Asociación Argentina de Mecánica Computacional

Mecánica Computacional Vol XXX, págs. 2169-2179 (artículo completo)
Oscar Möller, Javier W. Signorelli, Mario A. Storti (Eds.)
Rosario, Argentina, 1-4 Noviembre 2011

METAHEURISTICS APPLIED TO THE INTERCITY PUBLIC TRANSPORT PROBLEM

Diego A. Rodríguez[a,b,c], Ana C. Olivera[c] and Nélida B. Brignole[b,c]

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

[b]*Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET Camino La Carrindanga Km. 7, 8000 Bahía Blanca, Argentina*

[c]*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, Blanca, Argentina, drodriguez@plapiqui.edu.ar, aco@cs.uns.edu.ar, dybrigno@criba.edu.ar*

Keywords: metaheuristics, genetic algorithms, simulated annealing, MALLBA, intercity public transport, optimization.

Abstract. This paper presents a study on the use of metaheuristics for the treatment of the NP-hard problem known as the Intercity Public Transport Problem. The study aimed at optimizing the bus routes and frequencies through two basically different techniques: Genetic Algorithms and Simulated Annealing. The complete implementation of the algorithms was carried out on MALLBA, where traffic simulation was performed with SUMO (Simulation for Urban MObility). The testing case was an intercity passenger line linking the Argentinian cities of Bahía Blanca and Punta Alta.

1 INTRODUCTION

Every day thousands of commuters travel from one city to another for different purposes. In many cases, users share the same intercity line for a significant period of time. Therefore, a careful planning of the transport system becomes increasingly essential. Due to the rising demand, either increasing resources or reorganizing the fleet and staff allocation turn into valid options. There are many studies and results about this alternative (Wren, 1999). However, if it is impossible to incorporate new buses and drivers to the line, a reasonable policy consists in optimizing the bus routes and frequencies (Baaj and Mahmassani, 1991; Desaulniers and Hickman, 2007) in order to improve significantly the intercity transport system. This possibility is generally regulated by the government and it affects the resource allocation task (Ceder and Wilson, 1986).

Even though it is similar to urban traffic, the intercity traffic also has the following main differences that require special treatment: a) it usually has fewer and more remote stops; b) there is less interaction between vehicles resulting in fewer unplanned stoppages; c) there are route segments with special features (e.g. higher speed limits). Most of the research on route planning is related to urban transport, instead of concentrating on the intercity traffic (Guihaire and Hao, 2008, Su and Chang, 2010, Yan et al., 2006, Yan and Chen, 2002, Olivera et al., 2009). In this group there are only a few papers that give the demand a stochastic treatment (Yan et al., 2006, Olivera et al., 2009).

Among some of the studies about urban traffic, Salzborn (1980) models the frequency design in order to minimize the fleet and waiting times for the users. Van Nes et al. (1988) propose to evaluate all routes starting from zero frequency and increasing gradually the frequency according to marginal efficiency. However, more recent studies (Guihaire and Hao, 2008) concluded that it is appropriate to address the problem of frequencies and routes altogether.

In view of the problem complexity, it is appropriate to solve it through metaheuristic procedures (Colorni et al., 1996). Nevertheless, the particular technique is not well determined. A Genetic Algorithm (GA) was effectively applied for the Traveling Salesman Problem (Yan et al., 2008), but Geng et al. (2011) succeed in solving a similar problem with Simulated Annealing (SA). Tan et al. (2001) also made comparisons between heuristics, concluding that GA provides better results than SA at the expense of increased use of computing time. Based on the conclusions reported by previous research papers, we have decided to explore the application of both techniques. They are very popular, having been used for various studies related to transit network design problem. GAs are addressed in Xiong and Schneider (1993), Chakroborty (2003), Pattnaik et al. (1998), Fan and Machemehl (2006), Ngamchai and Lovell (2003) and Dhingra et al. (2000), while SAs are employed by Woch and Lebkowski (2009), Breedam (1995), and Yu et al. (2010).

GA's (Pattnaik et al., 1998) and SA's (Kirkpatrick et al., 1983) procedural behavior has been investigated in this paper. The generic procedures for both algorithms were implemented in MALLBA architecture (Alba et. al, 2007) in order to test our model for the treatment of intercity public transport problem (IPTP). Their performance has been analyzed and compared aiming at the improvement of an existing intercity line between the Argentinian cities of Bahía Blanca and Punta Alta. The actual traffic flow was considered by means of SUMO (Simulator of Urban Mobility) (Krajzewicz et al., 2006) that enabled the simulation of real scenarios.

The remainder of this paper is structured as follows. Section 2 presents the IPTP and the proposed model. Section 3 introduces both algorithms used for the treatment of an IPTP.

Then, section 4 details results and comparisons. Finally, the concluding remarks and suggestions for future research are summarized in section 5.

2 THE INTERCITY PUBLIC TRANSPORT PROBLEM

The demand for intercity transport is steadily growing worldwide. This increase does not correspond with an organizational structure because it does not give the users a service that meets their needs and it does not allow operators to maximize their profits. In particular, a disadvantage is that there are areas where only one operator is responsible for connecting two or more cities through a single interurban line, linking the cities to attain and maintain the passenger flow. IPTP resolution focuses on establishing the bus routes and frequencies (Yan et al., 2006; Yan and Chen, 2002) with a set of predetermined stops and interlocked in order to optimize the bus route and the journey times as much as possible.

Bahía Blanca is an Argentinian city with 400,000 inhabitants, two national universities, some governmental buildings and a populated surface of 2247 km^2. Punta Alta is a town with 60,000 inhabitants, an important Naval Base called Puerto Belgrano and a populated surface of 1312 km^2. Both are located to the South of Buenos Aires province. The distance between them is about 30 km. They are communicated through an interurban line with an approximate demand of two thousand commuters per day. Although the line is currently in operation, we could not get any official information about the bus routes and frequencies. In this study we had to resort to making inquiries through the websites of both municipalities and among line users in order to set stops and frequencies realistically.

The study case has a set of 22 stops that shape the line route linking Bahia Blanca and Punta Alta. Buses should start at a stop, visit all the others and conclude in a final stop. The objective is to find the path whose cost is the least, while establishing the vehicular frequency, which has been imposed between 25 and 50 minutes by municipal regulations. The case was analyzed by comparing metaheuristics. IPTP belongs to the NP-hard class; then, solving this problem by an exhaustive search method would require an enormous processing time. Therefore, it is desirable to employ metaheuristics in order to obtain satisfactory responses in a reasonable amount of time. Moreover, it should be noted that its computational complexity increases exponentially by increasing the number of stops. With a view to algorithmic comparison, the same representation of the problem was employed in order to encode all possible solutions and the same fitness function was chosen to evaluate the solution quality.

2.1 Solution representation

A solution to an IPTP is a sequence of stops; it is represented by a vector of *n* elements, where *n* is the number of stops to be visited by the bus (see Table 1). It also conserves another vector, indicating which city belongs to each stop.

A_1	A_2	…	…	A_i	…	…	A_{n-1}	A_n

Table 1: Representation of a solution.

For our problem:
$A_i = \{1, 2\}$ where 1=Bahía Blanca and 2=Punta Alta;
$i = \{1, 2, 3, \ldots, n\}$ where n=22.

2.2 The fitness function

The users' and operators' interests are represented by means of the fitness function Z (see

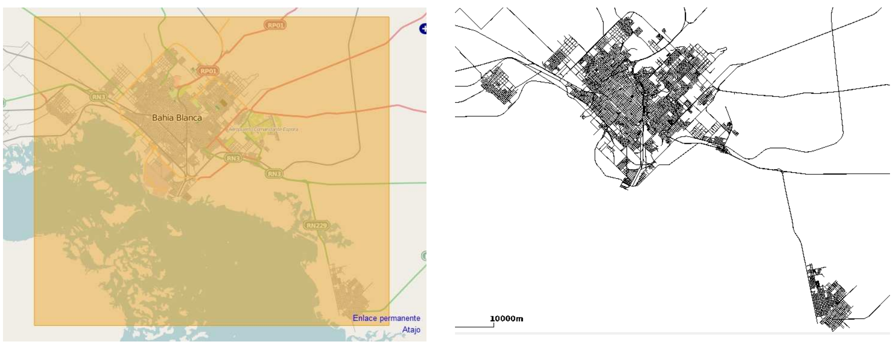
Eq. 1). Finding the routes and frequencies is formulated as an optimization problem that consists in minimizing the bus travel time, the route cost for the vehicle and the vehicular number that satisfactorily complies with the imposed frequencies.

$$Min \quad Z = \frac{(\mathrm{T} + \mathrm{R} + \mathrm{ST})}{V} \tag{1}$$

Symbols

- t_i The total travel time of a bus *i* in seconds
- T $\sum_{i=0}^{NV+V} ti$
- v_i A bus *i*
- V The number of vehicles that reach their destinies
- NV The number of vehicles that do not arrive at their destinies
- r_i The route amount for the vehicle *i*
- R $\sum_{i=0}^{NV+V} ri$
- s_i The number of times that a vehicle must stop.
- ST $\sum_{i=0}^{NV+V} si$
- S The total simulation time

For the fitness values associated with a feasible solution, we have made use of an available simulator called SUMO (Krajzewicz et al., 2006). This software lets one run simulations that emulate vehicular mobility at the microscopic level; it is open source, highly portable and capable of supporting full maps of real cities, like those obtained through GoogleMaps or OpenStreetMap (see Figure 1). It is possible to define properties by means of SUMO, such as vehicular acceleration and deceleration, driver´s ability, maximum vehicular speed, street directions, and waiting times. In this way, SUMO allows the accurate simulation of traffic behavior by providing all the data necessary to calculate the fitness of each solution.



OPENSTREETMAP **SUMO**

Figure 1: Bahía Blanca and Punta Alta cities: OpenStreetMap view (left) contrasted with the SUMO result (right).

There are many studies related to the use of metaheuristics for optimization problems. In the next section we have introduced the most relevant characteristics of the GA and SA approaches related to the problem under study.

3 METAHEURISTIC ALGORITHMS

Metaheuristics are non-deterministic algorithms that obtain high-quality solutions in a reasonable amount of time. Their main convenient features are their ability to escape local optima, their adaptability to changes and their problem independence.

In addition to their prestige, the present choice of both GA and SA metaheuristics is strongly linked to their marked differences. These techniques differ in the number of solutions processed at the same time and in the quantity of memory they use. On the one hand, GA is a population-based approach describing the evolution of a set of points in the search space; as it progresses, it uses a history of previous searches. On the other hand, the SA is a method based on the trajectory that uses only one solution at a time, marking a trajectory in the search space; its action is exclusively determined by the state of current information.

3.1 Genetic algorithm (GA)

GA is a search algorithm based on the principle of Darwin's biological evolution (Darwin, 1859). The algorithmic initialization is a set of individuals who constitute the population. Each represents a possible solution. Like in nature, individuals are selected favoring the best qualities, valued by means of a fitness function. Then, these individuals are usually crossed and their descendants take part of the new population. These offspring often mutate. This is repeated through generations until it reaches the algorithmic stopping point. This end condition may be either a predetermined number of iterations or when the algorithm has converged to a satisfactory solution.

GA Algorithm:

```
t:=0;
initialize P(t);          // a random population is generated
evaluate P(t);            // the quality of each solution is calculated
while not end do          // the number of generations has not been reached
   t:=t+1;
   P(t):=select P(t-1);   // the best solutions are selected
   recombine P(t);        // solutions are crossed
   mutate P(t);           // the offsprings are mutated
   evaluate P(t)          // the quality of each solution is calculated
end while
```

Table 2: Pseudo-code for Genetic Algorithm.

The algorithm starts with a randomly generated solution. The following three basic operators are used in each of the iterations:

Selection: Operator that chooses individuals (i.e., solutions) that will constitute the new population. For our particular case, we used roulette wheel selection (Baker, 1987), which consists in associating each individual with a value proportional to its fitness with respect to others in the population. Thus, individuals of better fitness have higher chances of being selected.

Crossover: Operator that allows passing genetic information of an original pair of chromosomes to their offspring, thus generating genetic diversity. In our algorithm we have selected the PMX (Partially Mapped Crossover) (Goldberg, D. E. and Linge, R, 1985), which produces children from two valid solutions by choosing two random cut points to form a subsequence of the mobile path; then, this information is exchanged between parents.

Mutation: Operator that exchanges two genes at random. This allows the intensification of a particular space touring neighboring places. For the IPTP in particular, we have made a small variation of the mutation by incorporating problem-specific knowledge to this operator, thus promoting exchanges between stops located in different cities.

3.2 Simulated Annealing (SA)

SA is a search method that is conceptually based on the physical process of heating a solid, followed by a gradual cooling, until a crystalline state with an almost perfect structure is reached (Goldberg, D. E. and Linge, R, 1985).

The algorithm uses a variable called temperature *T*, whose value indicates the extent that states the limit in order to accept a solution worse than the present one. This variable starts with a high value and it is reduced in each of the iterations, according to a cooling variable *α*. There is a set of neighbors with their corresponding fitness function. Whenever an individual is generated, the acceptance criterion is applied to check whether the current solution should be replaced. The criterion consists in accepting the neighboring solution if it is better than the current one- like in a classical search method. When the generated solution is worse than the best current solution, it is possible to accept it as the new current solution depending on the difference in quality between the two solutions (*&*), and the temperature (*T*). This difference (P_{accept}) is evaluated in Eq. 2, which enables escaping from local optima.

$$P_{accept} = e^{(-\frac{\&}{T})} \tag{2}$$

From the pseudo-code, it can be noticed that the higher the temperature, the greater the possibility of accepting worse solutions. This occurs in the first iterations (exploration), decreasing progressively as they advance (exploitation). In addition, the smaller the difference in quality, the greater becomes the probability of accepting a worse solution. When an iteration has ended, the temperature decreases exponentially to move towards the next.

SA Algorithm:

```
S_act ← Initialization ()          // a random solution is initialized
T ← T_0                            //a  temperature is initialized
while not end do                         // the number of generations has not been reached
     for Cont=1 to L(t ) do            // cooling speed
          S_cand ← SelectSolutionN(S_act) // neighbor selection
          & ← cost(S_cand)-cost(S_act)    // difference between current and neighboring
solution
          if ( U(0,1)< e^(- &/T) ) ) Ó (&<0) then // the criterion of acceptance is applied
               S_act ← S_cand              // the solution is updated
          end if
          T ← α(t)                         // cooling mechanism
     end for
end while
```

Table 3: Pseudo-code for Simulated Annealing.

4 DISCUSSION

For the analyses of both algorithms in this paper we have accomplished a series of empirical tests in order to adjust the parameters of each algorithm. 20 independent runs were carried out with the configuration shown in Tables 4 and 5. Implementations were made in MALLBA with C++ and they were run on a PC Intel Core 2 Duo, with a 2.53 GHz processor and 1GB of RAM.

Generations	Individuals	Offsprings	Crossover	Mutation
100	50	100	0.7	0.01

Table 4: Configuration parameters for the Genetic Algorithm.

Evaluations	Markov-Chain Length	Temperature Decay
300	10	0.99

Table 5: Configuration parameters for Simulated Annealing.

Before making comparisons, the results of both metaheuristics were analyzed with respect to the simulator usage in the calculation of the fitness function, as well as to limitations of currently available software for academic use. At the beginning of the testing, the measurements for both algorithms indicated that approximately 95% of the execution time was consumed in the fitness calculation. As to the simulator usage inside the fitness function, the stage that involves loading the test map area was detected as the most time consuming step. Since this step is made every time the fitness calculation is called, it is natural to establish guidelines to reduce the number of simulations for each generation. These patterns are directly related to the similarities among the individuals to be evaluated. We implemented a matrix assistant in order to reduce the time associated with fitness calculation. The goal was to store in the auxiliary memory the solutions with the highest probability to be chosen as the new solution. The choice is carried out by the internal methods in each algorithm. By means of this auxiliary memory, the fitness calculation is avoided for many solutions.

Based on empirical tests, we decided to use an auxiliary memory whose size was equal to twice the number of stops. The mechanism of solution selection included in the auxiliary memory consists in keeping solutions with the best fitness values and the latest solutions generated. As a result of this modification, we have managed to avoid about 30% of the simulator calls, thus significantly reducing the runtimes of the algorithms.

There is an extensive literature about GAs and SAs that aim at solving transportation problems; in general, GA has been accepted as better than SA for a wide variety of problems. Nevertheless, SA has exhibited an excellent performance in various small problems (Mayer et al., 1999).

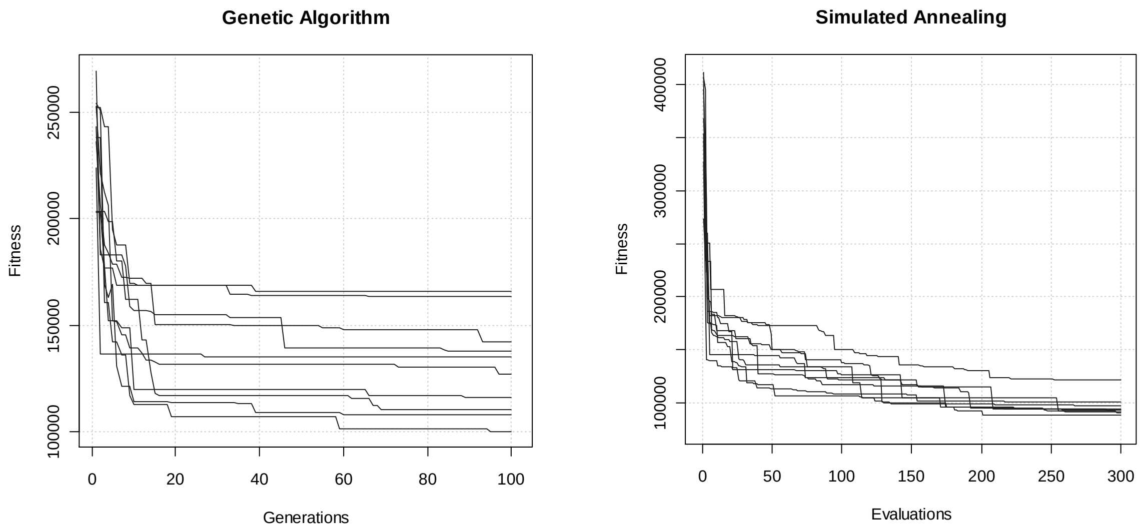
In this computational study, the GA provided satisfactory results for the resolution of this problem; even before reaching the 100 generations (see Figure 2a).

However, we have detected two situations well worth emphasizing:

1) When calculating the fitness solution, the excessive computation time of the algorithm is directly affected by the execution of the simulator. As GA is a population-based approach, this calculation is made on many occasions, which implies a higher processing time that we seek to reduce in order to increase the number of generations in search of a better solution.

2) The benefit of adding problem-specific knowledge in the mutation operator is remarkable. The main feature introduced aimed at encouraging the exchange of poorly located stops, taking advantage of the fact that we know in advance the locations of the stops for each city.

In contrast, SA is a trajectory method that makes fewer evaluations by iteration, thus reducing the computing time. Besides, it allows us to extend the number of iterations to 300, thus leading us successfully to obtain better results than through GA (see Figure 2b).



(a) (b)

Figure 2: Example of evolution of Genetic Algorithm (a) and Simulated Annealing (b) in ten runs.

In order to provide statistically meaningful comparisons, we have applied a Signed Ranked (Wilcoxon) test (Wilcox, 1987) to the numerical distributions of the results, instead of concentrating on other popular measures, like the mean and the standard deviation (Sheskin, 2007). Table 6 shows the corresponding values. The confidence level was set to 95%, which allows us to ensure that all these results are statistically different if they result in p-value<0.05. For Bahía Blanca-Punta Alta intercity line the differences between the distributions of GA and SA for 20 independent runs resulted in p-values much lower than 0.5. Therefore, it can be inferred that our SA obtained statistically better results than GA. A summary of these results can be seen in the boxplot shown in Figure 3. In this boxplot, we can confirm that SA shows better low and upper quartiles than GA.

	Mean	Median	Standard Deviation
GA	1.35E+05	1.32E+05	2.82E+04
SA	1.14E+05	1.06E+05	3.93E+04

Table 6: Mean, median and standard deviation for fitness values obtained by GA and SA in 20 runs for Bahía Blanca and Punta Alta instance.

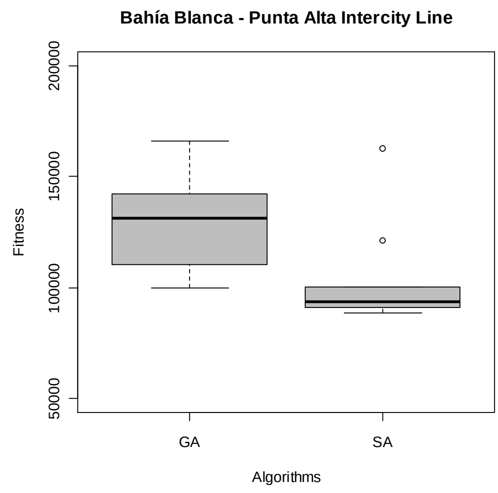


Figure 3: Boxplots of the results obtained with GA and SA.

The algorithms work by modifying the travel frequencies between 25 and 50. Nevertheless, the bus frequencies do not affect the quality of a solution directly, since the variables involved in the fitness function are compensated when the frequency changes. In fact, among the high quality solutions we found, some emerged with high frequency and others often appeared scarcely. If the frequency is low, there are fewer vehicles with a shorter covered distance, but more vehicles arrive to their destination with fewer stops. Therefore, in our algorithm the correct location of the stops really determines the solution quality.

5 CONCLUSIONS

For the Intercity Public Transport Problem (IPTP) we have evaluated two famous metaheuristics: Genetic Algorithm (GA) and Simulated Annealing (SA). Both GA and SA provide high-quality solutions; the former converges with fewer iterations, while the latter requires less computational time.

Based on this analysis, we can recommend the use of an SA procedure to solve simple versions of an IPTP. SA proved to exhibit better mean and median than GA. However, the application of GA should not be dismissed, especially in more complex instances of the IPTP.

It should be remarked that this is an introductory study on a complex research problem. Some key issues remain to be analyzed, such as the user`s demand for the service and the bus capacity. In addition, it is important to point out that at present we are developing a parallel version of the GA metaheuristics in order to obtain better solutions with less processing time.

REFERENCES

Alba, E. Luque, G. García-Nieto, J. Ordonez, G. and Leguizamón, G., MALLBA: A Software Library to Design Efficient Optimisation Algorithms Int. J. of Innovative Computing and Applications 2007 (IJICA), 1, 74-85, 2007.

Baaj, M. H. and Mahmassani, H. S., An AI-Based Approach for Transit Route System Planning and Design. Journal of Advanced Transportation, Vol 25(2), 187-210, 1991.

Baker James E., Reducing bias and inefficiency in the selection algorithm. In Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, pp. 14-21, 1987.

Breedam A.V. Improvement Heuristics for the Vehicle Routing Problem based on Simulated Annealing", European Journal of Operational Research, Vol. 86, pp. 480-490, 1995.

Ceder, A. and Wilson, N. H. M., Bus Network Design. Transportation Research, Vol 20B(4), 331-344, 1986.

Chakroborty P. Genetic Algorithms for Optimal Urban Transit Network Design. Computer-Aided Civil and Infrastructure Engineering, 18, 184–200, 2003.

Colorni A., Dorigo M., Maffioli F., Maniezzo V., Righini G. and Trubian M., Heuristics from nature for hard combinatorial optimization problems. International Transactions in Operational Research, Vol 3, 1-21, 1996.

Darwin, C. The Origin of Species, John Murray, London, 1859.

Desaulniers, G. and Hickman, M. Public Transit. Handbooks in Operation Research and Management Science, 14, 69 – 120, 2007.

Dhingra, S.L., Muralidhar, S.and Krishna Rao, K.V. Public transport routing and scheduling using genetic algorithms. In: Proceedings Presented at the CASPT 8th International Conference, Berlin, Germany, 2000.

Fan W. and Machemehl R., Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach. Journal of Transportation Engineering, 132, 122-132, 2006.

Geng X., Chen Z., Yang W., Shi D. and Zhao K. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. Applied Soft Computing, 11, 3680-3689, 2011.

Goldberg, D. E. and Linge, R., Alleles, loci, and the traveling salesman problem. In Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum, Hillsdale, New Jersey, pages 154-159, 1985.

Guihaire V.and Hao J. Transit Network Design and Scheduling: A Global Review. Transportation Research Part A: Policy and Practice, 42(10):1251-1273, 2008.

Kirkpatrick S., Gelatt C.D. and Vecchi M.P. Optimization by Simulated Annealing. Information Science, vol. 220, 1983.

Krajzewicz, D.; Bonert, M. and Wagner, P. The open source traffic simulation package SUMO *RoboCup 2006 Infrastructure Simulation Competition*, 2006.

Ngamchai, S. and Lovell, D.J., Optimal time transfer in bus transit route network design using a genetic algorithm. Journal of Transportation Engineering, 129(5), 510-521, 2003.

Olivera A., Frutos M, Carballido J, Ponzoni I, Brignole N., Bus Network Scheduling Problem: Grasp + eas With Pisa * simulation. In Bio-Inspired Systems: Computational and Ambient Intelligence, volume 5517 of *Lecture Notes in Computer Science*, pp. 1272-1279, 2009.

Pattnaik S. B., Mohan S. and Tom V. M.: Urban Bus Transit Route Network Design Using Genetic Algorithm. Journal of Transportation Engineering, 124(4):368-375, 1998.

Salzborn, F.J.M., Scheduling bus systems with interchanges. Transportation Science 14 (3), 211-220, 1980.

Sheskin, D. J. Handbook of Parametric and Nonparametric Statistical Procedures. *Chapman & Hall/CRC*, 2007.

Su J.M. and Chang C.H., The Multimodal Trip Planning System of Intercity Transportation in Taiwan. Expert Systems with Applications, 37(10):6850-6861, 2010.

Tan, K. C., Lee L. H.and Zhu K. Q. Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering, 15, 281-295, 2001.

Wilcox, R. New statistical procedures for the social sciences. Ed. *Hillsdale*, 1987.

Woch M. and Lebkowski P. Sequential Simulated Annealing for the Vehicle Routing Problem with Time Windows. Decision Making in Manufacturing and Services, 3, 87-100, 2009.

Wren, A., Heuristics Ancient and Modern; Transport Scheduling Through the Ages. Leeds Artificial Intelligence Seminar Series, University of Leeds, 1999.

Van Nes, R., Hamerslag R. and Immers, B.H., Design of Public Transport Networks. Transportation Research Record 1202, 74-83, 1988.

Xiong, Y. and Schneider, J.B., Transportation Network Design Using a Cumulative Algorithm and Neural Network. Transportation Research Record 1364, 37-44, 1993.

Yang J., Wu C., Lee H. P. and Liang Y. Solving traveling salesman problems using generalized chromosome genetic algorithm. Progress in Natural Science, 18, 887-892, 2008.

Yan S. and Chen H., A Scheduling Model and a Solution Algorithm for Inter-city Bus Carriers. Transportation Research Part A: Policy and Practice, 36(9):805-825, 2002.

Yan S., Chi C. and Tang C., Inter-city Bus Routing and Timetable Setting Under Stochastic Demands. Transportation Research Part A: Policy and Practice, 40(7):572-586, 2006.

Yu V., Lin S. W., Lee W., Ting C. J. A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering, 58, 288-299, 2010.