Adaptive Memory Programming for the Robust Capacitated International Sourcing Problem

José Luis González Velarde

Centro de Sistemas de Manufactura, ITESM Monterrey, México
Lugonzal@campus.mty.itesm.mx

Rafael Martí

Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain.
Rafael.Marti@uv.es

Latest revision: October 7, 2005

Abstract

The International Sourcing Problem consists of selecting a subset from an available set of potential suppliers internationally located. The selected suppliers must meet the demand for items of a set of plants, which are also located worldwide. Since the costs are affected by macroeconomic conditions in the countries where the supplier and the plant are located, the formulation considers the uncertainty associated with changes in these conditions. We formulate the robust capacitated international sourcing problem by means of a scenario-optimization approach. In this paper we propose a constructive method based on memory structures to solve this problem. The method is coupled with a local search procedure followed by a path relinking for improved outcomes. We propose innovative mechanism to achieve a good balance between intensification and diversification in the search process. Moreover, our path relinking implementation uses constructive neighborhoods for extrapolated relinking. The computational experimentation favors this method when compared with a recent tabu search approach.

Key Words: Memory structures, Path Relinking

1. Introduction

The Robust Capacitated International Sourcing Problem (ROCIS) consists of selecting a set of suppliers to meet the demand for items at several plants located in different countries. Different versions of the problem have been proposed depending on the assumptions. Table 1 summarizes the more relevant work in this and related problems.

Authors	Description	Year
Jucker and Carlson	Solve a single product, single period problem, with price and demand uncertainty	1976
Hodder and Jucker	Present a deterministic single period, single product model	1982
Hodder and Jucker	Optimally solve a single period, single product model with quantity setting firms	1985
Haug	Addresses the deterministic problem with a single product and multiple periods with discount factors	1985
Louveaux and Peters	Solve a scenario-based problem in which capacity is a first stage decision	1992
Gutierrez and Kouvelis	Explore the generation of scenarios to model price uncertainty and solve simple plant location problem	1995
Kouvelis and You	Propose an un-capacitated version robustness approach based on a *minimax regret* criterion	1997
González-Velarde and Laguna	Propose a tabu search method for the robust capacitated version with exchange rate uncertainty	2004

Table 1. Relevant previous work

In this paper we consider the variant introduced in González-Velarde and Laguna (2004), which deals with a single item in a single period and model uncertainty in the demand and the exchange rate via a set S of scenarios. A first formulation of the problem follows:

$$Min \quad \sum_{s \in S} p_s \left(\sum_{i \in M} e_{is} f_i y_i + \sum_{i \in M} \sum_{j \in N} e_{is} c_{ij} x_{ijs} \right)$$

subject to:

$$\sum_{i \in M} x_{ijs} \geq d_{js} \qquad \forall\ j \in N, s \in S$$

$$\sum_{j \in N} x_{ijs} \leq b_i y_i \qquad \forall\, i \in M, s \in S.$$

The *control* decision variables x_{ijs} reflect the shipping from supplier i ($i \in M=\{1,2,...,m\}$) to plant j ($j \in N=\{1,2,...,n\}$) in scenario s. The *design* variables y_i take the value 1 if supplier i is contracted, and 0 otherwise. The first set of constraints includes the condition that for each scenario s, the demand at plant j, d_{js}, must be satisfied. The second set considers that for each scenario s, the capacity of supplier i, b_i, cannot be exceeded.

The objective function reflects the fact that the total unit cost for delivering components from supplier i to plant j, c_{ij}, is known, however, the exchange rate at supplier's i location in scenario s, e_{is}, make cost data uncertainty under different scenarios (where p_s is the probability of occurrence of scenario s). Finally, it also incorporates the fixed cost f_i of development of supplier i.

González-Velarde and Laguna (2004) improve this formulation, replacing the objective function above with the following expression:

$$F(y)=\sum_{s\in S}p_s\left(\sum_{i\in M}e_{is}f_iy_i+z_s\right)+\omega\sqrt{\frac{\sum_{s\in S^+}p_s(z_s-E(z))^2}{\sum_{s\in S^+}p_s}}$$

where z_s is the optimal objective value associated with the transportation problem obtained when fixed the problem above for a particular scenario s, and:

$$S^+=\{s:z_s-E(z)\geq 0\}$$
$$E(z)=\sum_{s\in S}p_sz_s$$

This objective function penalizes only the positive deviations from the expected value (those situations in which the objective value in a given scenario exceeds the expected cost) and includes a normalizing term to account for the fact that not all the terms in the probability distribution are being added. The value of ω is a factor that the decision-maker can adjust to give more or less importance to the risk component of the objective function.

The authors develop a tabu search algorithm to obtain efficient solutions for this non-linear integer program. The proposed solution method can be viewed as a heuristic based on the paradigm of Benders decomposition. An initial set of values is assigned to the *y* binary variables, which makes the remaining problem linear. This problem in turn may be decomposed into |S| smaller linear sub-problems, one for each scenario. The optimal dual solution for each sub-problem is used to find a new set of values for the binary variables. Instead of generating valid cuts for an integer problem as in Benders method, these dual variables are combined to form neighborhoods of promising solutions and a search is conducted in the generated neighborhood. The search method is based on the short-term memory strategies of tabu search (Glover and Laguna, 1997). A new set of values is selected for the binary variables and the procedure continues until some termination criterion is reached.

In this paper we propose an alternative solution method for the robust capacitated international sourcing problem. Section 2 describes our solving methodology based on constructive memory structures and path relinking, to obtain high quality solutions for this problem. Section 3 is devoted to the computational experiments. It shows that our adaptive memory programming method outperforms the previous tabu search approach as well as a generic scatter search code. The paper finishes with the associated conclusions.

2. Solution Method

Our solution method for the ROCIS problem consists of three stages. The first one is a constructive procedure that incorporates memory structures for diversification purposes. The second one is a local search method, which is selectively applied to improve previously generated solutions. Here, the meaning of selectively is not limited to the objective function evaluation, but also includes the concept of influence associated with the solution structure. The third stage creates paths connecting improved solutions based on the path relinking methodology.

Most of the tabu search applications (Glover and Laguna, 1997) implement transition neighborhoods in the context of local search methods. Constructive neighborhoods have been rarely used with memory structures, although they were introduced from the very beginning of the methodology (Glover, 1989). In this paper we present a constructive method based on a greedy function modified with frequency based memory.

In common with other evolutionary methods, Path Relinking operates with a population of solutions, rather than with a single solution at a time, and employs procedures for combining these solutions to create new ones. From a spatial orientation, the process of generating linear combinations of a set of reference solutions may be characterized as generating paths between and beyond these solutions, where solutions on such paths also serve as sources for generating additional paths. This leads to a broader conception of the meaning of creating *combinations* of solutions. By natural extension, such combinations may be conceived to arise by generating paths between and beyond selected solutions in neighborhood space, rather than in Euclidean space. In this paper we propose a path relinking method that creates paths connecting all the pairs of solutions obtained from the selective application of the local search algorithm.

In the following subsections we describe the implementation of these three stages, as adapted in the context of the ROCIS problem.

2.1 Constructive Method

The first stage of our solution procedure is particularly important, given the goal of developing a method that balances diversification and intensification in the search. We implement a constructive method using frequency-based memory, as proposed in tabu search. This method is based on modifying a greedy measure of attractiveness by using a frequency counter that discourages the selection of suppliers frequently selected in previous solution generations.

The attractiveness of selecting supplier i is given by the greedy function $G(i)$ adding both, the fixed cost associated with this supplier f_i, and the sum of the shipping unit costs from this supplier to all the plants, relative to the supplier's capacity b_i. The shipping cost is multiplied by the probability of each scenario, making $G(i)$ a measure of expected attractiveness.

$$G(i) = \frac{f_i + \left(\sum_{s \in S} p_s \left(\sum_{j=1}^{n} c_{ij} e_{is} \right) \right)}{b_i}$$

A frequency counter $Freq_i$ is maintained to record the number of times supplier i has been selected in previous solutions. This frequency counter is used to penalize the "attractiveness" of an element, and therefore, inducing diversification with respect to the solutions already generated. We modify the value of $G(i)$ to reflect previous selections of element i, as follows:

$$G'(i) = G(i) + \beta \cdot \left(\frac{MaxG}{MaxFreq} \right) \cdot Freq_i \quad ,$$

where $MaxFreq$ is the maximum $Freq_i$ value for all i, and $MaxG$ is the maximum $G(i)$ value for all i.

Each construction starts by creating a list of unassigned suppliers U, which at the beginning consists of all the suppliers in the problem (i.e., initially $|U| = m$). Then, we restrict this candidate list considering the set U' with the k most attractive suppliers, according to the G'-value. In each construction step, the next supplier s is randomly selected from the set U', then U is updated ($U = U - \{s\}$) and U' is recalculated. The method finishes when the sum of the capacities of the selected suppliers is at least as large as D.

$$D = \max_{s \in S} \left(\sum_{j \in N} d_{js} \right)$$

Note that we stop the construction when the selected plants can satisfy the demand in any scenario. However, it is possible that an optimal solution have more selected plants than this minimum number. Since we do not have any information about the best number of selected plants, we do not implement here any mechanism to increase it in the current solution under construction because it would be absolutely artificial. In subsection 2.3 we describe a mechanism to explore variations on the cardinality of this set in the combination element of the path relinking method.

To avoid initial biases, the frequency mechanism is activated after the first *InitIter* constructions, and before this, selections are made with the G-value. Let P be the set of solutions generated with this method.

It is important to point out that although our construction method employs a candidate and a restricted candidate lists, the evaluation function $G'(i)$ is not adaptive since its value remains constant thorough the construction of a solution (and changes from one construction to the next one according to the frequencies). Therefore, strictly speaking, this is not a GRASP construction (Feo and Resende, 1995) and can be better classified as a memory-based construction.

Each generated solution is evaluated, by solving the scenario sub-problems and calculating $F(y)$. We also use a hash function to codify each solution as follows:

$$H(y) = \sum_{i \in M} y_i 2^i .$$

The values of $F(y)$ and $H(y)$ are stored to avoid evaluating a solution that has already been generated. The rational for this is that the objective function evaluation can be considerably expensive in terms of computational time as the number of scenario increases. It is also more convenient to store the hash value than the entire binary string in order to efficiently search for the membership of the solution in the list. We apply this evaluation filter based on the hash function in the entire procedure, including the improvement and path relinking methods described below.

2.2 Improvement Method

We partition the set P of generated solutions into classes according to the cardinality of the selected suppliers in the solutions. Specifically, let A_k be the set of solutions with k selected suppliers:

$$A_k = \left\{ y \in P \,/\, \sum_{i=1,..,n} y_i = k \right\}$$

Note that $P=A_1 \cup A_2 \cup .. \cup A_n$ where some A_k might be empty. In order to have a diverse set of solutions we want to have good representations of each cardinality set. Since the objective function evaluation is computationally expensive, the local search is only applied to a percentage *pr* of the best solutions in each set A_k.

The local search consists of exchanges of suppliers: a selected supplier is replaced by a non-selected one in the current solution. We examine the suppliers y_i from i=1 to n-1, and consider the most attractive un-selected suppliers for exchange, where attractiveness is now measured with $G''(j)$. The value $G(j)$ is modified adding the term $V(j)$ which measures the relative contribution of supplier j to the quality of the generated solutions (we scale this term to be in the same range than $G(j)$). We define $S_j \subseteq P$ as the set of generated solutions in which supplier j is selected: $S_j=\{y \in P \mid y_j=1\}$. Then, we compute $V(j)$ as the average value of the solutions in S_j.

$$G''(j) = G(j) + \left(\frac{MaxG}{MaxV} \right) \cdot V(j) \quad , \quad V(j) = \frac{\sum_{y \in S_j} F(y)}{\left| S_j \right|}$$

Where *MaxV* represents the maximum of the $V(j)$ values for j=1,…,m, and, as in previous expressions, *MaxG* is the maximum of the G-values.

The local search procedure examines, for each supplier y_i, the best alternative supplier for exchange. Non selected suppliers y_j are then scanned in the order given by the G''-value (where the supplier with the lowest G''-value is examined first). For each supplier y_j we test whether this exchange is feasible in terms of the capacity (we only admit those solutions where the supply exceeds the demand for every scenario). The first feasible exchange that results in a solution with a lower objective value is performed. The algorithm finishes when no further improvement is possible.

The local search is applied to the best solutions in each set A_k as determined by the parameter *pr*. Let A_k' be the set of improved solutions obtained with the application of the local search to the selected solutions in A_k. In the following sub-section we combine the solutions within each A_k' and between pairs of them.

2.3 Path Relinking

Path Relinking, PR, can be considered an extension of the classical combination mechanisms of other evolutionary methods. Instead of directly producing a new solution when combining two or more original solutions, PR generates paths between and beyond the selected solutions in the neighborhood space. The character of such paths is easily specified by reference to solution attributes that are added, dropped or otherwise modified by the moves executed. Examples of such attributes include edges and nodes of a graph, sequence positions in a schedule, vectors contained in linear programming basic solutions, and values of variables and functions of variables.

To generate the desired paths, it is only necessary to select moves that perform the following role: upon starting from an *initiating solution,* the moves must progressively introduce attributes contributed by a *guiding solution* (or reduce the distance between attributes of the initiating and guiding solutions). Then, consider the creation of a path that join two selected solutions y' and y'', restricting attention to the part of the path that lies 'between' the solutions, producing a solution sequence $y' = y(1), y(2), \ldots, y(r) = y''$.

Path Relinking starts from a given set of elite solutions obtained during a search process. Following the terminology given in Laguna and Martí (2003), we will let *RefSet* (short for "Reference Set"), refer to this set of *b* solutions that have been selected during the application of the previous search method. It has been well documented in the Scatter Search context (Glover 1998), that the size of the *RefSet* is the 10% of the size of the set *P*. Then, we will consider this ratio in our solution procedure and apply the PR method to all pairs of the best 10% improved solutions obtained after the application of the local search procedure. In mathematical terms: $RefSet = A_1' \cup A_2' \cup .. \cup A_n'$.

For each pair of solutions y' and y'' in A_j' we first compute two new solutions, $y_\cap$ and $y_\cup$. In the former we select the suppliers present in both solutions and the latter contains those suppliers present in at least one of them. In mathematical terms:

$$y_{\cap\, i} = \min\,(y'_i\,, y''_i)\ \ ,\ \ y_{\cup\, i} = \max\,(y'_i\,, y''_i)$$

Then, instead of creating a path between y' and y'', we create a path between these two new points $y_\cap$ and $y_\cup$. Let $y_\cap$ be the initiating solution and $y_\cup$.be the guiding solution in the path (which contains y' and y'' as shown in Figure 1). Let *S'* be the set of suppliers selected in *y'* but non-selected in *y''*. Symmetrically, let *S''* be the set of suppliers non-selected in *y'* and selected in *y''*.

$$S' = \{\text{ supplier } j \mid y_j'=1 \text{ and } y_j''=0 \}\ \ ,\ \ S''=\{\text{ supplier } j \mid y_j'=0 \text{ and } y_j''=1 \}$$

Starting with $y_\cap$, intermediate solutions in the first part of the path are generated by adding a supplier from *S'* to the current solution. Given an intermediate solution, non-selected suppliers are ordered according to their relative merit as measured by the *G''* value. This value was introduced in the improvement phase; however, we redefine $S_j \subseteq RefSet$ as the set of improved solutions in the *RefSet* in which supplier *j* is selected: $S_j=\{y \in RefSet \mid y_j=1\}$. Then, we compute *G''* with the expression given in the previous subsection.

Given an intermediate solution *y* in the first part of the path (initially $y= y_\cap$), non selected suppliers in *S'* are ordered according to their *G´´* value. Then, the supplier *j** in *S'* with the lowest *G´´* value is selected (y_{j*}=1), thus obtaining the next solution in the path. Once *y'* has been reached (after |*S'*| selections), we alternate between adding and deleting suppliers from the current solution to reach *y''*. Specifically, suppliers to be deleted are those in *S',* while suppliers to be added are those in *S''*. Even iterations correspond to additions and odd correspond to deletions. Given an intermediate solution *y*, the non selected supplier in *S'* with lowest *G''* value, *j**, is selected in an even iteration for inclusion in *y* (y_{j*}=1). Similarly, in an odd iteration, the non-selected supplier in *S'* with largest *G''* value, *j+*, is selected for deletion (y_{j+}=0). Finally, once y'' has been reached, in the third part of the path, we add the suppliers in *S'* to obtain $y_\cup$. As in the first part of the path, at each iteration we add the non selected supplier in *S'* with lowest *G''* value. The relinking finishes when the initiating solution matches the guiding solution (after 3|*S'*|+|*S''*| intermediate solutions have been generated).

As it is done in previous path relinking implementations (Laguna and Martí, 1999), we have also considered the inclusion of a extensive exploration at certain points of the relinking process. Specifically, an expanded neighborhood from some of the feasible solutions along the path is examined. It consists of exchanges of suppliers in which a selected supplier is replaced by a non-selected one until no more improvement can be made. This is the same exchange mechanism used in the improvement phase. Once the expanded neighborhood has been explored, the relinking continues from the solution before the exchanges were made.

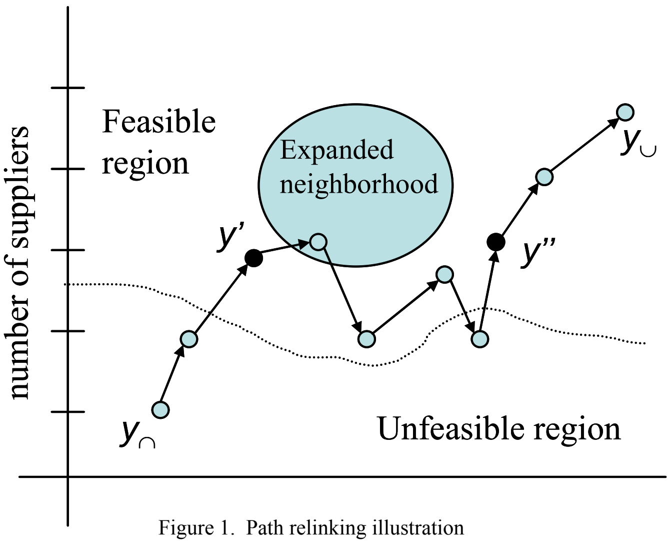

Figure 1. Path relinking illustration

Note that two consecutive solutions after a relinking step differ only in the selection of one supplier. Therefore, it is not efficient to apply the expanded neighborhood exploration (i.e., the exchange mechanism) at every step of the relinking process. As recommended in Laguna and Martí (1999), the exchange mechanism is applied every 10 steps of the relinking process.

Note that $y_\cap$ as well as the first solutions in the path can eventually be infeasible with respect to the capacity. However, once the feasibility is attained in an intermediate solution, we try to keep the feasibility in the remaining solutions in the path. This is why we alternate between adding and deleting suppliers in the second part of the path since successive deletions would cause unfeasibility. Note that we cannot guarantee that every solution in the path will be feasible but feasibility will be restored with the proposed mechanism.

Once the path has been traversed in the direction defined from y' to y'', the procedure is applied in reverse direction (form y'' to y') given that a different path is generated. The Path Relinking procedure terminates when all pairs of solutions have been examined in both directions.

3. Computational Experiments

For our computational testing we first use the set of 90 instances reported in González-Velarde and Laguna (2004). In this set the number of scenarios is fixed to 27, the number of plants to 10 and the number of suppliers to 10, 15 and 20, these numbers define three groups of 30 instances each. Within these three size categories, six subgroups of size 5 were formed by varying the parameter to define the relationship between demand at the plants and the capacity of the suppliers, as well as the parameter to define the relationship between fixed and variable costs. As in this previous work, we use a value of $\omega=2$ in the robust objective function, which is the one that penalizes positive deviations from the expected cost.

Note that the computational experiments in similar studies (e.g., Kouvelis and Yu, 1997), deal with problem instances of comparable size. However, the scenario sub-problems in such studies are trivially solved, given that they assume an infinite capacity for each supplier. Additionally to these 90 instances, we have generated 30 new larger instances with 20 plants and 40 suppliers. These instances have been generated with the same procedure reported in González-Velarde and Laguna (2004).

Laguna and Martí (2003) propose a generic scatter search for binary problems. Their method was originally designed to solve a knapsack problem; however it can be easily adapted to our problem. Basically we only need to use the evaluation function described in the introduction and modify the knapsack capacity constraint to control the feasibility. We have included this generic solver in our comparison as a baseline to measure the contribution of the specific solvers such as the previous tabu search method by González-Velarde and Laguna (2004) and our current implementation.

In our preliminary experimentation the value of *InitIter* was set to 10 and we have considered the key search parameters and 15 instances with 10 plants and 20 suppliers. In the first experiment we undertake to measure the value β. For each value of β (0.3, 0.5 and 0.7) Table 2 shows the average of the best objective value found with the constructive method, as well as the number of optima and average running time.

Table 2. Previous experiment.

β	**0.3**	**0.5**	**0.7**
Deviation	12.8%	12.2%	15.3%
Num. of Opt.	0	0	0
CPU sec.	0.81	0.84	0.81

Table 2 shows that the best solution is obtained, on average, with the constructive method with a β value of 0.5. In the second experiment we measure the contribution of the percentage parameter *pr* in the quality achieved by the local search method. Note that the local search is only applied to the best *pr%* solutions in each set A_k. We test three values for this parameter: 25%, 50% and 75% and use the same 15 instances that in the previous experiment. We do not produce tables for this experiment, since these three values provide the same solutions in the local search procedure. However, as expected, run time increases as *pr* increases. Therefore we set *pr*=25% in our solution method.

In the next experiment, we employ the 90 problem instances reported in González-Velarde and Laguna (2004). As mentioned, these instances have 10 plants and are grouped in three categories according to the number of suppliers (10, 15 and 20). Tables 3, 4 and 5 report for each group of 30 instances the average objective value, the average deviation from the optimal solutions, the number of optima achieved, and the average CPU seconds of the different methods under consideration. We compare the performance of the tabu search method (TS, González-Velarde and Laguna 2004), the generic scatter search method (SS), the constructive procedure described in section 2.1 (Const), the constructive procedure followed by the local search described in section 2.2 (Const+LS) and, the path relinking method (PR).

Table 3. n=10, m=10.

	TS	**SS**	**Const**	**Const+LS**	**PR**
Value	49706.3	52649.9	50326.0	49831.9	49706.3
Deviation	0.00%	6.49%	1.33%	0.35%	0.00%
Num. of Opt.	30	3	14	28	30
CPU sec.	0.59	5.27	0.20	0.60	3.33

Table 4. n=10, m=15.

	TS	**SS**	**Const**	**Const+LS**	**PR**
Value	41439.7	43832.0	42411.2	40487.7	40452.8
Deviation	2.74%	9.29%	5.27%	0.11%	0.00%
Num. of Opt.	4	0	2	27	30
CPU sec.	2.32	10.45	0.25	2.93	6.18

Table 5. *n*=10, *m*=20.					
	TS	**SS**	**Const**	**Const+LS**	**PR**
Value	38665.6	41396.3	39599.5	36342.5	36290.1
Deviation	7.42%	15.57%	9.95%	0.62%	0.44%
Num. of Opt.	0	0	0	19	21
CPU sec.	5.95	18.73	0.31	6.66	10.24

These tables show that the best solution quality is obtained by the path relinking method (PR), which is able to match a larger number of optimal solutions than the other methods. This is especially true in the instances with 20 suppliers in which PR matches 21 optimal solutions, Const+LS 19, and none of them the other methods. Considering the 90 instances in Tables 3, 4 and 5 together, TS matches 34, SS 3, Const 16, Const+LS 74 and PR 81. However, although in this problem run time is not a critical factor, it should be noted that the PR method consumes a running time 26 times higher than the simple construction method (Const). These tables also show that the performance of the SS method is clearly inferior with a significantly lower number of optimal solutions than those achieved by the other approaches. However, it is a generic method and its results are quite acceptable considering its wide applicability to any 0-1 optimization problem.

Regarding the relative deviation from optimality, Table 3 shows that both, the TS and the PR method present a 0.00% deviation on average, while SS, Const and Const+LS present 6.49%, 1.33% and 0.35% respectively. However, as shown in Tables 4 and 5, in larger graphs (relative to the number of suppliers) the methods quickly deteriorate presenting larger deviations from optimality. The ranking of the methods according to the average percentage deviation value across the 90 instances is SS (10.45%), Const (5.51%), TS (3.39%), Const+LS (0.36%) and PR (0.15%).

In our last experiment we undertake to compare the performance of our proposed procedures using relatively larger graphs (as compared to those in the first experiment). In specific, we generate 30 additional instances with 20 plants and 40 suppliers. We cannot obtain the optimal solution for these large instances. The BEST column represents the minimum value of the objective function for each instance after running all procedures during the experiment. (We cannot assess how close the BEST values are from the optimal solutions, and we are only using these values as a way of comparing the methods.)

Table 6. *n*=20, *m*=40.					
	Best	**TS**	**Const**	**Const+LS**	**PR**
Value	50359.15	51730.25	59427.64	50398.47	50359.15
Deviation	0.00%	2.93%	19.41%	0.11%	0.00%
Num. of Opt.	30	1	0	27	30
CPU sec.	--	381.25	2.68	188.64	218.23

Table 6 clearly shows that the proposed procedure outperforms the previous tabu search approach since it is able to obtain all the best known solutions in these large instances. The previous tabu search implementation also performs well since it presents an average deviation from the best solution known of 2.93%. It should be also noted that the construction with local search (without the path relinking phase) performs remarkably well in a relative short computational time (it achieves a 0.11% of average relative deviation and matches 27 out of 30 best known solutions in about 3 minutes of CPU time).

Conclusions

We have developed a heuristic procedure based on the Tabu Search methodology to provide high quality solutions to the Robust Capacitated Sourcing Problem (Rocis). Our solution method consists of three stages. A constructive procedure that incorporates memory structures for diversification purposes, a local search method, which is selectively applied to improve previously generated solutions with an associated low computational effort, and a path relinking implementation to create paths connecting improved solutions.

The proposed procedure was shown competitive in a set of problem instances for which the optimal solutions are known. For a set of larger instances, the proposed construction with its local search and path relinking variants performed remarkably well (outperforming the best procedure reported in the literature).

Acknowledgments

Research by José Luis González-Velarde is partially supported by the ITESM Research Chair in Industrial Engineering. Research by Rafael Martí is partially supported by the *Ministerio de Educación y Ciencia* (refs. TIN2004-20061-E and TIC2003-C05-01) and by the *Agencia Valenciana de Ciència i Tecnologia* (ref. GRUPOS03 /189).

References

Feo, T. and M. G. C. Resende (1995) "Greedy Randomized Adaptive Search Procedures," *Journal of Global Optimization,* Vol. 2, pp. 1-27.

Glover, F. (1989) "Tabu Search-Part I", *ORSA Journal on Computing*, Vol. 1, pp. 190-206

Glover, F. and M. Laguna (1997) *Tabu Search*, Kluwer Academic Publishers.

Glover, F., 1998. A Template for Scatter Search and Path Relinking. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (Eds.), Artificial Evolution, Lecture Notes in Computer Science 1363, Springer, pp. 13-54.

González-Velarde, J.L. and M. Laguna (2004), "A Benders-based heuristic for the robust capacitated international sourcing problem", *IIE Transactions*, vol. 36, pp. 1125-1133

Gutiérrez, G.J. and P. Kouvelis (1995), "A Robustness Approach to International Sourcing", *Annals of Operations Research*, vol. 59, pp. 165-193.

Haug, P. (1985), "A Multiple-Period, Mixed-Integer-Programming Model for Multinational Facility Location," *Journal of Management*, vol. 11, no. 3, pp. 83-96.

Hodder, J.E. and J. V. Jucker (1985), "A Simple Plant-Location Model for Quantity-Setting Firms subject to Price Uncertainty," *European Journal of Operational Research*, vol. 21, pp. 39-46.

Hodder, J.E. and J.V. Jucker (1982), "Plant Location Modeling for the Multinational Firm," Proceedings of the Academy of International Business Conference on the Asia-Pacific Dimension of International Business, Honolulu, December, 1982, pp. 248-258.

Jucker, J.V. and R.C. Carlson (1976) "The Simple Plant-Location Problem under Uncertainty," *Operations Research*, vol. 24, no. 6, pp. 1045-1055.

Kouvelis, P. and G. Yu (1997) *Robust Discrete Optimization and its Applications* Kluwer Academic Publishers, Dordrecht.

Laguna, M. and R. Martí (1999) "GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization," *INFORMS Journal on Computing,* Vol. 11, No. 1, pp. 44-52.

Laguna, M., Martí, R., 2003. Scatter Search – Methodology and Implementations in C*,* Kluwer Academic Publishers, Boston.

Louveaux F.V. and D. Peeters. (1992) "A dual-based procedure for stochastic facility location". *Operations Research*, vol. 40 no. 3, pp. 564—573.