first-stage variables. Some lower bounds on the second-stage value function were also presented in the paper. In Laporte *et al.* (1992), a branch and cut algorithm was proposed, and optimality cuts were generated in a similar way as in Laporte and Louveaux (1993). The integer L-shaped method has also been applied to the vehicle routing problem with stochastic demands. Hjorring and Holt (1999) derived more effective optimality cuts and a tight global lower bound on the second-stage value function based on the concept of partial routes for the single vehicle case. Laporte *et al.* (2002) studied the lower bound on the second-stage value function for the normal and Poisson distributed demands. They also constructed their optimality cuts based on the concept of partial routes presented in Hjorring and Holt (1999). Gendreau *et al.* (1995) applied the integer L-shaped method to the vehicle routing problem with stochastic demands and customers.

The purpose of this paper is to present an integer L-shaped algorithm for the TCTSP with stochastic travel and service times. In comparison with Laporte *et al.* (1992), their problem is a two-stage stochastic problem with complete recourse: though penalty may occur in the second-stage problem, the first-stage solution is always second-stage feasible. In our problem, however, due to the constraint that the total travel and service times of a tour in excess of time limit T cannot exceed ΔT, in addition to optimality cuts, feasibility cuts must also be considered in the second-stage problem, which makes the problem more difficult to solve. This paper is organized as follows: in Sec. 2, we give a formal description of the problem and formulate it as a stochastic program with recourse. Valid constraints used in the integer L-shaped algorithm are derived in Sec. 3. The integer L-shaped algorithm is described in Sec. 4. Section 5 presents the computational results and analysis. Finally, the results obtained in this study are summarized in Sec. 6.

2. Problem Description and Model Formulation

The TCTSP with stochastic travel and service times considered in this paper can be described as follows. Let $G = (V, A)$ be a complete graph, where $V = \{0, 1, \ldots, n\}$ is a vertex set, and $A = \{(i, j) \mid i, j \in V\}$ denotes a set of arcs. Vertex 0 represents the depot, and $1, \ldots, n$ denote n customers. Arc (i, j) represents the distance traveled between customers i and j. Associated with each customer i, there is a profit r_i and a service time τ_i; and associated with each arc, there is a travel time t_{ij}. Here, the travel and service times can be independent or dependent, discrete or continuous random variables. However, here we assume that the travel time for each arc and the service time for each node are independent discrete random variables. The "independent" assumption for the service time is quite obvious. For the travel time, it may or may not be independent depending on the various factors taken into consideration. For example, it is dependent if weather condition is the only factor considered. However, if some other factors, such as road repair, which affects a certain part of a road, are considered, the independent assumption may be more realistic.

Assume that T is the maximum effective working time, and ΔT is the maximum amount of time allowed to exceed T. The objective is to maximize the total profit realized from visiting a subset of the customers without violating the time limit constraint.

In stochastic programming, two versions of the problem are commonly considered: chance constrained programming and stochastic programming with recourse. In this paper, we present a recourse model for the TCTSP with stochastic travel and service times.

In addition to the notations V, A, r_i, T, and ΔT described above, the following notations are used in the model formulation.

$V' = V \backslash \{0\}$.

ξ = A vector of random variables corresponding to travel and service times. Assume that ξ has a finite number of realizations, $\xi^1, \xi^2, \ldots, \xi^K$, where K is the number of realizations of vector ξ.

p_k = The probability that the random vector ξ takes on the realization ξ^k.

$\theta(\xi^k)$ = The total travel and service times of the route in excess of T when the realization of the random variable is ξ^k.

$t_{ij}^{\xi^k}$ = A random variable representing time of traveling arc (i, j) when the realization of the random variable is ξ^k.

$\tau_j^{\xi^k}$ = A random variable representing service time of visiting node j when the realization of the random variable is ξ^k.

β = The unit penalty cost for total time of the route in excess of T.

$$x_{ij} = \begin{cases} 1 & \text{if arc}(i, j) \in A \text{ is traversed,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if node } j \in V' \text{ is visited,} \\ 0 & \text{otherwise.} \end{cases}$$

The recourse model can be formulated as follows:

$$Z = \max \left(\sum_{j \in V'} r_j y_j - \beta \sum_{k=1}^{K} p_k \theta(\xi^k) \right) \tag{2.1}$$

subject to:

$$\sum_{j \in V'} x_{0j} = 1, \tag{2.2}$$

$$\sum_{j \in V'} x_{j0} = 1, \tag{2.3}$$

$$\sum_{(i,j) \in A} x_{ij} = y_j \quad \forall j \in V', \tag{2.4}$$

$$\sum_{(i,j) \in A} x_{ij} = y_i \quad \forall i \in V', \tag{2.5}$$

$$\sum_{\substack{i \in S \\ j \in S}} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V', \ \ |S| \geq 3, \tag{2.6}$$

$$\theta(\xi^k) \geq \sum_{(i,j) \in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j \in V'} \tau_j^{\xi^k} y_j - T, \quad k = 1, 2, \ldots, K, \tag{2.7}$$

$$0 \leq \theta(\xi^k) \leq \Delta T, \quad k = 1, 2, \ldots, K, \tag{2.8}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in A, \tag{2.9}$$

$$y_j \in \{0, 1\} \quad \forall j \in V'. \tag{2.10}$$

In the above formulation, the objective function (2.1) is to maximize the expected profit generated (including the expected penalties incurred) from visiting a subset of n customers. Constraints (2.2) and (2.3) ensure that the route must start from depot and go back to depot in the end. Constraints (2.4) and (2.5) indicate that, if node j is not included in the tour, no arcs incident to j are included. Otherwise, there must be one arc going into and one arc coming out from node j. Constraint (2.6) is the sub-tour elimination constraint, which guarantees that except a sub-tour including the depot, no other sub-tour in set V' is allowed. Constraints (2.7) and (2.8) ensure that if the total time of a route including travel and service is greater than T, the excess amount must be within ΔT.

3. Valid Constraints Considered in the Integer L-Shaped Algorithm

In the above model, if we consider constraints (2.2)–(2.6) as one block, and write constraints (2.7) and (2.8) in extensive form for each $k = 1, 2, \ldots, K$, it is clear that the problem considered here has the block-angular structure, and therefore can be solved by the L-shaped method based on Benders' decomposition (Benders, 1962) technique. The basic idea of the L-shaped algorithm is to approximate the term $\beta \sum_{k=1}^{K} p_k \theta(\xi^k)$ (second-stage value function) in the objective function (2.1), which involves a solution of the second-stage recourse linear program. This is done by building a first-stage problem using an approximate term, and only evaluating the second-stage value function exactly in the second-stage sub-problem.

3.1. *The first-stage problem–current problem*

At a given phase of the algorithm, we call the following first-stage problem as the current problem (CP):

(CP)

$$Z = \max \left(\sum_{j \in V'} r_j y_j - \eta \right)$$

subject to:

$$\sum_{j \in V'} x_{0j} = 1,$$

$$\sum_{j \in V'} x_{j0} = 1,$$

$$\sum_{(i,j) \in A} x_{ij} = y_j \quad \forall j \in V',$$

$$\sum_{(i,j) \in A} x_{ij} = y_i \quad \forall i \in V'.$$

Set of illegal route elimination constraints (3.1)

Set of optimality constraints (3.2)

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A,$$

$$y_j \in \{0,1\} \quad \forall j \in V'.$$

The above CP is obtained by two relaxations. The sub-tour elimination constraints (2.6) and the second-stage feasibility constraints (2.7) and (2.8) are relaxed in a number of constraints in (3.1) known as feasibility cuts. Constraints (2.7) and (2.8) are relaxed because they are not known in advance. Once we get the first-stage solution, we know which of these constraints are violated. Constraints (2.6) are relaxed because, though they are known, the constraints are so numerous that it would be unrealistic to impose all of them to the problem. Finally, the second-stage expected value function $\beta \sum_{k=1}^{K} p_k \theta(\xi^k)$ is relaxed by an estimated bound η

and constraints (3.2) known as optimality cuts. Note that in the initial first-stage problem (first CP), the constraint set (3.1) may be empty and constraint set (3.2) may only contain the constraint $\eta \geq 0$. In the subsequent iterations, constraint set (3.1) includes the newly identified sub-tour elimination constraints and the second-stage feasibility constraints; while constraint set (3.2) includes the newly identified optimality cuts.

Given a first-stage solution (x, y, η) to the above CP, we can get the following second-stage problem, and derive the feasibility and optimality cuts based on it.

3.2. *The second–stage problem*

$$\min \quad w = \beta \sum_{k=1}^{K} p_k \theta(\xi^k) \tag{3.3}$$

$$\theta(\xi^k) \geq \sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T, \quad k = 1, 2, \ldots, K, \tag{3.4}$$

$$\theta(\xi^k) \leq \Delta T, \quad k = 1, 2, \ldots, K, \tag{3.5}$$

$$\theta(\xi^k) \geq 0, \quad k = 1, 2, \ldots, K. \tag{3.6}$$

Since the second-stage problem is a linear programming (LP) problem with continuous variables, we can derive the feasibility cut and the optimality cut from the dual problem.

Similar to the application of Benders' decomposition method for the mixed integer program, we add both the feasibility cuts and the optimality cuts when we get an integer first-stage solution, which corresponds to a set of selected customers to be visited.

3.3. *The feasibility cuts*

For each k, let I_+^k and I_-^k be the dual variables corresponding to constraints (3.4) and (3.5) respectively. Then the dual problem corresponding to the above second-stage problem can be formulated as follows:

(DSSP)

$$\text{Max} \quad \phi = \sum_{k=1}^{K} \left(\left(\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \right) \mathrm{I}_+^k - \Delta T * \mathrm{I}_-^k \right)$$

subject to:

$$\mathrm{I}_+^k - \mathrm{I}_-^k \leq \beta^* p_k, \quad k = 1, 2, \ldots, K,$$

$$\mathrm{I}_+^k \geq 0, \;\; \mathrm{I}_-^k \geq 0, \quad k = 1, 2, \ldots, K.$$

The above problem can be separated into K problems. For each k, $k = 1, 2, \ldots K$, we have,

(DSSP^k)

$$\text{Max} \quad \phi^k = \left(\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \right) \mathrm{I}_+^k - \Delta T * \mathrm{I}_-^k.$$

subject to:

$$\mathrm{I}_+^k - \mathrm{I}_-^k \le \beta * p_k,$$

$$\mathrm{I}_+^k \ge 0, \quad \mathrm{I}_-^k \ge 0$$

If the primal second-stage problem is infeasible, then at least one of the above problems (DSSP^k) is unbounded. An extreme ray of the feasible region of the dual problem (DSSP^k) is $\nu = (\mathrm{I}_+^k, \mathrm{I}_-^k) = (1, 1)$. Since the above problem (DSSP^k) is a maximization problem, if at the direction of the extreme ray,

$$\begin{pmatrix} \sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \\ -\Delta T \end{pmatrix}' \begin{pmatrix} 1 \\ 1 \end{pmatrix} > 0$$

holds, then the dual problem (DSSP^k) is unbounded and the primal second-stage problem is infeasible. To eliminate this first-stage solution, we can add a feasibility constraint as follows:

$$\begin{pmatrix} \sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \\ -\Delta T \end{pmatrix}' \begin{pmatrix} 1 \\ 1 \end{pmatrix} \le 0 \tag{3.7}$$

At the current first-stage solution, if we can find and add the strongest feasibility constraint, it is sufficient to eliminate all other violated feasibility constraints for the current solution. To get the strongest feasibility constraint, we choose the constraint corresponding to ξ^k, which makes the total travel and service times in excess of the time limit, $\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T$, the longest, and add the constraint into constraint set (3.1) of the first-stage problem (CP).

The above feasibility cut (3.7) indicates that, for this type of problem, the second-stage feasibility cut is very intuitive: the cut simply means that, at the current solution, under each realization of the random vector, the total travel and service times must be within the sum of the time limit (T) and the time tolerance (ΔT). Therefore, given a first-stage solution (x, y, η), we can simply check whether realization ξ', which produces the longest total travel and service times, is second-stage feasible. In the case it is not feasible, add a feasibility cut (3.7), which is constructed based on realization ξ', in terms of variables x_{ij} $((i,j) \in A))$, and y_j $(j \in V')$.

3.4. *The optimality cuts*

In case the first-stage solution (x, y, η) is feasible, by observing the feasible region of the dual problem (DSSP^k), we know that there are two extreme points: (0, 0) and $(\beta p_k, 0)$. From the objective function of DSSP^k, it is clear that if the following expression holds:

$$\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \geq 0$$

then the dual optimal solution occurs at the extreme point $(\beta p_k, 0)$. Hence, an optimality cut can be derived as

$$\phi^k \geq \left(\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \right) \beta p_k.$$

Otherwise, the optimal solution occurs at the extreme point (0, 0), and the optimal cut is

$$\phi^k \geq 0.$$

Therefore, in a single-cut algorithm, we can aggregate the cuts into a single optimality cut as

$$\eta \geq \sum_{k=1}^{K} \left(\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \right) \beta p_k \tag{3.8}$$

for all ξ^k with $\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \geq 0$.

In a multi-cut algorithm, we add the following cuts:

$$\eta \geq \sum_{k=1}^{K} \phi^k \tag{3.9}$$

$$\phi^k \geq \left(\sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \right) \beta p_k \quad \text{if} \sum_{(i,j)\in A} t_{ij}^{\xi^k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T \geq 0, \tag{3.10}$$

$$\phi^k \geq 0 \qquad \text{if} \sum_{(i,j)\in A} t_{ij}^{\xi_k} x_{ij} + \sum_{j\in V'} \tau_j^{\xi^k} y_j - T < 0. \tag{3.11}$$

For the optimality cut, similar to the feasibility cut case, we can also construct it intuitively. In the case of single optimality cut, at the current solution, we can calculate the penalty incurred for each realization of the random vector (the total travel and service times in excess of the time limit T multiplied by the unit penalty cost β). Then an optimality cut (3.8) can be constructed by ensuring that η is not less than the expected penalty incurred at the current solution. In other words, the approximate bound η is tightened by the amount of the expected penalty incurred at the current solution. In the case of multi-cut, again, η should be no less than the expected penalty incurred at the current solution; meanwhile, for each realization of the vector, an optimality cut is constructed to ensure that the amount of penalty corresponding to that realization of vector ξ is imposed on the current solution.

3.5. *Sub-tour elimination constraints*

For the TCTSP, since we only select a subset of customers in the solution, sub-tour elimination constraints only eliminate those sub-tours that do not contain the depot. At an integer solution, it is easy to detect a sub-tour, because once a main tour containing the depot is formed, other loops are sub-tours. Suppose we have a subset $S = (i_0, i_1, i_2, \ldots, i_u, i_{u+1} = i_0)$, which forms a sub-tour, we can add the following constraint to eliminate such a sub-tour:

$$\sum_{l=0}^{u} x_{i_l i_{l+1}} \leq u. \tag{3.12}$$

Since any node can be dropped from the solution in the TCTSP, we can also express the sub-tour elimination constraint in terms of the edge variables x_{ij} and the node variables y_i.

$$\sum_{l=0}^{u} x_{i_l i_{l+1}} \leq \sum_{i \in S \setminus \{j\}} y_i \quad \text{for } j \in S. \tag{3.13}$$

3.6. *The time limit constraint added to the initial first-stage problem*

In the initial first-stage (current) problem, after constraints (2.7) and (2.8) are relaxed, there is no time limit constraint. The feasibility cuts, playing a similar role as the time limit constraint, will only be gradually introduced into the current problem by solving the second-stage problems. To avoid including all customers into the first-stage solution, we add the following time limit constraint to constraint set (3.1) when solving the initial first-stage problem:

$$\sum_{(i,j) \in A} t_{ij}^{\xi^*} x_{ij} + \sum_{j \in V'} \tau_j^{\xi^*} y_j - T \leq \Delta T. \tag{3.14}$$

We use the minimum realization of the random travel and service time variables to construct the vector ξ^*; otherwise, we may eliminate some possible better tours when forming the initial tour at the very beginning.

The importance of adding this time limit constraint is obvious; otherwise, all the customers would be included in the initial first-stage solution. It is even more useful when the problem size (the total number of customers we choose from) is large, or given the time limit and tolerance, even in the best possible case (ξ^*), a small number of customers can be included in the initial first-stage solution.

4. The Integer L-Shaped Solution Method

The steps involved in the integer L-shaped algorithm can be described as follows:

Step 0: Set iteration index $u := 0$. Initialize the first-stage problem (the CP) with constraint set (3.1) only containing constraint (3.14) and constraint set (3.2) only containing $\eta \geq 0$.

Step 1: Set $u = u + 1$. Solve the CP. If the CP has no feasible solution, go to Step 6. Otherwise, let (x^u, y^u, η^u) be the optimal solution.

Step 2: Check for existing sub-tours, if any violation is detected, add the sub-tour elimination constraints found (Constraint (3.12) or (3.13)) to the constraint set (3.1) of the CP and go to Step 1.

Step 3: Check second-stage feasibility constraints. If violated, add the most violated one (Constraint (3.7)) to the constraint set (3.1) of the CP and go to Step 1.

Step 4: Compute the value of expected penalty for excess amount of time at the current first-stage feasible solution, denote it as $P(x^u, y^u)$.

Step 5: If $\eta^u \geq P(x^u, y^u)$, the CP satisfies the optimality criterion, go to step 6. Otherwise, introduce the optimality cuts (Constraint (3.8) for single-cut algorithm and Constraints (3.9)–(3.11) for the multi-cut algorithm) into the constraint set (3.2) of the CP, and go to Step 1.

Step 6: Output the best-known solution and stop.

5. Computational Results

To get the computational results, the algorithm is coded in VISUAL C++, and run on a 500 MHz Pentium II Processor with 128 MB RAM under Microsoft Windows 98. The MIPs are solved by calling the functions in the CPLEX optimization package into the VC++ program.

5.1. *Problem data generation*

Our computational results are the averages of the solutions obtained from ten randomly generated problem instances. For each problem instance, the customers are randomly generated from the square $20 * 20\,\text{km}^2$, according to a continuous uniform distribution, with the depot situated at the center. We assume that the effective working time T is 480 min a day; ΔT, the maximum amount of time allowed to exceed T, is assumed to be 120 min. Revenue of serving a customer j, r_j, is randomly generated from [0, 100]. The travel time between customers i and j is calculated based on the Euclidean distance d_{ij} and the speed of the vehicle. We assume that the distance matrix is symmetric and also that vehicle speed ν can take on five possibilities: 60, 50, 40, 30, and 20 km/h. Then the travel time between any two customers i and j can be approximated to be d_{ij}/v. The service time at each customer also takes on five possibilities: 10, 20, 30, 40, and 50 min. We define ξ as a vector of random variables corresponding to travel and service times. It has a finite number of realizations $\xi^1, \xi^2, \ldots, \xi^K$, with probabilities $p_1, p_2, \ldots, p_K$; with

ξ^k constructed by combining the five possibilities of travel time and five possibilities of service time.

In our computational experiment, we assume $K = 5$, and construct the data vectors for the travel time part as follows:

ξ^1: Best case; with 80% of the arcs among the customers traveling at speed 60 km/h, and the rest traveling at the other four speed possibilities.
ξ^2: Above average; with 80% of the arcs among the customers traveling at speed 50 km/h, and the rest traveling at the other four speed possibilities.
ξ^3: Average case; with 80% of the arcs among the customers traveling at speed 40 km/h, and the rest traveling at the other four speed possibilities.
ξ^4: Below average; with 80% of the arcs among the customers traveling at speed 30 km/h, and the rest traveling at the other four speed possibilities.
ξ^5: Worst case; with 80% of the arcs among the customers traveling at speed 20 km/h, and the rest traveling at the other four speed possibilities.

The service time part of the data vectors is constructed by uniformly selecting service time from the five possibilities: 10, 20, 30, 40, and 50 min.

Furthermore, we assume that the probabilities, p_1, p_2, p_3, p_4, p_5, corresponding to $\xi^1, \xi^2, \xi^3, \xi^4, \xi^5$, are 0.1, 0.2, 0.4, 0.2, 0.1, respectively.

5.2. *Computational results*

We examine the integer L-shaped algorithm from the following aspects:

- single optimality cut versus multiple optimality cuts;
- with different unit penalty cost β;
- with different ΔT — tolerance time allowed to exceed time limit T;
- with different number of states of the random vector ξ.

Results are presented in Tables 1–4 based on the following notations:

n	problem size (total number of customers)
β	unit penalty cost for total time of the route in excess of T
Profit	profit generated from serving the customers in the solution
F-cuts	number of feasibility cuts generated
O-cuts	number of optimality cuts generated
Sub-tour	number of sub-tour elimination constraints generated
S	single optimality cut algorithm
M	multi-cut algorithm

First, from Table 1, we note that as problem size increases, the problem becomes more difficult to solve, with more profits realized. This is due to the fact that, though we still have to satisfy the time limit constraint (the number of customers included in the solution may not change a lot), we now have more alternatives to choose from, i.e., the solution space becomes larger. Therefore, the problem difficulty increases,

Table 1. Performance of the algorithm with different unit penalty cost β.

n	β	Profit	F-cuts	O-cuts	Sub-tour	Time (s)
10		518.6	0.4	2.2	2.1	0.08
15		732.2	1.8	3.6	7.8	2.20
20	0.5	948.2	2.3	4.4	11.6	11.86
25		995.8	1.9	4.8	10.6	21.66
30		1081.9	2.5	4.8	26.8	372.55
35		1104.6	2.6	4.6	49.9	1083.90
10		517.2	0.4	2.2	2.0	0.25
15		704.2	1.8	3.6	8.0	0.96
20	2	890.9	2.3	4.5	19.0	25.50
25		936.5	1.9	4.7	10.5	20.84
30		1008.6	2.5	4.8	23.5	201.79
35		1021.4	2.6	4.6	64.3	1182.38

Table 2. Single optimality cut algorithm versus multi-cut algorithm.

n	Single/multiple	F-cuts	O-cuts	Sub-tour	Time (s)
10	S	0.4	1.7	2.5	0.33
	M	0.4	2.2	2.0	0.25
15	S	1.8	2.3	9.1	1.41
	M	1.8	3.6	8.0	0.96
20	S	2.3	3.4	24.2	39.62
	M	2.3	4.5	19.0	25.50
25	S	1.9	2.6	18.3	30.65
	M	1.9	4.7	10.5	20.84
30	S	2.5	3.1	28.5	235.99
	M	2.5	4.8	23.5	201.79
35	S	2.6	2.7	88.4	1688.35
	M	2.6	4.6	64.3	1182.38

as indicated by the increasing number of feasibility and optimality cuts, the number of sub-tour elimination constraints, and the computational time.

In Laporte *et al.* (1992), their computational result showed that, as the unit penalty cost increases the difficulty of the problem also increases. This is because in their problem all customers must be visited, and higher penalty means more penalty cuts needed and therefore more computational time taken to solve the problem. However, in the TCTSP with stochastic travel and service times, as unit penalty cost β increases, to maximize the profit realized, the number of customers visited in the solution may become lesser. There is no clear indication that difficulty of the problem increases as the unit penalty cost β increases. The computational time taken mainly depends on the number of constraints needed, especially the number of sub-tour elimination constraints.

Table 2 illustrates the effect of the single-cut and the multi-cut algorithms. We set unit penalty cost $\beta = 2$, and the number of states $K = 5$ in this case. Table 2 indicates that the multi-cut algorithm needs to generate more optimality cuts. However, the number of sub-tour elimination constraints needed for the multi-cut algorithm is less than that needed for the single-cut algorithm. Therefore, the multi-cut algorithm takes less computational time than the single-cut algorithm. The multi-cut algorithm is superior to the single-cut algorithm. This conforms to the findings in Birge and Louveaux (1988), though in their study, the superiority of the multi-cut algorithm over the single-cut algorithm is based on the stochastic two-stage linear problems.

In Table 3, we test the algorithm against the tolerance time ΔT. Intuitively, as ΔT increases, the time limit constraint becomes less restrictive; therefore, we need less number of feasibility cuts. On the other hand, as more customers may be included in the solution, the amount of time exceeding the time limit T may also increase, which may lead to more penalty incurred and therefore more optimality cuts generated. Columns F-cuts and O-cuts in Table 3 clearly support these facts. When the problem size is small (less than 20), as ΔT increases, the profit generated does not change much, and the computational time incurred does not increase; however, when the problem size becomes larger, with the increase of ΔT, profits generated increases, and more customers are likely to be included in the solution. Therefore, the number of sub-tour elimination constraints and the computational time also tend to increase.

The effect of the number of states of the random vector ξ on the algorithm is shown in Table 4. As the number of states of ξ increases, both the number of feasibility cuts and the number of optimality cuts tend to increase. The computational time taken depends mainly on the number of sub-tour elimination constraints added. Except for problem size 20 and 25, the computational time and the difficulty

Table 3. Performance of the algorithm with different ΔT.

n	ΔT (min)	Profits	F-cuts	O-cuts	Sub-tour	Time (s)
10	60	517.25	0.9	1.9	2.5	0.23
	120	517.25	0.4	2.2	2.0	0.25
	180	517.25	0.3	2.6	1.3	0.08
15	60	701.28	1.7	2.4	6.7	1.77
	120	704.23	1.8	3.6	8.0	0.96
	180	704.40	1.7	4.6	9.3	0.94
20	60	880.26	2.3	3.5	12.8	15.03
	120	890.92	2.3	4.5	19.0	25.50
	180	892.39	2.2	4.8	18.5	16.05
25	60	910.37	2.1	3.9	9.6	19.67
	120	936.48	1.9	4.7	10.5	20.84
	180	941.62	1.6	4.7	15.8	23.05
30	60	989.30	2.7	4.1	22.4	453.14
	120	1008.56	2.5	4.8	23.5	471.79
	180	1015.63	2.5	5.0	27.7	679.55
35	60	995.02	2.7	4.2	48.9	672.17
	120	1021.41	2.6	4.6	64.3	1182.38
	180	1033.49	2.7	4.8	73.1	5717.94

Table 4. Performance of the algorithm with different number of states of ξ.

n	States	F-cuts	O-cuts	Sub-tour	Time (s)
10	3	0.5	1.3	2.4	0.10
	5	0.4	2.2	2.0	0.25
	10	0.5	4.3	2.0	0.26
15	3	1.1	2.2	7.7	0.90
	5	1.8	3.6	8.0	0.96
	10	2.3	7.8	10.5	2.45
20	3	1.6	2.9	10.2	13.99
	5	2.3	4.5	19.0	25.50
	10	3.4	8.7	15.2	20.47
25	3	2.1	3.0	17.3	37.61
	5	1.9	4.7	10.5	20.84
	10	3.9	9.1	29.9	201.67
30	3	1.7	2.9	20.5	54.67
	5	2.5	4.8	23.5	201.79
	10	4.0	9.5	34.2	312.66
35	3	2.4	2.9	41.3	838.22
	5	2.6	4.6	64.3	1182.38
	10	4.0	8.9	94.9	23170.40

of the problem increase as the number of states increases. The computational time taken for the case of 35 customers and 10 states of the random vector ξ is extremely large, which indicates that the number of states of the random variables is the most influential factor.

6. Conclusion

In this paper, we considered the time-constrained traveling salesman problem with stochastic travel and service times, which can be encountered in a number of practical situations. We formulated it as a two-stage stochastic program with recourse, and presented an integer L-shaped algorithm for solving it. We examined the algorithm from a number of aspects. Computational results show that, for this particular problem, the difficulty mainly lies in the elimination of the sub-tours; therefore, the larger the number of sub-tours needed, the more computational time taken. The multi-cut algorithm showed its superiority to the single-cut algorithm in terms of the number of sub-tours imposed and the computational time required. As the unit penalty cost increases, the difficulty of the problem does not have a clear trend of increase due to the fact that lesser number of customers will be visited. With the increase of the number of states of random vector ξ or the increase of the tolerance time ΔT, when the problem size is large, the difficulty of the problem increases. Overall, the most influential factor of the algorithm is the number of states of the random vector.

As in reality, we usually face problems with large problem size and random vectors with more states, the exact algorithm is only suitable to be considered as a performance measurer. For larger problem, heuristic algorithms such as those presented in Tan *et al.* (2001) and Teng *et al.* (2003) can be used to provide an approximate solution to the problem.

References

Aarts, E and JK Lenstra (1997). *Local Search in Combinatorial Optimization*. New York: Wiley.

Benders, JF (1962). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4, 238–252.

Birge, JR and FV Louveaux (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34, 384–392.

Bland, RG and DF Shallcross (1989). Large traveling salesman problems arising from experiments in X-ray crystallography: a preliminary report on computation. *Operations Research Letters*, 8, 125–128.

Cloonan, J (1966). A heuristic approach to some sales territory problems. In *Proceedings of the Fourth International Conference on Operations Research*, JDC Little (ed.), pp. 81–84. Cambridge, Massachusetts: MIT Press.

Gendreau, M, G Laporte and F Semet (1998a). A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32, 263–273.

Gendreau, M, G Laporte and F Semet (1998b). A tabu search heuristic for the undirected selective traveling salesman problem. *European Journal of Operational Research*, 106, 539–545.

Gendreau, M, G Laporte and R Seguin (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29, 143–155.

Gensch, DH (1978). An industrial application of the traveling salesman's sub-tour problem. *AIIE Transactions*, 10, 362–370.

Gilmore, PC and RE Gomory (1964). Sequencing a one-state variable machine: A solvable case of the traveling salesman problem. *Operations Research*, 9, 849–859.

Golden, BL, L Levy and R Dahl (1981). Two generalizations of the traveling salesman problem. *Omega* 9, 439–445.

Hjorring, C and J Holt (1999). New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research*, 86, 569–584.

Jünger, M, G Reinelt and G Rinaldi (1994). The Traveling Salesman Problem. In *Handbook on Operations Research and the Management Sciences*, Ball, Magnanti, Monma, and Nemhauser (eds.), pp. 225–330, Amsterdam: North-Holland.

Korte, BH (1989). Applications of combinatorial optimization. In *Mathematical Programming: Recent Developments and Applications*, In M Iri and K Tanabe (eds.), pp. 1–55. Dordrecht: Kluwer.

Korte, BH and J Vygen (2000). *Combinatorial Optimization: Theory and Algorithms*. Berlin: Springer.

Laporte, G and FV Louveaux (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13, 133–142.

Laporte, G, FV Louveaux and H Mercure (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, 26, 161–169.

Laporte, G, FV Louveaux and Luc Van Hamme (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50, 415–423.

Laporte, G and S Martello (1990). The selective traveling salesman problem. *Discrete Applied Mathematics*, 26, 193–207.

Lawler, EL, JK Lenstra, AHG Kan Rinnooy and DB Shmoys (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Chichester, UK: John Wiley.

Ratliff, HD and AS Rosenthal (1981). Order-picking in a rectangular warehouse: a solvable case for the traveling salesman problem. PDRC Report Series No. 81-10, Georgia Institute of Technology, Atlanta, Georgia.

Reinelt, G (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*. Berlin: Springer.

Tan, KC, LH Lee and K Ou (2001). Hybrid genetic algorithms in solving vehicle routing problems with time window constraints. *Asia-Pacific Journal of Operational Research*, 18, 121–130.

Teng, SY, HL Ong and HC Huang (2003). A comparative study of metaheuristics for vehicle routing problems with stochastic demands. *Asia-Pacific Journal of Operational Research*, 20, 1–17.

Van Slyke, RM and R Wet (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17 638–663.

Ms S. Y. Teng is a PhD student in the Department of Industrial and Systems Engineering, National University of Singapore. She obtained her Master's Degree in Tianjin University, P. R. China. Her research interest includes heuristic algorithms for solving vehicle routing and scheduling problems.

Dr. H. L. Ong is an Associate Professor in the Department of Industrial and Systems Engineering, National University of Singapore. He obtained his PhD in Management Sciences from the University of Waterloo. His research interests are software development and design and analysis of algorithms for solving some scheduling problems. He is the chief editor of *Asia-Pacific Journal of Operational Research* and has published many papers in several OR journals.

Dr. H. C. Huang received her PhD in Operations Research from Yale University. She is an Associate Professor in the Department of Industrial and Systems Engineering at the National University of Singapore. Her research interests include combinatorial optimization and logistics.