Heuristic and preprocessing techniques for the robust traveling salesman problem with interval data

R. Montemanni*†, J. Barta, L.M. Gambardella

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)

Galleria 2, CH-6928 Lugano-Manno, Switzerland

Technical report IDSIA-01-06

Abstract

We study a version of the traveling salesman problem where travel times are specified as a range of possible values. This model reflects the difficulties to estimate travel times exactly in reality. Robustness concepts are used to drive optimization.

We propose some efficient heuristic and preprocessing techniques. Computational experiments are presented.

Keywords: Traveling salesman problem, Robust optimization, Interval data, Heuristic methods.

*Partially supported by the Swiss National Science Foundation through project 200020-109854/1.

†Corresponding author. Tel: +41 91 610 8667. Fax: +41 91 610 8661. Email: roberto@idsia.ch.

1 Introduction

Given the cost of travel between each pair of a finite number of cities, the *traveling salesman problem* (TSP) is to find the cheapest tour passing through all of the cities and returning to the point of departure. Many variants and generalizations of the problems have been deeply studied along the years, starting from the pioneer work presented by Dantzig et al. [1] in 1954. Here we consider a variation - motivated by the intrinsic difficulty in estimating travel times exactly in the reality - of the most classic, and most studied, version of the problem, namely the symmetric traveling salesman problem, where the cost of travel between two cities is independent from the traveling direction. Surveys on the results achieved over the years for the TSP can be found in Lawler et al. [7] and Reinelt [11]. A vast collection of resources and information concerning the traveling salesman problem can be found at the Traveling Salesman Problem web-page (http://www.tsp.gatech.edu) maintained by William Cook.

Estimating travel times (i.e. edge costs) exactly is typically a difficult task, since they depend on many factors that are difficult to predict, such as traffic conditions, accidents, traffic jams or weather conditions. For this reason, the simple model previously introduced may be inadequate. To overcome this problem, more complex models, that take into account uncertainty, have to be considered. Montemanni et al. [8] recently proposed an approach where an interval of possible travel times is associated with each edge. They adopted the robust deviation criterion (see Kouvelis and Yu [6]) to drive optimization.

In this paper we present efficient tools for:

- estimating the robustness cost of a given tour;
- producing heuristic solutions to the problem;
- preprocessing the problem in order to identify edges that will never be on an optimal robust tour.

It is important to point out that the methodological approaches we discuss in this paper, can be adapted for the robust counterpart of many other NP-hard

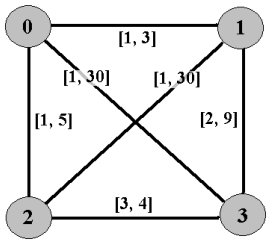

Figure 1: Example of undirected graph with interval costs.

combinatorial optimization problems.

2 Problem description

The robust deviation symmetric traveling salesman problem with interval data is defined on an undirected graph $G = \{V, E\}$, where V is a set of vertices, with vertex 0 associated with the depot, and vertices $1, \ldots, |V|$ representing the cities to be visited, and E is the set of edges of the graph. All the problems considered in this paper are based on complete graphs: $\forall i, j \in V \; \exists \{i, j\} \in E$. In the paper we will refer to an edge either as e or as $\{i, j\}$, depending on the the context, choosing the second notation when the two nodes of the edge have a role within the description. An interval $[l_{ij}, u_{ij}]$, with $0 \leq l_{ij} \leq u_{ij}$, is associated with each edge $\{i, j\} \in E$. An example of undirected graph with interval costs is given in Figure 1.

The objective of the optimization is to find a Hamiltonian circuit (indicated as *tour* in the remainder of the paper) with the minimum cost, according to the cost function associated with the chosen notion of robustness. We consider the *robust deviation criterion* (Kouvelis and Yu [6]).

In order to formally describe the *robust deviation traveling salesman problem*, we need the following definitions.

Definition 1. *A* scenario R *is a realization of the edge costs, i.e. a cost* $c_{ij}^R \in [l_{ij}, u_{ij}]$ *is chosen for each edge of the graph.*

Definition 2. *The* robust deviation *of a tour t in scenario R is the difference between the cost of t in scenario R and the cost of a shortest tour in R.*

Definition 3. *A tour t is said to be a* robust tour *if it has the smallest (among all possible tours) maximum (among all possible scenarios) robust deviation.*

A scenario can be seen as a snapshot representing the real network situation, and a robust deviation tour can be seen as a path that should guarantee reasonably good performance (compared to optimal solutions) under any possible arc costs configuration.

The following theorem establishes a criterion that makes the robust traveling salesman problem much more tractable from a combinatorial point of view.

Theorem 1 (Montemanni et al [8])**.** *Given a tour t, the scenario R that maximizes the robust deviation for t is the one where all the edges of tour t have the highest possible cost, and the costs of the remaining edges are at their lowest possible value, i.e. $c_{ij}^R = u_{ij}\ \forall\{i,j\} \in t$ and $c_{ij}^R = l_{ij}\ \forall\{i,j\} \notin t$.*

Theorem 1 defines the concept of scenario induced by a tour t, that will be referred to as $Ind(t)$ in the remainder of the paper, and states that this scenario $Ind(t)$ produces the maximal robust deviation of tour t. The very important implication is that, given a tour t, we can compute its robustness cost by solving a classic traveling salesman problem on its induced scenario.

Figure 2 depicts the scenario induced by the tour $t = \{\{0,1\},\{1,3\},\{3,2\},\{2,0\}\}$ on the graph of Figure 1. Since the tour with minimum cost on this scenario is $t' = \{\{0,1\},\{1,2\},\{2,3\},\{3,0\}\}$, the robustness cost of t is $\underbrace{(3+9+4+5)}_{\text{cost of } t} - \underbrace{(3+1+4+1)}_{\text{cost of } t'} = 21-9 = 12$. It is also easy to see that tour t is the robust tour of the example of Figure 1.

3 Estimates for the robustness cost

According to Theorem 1, once we have a tour t, its robustness cost can be evaluated by solving a classic traveling salesman problem in the scenario induced

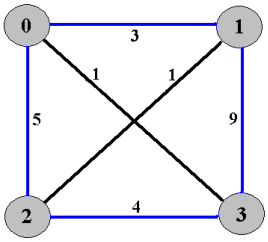

Figure 2: Scenario induced by the tour $t = \{\{0,1\},\{1,3\},\{3,2\},\{2,0\}\}$ on the graph with interval costs of Figure 1.

by tour t. In case of large problems, it can however be difficult to solve such a traveling salesman problem to optimality (we remind that it is an NP-hard problem itself). We can partially overcome this problem by providing very efficient lower and upper bounds for the robustness cost. The tools we need to provide these bounds are a lower bound procedure and an heuristic procedure for the classic traveling salesman problem, that will be run on $Ind(t)$, the scenario induced by tour t. Since heuristic and lower bound procedures for the traveling salesman problem are very efficient in practice, we can expect to have very tight estimates also for the value of the robustness cost of a given tour. In our implementation we used the procedures described in Helsgaun [2].

Let $UppBnd(R)$ and $LowBnd(R)$ be an upper and a lower bound for the (classic) cost of the shortest tour in scenario R, respectively. Let $ClCost(U,t)$ be the (classic) cost of tour t in scenario U, where all the costs are at the highest possible value (i.e. $c^U_{ij} = u_{ij} \quad \forall\{i,j\} \in E$). We can formulate the following result:

Proposition 1. *Given a tour t, its robustness cost $RobCost(t)$ satisfies the following inequality:*

$$Clcost(U,t) - UppBnd(Ind(t)) \leq RobCost(t) \leq$$

$$\leq ClCost(U,t) - LowBnd(Ind(t)) \tag{1}$$

Proof. Follows directly from Theorem 1, that defines how the robustness cost of a tour can be calculated. □

Given a tour t, evaluating its (classic) cost in scenario U has a complexity in the order of $O(|V|)$, since each edge on the tour has to be processed. Building scenario $Ind(t)$ has a complexity of $O(|V|^2)$, since each edge of the graph has to be processed. The approximate running time for estimating for the (classic) cost of the optimal tour in scenario $Ind(t)$ is, in our implementation, $O(|V|^{2.2})$ (see Helsgaun [2]). This last term dominates the others.

It is worth to observe that fully polynomial time upper and lower bounds for the cost of a classic traveling salesman problem - they exist - would have guaranteed a fully polynomial execution time for the routines described in this section. We preferred to use the methods described in Helsgaun [2] since they are known to be the most effective in practice, notwithstanding they do not run in polynomial time in the worst case. Similar reasonings apply for the methods we will describe in the following sections.

4 Heuristic algorithms

Exact algorithms for the robust traveling salesman problem have been described in Montemanni et al. [8]. It is however reasonable to expect these exact algorithms to efficiently handle problems up to a certain size only. For this reason heuristic procedures are of interest. The methods we present here heavily rely on well-known heuristic algorithms for the classic traveling salesman problem.

The method we propose is based on the construction of an ad-hoc scenario and on the solution of the related classic traveling salesman problem with an heuristic approach. The resulting tour will also be a feasible solution for the robust problem. We consider in particular two scenarios, that according to Kasperski and Zieliński [4] and [5] are the most promising ones. They are scenario U (all the costs at the highest possible value) and scenario M, defined as follows: $c_{ij}^M = \frac{(u_{ij}+l_{ij})}{2} \quad \forall\{i,j\} \in E$. According to Kasperski and Zieliński

[4], the optimal solution of the classic TSP on scenario M, if available, would guarantee a 2-approximation for the optimal solution of the robust TSP itself, while scenario U has been proven to perform well (often better than M) for the robust counterpart of many combinatorial optimization methods (see Kasperski and Zieliński [5], Montemanni and Gambardella [9] and Montemanni et al. [10]). It is important to observe that we cannot guarantee an approximation ratio for our algorithm, unless we solve to optimality the classic problem on scenario M, since it is well-known that no heuristic algorithm with a fixed approximation ratio exists for the generic classic TSP, unless P = NP. In our implementation we will use the heuristic routines for the classic traveling salesman problem described in Helsgaun [2].

In the reminder of this paper we will refer to the heuristic algorithm based on scenario U as HU, and to that based on scenario M as HM.

The construction of scenarios M and U has a complexity of $O(|V|^2)$, since each edge of the graph has to be processed. Producing an upper bound for the (classic) cost of the optimal tour in a given scenario has, in our implementation, an approximate running time of $O(|V|^{2.2})$ (see Helsgaun [2]). This last term dominates the other.

5 Preprocessing

We present a preprocessing routine able to identify edges that will never be on a robust tour. This operation is expected to speed up the execution time of exact algorithms and of heuristic approaches more complex than the ones we propose in this paper (e.g. metaheuristic algorithms).

We adopt the following notation: $RobCost(t)$ is the robustness cost of tour t, $UppBnd(U)$ is an upper bound for the cost of a shortest tour in scenario U and $LowBnd(U, e)$ is a lower bound for the cost in scenario U of every tour that contains edge e.

We are now ready to give the following preliminary result:

Lemma 1. *The following inequality holds:*

$$LowBnd(U,e) - UppBnd(U) \leq RobCost(t) \quad \forall\ t \in \mathcal{TSP} \mid e \in t \tag{2}$$

Proof. $LowBnd(U,e)$ is, by definition, a lower bound for the cost of each tour containing edge e. To obtain (2), it is enough to observe that $UppBnd(U)$ is an upper bound for the regret term associated with each possible tour. □

It is important to observe that very efficient tools are available for the calculation of $LowBnd(U,e)$ and $UppBnd(U)$. In particular, in our implementation those presented in Helsgaun [2] are used.

Lemma 1 provides a lower bound for the robustness cost of each tour containing a given edge e. However, the lower bound can be sometimes further refined, according to the following result.

The idea is to improve the accuracy of the estimate for the regret term in the left hand side of inequality (2). We are therefore looking for an appropriate numeric term $\gamma(e)$, based on some intrinsic properties of the problem and of the estimate provided in (2), that can be added to the estimate itself.

We now give the following definition:

$$\gamma(e) := \begin{cases} u_{e_1} - l_{e_1} + u_{e_2} - l_{e_2} & \text{if } LowBnd(U,e) - UppBnd(U) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

with $e_1, e_2 \in E$ are the two edges of graph G with the smallest cost intervals, i.e. $(u_{e_1} - l_{e_1}) \leq (u_e - l_e)\ \forall e \in E$ and $(u_{e_2} - l_{e_2}) \leq (u_e - l_e)\ \forall e \in E \setminus \{e_1\}$.

We are ready to give the following result, in which $St(S)$ indicates the (classic) shortest tour in scenario S:

Theorem 2. *The following inequality holds:*

$$LowBnd(U,e) - UppBnd(U) + \gamma(e) \leq RobCost(t) \quad \forall\ t \in \mathcal{TSP} \mid e \in t \tag{4}$$

Proof. If $LowBnd(U,e) \leq UppBnd(U)$ then $\gamma(e) = 0$ and (4) is automatically implied by Lemma 1.

Let now t_e^* be the shortest tour in scenario U among those including edge e. If $LowBnd(U,e) > UppBnd(U)$, then the tour $\hat{t} = St(Ind(t_e^*))$, that defines the regret term for tour t_e^*, cannot coincide with t_e^*, having a lower cost in scenario U:

$$ClCost(t_e^*) \geq LowBnd(U,e) > UppBnd(U) \geq ClCost(\hat{t}) \tag{5}$$

Therefore tour $\hat{t}$ has at least two edges that are not in common with tour t_e^*. Because of Theorem 1, at least two of the edges traversed by $\hat{t}$ will then be at their lower possible cost. A valid correction term for the overestimation provided by $UppBnd(U)$ for the cost of $\hat{t}$, is given by $\gamma(e)$ (that is the smallest possible error in the estimate). Inequality (4) is consequently valid. □

Proposition 2. *Given an heuristic estimate ub for the robustness cost of the optimal robust tour, if the following inequality holds*

$$LowBnd(U,e) - UppBnd(U) + \gamma(e) > ub \tag{6}$$

then edge e is not traversed by any optimal robust tour.

Proof. The thesis follows directly from Theorem 2. □

We will refer to an edge that, according to Proposition 2, is not part of any optimal robust tour as a *dominated* edge.

Sorting the edges of the graph in terms of their value of $u_e - l_e$ has a complexity of $O(|V| \log |V|)$ (Hoare [3]). The upper bound for the cost of the classic traveling salesman problem in scenario U is obtained approximately in $O(|V|^{2.2})$ (Helsgaun [2]). For each one of the $O(|V|^2)$ edges of the graph, the lower bound for the (classic) cost in scenario U of tours including edge e is obtained approximately in $O(|V|^{2.2})$ (Helsgaun [2]). This last term dominates the others, and produces a total approximate running time of $O(|V|^{4.4})$.

6 Computational experiments

In this section we measure from an empirical point of view the performance the heuristic and preprocessing routines presented in Sections 4 and 5, respec-

tively. Characteristics of the instances adopted for the tests are first described in Section 6.1.

All the algorithms have been coded in *ANSI C*. Callable libraries based on algorithms *LKH 1.3* (http://www.akira.ruc.dk/~keld/research/LKH), that represents the state of the art of heuristic and lower bounding (with and without compulsory edges) algorithms for the classic symmetric traveling salesman problem, is used in our implementation.

All the tests we present have been carried out on a Intel Pentium 4 1.5 GHz / 256 MB machine.

6.1 Description of the test problems

The benchmark problems proposed in Montemanni et al. [8] (available as a zip archive at http://www.idsia.ch/~roberto/RTSPinst.zip) have been adopted for the experiments presented in this paper.

6.1.1 Random instances

The first family is a set of non-euclidian random instances, generated according to the following schema: a problem of type *R-N-M* has N nodes ($|V| = N$) and $\forall i, j \in V$ cost u_{ij} is picked up at random from the set $\{0, 1, \ldots, M\}$, while l_{ij} is selected again at random from the set $\{0, 1, \ldots, u_{ij}\}$.

6.1.2 TSPLIB instances

The second family of problems has been generated from some classic traveling salesman problems available at the web-site *TSPLIB* (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95). Given a problem *Prob* from TSPLIB, each new instance named *Prob-*β (with $\beta \in [0, 1]$) will have the same vertices of the original instance *Prob*, while interval costs will be generated at random, in such a way that $\forall i, j \in V$, l_{ij} is selected at random as an integer number from the interval $\left[\lfloor (1-\beta)c^P_{ij} \rfloor, c^P_{ij}\right]$, while u_{ij} is an integer number

picked up from $\left[c_{ij}^P, \lceil (1+\beta)c_{ij}^P \rceil\right]$, with c_{ij}^P being the original cost in problem *Prob.*

6.2 Heuristic algorithms

The quality of the solutions provided by the heuristic approaches described in Section 4 is compared in Table 1. Algorithm HMU, not previously mentioned, also appears in Table 1. It runs in sequence methods HM and HU and returns the solution with the lowest upper bound for the cost, between the two. We decided to include also this method since the computation times required by both algorithms HM and HU are always extremely short. On our machine they are strongly below 0.1 seconds (comprehensive of the time required to estimate the robustness cost of the solution, as described in Section 3). It is worth to mention that HMU maintains the same computational complexity of HM and HU, since it sequentially runs them.

In Table 1 we report, for each problem and for each algorithm considered, average and standard deviation for the following indicator:

$$100 \cdot \frac{UB_{heu} - LB_{ex}}{UB_{heu}} \tag{7}$$

where UB_{heu} indicates an upper bound for the cost of the heuristic solution provided by the heuristic algorithm (according to Section 3) and LB_{ex} is the best lower bound for the cost of the optimal robustness solution available (found by the exact algorithms described in Montemanni et al. [8]).

It is interesting to mention here that the lower and upper estimates for the robustness cost obtained as described in Section 3 are very tight for the problems considered. due to the accuracy of the embedded upper and lower bound for the classic traveling salesman problem. For the problems considered, the average gap between lower and upper bound is only 0.56%.

Indicator (7) provides then an upper bound for the gap between the cost of the heuristic solution and that of the optimal one.

In the second column of Table 1 it is indicated whether an optimal solution is available for all the instances of each problem considered. In case this is true

(*Yes* in the corresponding entry of the table), the value LB of formula (7) will coincide with the cost of the optimal solution. Rows with entry *No* in the second column may overestimate the gap between the cost of the heuristic and that of an optimal one.

Notwithstanding the very short computation times required by the algorithms, the quality of the heuristic solutions provided by methods HU, HM and HMU is impressive. All of them hare on average no more than 5% above the lower bounds for the optimal costs.

We can observe that HU seems to perform slightly better than HM. It is however very interesting to notice that HMU, that combines HM and HU is able to clearly improve (both in terms of average and standard deviation) the results of the two basic heuristics. This is important, since it indicates that HM and HU tend to find solutions with different characteristics.

It is finally interesting to observe that the quality of the heuristic solutions does not seem to be strictly related to the dimension of the problems, since it does not decade for the biggest problems considered.

6.3 Preprocessing

In Section 5 a technique able to identify edges that will never be on an optimal robust tour has been described. In Table 2 we summarize the results obtained by this technique. The preprocessing routine needs un upper bound for optimal robustness cost as an input. In our implementation this information is provided by the heuristic algorithm HMU (see Section 6.2). We will not include the time required by this algorithm within the time required by the preprocessing technique itself. It is however important to remind that HMU is extremely fast, and its computation time are negligible.

In Table 2 we report, for each problem, information about the percentage of edges that are identified as dominated and about the computation time required by the preprocessing procedure.

Table 2 indicates that the percentage of edges detected as dominated heavily

Table 1: Heuristic algorithms. Percentage deviations from best lower bounds (7). Averages over 10 instances.

Problems	Optimal	Algorithm HM		Algorithm HU		Algorithm HMU	
		Avg	StDev	Avg	StDev	Avg	StDev
R-10-100	Yes	4.16	8.97	2.38	3.00	0.25	0.56
R-10-1000	Yes	6.96	7.61	4.28	5.57	1.44	2.45
R-20-100	Yes	3.51	5.14	1.48	2.37	0.73	1.39
R-20-1000	Yes	2.44	3.13	1.03	1.51	0.50	1.10
R-30-100	Yes	2.30	3.50	1.66	1.98	0.37	0.79
R-30-1000	Yes	3.06	2.95	1.03	1.35	0.94	1.36
R-40-100	Yes	3.99	3.32	1.02	0.98	0.96	0.89
R-40-1000	Yes	3.84	3.51	1.08	1.22	0.72	0.67
R-50-100	No	5.55	4.78	2.18	3.87	2.14	3.89
R-50-1000	Yes	4.87	2.22	1.11	1.19	1.08	1.15
R-60-100	No	6.33	4.26	3.41	2.82	3.35	2.90
R-60-1000	No	4.54	2.09	1.33	0.99	1.33	0.99
R-30-10	Yes	3.37	2.75	0.49	1.03	0.49	1.03
R-30-10000	Yes	2.23	1.87	1.79	2.61	0.61	0.64
gr17-0.25	Yes	4.13	5.42	4.44	4.20	2.42	3.07
gr17-0.50	Yes	3.37	3.67	3.82	3.86	2.35	3.24
gr21-0.25	Yes	5.53	8.01	7.98	10.04	5.53	8.01
gr21-0.50	Yes	0.87	2.64	1.02	1.53	0.44	1.27
gr24-0.25	Yes	5.01	4.45	4.13	4.48	3.32	3.39
gr24-0.50	Yes	2.10	1.94	0.88	1.11	0.70	1.08
fri26-0.25	Yes	2.11	2.15	10.38	10.78	1.78	1.97
fri26-0.50	Yes	2.20	2.34	0.82	0.99	0.73	0.78
swiss42-0.25	Yes	6.05	4.06	6.26	4.21	4.86	3.59
swiss42-0.50	No	2.10	2.09	2.12	1.42	1.62	1.44
dantzig42-0.25	Yes	3.22	3.08	3.33	3.30	2.87	2.93
dantzig42-0.50	Yes	2.99	2.87	2.28	1.49	1.65	1.38
gr48-0.25	Yes	8.93	5.33	10.69	4.88	6.76	2.44
gr48-0.50	No	11.13	5.51	14.61	4.45	11.04	5.54
hk48-0.25	Yes	3.85	4.10	5.66	6.02	1.53	1.74
hk48-0.50	No	4.92	3.83	6.63	4.86	4.73	3.88
brazil58-0.25	Yes	6.05	7.37	3.74	1.68	2.52	1.27
brazil58-.50	No	7.99	4.74	4.27	1.52	3.78	1.61
Averages		4.37	4.05	3.67	3.17	2.30	2.14

Table 2: Preprocessing. Dominated edges and computation time. Averages over 10 instances.

Problems	Dominated edges (%)		Seconds	
	Avg	StDev	Avg	StDev
R-10-100	3.33	9.03	0.27	0.02
R-10-1000	4.00	6.93	0.27	0.01
R-20-100	0.00	0.00	1.84	0.07
R-20-1000	1.47	2.78	1.78	0.08
R-30-100	0.00	0.00	5.89	0.22
R-30-1000	0.00	0.00	5.87	0.29
R-40-100	0.00	0.00	15.11	0.72
R-40-1000	0.00	0.00	14.77	0.63
R-50-100	0.00	0.00	30.54	0.79
R-50-1000	0.00	0.00	30.96	1.17
R-60-100	0.00	0.00	54.86	1.46
R-60-1000	0.00	0.00	56.35	1.62
R-30-10	0.00	0.00	6.86	0.16
R-30-10000	5.61	9.16	5.86	0.25
gr17-0.25	11.32	7.21	1.12	0.09
gr17-0.50	0.00	0.00	1.09	0.08
gr21-0.25	71.71	6.17	2.37	0.15
gr21-0.50	8.57	7.32	2.31	0.07
gr24-0.25	33.62	10.89	3.24	0.08
gr24-0.50	0.33	0.36	3.24	0.16
fri26-0.25	35.17	12.49	4.06	0.28
fri26-0.50	0.71	1.33	4.05	0.33
swiss42-0.25	21.34	6.32	18.31	0.60
swiss42-0.50	0.00	0.00	18.57	0.92
dantzig42-0.25	20.46	4.27	19.02	0.68
dantzig42-0.50	0.03	0.08	19.09	0.57
gr48-0.25	4.89	3.68	31.62	0.32
gr48-0.50	0.00	0.00	30.65	0.59
hk48-0.25	26.13	5.27	26.79	0.87
hk48-0.50	0.01	0.03	27.37	1.52
brazil58-0.25	4.25	3.03	60.22	2.13
brazil58-.50	0.04	0.03	57.75	3.86
Averages	7.91	3.01	17.57	0.65

depends on the characteristics of the problem. For the random problems the preprocessing routine works better for instances with large values for the highest possible cost (e.g. R-30-10000). This can be justified by observing that these problems have edges with very large values for the interval costs. These edges are very likely to be dominated. More interesting is the situation for the TSBLIB problems. In this case the preprocessing technique works better for problems with a smaller uncertainty factor (e.g. gr21-0.25 instead of gr21-0.50). This indicates an effort in providing tighter interval costs to the model, will translate into a more effective preprocessing phase, that in turn will translate into an easier problem.

Table 2 also indicate that the computation times of the preprocessing routine are always short (on average within 1 minute for all the problems but one). These computation times are obviously related to the dimension of the problems, since each edge has to be examined.

7 Conclusion

The robust traveling salesman problem with interval data, a version of the problem where uncertainty about edge costs is taken into account, has been studied in this paper.

Some efficient heuristic algorithms and a preprocessing technique have been discussed and evaluated from a computational point of view.

The methodological approach presented can be easily extended to the interval data counterpart of many other classic NP-hard combinatorial optimization problems.

Acknowledgements

The authors would like to thank Keld Helsgaun for his suggestions and help in the use of the routines of *LKH*.

References

[1] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson. Solutions of a large scale travelling salesman problem. *Operations Research*, 2:393–410, 1954.

[2] K. Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[3] C.A.R. Hoare. ACM algorithm 64: Quicksort. *Communications of the ACM*, 4(7):321, 1961.

[4] A. Kasperski and P. Zieliński. An approximation algorithm for interval data minmax regret combinatorial optimization problems. Submitted for publication.

[5] A. Kasperski and P. Zieliński. The shortest path problem with interval data. Submitted for publication.

[6] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its applications.* Kluwer Academic Publishers, 1997.

[7] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem.* Wiley, Chichester, 1985.

[8] R. Montemanni, J. Barta, and L.M. Gambardella. The robust traveling salesman problem with interval data. submitted for publication.

[9] R. Montemanni and L.M. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers and Operations Research*, 31(10):1667–1680, 2004.

[10] R. Montemanni, L.M. Gambardella, and A.V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32(3):225–232, 2004.

[11] G. Reinelt. *The Traveling Salesman: Computational solutions for TSP applications*. Springer-Verlag, Berlin, 1994.