

VEHICLE ROUTING FOR PUBLIC TRANSPORT WITH ADAPTED SIMULATED ANNEALING

D.A. RODRÍGUEZ^{†‡§}, A.C. OLIVERA[§] and N.B. BRIGNOLE^{‡§}

[†] *Facultad de Ciencias Exactas, Universidad Nacional de Salta (UNSa), Av. Bolivia 5150, Salta, Argentina.*

[‡] *Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET.*

Camino La Carrindanga Km. 7, 8000 Bahía Blanca, Argentina.

[§] *Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC), Departamento de Ciencias e Ingeniería de la Computación (DCIC), Universidad Nacional del Sur (UNS), Av. Alem 1253, Bahía Blanca, Argentina.*

Abstract— This paper presents an Adapted Simulated Annealing (ASA) algorithm to solve an instance of the vehicle routing problem (VRP): the intercity public transport problem (IPTP). This combinatorial optimization problem was effectively solved by means of a robust method. Its performance was achieved thanks to the incorporation of an auxiliary memory and a novel choice of the neighbours. The model is based on initial random solutions capable of generating appropriate bus routes and frequencies in a large solution space. We have established a search strategy that provides excellent responses at the process level. The intercity line linking Bahía Blanca and Punta Alta was chosen with the intention of evaluating ASA performance. The real traffic-behavior has been represented by means of the simulation software called SUMO. The computational results clearly indicate that the proposed approach constitutes an improvement in the ability to search for high quality solutions that facilitates the convergence.

Keywords— Vehicle Routing Problem, Intercity Public Transport, Simulated Annealing, Scientific Computing, Optimization.

I. INTRODUCTION

Vehicle Routing Problems (VRPs) aim at the determination of the optimal set of routes to be covered by a fleet of vehicles in order to service multiple customers (Toth and Vigo, 2002). Several methods have been proposed to solve problems related with the VRP (Maffioli, 2003; Dondo *et al.*, 2003; Giaglis *et al.*, 2004; Sa'adah and Paecher, 2004; Dooley *et al.*, 2005; Alvarenga *et al.*, 2007). In particular, the Intercity Public Transport Problem (IPTP) can be modeled as a variant of the VRPs (Rodríguez *et al.* 2011).

Transport Logistics is a vital aspect for the success of a region. This activity has a significant influence on the economy, the society and the environment. The intercity public transport is essential for job creation, touristic boosting and trade increase. It helps to reduce road congestions and transit accidents. Then, it is worthwhile to improve this service by reducing operation costs, thus being advantageous to both the operator and the users. Besides, when the measures become obsolete and inefficient, it is necessary to reconsider them. New ways of mobility, efficiency, cost and convenience, should be contemplated in order to meet existing demands and

mobility needs. In this work a systematic analysis to define vehicle frequencies, routes and stops is provided, being useful to achieve a cheap rational transport system.

The economic importance of people's mobility creates the need to develop techniques and strategies that provide competitive solutions, solving the Transport Problems with a reasonable speed. In this regard, there are some important contributions that deserve highlighting. Yan and Chen (2002) have proposed a space-time network that reflects the movement of vehicles and passengers. Mauttone and Urquhart (2009) have developed a heuristic algorithm in order to generate initial solutions that improve the process of solving the Transit Network Design Problem. Olivera *et al.* (2009) have presented a novel method in order to evaluate possible solutions to the Bus Network Scheduling Problem. Xinchao (2011) has addressed the design of routes and the adjustment of frequencies simultaneously by using a genetic algorithm hybridized with a neighbourhood-search heuristics. Finally, Cipriani *et al.* (2012) has proposed a set of heuristics that solve a Bus Network Design Problem in an area with a multimodal transportation system.

In this work we are presenting a study that will allow businessmen to make decisions that provide benefits in transport services. This treatment is focused on optimizing the route and frequency to be followed by intercity public transport buses. Guihaire and Hao (2008) addressed similar goals concerning urban transport. In contrast, by using a traffic simulation software, in our model we have included the features that differentiate intercity from urban transports. Intercity transport is mainly characterized by the following features:

- It usually has fewer and more remote stops;
- There is less interaction between vehicles resulting in fewer unplanned stoppages;
- There are route segments with special features (e.g. higher speed limits and more lines).

Ortúzar and Willumsen (2011) presented an introduction to general-purpose transport models in urban scenarios. In particular, zoning and network building were discussed. Mauttone (2005) highlighted that the adopted level of detail is one of the most important decisions to be made at the beginning of process modeling. There are some widely accepted criteria in order to

divide the area under study into zones. These standards are focused on general-purpose transport models, which generally work on the bases of several stages: generation, distribution, modal division and assignment. Zoning, which is defined under these criteria, constitutes an input for general-purpose transport models and is also involved in both collecting and processing of transport data. Moreover, Martinez *et al.* (2007) stated that in most of the transport planning studies, one of the first steps is the definition of a zoning scheme in order to divide the study area and consequently, the corresponding space is discretized.

Planning vehicle operations involves a large number of decisions. In addition, the number of possible plans grows exponentially with the size of the problem. Then, it is often nonviable the resolution with exact methods. In consequence, it becomes necessary to use metaheuristics. Although these alternative procedures do not guarantee the optimal solution, they provide quality solutions within a reasonable computing time (Colorni *et al.*, 1996). In particular, in this paper we have combined an exhaustive search by Simulated Annealing (SA) with robust traffic-simulation software. With a view to solving simple versions of an IPTP), we have recommended the use of an SA procedure, instead of choosing Genetic Algorithms, since SA works faster (Rodriguez *et al.*, 2011). An evolution of that introductory study is presented in this work. We have implemented an auxiliary memory whose content was adapted to the current best solution, together with a dynamic neighbourhood search that speeds up the exploitation of search space in the final stage of process. In this way, we have achieved greater efficiency in our model. For model testing, we have implemented the algorithm in MALLBA architecture (Alba *et al.*, 2007), performing the simulations with SUMO (Simulation of Urban MObility) software (Behrisch *et al.*, 2011).

The case study analyzed in this paper corresponds to an intercity line connecting Bahía Blanca and Punta Alta, both of these cities located in Argentina (see <http://www.gpsurbana.com/>, choose line 319). The current service has major defects that induce people to live in the nearest city - usually in Bahía Blanca - to be able to work or study. Therefore, this study is a contribution to decentralization. Our approach will help to achieve a flexible transport system that will encourage people to live in Punta Alta, giving them the possibility of working or studying in Bahía Blanca without any major transportation problems. The remainder of this paper is organized as follows. Section 2, presents the general formulation of the mathematical problem, followed by the case study particularization. In Section 3, we describe the adaptation of the SA technique. The main details of the model implementation are given in Section 4. Next, the results of the computational study are discussed in Section 5. Finally, Section 6 states the conclusions and highlights future research directions.

II. PROBLEM STATEMENT

An intercity line is described as a sequence of stops that

the buses must go by on a certain frequency. Our goal is

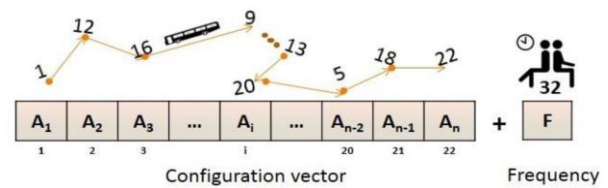


Figure 1: Solution Representation

to obtain an algorithm that finds the order of the stops to be visited and the output frequencies so that costs are minimized. In this paper the costs we have considered are: travel time, distance traveled and the number of detentions for each of the buses.

Representation: In these problems the structure of a solution representation requires close attention in order to try to avoid complications in its handling during the process. Prospective solutions should be feasible, i.e. available for its use.

In our problem, we have represented a solution as a configuration vector together with a frequency value. The configuration vector is a vector of integers that identify each of the existing stops. The frequency value is an integer that was incorporated for each solution vector in order to represent the bus frequency required for that configuration.

For example, in Fig. 1 the buses that go by the route $\{A_1, A_2, \dots, A_n\}$ should depart every F minutes. Besides, we also maintain a vector $\mathbf{v} \in \mathbb{Z}^n$ that preserves the information necessary to identify the city that each stop belongs to.

Fitness function: To evaluate each of the solutions proposed by the algorithm we have designed a fitness function (Eq. 1), which provides key properties through simulation. They are key aspects because they directly affect the quality of the solution. We have employed a simulator called SUMO in order to obtain the values associated with fitness. By means of SUMO it was possible to define important properties, such as acceleration and deceleration of a vehicle, driver's ability, maximum vehicle speed, direction of the streets and driver's timeouts. In this way, accurate simulations of traffic behavior were obtained.

Equation (1) shows how the fitness value Z is evaluated:

$$\text{Min } Z = \sum_{i=1}^{NB+B} \frac{T_i + R_i + S_i}{B} \quad (1)$$

where B is the number of buses that reach their destinies, NB is the number of buses that do not arrive at their destinies, T_i is the total travel time of an i -th bus (b_i) in seconds. R_i is the route length for b_i in meters. S_i is the number of times that b_i must stop (unplanned stops).

Case study parameters: The proposed procedure has been implemented on a real-life network, joining Bahía Blanca (1) and Punta Alta (2). Then, the additional vector \mathbf{v} whose components are identifying the city the stops belong to will have values either 1 or 2.

It has an approximate service demand of 2.000 pas-

sengers per day. Most of these trips are made for work or study. There are universities and large companies settled in Bahía Blanca. So, the buses are mainly used by people commuting from Punta Alta to Bahía Blanca. The distance between them is approximately 30 km. and the existing stops are 22. We have assumed that a bus arrives at its destination when it has already gone by all the existing stops during the simulation time.

The output frequency of the buses should be between 20 minutes and 50 minutes. These frequency limits were accorded by the company transport with municipal authorities regulating the service. In this paper for the runs of the proposed algorithm the simulation time set was 8 hours.

For example, Table 1 shows the characteristic values for two different executions of the proposed method. The combination of initial stops (Route) and its associated frequency (F) are both chosen at random. The total number of buses (BA) is equal to the number of buses that have arrived to their destination (B) plus the number of buses that are still running when the simulation time (t_{sim}) is over (NB). According to F, BA varied for each iteration. It was calculated by $BA = t_{sim}/F$. The following values that also appear in Eq. 1 are reported too: travelling time (t_i), route length (r_i) and number of unplanned stops (s_i). For each execution both the initialization and the final values for the best combination found are reported.

III. SIMULATED ANNEALING: AN ADAPTED MODEL

Simulated Annealing (SA) (Kirkpatrick *et al.*, 1983) is an algorithm intended to simulate the conditioning process that the physics apply to metals. This procedure consists in firstly raising suddenly the temperature and then, cooling the metal slowly so that the final structure becomes “frozen”. This ultimate situation corresponds to an “almost perfect” structure (Bertsimas and Tsitsiklis, 1993).

By mimicking this process, our Adapted Simulated Annealing (ASA) algorithm starts with a randomly selected solution, together with a cooling variable T . This variable starts with a high value, and decreases after a certain number of iterations given by the Markov-chain length. When a potential solution (PS) with a worse fitness value appears, T controls directly its acceptance.

The SA technique tries to avoid being trapped in a local optimum by sometimes accepting a neighbour of low quality that increases the value of the objective function. In general, the commonest approach is to consider that the neighbour of a solution is different from the solution vector only in an exchange of elements- i.e., when two elements have been switched- that has been done at random.

In contrast, in this work we have designed an ad-hoc neighbour-selection method, where the switch is guided. We have given greater probability of being chosen to the vector resulting from an exchange between two stops that belong to different cities (see Fig. 2). In addition, we have noted that it is beneficial to make this

probability rise as the iterations progress. In this way, at the end of the process the algorithm tends to end up with a solution vector whose elements corresponding to a given city are gathered together.

Figure 3, is a representation of the basic structure of our ASA algorithm. Firstly, PS is evaluated; next, one of its neighbours is selected. In turn, the neighbouring solution (NS) is evaluated. If NS is better than PS, NS replaces PS; otherwise the worst solution is accepted with a probability controlled by Eq. 2

$$U(0,1) < e^{(-\delta/T)} \quad (2)$$

where $U(0,1)$ is a random number between 0 and 1, δ is the difference between PS and NS, and T is the temperature value.

At first, the search space is explored because the higher the temperature, the greater the probability of accepting worse solutions. In the end, the search space is exploited by decreasing the temperature. After having reduced the temperature exponentially with each iteration, worse solutions are hardly accepted.

IV. ASA IMPLEMENTATION

We have implemented the ASA algorithm in Mallba architecture in order to test our model. The outcome of Mallba project (Alba *et al.*, 2007) - which was carried out by University of Malaga, La Laguna and Barcelona- is a library of algorithmic skeletons that are implemented in C++ as a set of required classes and provided classes. We scheduled the required classes according to the needs associated with our problem. This code works together with the classes provided by the library Mallba, thus making our model easier to implement. We have chosen a Simulator of Urban Mobility called SUMO (Kirkpatrick *et al.*, 1983) in order to obtain the values associated with fitness. This software lets you run simulations that emulate the vehicular mobility at the microscopic level. It is open source, highly portable and withstands complete maps of real cities obtained from GoogleMaps or OpenStreetMap. The programmed ASA algorithm generates an XML file containing all data necessary to calculate the fitness function; the XML file is sent to the simulator; SUMO processes this file, and returns a value that indicates the solution quality.

Several tests were performed with different parameter combinations in order to find the best values (Rodriguez *et al.*, 2011). Then, the adopted parameters were: Temperature decay=0.99; Markov-Chain Length=10 and Number of iterations=300.

Despite having made a good choice of parameters, the computational times were not as satisfactory as expected. After an analysis of the ASA algorithm, we found that the largest computational time (over 90% of the global time in all cases) is spent during simulation. Then, the execution time grows significantly whenever the algorithm has to call the simulator. Therefore, with a view to achieving an optimum performance, it is worthwhile concentrating on any simulator that provides the quickest computing times.

Table 1: Characteristic variables associated with the model.

Variables:	F	N	NB	T_i	R_i	S_i	Z
First execution							
Initial Route				1 17 8 10 13 9 5 6 16 15 19 20 18 3 2 12 21 7 11 4 14 22			
Initial Values	30	13	3	3,35E+05	4,67E+06	1,04E+05	3,93E+05
Final Route				1 2 3 7 8 4 5 6 9 10 12 14 15 16 17 19 18 21 11 13 20 22			
Final Values	33	14	1	1,35E+05	1,02E+06	8,14E+04	8,82E+04
Second execution							
Initial Route				1 15 12 2 4 8 6 3 10 9 17 21 5 13 14 16 19 18 11 20 7 22			
Initial Values	26	15	4	3,51E+05	4,47E+06	1,17E+05	3,29E+05
Final Route				1 2 5 6 3 4 7 8 9 10 12 13 15 14 11 16 17 18 22 21 19 20			
Final Values	39	12	1	1,12E+05	7,62E+05	6,33E+04	7,81E+04

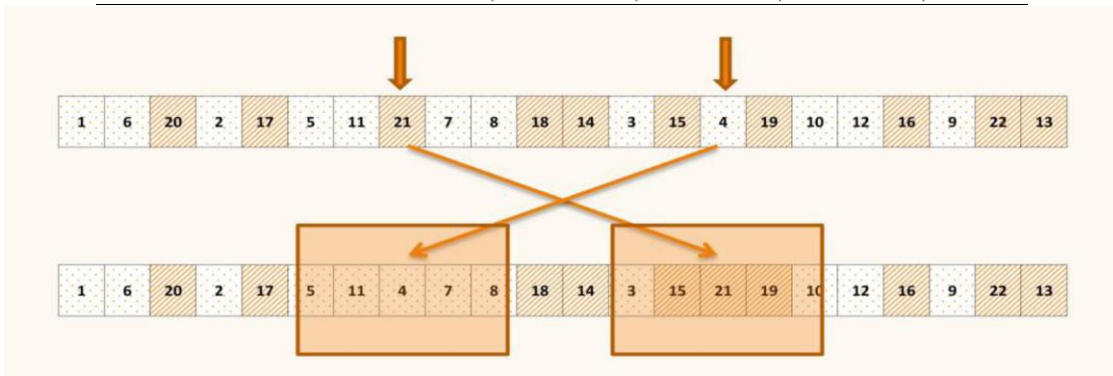


Figure 2: An example of the neighbour-selection method.

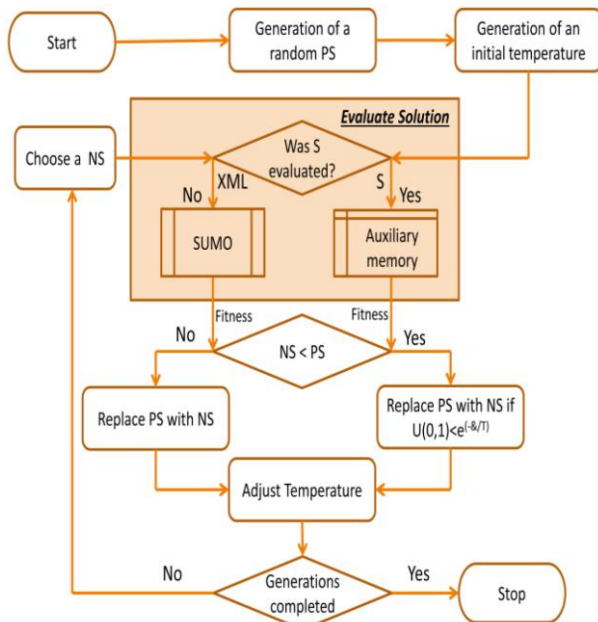


Figure 3: A simple flowchart representing our ASA process.

Furthermore, it was found that during the last iterations of the algorithm, the solutions that had already been evaluated were sent to the simulator again, resulting in unnecessary resource consumption. To overcome this shortcoming, the SA was adapted by incorporating an auxiliary memory. In this way those solutions most likely to be retested are kept. Then, before calling the simulator, the algorithm checks whether the solution is included in the auxiliary memory. If it is so, the fitness value is rapidly recovered. Thus, satisfactory behaviour in algorithmic performance was achieved.

However, we have tried to improve this performance

even more. Instead of keeping the solutions with the best fitness value in the auxiliary memory, the neighbour quality was taken into account. Due to the characteristics of the neighbour-selection method, we have found that it is appropriate for our ASA algorithm that the dominant solutions in the auxiliary memory are those with the greatest probability of being chosen as a neighbour solution, only considering its fitness value as a secondary attribute. For this implementation it is necessary to maintain a value equal to the number of coincidences between the PS and NS, which we have called Likely Neighbour (LN). The auxiliary memory is updated for each generation of a neighbour, and it is sorted by LN whenever a better solution appears.

The purpose was to save the solutions that were more similar to the accepted one. Therefore, it was necessary to introduce the LN index. This index shows the amount of coincident elements of an NS vector by comparison with the PS. The memory is ordered as a function of LN and remaining priorities are established. The major benefit is achieved by keeping in memory neighbour solutions as a function of LN. The auxiliary memory size ($DimM=2p$) can be established in accordance with problem dimensions. After several trials (Rodríguez *et al.*, 2011), a valid alternative consists of storing in memory double the amount of stops p . In this work, 44 neighbour solutions are kept and each stop is represented by an integer, and the corresponding index NL is also an integer. This application makes use of 4 bytes to store an integer number. Then, the memory consumption ($ConM=4(p+1)(2p)$) results are not transcendental.

V COMPUTATIONAL RESULTS

The executions were performed on a standard PC: Intel Core 2 Duo 2.53 GHz and 1 GB of RAM. Thirty independent ASA runs were carried out. Figure 4 shows the evolution of the obtained results. The fitness values decrease whenever the evaluations progress. Then, this representative behavior exhibits the satisfactory robustness of ASA when solving the IPTP.

A genetic algorithm was also used as a benchmark to measure the quality of the ASA algorithm. The Wilcoxon signed rank test was employed to compare the numerical distributions of the results. Both techniques showed statistically significant differences ($p < 0.05$). In Fig. 5, the results clearly indicate that the ASA overcomes the genetic algorithm. It is possible to observe that the mean values of ASA are better. It is important to remark that even the best results obtained by the GA are above the ASA's median.

The benefit of the ASA method working with an auxiliary memory is reflected in an average reduction of 20% in the runtime. The modification that gives greater importance to the neighbour quality has shown an extra improvement of 12% in the performance computing. Figure 6 reflects the avoided calls to the simulator.

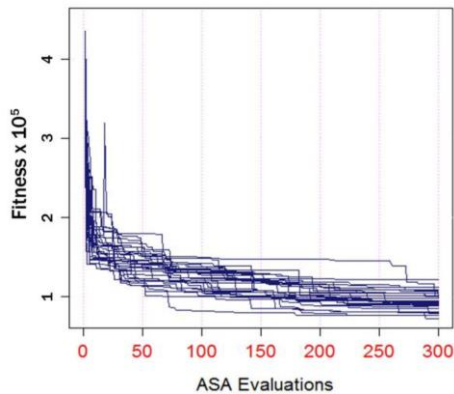


Figure 4: ASA Evolution.

Bahía Blanca - Punta Alta Intercity Line

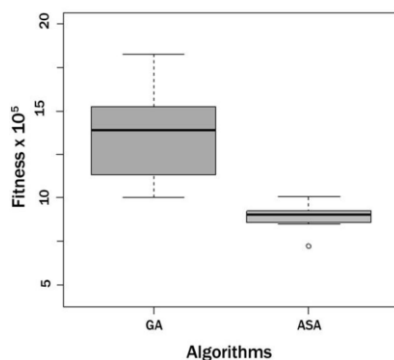


Figure 5: Boxplots comparing ASA with GA.

VI. CONCLUSIONS

In this work we have chosen a Simulated Annealing core in order to solve the intercity public transport problem. This proposal has resulted in an effective, efficient option. Our implementation of the ASA algorithm proved to be capable of yielding high quality solutions at appropriate times.

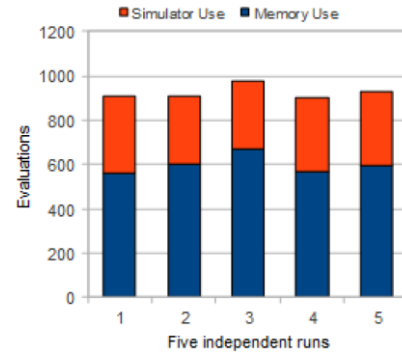


Figure 6: Calls to the simulator.

We have built an efficient model that minimizes the number of calls to the simulator by both the addition of an auxiliary memory and a novel neighbour-selection method. We have also demonstrated that our technique can be applied to a real scenario.

This study is focused on a single-thread approach. The implementation of a parallel strategy requires a particular evaluation of the performance of metaheuristics based on population, in comparison with those based on trajectory. Then, it is an interesting subject for future work.

REFERENCES

- Alba, E., G. Luque, J. García-Nieto, G. Ordóñez and G. Leguizamón, "Mallba: a software library to design efficient optimisation algorithms," *Int. J. of Innovative Computing and Applications (IJICA)*, **1**, 74–85 (2007).
- Alvarenga, G.B., G.R. Mateus and G. Tomi, "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows," *Comp. & Ops. Res.*, **34**, 1561-1584 (2007).
- Behrisch, M., L. Bieker, J. Erdmann and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview, in: SIMUL," *The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 63–68 (2011).
- Bertsimas, D. and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, **8**, 10–15 (1993).
- Cipriani, E., S. Gori and M. Petrelli, "Transit network design: A procedure and an application to a large urban area," *Transportation Research Part C: Emerging Technologies*, **20**, 3–14 (2012).
- Colomi, A., M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini and M. Trubian, "Heuristics from nature for hard combinatorial optimization problems," *International Transactions in Operational Research*, **3**, 1–21 (1996).
- Dondo, R., C.A. Mendez and J. Cerdá, "An optimal approach to the multiple-depot heterogeneous vehicle routing problem with time window and capacity constraints," *Latin American Applied Research*, **33**, 129-134 (2003).
- Dooley, A.E., W.J. Parker and H.T. Blair, "Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying," *Computers*

- and *Electronics in Agriculture*, **48**, 75–91 (2005).
- Giaglis, G.M., I. Minis, A. Tatarakis and V. Zeimpekis, “Minimizing Logistics Risk through Real-Time Vehicle Routing and Mobile Technologies: Research To-Date and Future Trends,” *International Journal of Physical Distribution and Logistics Management*, **34**, 749-764. (2004).
- Guihaire, V. and J. Hao, “Transit network design and scheduling: A global review,” *Transportation Research Part A: Policy and Practice*, **42**, 1251–1273 (2008).
- Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, “Optimization by simulated annealing,” *Science*, **220**, 671–680 (1983).
- Maffioli, F., “The vehicle routing problem: A book review,” *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, **1**, 149–153 (2003).
- Martínez, L.M., J.M. Viegas and E.A. Silva, “Zoning decisions in transport planning and their impact on the precision of results,” *Transportation Research Record, Journal of the Transportation Research Board*, **1994**, 58-65 (2007).
- Mauttone, A., *Optimización de recorridos y frecuencias en sistemas de transporte público urbano colectivo*, Master thesis in computer science, Universidad de la República, Uruguay (2005).
- Mauttone, A. and M.E. Urquhart, “A route set construction algorithm for the Transit Network Design Problem,” *Computers & Operations Research*, **36**, 2440–2449 (2009).
- Olivera, A.C., M. Frutos, J.A. Carballido, I. Ponzoni and N.B. Brignole, “Bus network scheduling problem: Grasp + eas with pisa * simulation,” *Lecture Notes in Computer Science*, **5517**, 1272–1279 (2009).
- Ortúzar, J. de D. and L.G. Willumsen, *Modeling Transport*, 4^a Ed., John Wiley & Sons, Chichester, (2011).
- Rodriguez, D.A., A.C. Olivera and N.B. Brignole, “Metaheuristics Applied to the Intercity Public Transport Problem,” *Mecanica Computacional*, **XXX**, 2169-2179 (2011).
- Sa’adah, S. and P.B. Paechter, “Improving Vehicle Routing Using a Customer Waiting Time Colony,” *EvoCOP 2004, LNCS 3004*, J. Gottlieb and G.R. Raidl (Eds.), Springer-Verlag, 188–198 (2004).
- Toth, P. and D. Vigo D, “Models, relaxations and exact approaches for the capacitated vehicle routing problem,” *Discrete Applied Mathematics*, **123**, 487-512 (2002).
- Xinchao, Z., “Simulated annealing algorithm with adaptive neighbourhood,” *Applied Soft Computing*, **11**, 1827-1836 (2011).
- Yan, S. and H. Chen, “A scheduling model and a solution algorithm for inter-city bus carriers,” *Transportation Research Part A: Policy and Practice*, **36**, 805–825 (2002).

Received: March 7, 2013

Accepted: January 10, 2014

Recommended by Subject Editor: José Guivant