SOLVING THE TRAVELLING SALESMAN PROBLEM WITH TIMEWINDOWS BY LAGRANGEAN RELAXATION

R. Herrero[a], D. Guimarans[b] and J. J. Ramos[c]

Department of Telecommunication and System Engineering
Universitat Autònoma de Barcelona
08193, Bellaterra, Spain

[a] Rosa.Herrero@uab.cat, [b] Daniel.Guimarans@uab.cat, [c] JuanJose.Ramos@uab.cat

ABSTRACT
This article presents a Lagrangean Relaxation-based methodology to solve the Travelling Salesman Problem with Time Windows. The Lagrangean function exploits the structure of the problem. The proposed method, which elaborates on the Subgradient Optimization, presents a simple heuristics aimed to improve algorithm's convergence on the optimal solution. Furthermore, if the optimal solution is not found in a reasonable number of iterations, this method is able to provide a feasible solution while guaranteeing a tight gap between the primal and the optimal cost.

Keywords: Travelling Salesman Problem, Time Windows, Lagrangean Relaxation, Subgradient Optimization

1. INTRODUCTION

The Travelling Salesman Problem (TSP) is the following: "A salesman is required to visit once and only once each of n different cities starting from a base city, and returning to this city. What is the path that minimizes the total distance travelled by the salesman?" The TSP is probably the most famous and extensively studied problem in the field of Combinatorial Optimization (Lawer et al. 1985).

This work considers the Travelling Salesman Problem with Time Windows (TSP-TW) which is a generalization of the TSP. In the TSP-TW, the service at each city must start within an associated time window (Desrosiers et al. 1988).

The TSP belongs to the class of $\mathcal{NP}$-Hard optimization problems. Therefore, the TSP-TW with general time windows is at least $\mathcal{NP}$-hard too. Indeed, it is strongly $\mathcal{NP}$-complete to find a feasible solution for the TSP-TW (Savelsbergh 1985).

Lagrangean relaxation (LR) is a well-known method to solve large-scale combinatorial optimization problems. It works by moving hard to satisfy constraints into the objective function associating a penalty on the objective if they are not satisfied. For a recent review of LR techniques and applications, see Guignard 2003.

In the proposed approach to the TSP-TW, LR relaxes the constraint set requiring that all customers must be served and the constraint set requiring that the arrival time must be greater than the earliest time.

The Lagrangean function exploits the structure of the problem, so it reduces considerably the formulation's size. Thus, the Lagrangean Problem needs less computational effort. However, it is often a major issue to find the optimal Lagrangean multipliers. The commonly used approach is the subgradient method. It guarantees convergence but it is too slow to be real practical interest.

The proposed method improves the convergence on the optimal solution in the Subgradient Optimization by using an Insertion heuristic to obtain a feasible solution from a LR solution. If the optimal solution is not found in a reasonable number of iterations, it is able to provide a feasible solution obtaining a tight gap between the primal and the optimal cost.

The present paper is structured as follows. Section 2 introduces some notation and terminology and presents a formulation of the problem. Section 3 describes the proposed Lagrangean relaxation. Section 4 describes the proposed method and heuristics used to improve its efficiency. Section 5 presents some results obtained by using the method proposed in the previous section. Finally, application and further research directions are outlined.

2. MATHEMATICAL FORMULATION

The TSP can be considered as a routing network, represented by a directed graph $G = (I, E)$, connecting city nodes $I = \{i_1, i_2, \dots, i_n\}$ through a set of directed edges $E = \{(i, j) : i, j \in I\}$, where $i_1 = 1$ is the base city. The tour is a path, $P = (1, i_2, \dots, i_n, 1)$

connected by edges belonging to E that starts and ends at a base city and visits all the cities.

An edge $e=(i,j)\in E$ is associated with a matrix $C=\{c_e\}$ denoting the travel time from node i to node j. It is assumed that distances defined in the problem satisfy the triangular inequality, that means the value c_e is supposed to be the lowest time route connecting node i to node j.

In the TSP-TW problem, each city $i\in I$ has an associated service time st_i and an associated time window $[a_i,b_i]$, defined by the earliest and the latest time to start the service in this city. The base city may also have a time window defining the scheduling horizon, but not a service time $(st_1=0)$. Moreover there is a matrix (t_e) denoting the travelling time from node i to node j plus the service time st_i at node i, $t_e = st_i + c_e$.

Then a feasible solution to the TSP-TW should satisfy the following constraints:

- Each city should be visited only once.
- The route must start and finish at the base city.
- Each city is visited within its time window.

The problem goal is to minimize the total time of the route.

The proposed mathematical formulation requires defining the binary variable x_e to denote that the edge $e=(i,j)\in E$ is used in the tour

$$x_e = \begin{cases} 1 & \text{if city j is visited immediately after i;} \\ 0 & \text{otherwise.} \end{cases}$$

It requires to define also another variable s_i to denote the arriving time at node i. These variables are nonnegative integers.

The proposed mathematical formulation for the TSP-TW problem is as follows:

$$\min \sum_{e\in E} t_e x_e \tag{1}$$

subject to

$$\sum_{e\in\delta^+(i)} x_e = 1, \quad \forall i\in I \tag{2}$$

$$\sum_{e\in\delta^-(i)} x_e = 1, \quad \forall i\in I \tag{3}$$

$$s_i + t_e - M(1-x_e) \le s_j, \quad \forall e=(i,j)\in E : j\ne 1 \tag{4}$$

$$s_i + t_e - M(1-x_e) \le b_1, \quad \forall e=(i,1)\in E \tag{5}$$

$$s_i \ge a_i, \quad \forall i\in I \tag{6}$$

$$s_i \le b_i, \quad \forall i\in I \tag{7}$$

where

- $\delta^+(i)=\{e\in E : \exists j\in I, e=(i,j)\}$ represents the set of arcs whose starting node is i.
- $\delta^-(i)=\{e\in E : \exists j\in I, e=(j,i)\}$ represents the set of arcs whose ending node is i.
- M is a large real value, satisfying $M\ge b_i + t_{ij} - a_j \quad \forall (i,j)\in E$.

Constraint (2) guarantees that every city must have one starting edge, whereas constraint (3) guarantees that every city must have one finishing edge.

If node i is visited immediately before node j, then constraint (4) states that the arriving time at node j, s_j, should be greater than s_i plus t_e, the sum of the travelling time and the service time at node i, except for the base city.

Constraint (5) states that the total time should be lower than the maximum allowed travelling time b_1.

Constraints (6)-(7) impose that each city should be visited within the defined time window.

Note that constraint (4) replaces the subtour elimination constraints for the TSP. It was proposed by (Miller et al. 1960).

3. LAGRANGEAN RELAXATION

A feasible solution of a TSP is a 1-tree having two edges incident to each node, one starting edge and one ending edge. A 1-tree is a tree on the graph induced by $\{2,\ldots,n\}$ nodes plus two edges incident to node 1, as explained on (Held and Karp 1970).

Note that summing constraints (2) and (3), the result is $\sum_{e\in E} x_e = n$. The 1-tree is determined by adding to the latter expression the subtour elimination constraints. Finding a minimum weight 1-tree is relatively easy, so this is potentially an interesting relaxation.

However in the TSP-TW, the 1-tree should satisfy time windows constraints. Constraint (6) can increase the effort to find a minimum weight 1-tree.

The proposed Lagrangean function is as follows:

$$\begin{aligned} L(u,v,w) = \min_{x1-tree} \sum_{e\in E} t_e x_e \quad &+ \sum_{i\in I} u_i \left(1 - \sum_{e\in\delta^+(i)} x_e\right) \\ &+ \sum_{i\in I} v_i \left(1 - \sum_{e\in\delta^-(i)} x_e\right) \\ &+ \sum_{i\in I} w_i (a_i - s_i) \end{aligned} \tag{8}$$

$$\text{s.t.} \quad s_i + t_e - M(1-x_e) \le s_j, \quad \forall e=(i,j)\in E : j\ne 1 \tag{9}$$

$$a_i + t_e - M(1-x_e) \le s_j, \quad \forall e=(i,j)\in E : j\ne 1 \tag{10}$$

$$s_i + t_e - M(1-x_e) \le b_1, \quad \forall e=(i,1)\in E \tag{11}$$

$$a_i + t_e - M(1-x_e) \le b_1, \quad \forall e=(i,1)\in E \tag{12}$$

$$s_i \le b_i, \quad \forall i\in I \tag{13}$$

This LR relaxes constraints (2)-(3) weighting them with multiplier vectors u and v of appropriate dimensions and unrestricted sign. It also relaxes the constraint (6) weighting it with a multiplier vector w of appropriate dimensions and nonnegative components.

Note that, in this Lagrangean Dual Problem, waiting times are allowed. That means one may arrive at a city i earlier than a_i and wait until the node is released at time a_i. Thus, waiting time has influence on the dual cost of a solution. For this reason, the proposed primal formulation considers to minimize the total time of the tour.

As mentioned, finding a feasible solution for the TSP-TW is $\mathcal{NP}$-complete. However, this is a slightly relaxed problem and feasible solutions may be obtained more easily.

3.1. Strengthened Constraints

Since constraints (9), (11) and (13) are not very strong and they can be lifted in several ways, constraints (10) and (12) were added to avoid error propagation. Note that, constraints (9), (11) and (13) are the same as constraints (4), (5) and (7) respectively.

Also, to avoid the optimization to consider s_i bigger than the arrival time associated to x_e, Lagrangean function (8) is replaced by:

$$L(u,v,w) = \min_{x1-tree} \sum_{e \in E} t_e x_e + \sum_{i \in I} u_i \left(1 - \sum_{e \in \delta^+(i)} x_e\right) + \sum_{i \in I} v_i \left(1 - \sum_{e \in \delta^-(i)} x_e\right) + \sum_{i \in I} w_i \Delta_i \quad (14)$$

And constraints (15) and (16) are added:

$$\Delta_i \geq a_i - s_i \quad \forall i \in I \quad (15)$$

$$\Delta_i \geq 0 \quad \forall i \in I \quad (16)$$

3.2. Subgradient Optimization

Algorithm 1 shows the standard subgradient algorithm, as explained on (Wolsey 1998).

If $\lambda_k = \lambda_0 \rho^k$ for any parameter $\rho < 1$, then convergence is guaranteed if λ_0 and ρ are sufficiently large. Otherwise, geometric series $\lambda_0 \rho^k$ tend to zero too rapidly and sequences $\{u^k\}$, $\{v^k\}$ and $\{w^k\}$ converge before reaching an optimal point. In practice, rather than decreasing λ_k at each iteration, the parameter λ_k is reduced by ρ every n consecutive iterations without improving the lower bound.

Algorithm 1: Subgradient Optimization

Iteration 0
Initialize $u^0 \leftarrow v^0 \leftarrow w^0 \leftarrow 0$;
Iteration k
Solve the Lagrangean function $L(u^k, v^k, w^k)$ Evaluate the subgradient $\gamma(u^k, v^k, w^k)$ Calculate the step size λ_k $(u^{k+1}, v^{k+1}, w^{k+1}) = (u^k, v^k, w^k) + \lambda_k \gamma(u^k, v^k, w^k)$ $k \leftarrow k+1$

4. THE PROPOSED METHOD

Algorithm 2 shows the proposed method.

First, the method tries to obtain a feasible solution applying the Nearest-Feasible-Neighbour Heuristic. As mentioned above, it is important to notice that it is an $\mathcal{NP}$-complete problem to find a feasible solution to the TSP-TW. Therefore, if the heuristic only finds a subpath then it applies the Insertion Heuristic. Sorting visits dominates the time complexity of these heuristics.

Algorithm 2: The Proposed Method

Step 0
Initialize $u^0 \leftarrow v^0 \leftarrow w^0 \leftarrow 0$; $\delta_0 \leftarrow 2$ Apply the NFN Heuristic and the Insertion Heuristic.
Step 1
Solve the Lagrangean function $L(u^k, v^k, w^k)$ Evaluate the subgradient $\gamma(u^k, v^k)$
Step 2
if ($\left\|\gamma(u^k, v^k)\right\|^2 = 0$) then It finds a tour perhaps it has a waiting time.
Step 3
if ($\left\|\gamma(u^k, v^k)\right\|^2 < \zeta$) then Consider the subpath $P := (1, i_1, i_2, \ldots, i_k, 1)$ from the dual solution and apply Insertion Heuristic
Step 4
Calculate the step size λ_k $(u^{k+1}, v^{k+1}) = (u^k, v^k) + \lambda_k \gamma(u^k, v^k)$ $w_i^{k+1} = w_i^k + \lambda_k \max\{0, a_i - s_i\}$ $k \leftarrow k+1$ Return to step 1.

In step 2, the method finishes if a tour is found. The dual solution may contain a waiting time, but the dual cost is the optimal cost. In this case, the solution is an equivalent of the optimal solution with some swapped nodes but identical cost. The waiting time can be eliminated afterwards by applying a Swap Heuristic or Two-Node-Exchange Heuristic, as explained on (Ascheuer et al. 2001).

In step 3, if the solution is nearly a tour, the method applies the Insertion Heuristic in the associated subpath. The parameter ζ depends on the number of variables. In the beginning, it may be large equals to $N/2$. However, it should be tight enough to not increase the running time excessively. Moreover, the method does not allow applying step 3 again in next consecutive iterations to not increase the running time significantly.

As mentioned above, the proposed method uses the step size λ_k, reducing by ρ every n consecutive iterations without improving the lower bound.

It should be remarked that the subgradient depends on constraint sets requiring that all customers must be served.

4.1. Nearest-Feasible-Neighbour Heuristic

Starting with each feasible edge $(1,i) \in E$, Nearest-Feasible-Neighbour Heuristic (NFN) enlarges the current subpath $(1,i_1,i_2,\ldots,i_k)$ by an arc (i_k,i_l) resulting in the smallest increase in the objective value while guaranteeing feasibility, but the final path may not include all $j \in I$.

4.2. Insertion Heuristic

Starting with a shortest path in E obtained by the Nearest-Feasible-Neighbour Heuristic, Insertion Heuristic enlarges the current partial path $P := (1,i_1,i_2,\ldots,i_k,1)$ by a node j satisfying a certain insertion criterion. Let $S := I - \{i_1,\ldots,i_k\}$ be the set of unvisited cities and

$$d_{\min}(j) := \min\{c_{i_l j} + c_{j i_{l+1}} - c_{i_l i_{l+1}} \mid i_l \in P \text{ and } (1,i_1,\ldots,i_l,j,i_{l+1},\ldots,i_k,1) \text{ is feasible}\}$$

Among all unsequenced cities, it chooses the node $j \in S$ that causes the lowest increase in the path length.

If the current partial path obtained from the dual solution has waiting time, the Insertion Heuristic enlarges the current partial path $P := (1,i_1,i_2,\ldots,i_k,1)$ in the first i_l which has $s_i > a_i$ by a node j satisfying a certain insertion criterion. Let $S := I - \{i_1,\ldots,i_k\}$ be the set of unvisited cities and

$$d_{\min}(j) := \min\{c_{i_l j} + c_{j i_{l+1}} - c_{i_l i_{l+1}} \mid i_l \in P \text{ is the first wainting node, } a_j \le s_j \le b_j \text{ and } (1,i_1,\ldots,i_{l-1},j,i_l,\ldots,i_k,1),\ s_m \le b_m \forall m \ge l\}$$

Among all unsequenced cities, it chooses the node $j \in S$ that causes the lowest increase in the path length. Of course, this reduces the waiting time in node i.

5. RESULTS

The method has been tested using the problems proposed by (Potvin and Bengio 1996) as individual route instances on Solomon's RC2 VRPTW (Vehicle Routing Problem with Time Windows). Solomon's RC2 instances contain a combination of randomly-spaced and clustered customers. They have large time windows and large vehicle capacity but have very few routes and significantly more customers per route.

The Lagrangean function (14) has been programmed using the software ECLiPSe and implemented using *eplex* library included in this package.

5.1. Heuristics Results

The heuristics were applied in the rc_201.1 individual route instance on Solomon's RC2 VRPTW instances.

In Fig. 1 the resulting path from using the Nearest-Feasible-Neighbour Heuristic (NFN) is shown. The path that was found is feasible but incomplete with $S = \{3, 16\}$.

In Fig 2. the resulting path using the Insertion Heuristic starting from the previous path is shown, but in this case the path found is complete but unfeasible.

In Fig. 3 the resulting path from using the Insertion Heuristic from the dual solution of $k = 13$ iteration is shown.

The results are shown on the table 1. Insertion Heuristic has completed the path from NFN Heuristic and obtained the optimal known travel time 592, but the path found has a waiting time. Also, the path found from the dual solution has obtained the optimal known travel time but with less travel cost.

Table 1: Heuristics' Results

Heuristic	Travel cost	Travel Time	Waiting Time
NFN	389	569	-
NFN + Insertion	366	592	36
Insertion	334	592	68

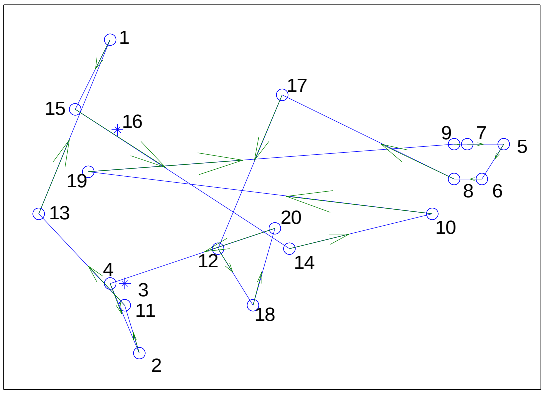

Figure 1: Nearest - Feasible - Neighbour Heuristic

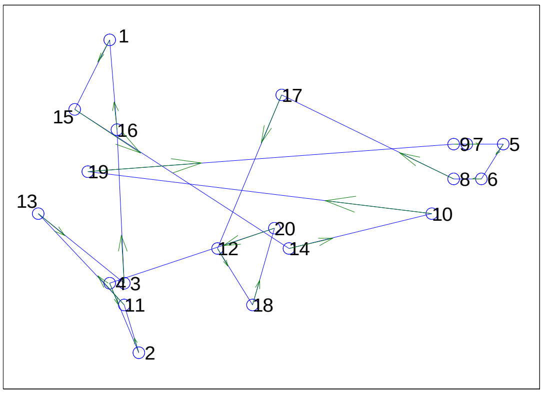

Figure 2: NFN Heuristic + Insertion Heuristic

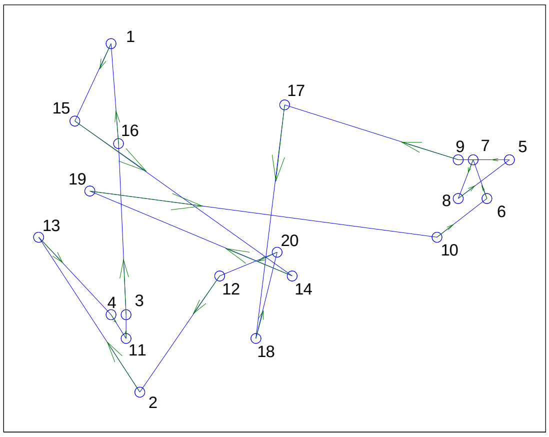

Figure 3: Insertion Heuristic

5.2. Method Results

The proposed method was applied in the individual route instances on rc_201 Solomon's RC2 VRPTW instance.

The results are shown on the table 2. Remember that the method minimizes the travel time, in that way the solutions found has obtained tight values in spite of the waiting time. But the travel cost has obtained a 27% deviation from the best known solution. The best known solution of the rc_201 Solomon's RC2 VRPTW is 1406.94 dist. with 4 vehicles, that was solved by Cordeau et. al. 2000.

Table 2: Method Results

Instances	Travel cost	Travel Time	Waiting Time
rc_201.1	366	592	36
rc_201.2	500	860	99
rc_201.3	521	862	31
rc_201.4	557	889	82
Total	1944	3203	248

Aiming to increase the proposed method's efficiency, the parameter $\lambda_k = \delta_k \dfrac{UB - L(u^k, v^k, w^k)}{\left\|\gamma(u^k, v^k, w^k)\right\|^2}$ with $0 < \delta_k \leq 2$ was introduced and UB denotes a known upper bound on the optimal value of the original problem. This step size guarantees convergence and leads to much faster convergence. However, depending on the problem instance, the method needs a too tight upper bound to ensure algorithm's convergence.

This step size was applied in the rc_201.1 individual route instance on Solomon's RC2 VRPTW instances. In Fig 4. the result using the UB equals to 592 is shown, in this case the sequences do not converge.

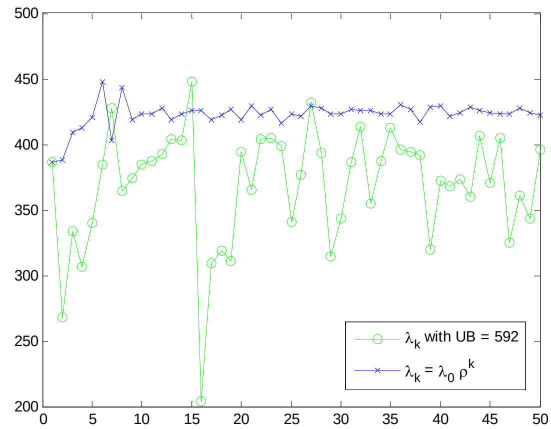

Figure 4: This step size do not converge

6. CONCLUSIONS AND FUTURE RESEARCH

The present paper presents an aproximation to sole the Travelling Salesmen Problem with Time Windows using Lagrangean Relaxation. Moreover, the presented methodology, proposes an insertion heuristic to improve LR results.

The proposed method was found to be able to find feasible solutions of the problem allowing waiting time. However, the algorithm may not guarantee finding the optimal solution.

Since the future objective is to solve the VRPTW. Firstly, dividing it in instance of TSPTW clustering nodes in instances satisfy the constraint of vehicle capacity. Secondly, the proposed method will solve the instances. The solutions can be allow to have waiting times, even thought, it may be not the optimal solution. Finally, the instances will be rescheduled according to the waiting time and time windows.

Near further research is to be addressed in the directions to improve convergence. Also introduce an algorithm to solve the Lagrangean function aiming to reduce the computational time.

REFERENCES

Ascheuer, N., Fischetti, M., Grötschel, M., 2001. Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut. *Math. Program.*, Ser. A 90: 475-506.

Cordeau, J. F., Laporte, G., Mercier, A., 2000. A unified tabu search heuristic for Vehicle routing problems with time Windows. Technical Report CRT-00-03, Centre for Research on Transportation, Montréal, Canada.

Desrosiers, J., Sauvé, M., Soumis, F., 1988. Lagrangean relaxation methods for solving the minimum fleet size multiple travelling salesman problem with time windows. *Management Science* 34: 1005-1022.

Guignard, M., 2003. Lagrangean Relaxation. *Top* 11(2): 151-228.

Held, M., Karp, R. M., 1970. The traveling salesman problem and minimum spanning trees. *Operations Research* 18: 1138-1162.

Lawer, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., Shmoys, D.B., 1985. *The Traveling Salesman problem: a guided tour of combinatorial optimization*. New York: John Wiley & Sons.

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation and travelling salesman problems. *J. Assoc. Comput. Mach.* 7: 326-329.

Potvin, J. Y., Bengio, S., 1996. The Vehicle Routing Problem with Time Windows Part II: Genetic Search. *INFORMS Journal on Computing* 8: 165-172.

Savelsbergh, M.W.P., 1985. Local search for routing problems with time windows. *Annals of Operations Research* 4: 285-305.

Wolsey, L. A., 1998. *Integer programming*. New York : John Wiley & Sons.