Improving the Intensification and Diversification Balance of the Tabu Solution for the Robust Capacitated International Sourcing Problem (RoCIS)

Héctor Joaquín Fraire Huacuja, José Luis González-Velarde, Guadalupe Castilla Valdez

Abstract:

This paper addresses the robust capacitated international sourcing problem (RoCIS), which consists of selecting a subset of suppliers with finite capacity, from an available set of potential suppliers internationally located. This problem was introduced by González-Velarde and Laguna in [1], where they propose a deterministic solution method based on tabu search memory strategies. The process consists of three steps: build an initial solution, create a neighborhood of promising solutions and perform a local search in the neighborhood. In this work we propose improving the construction of the initial solution, the construction of the neighborhood, the local search, and the intensification and diversification balance. Experimental evidence shows that the improved tabu solution with diversification outperforms the best solutions reported for six of the instances considered, increases by 18% the number of best solutions found and reduces by 44% the deviation of the best solution found, respect to the best algorithm reported.

Keywords: *RoCis, heuristic approach, optimization, initial solution, tabu search, memory strategies.*

1. Introduction

The international sourcing problem consists of selecting a subset of suppliers, with a finite production capacity, from an available set of potential suppliers located internationally. In this paper we analyze the variant proposed in [1], which considers only a product in a single period and uncertainty on the demand and the exchange rate are modeled *via* a set of scenarios. In the formulation of this problem it is assumed that the costs depend on the economic conditions in the countries where the suppliers and the plants are located and that the production capacity of suppliers is finite. The robust formulation considers that a solution is feasible if and only if it is feasible in all the scenarios. The objective function minimizes the expected value of the costs and penalizes the solutions whose optimal cost in some scenario surpasses the expected value of the optimal costs in all the scenarios. Through this mechanism the associated risk is incorporated.

The rest of our paper is organized as follows: related work, problem formulation, solution proposal and experimental results.

2. Related work

Now we summarize the most relevant works from the literature about the plant location problem, because it is closely related to the international sourcing problem. Jucker and Carlson solve a single product, single period problem, with price and demand uncertainty [2]. Hodder and Jucker present a deterministic single period, single product model [3]. Hodder and Jucker optimally solve a single period, single product model, setting the plants quantity [4]. Haug approaches the deterministic problem with a single product and multiple periods with discount factors [5]. Louveaux and Peters solve a scenario-based problem in which the capacity is a first stage decision [6]. Gutierrez and Kouvelis explore the generation of scenarios to model price uncertainty and solve a simple plant location problem [7]. Kouvelis and You propose an un-capacitated version robustness approach based on a minimax regret criterion [8] .

Now we describe the most relevant work about the international capacitated sourcing problem. The robust formulation of the international capacitated sourcing problem was proposed by Gonzalez-Velarde and Laguna [1]. In this work they propose a solution method based on the Benders paradigm, incorporating Tabu Search (TS) mechanisms. The process consists of building an initial solution, creating a neighborhood of promising solutions and performing a local search on the neighborhood. As the choice of the initial solution determines the efficiency of the process, this solution is constructed by applying a heuristic that gives preference to suppliers with lower fixed costs and greater production capacity.

González-Velarde and Martí propose a non-deterministic solution method based on GRASP, without incorporating the adaptive element, so the algorithm is classified as adaptive memory programming (AMP) type and path relinking is used to post processing the built solutions [9]. In the heuristic used to build a set of initial solutions, the shipping cost of each supplier to all plants is considered. The authors suggest that this way of incorporating the shipping cost seems too pessimistic.

In this work we propose to modify the TS based solution by improving the construction of the initial solution, the construction of the neighborhood, the local search, and the intensification and diversification balance.

3. Problem formulation

AThe robust capacitated international sourcing problem (RoCIS) consists of selecting a set of suppliers to satisfy the demand for products at several plants located in different countries. The model deals with a single item in a single period. The uncertainty in the demand and the exchange rates are modeled *via* a set of scenarios. The model uses the following definitions:

Parameters

N: international plants set $\{1,2,...,n\}$.
M: international suppliers set $\{1,2,...,m\}$.
S: scenarios set.
f_i: fixed cost associated with supplier i.
c_{ij}: total unit cost for delivering items from supplier i to plant j.
b_i: capacity of supplier i.
d_{js}: demand at plant j under scenario s.
e_{is}: exchange rate at supplier's i location under scenario s.
p_s: occurrence probability of scenario s.

Variables

x_{ijs}: product shipment from supplier i to plant j under scenario s.
y_i: 1 if supplier i is contracted and 0 otherwise.

Given a supplier selection $y=[y_i]\ i=1,2,...m$, then the problem becomes separable, and the following transportation problem must be solved for each scenario s:

Minimize

$$z_S = \sum_{i\in M} \sum_{j\in N} e_{is} c_{ij} x_{ijs} \tag{1}$$

subject to:

$$\sum_{i\in M} x_{ijs} \geq d_{js}, \qquad \forall j \in N \tag{2}$$

$$\sum_{i\in N} x_{ijs} \leq b_i y_i, \qquad \forall i \in M \tag{3}$$

$$x_{ijs} \geq 0, \qquad \forall i \in M \quad \forall j \in N \tag{4}$$

Then, the problem consists of minimizing:

$$F(y) = \sum_{s\in S} p_s \left(\sum_{i\in M} e_{is} f_i y_i + z_s \right) + w \sqrt{\frac{\sum_{s\in S^+} p_s (z_s - E(z))^2}{\sum_{s\in S^+} p_s}} \tag{5}$$

where $S^+ = \{s : z_s - E(z) \geq 0\}$ and $E(z) = \sum_{s\in S} p_S z_S$.

4. Solution proposal

The solution method reported in [1] is a heuristic search based on Benders decomposition paradigm. An initial solution is constructed giving priority to the suppliers of smaller fixed cost and larger production capacity. For each supplier selection, the problem is decomposed into transportation subproblems, one for each scenario. The optimal dual solution for each sub problem is used to find a promissory neighborhood and a local search in the neighborhood is carried out. The method uses several short term tabu memories, to monitor the suppliers used in the visited solutions [3]. As the search goes, the best found solution is updated and continues until finishing the exploration of the neighborhood. When the search stops, a new search begins in the best found solution neighborhood, the procedure continues during a certain number of iterations (50). Figure 1 shows the detailed algorithm for this solution method. As we can see the tabu solution, except for the diversification induced by the short term memory, it does not include a long term diversification process.

4.1. Improving the initial solution construction

The reported solutions to RoCIS problem considers two strategies to select the suppliers that must be incorporated into an initial solution [1, 9]. The first one gives priority to the smaller fixed cost suppliers and greater production capacity the second one incorporates the expected value of the products shipment cost from the selected supplier to *all* plants. The main limitation of the first strategy is that it does not consider the shipment cost, and even though this factor is considered the second one, the mechanism used is too pessimistic. In this work we propose to modify the incorporation mechanism of the shipment cost, *to include only the plants towards which the products shipment from the site of the supplier is less expensive*.

To describe this proposal, let $C_i=\{c_{ij} \mid j=1,2,...n\}$ the shipment costs set from supplier i to all the plants and BCi a threshold cost defined on C_i. Then, the set of plants toward which the products shipment from the site of the supplier i is less expensive, can be defined as:

$$P_i^+ = \left\{ j \in N \middle| c_{ij} < BC_i \right\}$$

Now if c_{min} and c_{max} are the minimum and maximum of the shipment costs in C_i, then BC_i can be modeled as:

$$BC_i = c_{\min} + \alpha(c_{\max} - c_{\min}) \qquad \forall i \in M$$
$$where \qquad 0 < \alpha < 1$$

To include in G_i only the plants in P_i^+, must be defined as:

$$G_i = \frac{f_i + \left(\sum_{s\in S} p_s \left(\sum_{j\in P_i^+} c_{ij} e_{is} \right) \right)}{b_i}, \qquad \forall i \in M$$

The a α value locates the initial solution on different regions of the search space, and it can be used as a long term diversification mechanism in the local search by dynamically changing its values.

4.2. Improving the neighborhood construction

To improve the quality of generated neighbors the mechanism used to determine the relative cost of the three types of movements that are applied to generate the neighborhood (insert, delete and suppliers exchange) is modified. With the current r_i definition, the movements are selected based on their impact on the growth rate of the objective function, which could return in some cases an inappropriate choice. Currently r_i is defined as:

$$r_i = \begin{cases} \dfrac{E(\pi_i)}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

where f_i is the fixed cost of supplier i, $E(\pi_i) = \sum_{s\in S} p_s \pi_{is}$ is the expected dual price of supplier i, p_s is the occurrence probability of scenario s and π_{is} is the supplier i dual price in scenario s.

In opposite would be more appropriate to select the movements *based on the net increase of the objective function generated when the movements are applied*. Then

r_i is redefined as:

$$r_i = \begin{cases} \frac{E(\pi_i) \cdot b_i}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

where b_i is the production capacity of supplier i.

4.3. Improving the neighborhood local search

To improve the local search process is proposed to apply path re-linking on the two best global solutions found when each iteration ends, after the second iteration. The path re-linking strategy used is basically the described in [9]. The algorithm used to perform the process is shown in Figure 1.

Data structures used:

y' prior best global solution found.
y'' actual best global solution found.

For each pair of solutions y' and y'':
Step 1: Determine the $y_\cap$ and $y_\cup$ solutions considering:
a. $y_{\cap i} = 1$ if $y'_i = 1$ and $y''_i = 1$, otherwise $y_{\cap i} = 0$,
b. $y_{\cup i} = 1$ if $y'_i = 1$ or $y''_i = 1$, otherwise $y_{\cup i} = 0$.
Step 2: Determine the S' set of selected suppliers in y' but not selected in y''
Step 3: Determine the S'' set of not selected suppliers in y' and selected in y''
Step 4: Add to the $y_\cap$ solution the suppliers in S' in appropriated order to reach y'
Step 5: Alternate between delete of y' a supplier of S' and add a supplier of S'' until reach the y'' solution
Step 6: Append to y'', one by one the suppliers of set S' in the appropriate order to reach $y_\cup$.

Fig. 1. Path re-linking algorithm.

4.4. Incorporating a long term diversification process

The evaluation of the ROCIS problem objective function requires applying $|S|$ times the linear optimizer. This constitutes the main source of computational cost of the TS method reported in [1]. To reduce this cost, the candidate solutions evaluated and their objective values are recorded. Each time that a candidate solution must be evaluated, the record is reviewed and if the candidate already is recorded, the objective value is retrieved; otherwise, the candidate solution evaluation is performed and recorded. This record can be used to determine the behavior of local search carried out in the neighborhood built in the iteration. If the number of candidate solutions that were recorded is counted in each iteration, then we can determine what neighborhood percentage has already been explored. High values of this percentage indicate that the local search is stangned and that a diversification process is required. When iteration ends, we can determine if the neighborhood percentage that has been explored, exceeds a specified level (50% in our case). If this limit is reached, a new initial solution is built by assigning to the α parameter, a value that has not been used; otherwise, the iteration continues building a new neighborhood.

To determine the sequence of α values used in the diversification process a preliminary experimentation was done. Table 1 shows the experimental results obtained with the improved tabu solution (ITS α), solving the instances with different α values. The first column contains an identifier of the instance solved. The second one contains the best solution reported with AMP [9]. The following 8 columns contain the best solution found and the number of iterations required to find it with each algorithm used (α=0.2, 0.4, 0.6, and 0.8). The last column contains the best global solution found by the algorithms evaluated. For each algorithm, the best solution found is emphasized when this is the best overall solution. As we can see on average approximately 12 iterations are needed for all α? values. As we can see the instances 25 and 29 for α=0.6 are which consume the largest number of iterations without reach the best known, and for α=0.4 the best known are achieved in few iterations. Then if we discard these instances the average iterations to achieve the best solution is reduced as shown in Table 10. As we can see the iterations required for α=0.2 and 0.4 remains around 12, but for α=0.6 it is reduced to 10. Therefore the sequence of α values considered to use in the diversification process is the following:

$$S = \{\alpha_0{=}0.6, \alpha_?{=}0.2, \alpha_2{=}0.4\}$$

This sequence takes advantage of the speed to reach the greatest number of best solutions with α= 0.6 and helps to refine the results with the following values α=0.2, 0.4.

Data structure used:

Supplier selection: $y = [y_1, y_2, \ldots, y_m]$

Hashing solution representation: $H(y) = \sum_{i \in M} y_i 2^i$

List of evaluated solutions: $coded\_sol[H[y]]$

Tabu suppliers lists: *insertion, delete and swap*

Main algorithm

Step 1: Building initial solution y

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

1.2 Build a list of suppliers in ascending order by $G_i = \frac{f_i}{b_i}$

1.3 Build the solution y, selecting suppliers in the sorted list until the sum of the capacities of the selected suppliers is greater than D.

1.4 Determine $F[y]$, solving the distribution subproblems generated for all scenarios.

1.5 Record the solution y and its objective value $F[y]$ in the list of evaluated solutions $coded\_sol[H[y]]$

Step 2: Repeat until reach 50 iterations

2.1 Generate a promising solution neighborhood of y

2.1.1 Determine the expected value of shadow prices (π_{is}) linked to the constraint corresponding to each supplier, in the solution of the $|S|$ sub problems (for all the scenarios).

$$E(\pi_i) = \sum_{s\in S} p_s \pi_{is}$$

2.1.2 Calculate the relative cost of the suppliers $(r_i.)$

$$r_i = \begin{cases} \dfrac{E(\pi_i)}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

2.1.3 For each of the possible insertions, deletions and swaps of suppliers that can be made from the solution, validate the appropriate feasible configuration with respect to the maximum demand D.

2.1.4 Build lists of candidates movements of insertion, deletion and swap, with *the movements identified in the previous step that are not stored in the tabu list for each type of movement*.

- The list of candidates for insertion contains the suppliers with the 3 lowest values of r_i.
- The list of candidates for deletion contains the suppliers with the 3 highest values of r_i.
- The list of candidates for swap contains the suppliers with the $\frac{(m^2-m)}{8}$ lowest values of r_j - r_i corre-sponding to the swap between supplier i by supplier j in configuration y.

2.2 Do local search

2.2.1 For each configuration y' generated from the movements of the candidate lists of insertion, deletion and swapping:

2.2.1.1 The suppliers involved in the movement used to generate the configuration y' are: appended to the insertion tabu list (if the movement was for deletion), or removed (if the movement was to insertion). This two tabu lists are used too when a swap movement is applied, cnsidering that a swap movement requires a deletion and an insertion. The number of iterations during which a supplier involved in a movement is considered tabu are: $\frac{m}{3}$ for insertions and eliminations and $\frac{m(1)-m}{16}$ for swaps.

2.2.1.2 If the solution y' is already saved in the list of evaluated solutions $coded\_sol[H[y]]$, its objective value $F[y']$ is retrieved, otherwise $F[y']$ is calculated and appended to the list.

2.2.1.3 Update the best solution found y_{best}.

2.2.2 $y = y_{best}$

Fig. 2. Tabu solution algorithm TS [1].

Data structures used:

Supplier selection: $y = [y_1, y_2, \ldots, y_m]$

Hashing solution representation: $H(y) = \sum_{i\in M} y_i 2^i$

List of evaluated solutions: $coded\_sol[H[y]]$

Tabu suppliers lists: *insertion, delete and swap*

As part of the diversification mechanism the following variables and constants are defined:

RC: The counter of candidate solutions generated in the current neighborhood already registered in the hashing list.

The number of candidate solutions in the current neighborhood:

$$\text{NCS} = 3 + 3 + \frac{(m^2 - m)}{8}$$

The percentage of the current neighborhood had already been reviewed in previous iterations (stagnation level observed):

$SL = RC/NCS$

Function InitialSolution(α)

Step 1: Calculate $D = \max_{s\in S}(\sum_{j\in N} d_{js})$

Step 2: Build a suppliers list in ascending sorted by

$$G_i = \frac{f_i + \left(\sum_{s\in S} p_s \left(\sum_{j\in P_i^+} c_{ij} e_{is}\right)\right)}{b_i}, \qquad \forall i \in M$$

where $P_i^+ = \{j \in N | c_{ij} < CF_i\}$ and

$$CF_i = c_{\min} + \alpha(c_{\max} - c_{\min}) \qquad \forall i \in M$$

where $0 < \alpha < 1$

Step 3: Build the solution y, selecting suppliers in the sorted list until the sum of the capacities of the selected suppliers is greater than D.

Step 4: Determinate $F[y]$, solving the transportation sub problems generated in all scenarios.

Step 5: Record the y solution and its objective value $F[y]$ in the list of evaluated solution $coded\_sol[H[y]]$

Step 6: Return the actual solution y.

Main algorithm

Step 1: $change = 0$

Step 2: $y = SoluciónInicial(\alpha=0.6)$

Step 3: Repeat until 50 iterations

3.1 Generate a promising solution neighborhood of y

2.1.1 Determine the expected value of shadow prices (π_{is}) linked to the constraint corresponding to each supplier, in the solution of the 27 sub problems (for all the scenarios).

$$E(\pi_i) = \sum_{s\in S} p_s \pi_{is}$$

3.1.2 Calculate the suppliers relative cost $(r_i.)$

$$r_i = \begin{cases} \dfrac{E(\pi_i)\cdot b_i}{f_i} & si \quad E(\pi_i) < 0 \\ f_i & si \quad E(\pi_i) = 0 \end{cases}$$

3.1.3 For each of the possible insertions, deletions and swaps of suppliers that can be made from the solution, validate the feasibility of configuration with respect to the maximum demand D.

3.1.4 Build lists of candidate movement for insertion, deletion and swap, with *aspiration criteria*.
- The list of candidates for insertion contains the suppliers with the 3 lowest values of r_i.
- The list of candidates for deletion contains the suppliers with the 3 highest values of r_i.
- The list of candidates for swap contains the suppliers with the $\frac{(m^2-m)}{8}$ lowest values of $r_j - r_i$ corresponding to the swap between supplier i by supplier j in configuration y

3.2 Local search process

3.2.1 For each configuration y' generated from the movements of the candidate lists of insertion, deletion and swapping:

3.2.1.1 The suppliers involved in the movement used to generate the configuration y' are: appended to the insertion tabu list (if the movement was for deletion), or removed (if the movement was to insertion). This two tabu lists are used too when a swap movement is applied, cnsidering that a swap movement requires a deletion and an insertion. The number of iterations during which a supplier involved in a movement is considered tabu are: $\frac{m}{3}$ for insertions and eliminations and $\frac{m(1)-m}{16}$ for swaps.

3.2.1.2 If the solution y' is already saved in the list of evaluated solutions, its objective value $F[y']$ is retrieved and the repeated candidate solutions counter RC is incremented, otherwise $F[y']$ is calculated and appended to the list.

3.2.1.3 Update the best solution found y_{best}.

3.2.2 Path relinking process

3.2.2.1 From the second iteration, update the two best solutions found.

3.2.2.2 Apply path relinking to the two best solutions found and update y_{best}

3.2.3 Diversification process

3.2.3.1 Calculate the stagnation level $SL = RS/NCS$

3.2.3.2 If $(SL > 0.5)$
then
 $change = change + 1$
 If $(change = 1)$ then
 $y = \text{InitialSolution}(\alpha = 0.2)$
 If $(change = 2)$ then
 $y = \text{InitialSolution}(\alpha = 0.4)$
Else
 $y = y_{best}$

Fig. 3. Improved tabu solution with diversification ITSD algorithm.

Table 1. Performance of the improved tabu solution ITS (with α= 0.2, 0.4, 0.6, 0.8 and 50 iterations).

	AMP	ITS 0.2		ITS 0.4		ITS 0.6		ITS 0.8		Best
1	33178.63458	33178.63458	12	33178.63458	10	33178.63458	9	33178.63458	8	33178.63458
2	44181.48210	44181.48210	12	44181.48210	10	44181.48210	9	44181.48210	10	44181.48210
3	39558.82437	39558.82437	12	39558.82437	9	39558.82437	7	39558.82437	6	39558.82437
4	47120.47639	47120.47639	10	47120.47639	7	47120.47639	9	47120.47639	16	47120.47639
5	41515.93392	41515.93392	12	41515.93392	8	41515.93392	10	41515.93392	13	41515.93392
6	41285.57388	41285.57388	20	41285.57388	17	41285.57388	5	41285.57388	13	41285.57388
7	42015.04585	42015.04585	7	42015.04585	12	42015.04585	7	42015.04585	11	42015.04585
8	55627.07483	55627.07483	10	55627.07483	9	55627.07483	11	55627.07483	7	55627.07483
9	46055.98672	46055.98672	6	46055.98672	5	46055.98672	10	46055.98672	9	46055.98672
10	57188.41647	57188.41647	32	57188.41647	28	57188.41647	2	57188.41647	3	57188.41647
11	60692.58875	60692.58875	6	60692.58875	11	60692.58875	9	60692.58875	20	60692.58875
12	55603.79858	55603.79858	11	55617.16356	9	55603.79858	10	55617.16356	23	55603.79858
13	67389.80329	67389.80329	14	68158.76152	13	68158.76152	9	68158.76152	7	67389.80329
14	65420.80667	65595.87523	9	65427.00810	24	65420.80667	10	65420.80667	10	65420.80667
15	78184.02415	78184.02415	6	78184.02415	7	78184.02415	21	78184.02415	36	78184.02415
16	38094.86669	37809.00955	22	37809.00955	12	37809.00955	11	37820.65015	11	37809.00955
17	34109.31059	34109.31059	10	34109.31059	12	34109.31059	9	34109.31059	9	34109.31059
18	34127.48022	33814.09910	10	33814.09910	9	33814.09910	9	33814.09910	12	33814.09910
19	40558.79816	40558.79816	9	40558.79816	11	40558.79816	11	40570.84487	30	40558.79816
20	32210.96759	31496.84804	11	31496.84804	12	31496.84804	9	31496.84804	12	31496.84804
21	41551.65039	41741.15551	11	41527.77087	7	41741.15551	7	41741.15551	10	41527.77087
22	38833.67675	38833.67675	8	38833.67675	25	38833.67675	11	38833.67675	10	38833.67675
23	44391.63693	44391.63693	7	44391.63693	12	44391.63693	13	44391.63693	11	44391.63693
24	41831.94585	41831.94585	18	41831.94585	6	41831.94585	6	41831.94585	28	41831.94585
25	53709.18863	53709.18863	7	53709.18863	8	54180.96605	43	53709.18863	10	53709.18863
26	61377.26091	61377.26091	11	61377.26091	10	61377.26091	8	61377.26091	19	61377.26091
27	69464.05787	69541.17681	27	69496.30247	9	69464.04654	26	69464.04654	45	69464.04654
28	75482.59766	75482.59766	5	75482.59766	20	75952.11365	4	75482.59766	20	75482.59766
29	61818.89140	62170.32689	28	61818.89140	5	61963.37426	44	61851.60940	30	61818.89140
30	68193.73131	68073.37865	11	68193.72023	24	68193.72023	18	68292.71756	17	68073.37865

As it is showed in Figure 3, the InitialSolution(α) function is used to diversify the search and is initially invoked with α_0. When the observed stagnation level reaches 50%, after a complete exploration of the current neighborhood, the InitialSolution(α) function is invoked with $\alpha_?$. The search continues and when the stagnation level reaches 50% again, the InitialSolution(α) function is called with α_2. Then the search continues until finish, without changing the α value.

Figure 3 shows the detailed proposed tabu solution algorithm, which incorporates the improvements in the construction of the initial solutions, in the construction of neighborhoods, in the local search, and the long term diversification mechanism.

5. Experimental results

The experiments were done in a computer Dell Optiplex 160L with a Pentium IV processor to 2.4 GHZ and 1 GB ram. The source code was compiled using Visual C 6.0 and the operating system Windows XP. For the solution of the transportation sub problems LINDO API 2.0 was used. To evaluate the performance of the algorithms, the larger instances reported in [9] were used. The instances were generated with 20 plants, 40 suppliers and 27 scenarios, and constitute a representative sample of instances with the same size and different hardness degrees. As the optimal solutions for these instances are not known, the results obtained in this work are compared with the best solutions reported in [9]. To evaluate the impact of the proposed improvements on the TS performance, four types of experiments were carried out.

In the first one the proposed improvements for the initial construction, the construction of the neighborhood and local search were evaluated. Table 1 shows the results obtained with the improved tabu solution (ITS α), for different values. The table shows that the improved TS found better solutions than those reported for instances 16, 18, 20, 21, 27 and 30. As we can observe the best global solutions found by AMP are also found by ITS using one or more of the α values. However there is not a single value which allows finding the best global solution for all instances. Experimental evidence confirms that the value operates as a diversification mechanism on the search process.

Table 2 shows a summary of the experimental results, including: the average cost of the found solutions, the number of overall best solutions found, the error rate over the average cost of best solution and the average time used to solve each instance (in CPU seconds).

In other hand, in Table 1 we can observe that 4 ITS algorithms obtains the best known solutions for 19 instances, and similarly 3 ITS algorithms for 3 instances, 2 ITS algorithms for 4 instances and 1 ITS algorithm for 4 instances. Then if four groups of instances are considered: I_1, I_2, I_3 and I_4, where the group I_n contains all the instances for which n ITS algorithms obtains the best known solutions. We could consider that for $n>m$, the instances in I_n are easiest than the instances in I_m. Then the question is ¿ a structural and landscape analysis of the instances can help us to explain the relative hardness observed?

In the second experiment the structural analysis of the instances was done. For all instances the sparsity, the variation coefficient and the skewness of the instance parameters (shipment cost matrix, fixed cost vector, capacity supplier vector, plants demand matrix and currency rate matrix) were calculated. The sparsity measures the percentage of parameter structure elements that are equal to zero; the main interest in this measure is that according to Mitchell and Borchers, it has a strong influence on algorithm behavior [11]. The *variation coefficient* (VC) is defined as σ/X where σ is the standard deviation and X the mean of the structure elements. VC gives an estimate of the variability of the structure elements, independent of their size. The *skewness* is the third moment of the mean normalized by the standard deviation; it gives an indication of the degree of asymmetry of the structure elements. In the experiment all the instances were considered grouped in I_1, I_2, I_3 and I_4. Then the sparsity, the variation coefficient and skewness were calculated for the five components of each instance: shipment cost matrix, fixed cost vector, capacity supplier vector, plants demand matrix and currency rate matrix. For all the instances and components the observed sparsity percentage was zero. Table 3 contains the obtained results for the variation coefficient and Table 4 the results for the skewness. As we can observe the four instances groups shows a similar structure, because the differences between the values of the variation coefficient and of the skewness are minimal. In the third experiment a ruggedness analysis of the landscape was done. The central idea of the landscape analysis in combinatorial optimization is to represent the space searched by an algorithm as a landscape formed by all feasible solutions and the objective value assigned to each solution [12]. The information generated with the landscape analysis is used to gain knowledge about: the search space characteristics and their relation with the behavior of local search or metaheuristic algorithms [13],[14], problem or problem instance hardness [15], [16], or useful parameterizations of local search algorithms [17]. A search landscape is considered rugged if there is a low correlation between neighboring points. To measure this correlation a *random walk* of length m, is performed in the search landscape to interpret the resulting series of m points $\{f(x_t), t=1,\ldots, m\}$ as a time series. The autocorrelation $r(s)$ of the points in the series that are separated by s steps is defined as:

$$r(s) = \frac{1}{\sigma^2(f)(m-s)} \sum_{t=1}^{m-s} (f(x_t) - \bar{f})(f(x_{t+s}) - \bar{f})$$

where $\sigma^2(f)$ and $\bar{f}$ are the variance and the mean of the points in the series. Now the *search landscape correlation length* is defined as:

$$l = -\frac{1}{\ln(|r(1)|)}$$

where $|r(1)| \neq 0$. Then the lower is the value of l, the more rugged is the landscape [18].

Previously to the determination of the search space correlation length (l) values for the instances in the considered groups, we must determine the length of the random walk to be applied. For this purpose were calculated five times the average of the l values of all the instances with a random walk length given. The obtained results,

Table 2. Comparative summary of the performance of the improved tabu solution ITS with 50 iterations.

	AMP	TS	ITS 0.2	ITS 0.4	ITS 0.6	ITS 0.8	Best
Val.	50359.15105	51730.25	50337.79759	50341.93472	50383.51267	50352.94571	50310.56363
# Bests	24	1	26	25	24	23	30
Dev.	0.10%	2.93%	0.05%	0.06%	0.14%	0.08%	0%
CPU secs	218.23	381.25	374.22	392.92	371.75	368.97	

Table 3. Structural information of the instances groups I_1, I_2, I_3 and I_4 (variation coefficient).

Shipment cost	I_1	I_2	I_3	I_4
Min	0.45	0.46	0.46	0.45
Median	0.48	0.48	0.47	0.47
Max	0.48	0.49	0.49	0.50
Mean	0.47	0.48	0.47	0.47
Supplier capacity	I_1	I_2	I_3	I_4
Min	0.10	0.11	0.11	0.10
Median	0.12	0.11	0.12	0.12
Max	0.12	0.12	0.12	0.13
Mean	0.12	0.11	0.12	0.11
Fixed cost	I_1	I_2	I_3	I_4
Min	0.10	0.11	0.11	0.10
Median	0.12	0.11	0.12	0.12
Max	0.12	0.12	0.12	0.13
Mean	0.12	0.11	0.12	0.11
Demand	I_1	I_2	I_3	I_4
Min	0.15	0.15	0.16	0.15
Median	0.16	0.16	0.16	0.16
Max	0.16	0.16	0.16	0.17
Mean	0.16	0.16	0.16	0.16
Exchange rate	I_1	I_2	I_3	I_4
Min	0.10	0.10	0.10	0.10
Median	0.10	0.10	0.10	0.10
Max	0.10	0.10	0.10	0.10
Mean	0.10	0.10	0.10	0.10

Table 4. Structural information of the instances groups I_1, I_2, I_3 and I_4 (skewness).

Shipment cost	I_1	I_2	I_3	I_4
Min	0.08	0.13	0.17	-0.03
Median	0.17	0.17	0.25	0.19
Max	0.23	0.25	0.38	0.35
Mean	0.16	0.18	0.27	0.17
Supplier capacity	I_1	I_2	I_3	I_4
Min	-0.10	0.00	-0.30	-0.56
Median	-0.03	0.13	0.00	-0.05
Max	0.50	0.49	0.22	0.64
Mean	0.09	0.19	-0.03	-0.01
Fixed cost	I_1	I_2	I_3	I_4
Min	-0.10	0.00	-0.30	-0.56
Median	-0.03	0.13	0.00	0.08
Max	0.50	0.49	0.22	0.64
Mean	0.09	0.19	-0.03	0.09
Demand	I_1	I_2	I_3	I_4
Min	-0.05	-0.10	-0.06	-0.12
Median	-0.04	-0.01	0.02	-0.03
Max	0.03	0.12	0.05	0.10
Mean	-0.02	0.00	0.00	-0.01
Exchange rate	I_1	I_2	I_3	I_4
Min	-0.03	-0.08	0.05	-0.06
Median	0.00	-0.01	0.06	0.02
Max	0.09	0.05	0.07	0.06
Mean	0.01	-0.01	0.06	0.01

Table 5. Average l value obtained in five random walks with two lengths: 1000 and 50000 steps.

Walk	l average (Length=1000)	l average (Length=50000)
1	0.25	0.19
2	0.26	0.18
3	0.28	0.20
4	0.28	0.19
5	0.25	0.20

Graph 1. Average l value obtained in five random walks with two lengths: 1000 and 50000 steps.

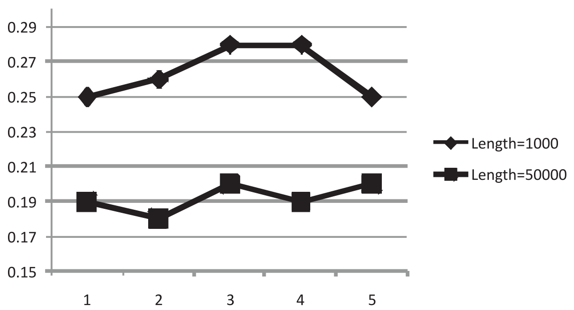

Table 6. Minimum, median, maximum and mean of the search landscape correlation length (l value) obtained with a random walk of 50000 steps, for the groups I_1, I_2, I_3 and I_4.

	Search landscape correlation length (*l* value)			
	I_1	I_2	I_3	I_4
Min	0.17	0.16	0.13	0.14
Median	0.20	0.18	0.16	0.19
Max	0.22	0.21	0.17	0.23
Mean	0.19	0.18	0.15	0.19

Table 7. Performance of the improved tabu solution with diversification ITSD and 50 iterations.

	AMP	ITS 0.2		ITS 0.4		ITS 0.6		ITSD		Best
1	33178.63458	33178.63458	12	33178.63458	10	33178.63458	9	33178.63458	9	33178.63458
2	44181.48210	44181.48210	12	44181.48210	10	44181.48210	9	44181.48210	9	44181.48210
3	39558.82437	39558.82437	12	39558.82437	9	39558.82437	7	39558.82437	7	39558.82437
4	47120.47639	47120.47639	10	47120.47639	7	47120.47639	9	47120.47639	9	47120.47639
5	41515.93392	41515.93392	12	41515.93392	8	41515.93392	10	41515.93392	10	41515.93392
6	41285.57388	41285.57388	20	41285.57388	17	41285.57388	5	41285.57388	5	41285.57388
7	42015.04585	42015.04585	7	42015.04585	12	42015.04585	7	42015.04585	7	42015.04585
8	55627.07483	55627.07483	10	55627.07483	9	55627.07483	11	55627.07483	11	55627.07483
9	46055.98672	46055.98672	6	46055.98672	5	46055.98672	10	46055.98672	10	46055.98672
10	57188.41647	57188.41647	32	57188.41647	28	57188.41647	2	57188.41647	2	57188.41647
11	60692.58875	60692.58875	6	60692.58875	11	60692.58875	9	60692.58875	9	60692.58875
12	55603.79858	55603.79858	11	55617.16356	9	55603.79858	10	55603.79858	10	55603.79858
13	67389.80329	67389.80329	14	68158.76152	13	68158.76152	9	67433.60472	19	67389.80329
14	65420.80667	65595.87523	9	65427.00810	24	65420.80667	10	65420.80667	10	65420.80667
15	78184.02415	78184.02415	6	78184.02415	7	78184.02415	21	78184.02415	18	78184.02415
16	38094.86669	37809.00955	22	37809.00955	12	37809.00955	11	37809.00955	11	37809.00955
17	34109.31059	34109.31059	10	34109.31059	12	34109.31059	9	34109.31059	9	34109.31059
18	34127.48022	33814.09910	10	33814.09910	9	33814.09910	9	33814.09910	9	33814.09910
19	40558.79816	40558.79816	9	40558.79816	11	40558.79816	11	40558.79816	11	40558.79816
20	32210.96759	31496.84804	11	31496.84804	12	31496.84804	9	31496.84804	9	31496.84804
21	41551.65039	41741.15551	11	41527.77087	7	41741.15551	7	41527.77087	30	41527.77087
22	38833.67675	38833.67675	8	38833.67675	25	38833.67675	11	38833.67675	11	38833.67675
23	44391.63693	44391.63693	7	44391.63693	12	44391.63693	13	44391.63693	13	44391.63693
24	41831.94585	41831.94585	18	41831.94585	6	41831.94585	6	41831.94585	6	41831.94585
25	53709.18863	53709.18863	7	53709.18863	8	54180.96605	43	53709.18863	38	53709.18863
26	61377.26091	61377.26091	11	61377.26091	10	61377.26091	8	61377.26091	8	61377.26091
27	69464.05787	69541.17681	27	69496.30247	9	69464.04654	26	69464.04654	49	69464.04654
28	75482.59766	75482.59766	5	75482.59766	20	75952.11365	4	75482.59766	15	75482.59766
29	61818.89140	62170.32689	28	61818.89140	5	61963.37426	44	61818.89140	40	61818.89140
30	68193.73131	68073.37865	11	68193.72023	24	68193.72023	18	68073.37865	24	68073.37865

Table 8. Comparative summary of the performance of the improved tabu solution with diversification ITSD and 50 iterations.

	AMP	TS	ITS 0.2	ITS 0.4	ITS 0.6	ITS 0.8	Best
Val.	50359.15105	51730.25	50337.79759	50341.93472	50383.51267	50312.02403	50310.56363
# Bests	24	1	26	25	24	29	30
Dev.	0.10%	2.93%	0.05%	0.06%	0.14%	0.003%	0%
CPU secs	218.23	381.25	374.22	392.92	371.75	386.94	

with random walk lengths of 1000 and 50000 steps, are showed in Table 5 and Graph 1. As we can observe, with 1000 steps the average of the l values varies from 0.25 to 0.28 and for 50000 steps varies from 0.18 to 0.20. Given the high resource consumption required to solve the ROCIS instances, we consider that with 50000 steps the average of the l values shows an appropriated precision level and stability. Now we calculate the l values for the instances in each group using a random walk with 50000 steps. Table 6 shows the minimum, median, maximum and the mean of the average l values for each group (I_1, I_2, I_3 and I_4). As we can observe that do not exist a significant difference respect to the landscape ruggedness generated for the random walk with the instances of the different groups. All the average l values are very similar and closer to zero. The random walk algorithm seems perceive that all the instances have a high hardness level regardless of the group that they belong. Then it seems more appropriate to incorporate in the tabu solution a long term diversification process to avoid to get stuck in the local optimums.

In the last experiment the performance of the improved tabu solution with diversification (ITSD) was evaluated. As we can see in the section 4.4. (Incorporating a diversification process), the sequence of a values used in the diversification process is the following:

$$S = \{\alpha_0=0.6, \alpha_?=0.2, \alpha_2=0.4\}$$

The ITSD algorithm starts with α=0.6 and the first time that stagnation is detected, is changed to 0.2 and in the second one α switches to 0.4. The search continues with the last one α value, until reaching the stopping condition

Table 7 shows the comparative performance of the ITSD algorithm with respect to ITS algorithm (for α=0.2, 0.4 and 0.6) and AMP algorithm. As we can see, the solution with diversification is able to find 29 of the 30 global best solutions, outperforming all the algorithms evaluated. Table 8 shows a summary of the experiment results. Tables 2 and 8 shows that with 50 iterations ITS (with α= 0.2 and 0.4) and ITSD are better in quality than AMP, but the last one is better than ITS and ITSD in efficiency.

To reduce the resources consumption of ITS and ITSD, the number of iterations was reduced to 30. Table 9 shows the obtained results and we can observe that now ITSD outperfoms in quality and efficiency to ITS (for α=0.2) and to AMP.

Table 9. Comparative summary of the performance of the improved tabu solution with diversification ITSD and 30 iterations.

	AMP	ITS 0.2	ITSD
Val.	50,359.15105	50,344.08684	50,310.56363
# Bests	22	25	26
Dev.	0.10%	0.066%	0.056%
CPU secs	218.23	235.76	218.20

Table 10. Average iterations needed for the improved tabu to reach the best solution without instances 25 and 29.

	ITS 0.2	ITS 0.4	ITS 0.6
Average Iterations until the best reached	12.10714	12.42857	10.00000

6. Conclusions and future work

This paper approaches the robust capacitated international sourcing problem (RoCIS) which consists of selecting a subset of suppliers with finite capacity, from an available set of potential suppliers internationally located. The tabu solution proposed in [1] consists of three phases: build an initial solution, create a neighborhood of promising solutions and perform an extensive search in the neighborhood. In this work the construction of the initial solution, the construction of the neighborhood, and the local search were improved. Also the intensification and diversification balance of the tabu solution was improved, incorporating a long term diversification process. Experimental evidence shows that the improved tabu solution with diversification outperforms the best solutions reported for six of the instances considered, increases 18% the number of best solutions found and reduces 44% the deviation from the best solution found, respect to the best algorithm solution reported.

Future work includes improving the efficiency of the proposed solution, incorporating different diversification mechanisms and stopping conditions based on the stagnation detection.

ACKNOWLEDGMENTS

Authors thank the support received from Tecnológico de Monterrey, Consejo Nacional de Ciencia y Tecnología (CONACYT) and Consejo Tamulipeco de Ciencia y Tecnología (COTACYT) through projects CAT128, CONACYT-67032 and TAMPS-2007-C15-106096 respectively, for the research reported in this paper.

AUTHORS

Héctor Joaquín Fraire Huacuja, Guadalupe Castilla Valdez - Instituto Tecnológico de Ciudad Madero. 1° de Mayo y Sor Juana I. de la Cruz S/N, Ciudad Madero Tamaulipas, México CP. 89440.

José Luis González-Velarde - Centro de Calidad y Manufactura, Tecnológico de Monterrey, Monterrey Nuevo León, México.

* Corresponding author

References

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

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

[3] Hodder J.E., Jucker J.V., "Plant Location Modeling for the Multinational Firm". In: *Proceedings of the Academy of International Business Conference on the Asia-Pacific Dimension of International Business*, Honolulu, December 1982, pp. 248-258.

[4] Hodder J.E., Jucker J.V., "A Simple Plant-Location Model for Quantity-Setting Firms subject to Price Uncertainty", *European Journal of Operational Research*, vol. 21, 1985.

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

[6] Louveaux F.V., Peters D., "A dual-based procedure for stochastic facility location", *Operations Research*, vol. 40, no. 3, 1992, pp. 564-573.

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

[8] Kouvelis P., Yu G., *Robust Discrete Optimization and its Applications*, Dordrecht: Kluwer Academic Publishers, 1997.

[9] González-Velarde J.L., Martí R., "Adaptive Memory Programming for the Robust Capacitated International Sourcing Problem", *Computers and Operations Research*, vol. 35, no. 3, 2008, pp. 797-806.

[10] Glover F., Laguna M., *Tabu Search*, Kluwer Academic Publishers, 1997.

[11] Mitchell J.E., Borchers B., "Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm". In: H. L. Frenk *et al.*, editor, *High Performance Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 349-366.

[12] Merz P., Freisleben B., "Fitness landscapes and memetic algorithm design". In: D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, McGraw-Hill, London, 1999, pp. 245-260.

[13] Boese K. D., *Models for Iterative Global Optimization*. PhD thesis, University of California, Computer Science Department, Los Angeles, CA, USA, 1996.

[14] Stutzle T., Hoos H.H., MAX-MIN Ant System. *Future*

Generation Computer Systems, vol. 16(8), 2000, pp. 889-914.

[15] AngelE., Zissimopoulos V., "On the classification of NP-complete problems in terms of their correlation coefficient". *Discrete AppliedMathematics*, no. 99, 2000, pp. 261-277.

[16] Stadler P.F., Schnabl W., "The landscape of the travelling salesman problem", *Physics Letters A*, 161, 1992, pp. 337-344.

[17] Angel E., Zissimopoulos V., "Autocorrelation coefficient for the graph bipartitioning problem", *Theoretical Computer Science*, no. 191 1998, pp. 229-243.

[18] Weinberger E. D., "Correlated and uncorrelated fitness landscapes and how to tell the difference", *Biological Cybernetics*, no. 63, 1990, pp. 325-336.