

Chapter 13

A Multi-Objective Optimization via Simulation Framework for Restructuring Traffic Networks Subject to Increases in Population

Enrique Gabriel Baquela and Ana Carolina Olivera

Abstract Traffic network design is a complex problem due to its nonlinear and stochastic nature. The Origin-Destiny Traffic Assignment Problem is particular case of this problem. In it, we are faced with an increase in the system vehicle population; and, we want to determine where to set the generating nodes and the traffic consumers, minimizing the current system and trying to reduce necessary investment. Performing optimizations in an analytical way in this kind of problems tends to be really complicated and a bit impractical, since it is difficult to estimate vehicle flows. In this chapter, we propose the use of a Multi-Objective Particle Swarm Optimization together with Traffic Simulations in order to generate restructuring alternatives that optimize both, traffic flow and cost associated to this restructure. This approach allows to obtain a very good approximation of the Pareto Frontier of the problem, with a fast convergence to the low infrastructure cost solutions and a total coverage of the frontier when the number of iterations is high.

Keywords Traffic net design • Particle swarm optimization • Metaheuristics • Traffic simulation • Simulated optimization

13.1 Introduction

Nowadays, population growth seems to have a direct impact on daily life due mainly to society progress. Traffic jams, pollution and parking problems are some of the most common problems in relation to traffic networks. In this chapter, we present a

E.G. Baquela (✉)

Facultad Regional San Nicolás, Universidad Tecnológica Nacional, San Nicolás de los Arroyos, Buenos Aires, Argentina

e-mail: ebaquela@frsn.utn.edu.ar

A.C. Olivera

Departamento de Ciencias Exactas y Naturales - Unidad Académica Caleta Olivia, Caleta Olivia, Santa Cruz, Argentina

Universidad Nacional de la Patagonia Austral. CONICET, Caleta Olivia, Santa Cruz, Argentina

e-mail: acolivera@conicet.gov.ar

methodology based on Optimization via Simulation (OvS) to select, from a group of nodes of a traffic network, the best alternative to increase the capacity of that network in the “Origin-Destiny Traffic Assignment Problem” (ODTAP), a subtype of the “Facility Location Problem” (FLP) and the “Network Traffic Design Problem” (NTDP) [17, 16]. Given a specific traffic network, which will suffer an increase in population, the main goal is to decide where to foment the installation of new urban infrastructure in order to minimize the cost of this and minimize the impact over the traffic system itself. In this context, we consider that the network already has a certain structure and a dynamic flow restricting possible configurations. To deal with the resolution of ODTAP, a Multi-Objective Particle Swarm Optimization Algorithm (MOPSO) was used together with Traffic Simulations (TS) in OvS based formalism.

The structure of this chapter goes as follow: in Sect. 13.1.1, a review of related works in the literature is presented; Sect. 13.2 explains the origin-destiny assignment problem and its formulation such as the optimisation problem; then, Sect. 13.3 shows an introduction to traffic simulation; Sect. 25.4 introduces the PSO algorithms for single and multi-objective problem; in Sect. 13.5, our optimisation approach is described; experiments and analysis of results are detailed in Sects. 16.6 and 16.7; finally, concluding remarks and future work are given in Sect. 16.8.

13.1.1 Literature Review

City growth reflects a progressive dynamic that is hardly ever planned, resulting in a decrease in the general performance of its sub-systems [41, 58]. In the case of the traffic sub-system, a population growth means more vehicles in the system, possible traffic jams, longer travel times, etc. [30, 41]. Even though this is not easy to control, it is possible to establish policies related to urbanization permits, to create industrial centers and to encourage activities in strategic zones, which will allow us to control the way the city grows and its associated traffic sub-system [18, 17, 28, 58]. In this line, the simulation [45, 40] reveals itself as a useful tool to quantify the impact of a certain traffic network topology and the corresponding population assignment and, also, other problems related to traffic systems [56, 9, 29, 53, 26].

In this work, we present the problem of deciding where to foment the installation of urban, industrial and/or commercial complexes so that it has a minimum effect over the total traffic system with the minimum possible cost in infrastructure. We take into account that the network already has a certain structure and a dynamic flow restricting possible configurations. We name this proposed decision problem restatement of the traffic network and the underlying problem Origin-Destiny Traffic Assignment Problem (ODTAP). The ODTAP is an underlying problem of the “Network Traffic Design Problem” (NTDP). The NTDP is the NP-Hard problem [16] of building a traffic network in such a way that it minimizes the performing function of the system, i.e. the mean travel time in general [57]. In this context, several literature works exist in relation to NTDP, some of them related to the ODTAP issue and urban planning.

The metaheuristic approach has proven to be useful for different optimization problems [43, 6, 33, 42, 47, 25, 54, 1, 32, 36, 59, 27]. For the NTDP, there are some works for which the solution is obtained by heuristics and bio-inspired algorithms [24, 7, 19, 20, 10, 13, 44].

In this chapter, we explore the Optimization via Simulation (OvS) capacity to obtain good solutions for ODTAP considering that this tool has proved to be useful to solve a lot of problems that, due to their complexity, are difficult to mold analytically.

In [26] the authors apply OvS optimizing with Particle Swarm Optimization (PSO) and simulating with the SUMO- Simulated of Urban Mobility tool [5] to solve the scheduling of traffic lights cycle. Kalganova et al. [34] uses genetic algorithms in an OvS to solve the traffic lights synchronization problem. Finally, Horvat and Tosic [31] utilize OvS together with genetic algorithms for the design of non-restrictive traffic networks in pre-existing networks.

Regarding what we have said, our work considers land use in the suburbs and cities in order to plan the urban expansion and predicts its growth considering its final impact on the traffic organization. The proposed methodology combines a popular traffic simulator SUMO [5] with “Speed-constrained Multi-objective Particle Swamp Optimization” (SMPSO) in a Optimization via Simulation context [22] in order to improve the traffic network. The propose methodology take advantage of the infrastructure in the urban sprawl. Our objectives are minimizes the impact on the landscape to obtain accessibility and mobility facilities. The experiments and comparisons with other techniques reveal that our proposed OvS approach obtains significant profits in terms of traffic network improvement. By its own nature, the OVS algorithm presented here can be easily adapted to solve different multi-objective problems [23, 21].

13.2 Origin-Destiny Traffic Assignment Problem

In this section, we define our case of study (traffic systems) and the ODTAP. Then, we expose the characteristics of the problem to be optimized.

13.2.1 Traffic Systems

A traffic system is a complex system made up of at least two elements: a traffic network and a traffic demand.

A traffic network is an oriented graph in which the edges represent the streets, and the nodes, the changing points of those streets. In Fig. 13.1, we can see a simple traffic network, composed of two one-way streets at each intersected at the corner. It is represented by a graph with 4 edges and 5 nodes, in which nodes “O” are traffic origins, nodes “D” are traffic destinations and nodes “I” are corners. As it can be

observed in the previous figure, there are different types of nodes and edges in one traffic network. Each edge is anathematized by a group of parameters: its length, the number of tracks of the traffic route, maximum circulating speed, and so on. The nodes are also defined by certain features which determine their participation in the model: kind of node (origin, destiny, lane extension, corner, etc.) and spatial coordinates, and so on. In our case, we are interested in identifying origin nodes or demand destiny nodes, as they generate or receive traffic flow. This kind of nodes needs to have associated information about maximum capacity and its rate of emitting and receiving vehicles.

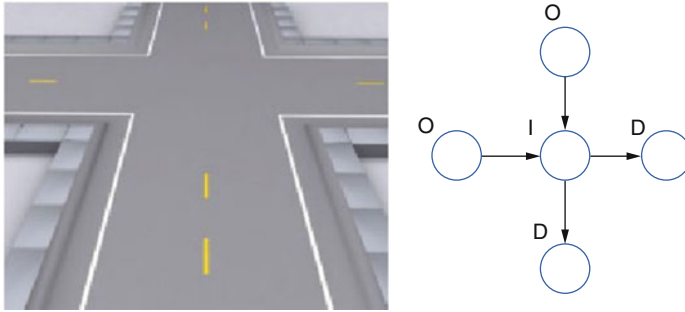


Fig. 13.1 Representation of streets intersection

Traffic demand, on the other hand, is the set of all circulating vehicles in the system (taking into account the routes they follow and their starting time). In a practical way, traffic demand is more difficult to deal with than the network structure. As it is, in a real implementation it is highly difficult not to say impossible to know the routes and starting points of the total number of vehicles of the system. Besides, generally, traffic studies normally imply having to determine how routes configuration can vary against changes in the system, turning the previous group useless. That is the reason why the demand tends to be characterized by a generating group, like traffic flows (which is the characterization used in this work). A traffic flow is a group of vehicles that in a specific time window circulating from a node i to a node j . The previous definition does not include the routes followed by the vehicles; so, flows tend to be invariable to most of the changes to be done in a traffic system. Changes which imply an alteration in the flows (as it is the case of ODTAP) turn to be easier to handle, because one only has to vary the number of vehicles associated to each flow. In order to evaluate a traffic system, it is necessary to generate the routes for the involved vehicles taking into account the flows, for which you have several mechanisms: proportional assignment, shorter paths, balancing assignments.

13.2.2 Origin-Destiny Traffic Assignment Problem

“Origin-Destiny Traffic Assignment Problem” (ODTAP) [4] tries to absorb the impact of a variation in the traffic demand (by increasing or redistributing flows) through a restructuration of the origin nodes and destinations of traffic flows. For instance, if you want to determine where to set a new shopping mall in the city. In this situation, it will not necessarily increase the number of vehicles, but it will modify the routes followed by the same vehicles at some particular times. The decision of where to set the new mall (regulated by the city hall through permits and authorizations) can be assimilated to vehicle reassignment to new traffic flows that have as one of their extremes the shopping mall (either the destiny or the origin, depending on the time frame). Another example can be the installation of an industrial depot (in this the city hall directly controls the location). Apart from redirecting vehicle routes already existing in the system, the installation of the industrial depot will probably generate new vehicle flows from an to other cities (increased by the total number of vehicles circulating). Once again, from a traffic point of view, this is translated in a restructuration of existing traffic flows and in the generation of new ones having the depot as origin and destiny. As a final example we can name the setting off a new neighborhood in town, which will have a similar effect as the previous of producing new traffic flows arriving and departing from and to the neighbor.

The three examples stated in the previous paragraph become new traffic origins and destinies. In general, if the change in the system is carried out in an organized way, it is possible to establish areas in which this new origins and destinies can establish themselves, either through regulations or plans encouraging its reside). This tend to have a meaningfully high infrastructure cost, since they demand zone paving, basic service supply, an improvement in the entrance, and so on. Therefore, the problem is now how to do this assignment (i.e. selecting the nodes of a traffic network which will function as the new origins and destinies) with the minor possible effect on current traffic, on the one hand, and using the less possible amount of money for infrastructure. Measuring the cost impact is quite simple, but evaluating the impact on the traffic system is much more complicated. It normally uses a mean time metric, thus, the size of the impact on traffic is estimated by time variations of mean time of all the system vehicles.

Formally, the ODTAP is defined on n nodes belonging to the set $N = \{1, \dots, n\}$, with arc (edges) set E , and weight costs w_k with $k = 1, \dots, m$; associated with the arcs, then, the ODTAP can be resumed as follow:

$$\min Z_1 = \sum (t_{ijp} \cdot (Pop_{ijp} + X_{ijp})) \quad (13.1)$$

$$\min Z_2 = \sum (c_i \cdot X_i) \quad (13.2)$$

subject to

$$\sum (X_i + Pop_i) \leq \text{Max Population at Origin } i \quad (13.3)$$

$$\sum (X_j + Pop_j) \leq \text{Max Population at Destination } j \quad (13.4)$$

$$\sum (X_{ijp}) = \text{Increase in Population} \quad (13.5)$$

Where, $i \in N$ are the origins, $j \in N$ are the destinies, p are the paths that connect i with j , Pop_{ijp} is the current population that needs to travel from i to j through the path p , X_{ijp} is the new assignment of population that will need to travel from i to j through the path p , t_{ijp} is the travel time from i to j in the path p of the population and c_i infrastructure cost necessary to support a population growth X_i in node i . It is important to note that this characterization of the ODTAP should be enough for small instances in simple models. However, there is not always accurate information available about traffic travel times associated with the land use in urbanization planning. The genuine function of traffic travel time depends on many uncontrollable factors such as the shape of the roads, the amount of the street, the velocity of the vehicles, etc. Moreover, traffic congestion can be an important factor in the total journey of the population. For these reasons, the total travel time can be reformulated in Eq. (13.6).

$$\min Z_1 = \sum (t_{ijp}((Pop_{i11} + X_{i11}), \dots, (Pop_{inn} + X_{inn})) \cdot (Pop_{ijp} + X_{ijp})) \quad (13.6)$$

Hence, it is considered the impact between the origin-destiny assignments into the traffic network and how it affects the Land Use planning

13.3 Traffic Simulations

Because we want to represent correctly the effect of variations in vehicle flows have on mean travel time, we will test that assignation by doing a simulation of the traffic flow. There are several ways in which you can simulate traffic flow in a traffic network, but in this work we choose to use continuum microscopic simulations.

A microscopic traffic simulation is a type of simulation of discrete time in which each vehicle behavior is modeled and simulated individually. Each vehicle in the traffic system is characterized with an identification code (“id”), a group of constant parameters regulating the route to follow, the moment in which the trip starts, maximum speed and the rest of the vehicle circulation criteria, and a group of variables with vehicle and speed information (they are updated in every interval t). The behavior of each vehicle follows simple laws, represented in an equation of the type stimuli-response where the main factor that determines the behavior of vehicles is the vehicle situated in front. These kinds of logics are known as “Car-Following” [40]. An example of this logic for a vehicle i is the following:

- If there is no car in front (from the vehicle position of vehicle i to a critical distance), accelerate until reaching maximum speed.

- If there is one car in front and its speed is lower than that of vehicle i , slow down to equal the speed of that vehicle.
- If there is a car front and its speed is higher or the same as that of vehicle i , accelerate to reach the same speed or to reach the maximum speed of vehicle i , whatever is lower.

Apart from interactions with other vehicles, interactions with the rest of the system elements are added, like the traffic lights system.

Taking into account each vehicle information, it is possible to obtain afterwards variables aggregated to the system, as average speed, maximum waiting time [11, 37].

The main advantage this type of simulations offers is that modeling all its elements is comparatively easier than in the rest of the simulations, without the need of making assumptions about global system behavior. In fact, global behavior emerges as a consequence of interactions between each of its components.

13.4 Multiobjective Particle Swarm Optimization

As it was previously mentioned, with the objective of optimizing the ODTAP we use a multi-objective version of the “Particle Swarm Optimization” (PSO). In this section, we make an introduction to this metaheuristic and its adaptation to solve multi-objective problems.

13.4.1 Particle Swarm Optimization

The PSO algorithm first appeared in 1995 [35] and since then it has been used in a high number of problems [15, 2, 52, 3, 51]. PSO is based on bird flock behavior when they go searching for their food. In a flock, you can observe individual and group behavior. PSO algorithm is a population optimization algorithm where in each iteration we have a group of potential solutions (particles expressed in PSO terminology) that function as the birds of the flock. Every particle is defined based on their position p and its speed v in a space $n - dimensional$. Interaction after interaction, each one of them evolve according to the historic individual information recollected and to global historic information of the whole flock, updating its position and speed. This evolution is regulated by Eqs. (13.7) and (13.8):

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_1 \cdot (p_{i,j}(t) - x_{i,j}(t)) + c_2 \cdot r_2 \cdot (p_{g,j}(t) - x_{i,j}(t)) \quad (13.7)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (13.8)$$

Where w is the inertia factor (which regulates the updating parameters speed), $v_{i,j}$ is the component of particle i in dimension j , c_1 and c_2 are the importance of individual and global information in each update, r_1 y $r_2 \sim U(0,1)$ represent the differential acceleration of each particle, p_i is the best value found by particle i ($pbest$) and p_g is the best solution found in the neighborhood ($gbest$). During each interaction, the speed value is updated first, and then the position value of each particle.

Even though it is a kind of evolutionary algorithm, PSO presents the distinctive feature that the individuals belonging to the population do not compete with each other to monopolize the following interactions, but they even cooperate to find the global optimum.

13.4.2 Multi-Objective Particle Swarm Optimization

The original definition of PSO was thought to deal with mono-objective problems. However, there are several adaptations to deal with problems in which it is required to optimize several objectives simultaneously [50, 48, 38, 46].

In a multi-objective problem, the focus is no longer finding the global optimum but to obtain the best approximation to the Pareto Frontier of the system. For each point in the “Decision Space” there is an associated point in the “Objective Space”, being the latter a m -dimensional space with m equal to the number of objectives of the problem. Given that it exists several “Trade-Off” possible among the different objectives, the function of the optimization algorithm is finding the section in the frontier of the “Objective Space” where these “Trade-Off” are better than any other belonging to that space. In multi-objective optimization terminology, a solution i dominates a solution j if it is at least better in one of their objectives and better or the same in the rest. The aim of a multi-objective optimization algorithm is, then, to find the group of all solutions that are not dominated by any other solution of the space (thus, they are better than the rest) [38].

In this work, we decided to use the “Speed-constrained Multi-objective PSO” (SMPSO). This algorithm was presented in [14] and it is an adaptation of “Optimized Multi-Objective PSO” (OMOPSO) presented in [49]. Both have proven to have a good performance in comparison to other versions of the multi-objective PSO, and even compared to NSGA-II [14, 8]. The main problem in any multi-objective alteration based on Pareto Frontiers of the PSO consists in how to select the leader of each interaction ($gbest$). Both algorithms handle the leader selection with a fix size list of leader solutions, selected from the group of non-dominated solutions. In each interaction, solutions are added or eliminated from the list following the non-dominating criteria. To all the leaders, a “Crowding Factor” is calculated. In case the size of the list increases, the solutions with the worst “Crowding Factor” value are eliminated; this value performing as second classificatory factor of solutions. To estimate the speed v_i of each swarm particle, a leader is chosen through a binary tournament based on the crowding value of the leaders.

Apart from this leader selection mechanism, the OMOPSO and the SMPSO have other features. The calculation of non-dominance follows non-strict dominance criteria o ϵ -dominance. is considered to be dominant with respect to a solution s if for all objectives $f(q)/(1 + \epsilon) \leq f(s)$ and to at least one objective $f(q)/(1 + \epsilon) < f(s)$. Also, to avoid bias, the parameters c_1 and c_2 are selected randomly. Finally, mutation process of solutions is incorporated, taken from the namesake process used in Genetic Algorithms.

Unlike OMOPSO, SMPSO also adds a speed calculation restriction; with the goal of avoiding particles positioning varies from its feasible extremes without evaluating intermediate points. To do this, speeds are restricted according to the Eqs. (13.9) and (13.10).

$$-delta_j \leq v_{i,j}(t) \leq delta_j \quad (13.9)$$

$$delta_j = \frac{upperLimit_j - lowerLimit_j}{2} \quad (13.10)$$

13.5 Optimization Framework

In this section we explain the OvS Procedure developed to solve the Multi-Objective version of ODTAP.

13.5.1 General Procedure

In order to solve the ODTAP, we have chosen a formalism based on OvS [23]. In the OvS, the evaluation of the goodness of solutions is done by the execution of a simulator, which functions as the generator of necessary values to calculate the objective function. Both the optimizer and the simulator are independent from each other, connecting themselves in a black box way, having a system functioning as a controller between the two (Fig. 13.2). Apart from regulating the process, the controller has the function of translating the resulting solutions brought by the optimizers to configuration parameters for the simulator, at the same time it converts this last one outputs (data samples) into a value the optimizer can accept as an aptitude or fitness function. In this context, SUMO is used for simulation purpose [5] and the SMPSO to the optimization part. This framework is adapted from a framework used to solve the mono-objective version of the ODTAP, presented in [4], in which it replaces the use of Genetic Algorithms by the PSO. The OvS procedure was implemented in the popular software R [39, 55] with the library RSUMO (<http://www.modelizandosistemas.com.ar/p/rsumo>).

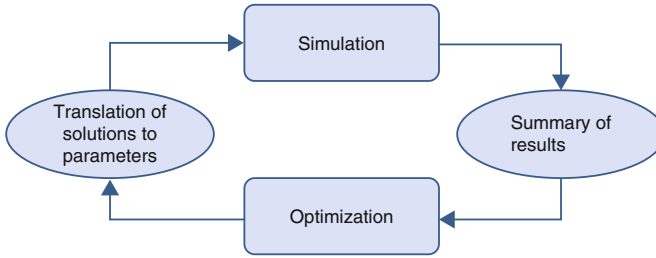


Fig. 13.2 OvS procedure as black box

Figure 13.3 shows the interrelations between the R language, SUMO package and the PSO Procedure. Defined current traffic net and current traffic density, in order to absorb population growth and restrictions on the capacity increase in the nodes, the procedure generates SMPSO solutions to assess, in the form of increases in capacity per node. These solutions are translated into traffic flows and used to run the simulation. When the simulation process ends, the travel time statistics are consolidated in an average value of travel time and reentered the SMPSO to determine the quality of each solution and generate new solutions in the next iteration.

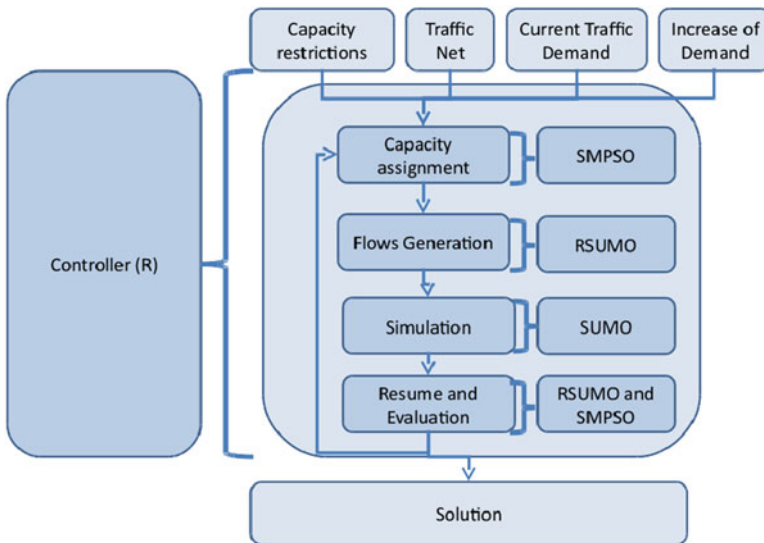


Fig. 13.3 OvS procedure

13.5.2 Solution Evaluation

In our model, SMPSO algorithm only assigns values to variables $X_{i,j}$. That is to say, it defines the population growth (or decrease) in each origin-destiny pair of the

potential origin-destiny pair group in which to invest on infrastructure. Based on this assignation, we can calculate each traffic flow $Pop_{i,j} + X_{i,j}$ (keeping $Pop_{i,j}$ constante durante todas las evaluaciones). constant during all evaluation). With the group of traffic flows, SUMO can estimate the distribution of that flow among every path p that link i with j , and then assign starting moments to each of the vehicles involved in the routes. Once each vehicle is programmed, simulations starts and statistics are recollected about mean travel time.

The evaluation of the objective cost is much simpler. You only have to compute the number of vehicles assigned to each node and multiply it with the infrastructure cost. In case it is requested, you can modify the calculation by assuming the non-linearity of these costs, making them vary step by step with respect to the number of assigned vehicles, without altering the optimization model.

13.6 Experiments

In this section, we detail the experiments carried out to evaluate the quality of the solutions of the proposed optimization algorithm.

13.6.1 Scenarios

Building scenarios can be divided in two stages: construction of traffic networks and generation of initial traffic flows. To test the quality of the algorithm, a group of ten scenarios was designed based on traffic network like the “grid” type. Its network presents the peculiarity of having a great number of possible routes between two nodes, whichever they are. Doing small variations in this network, you can get traffic networks similar to those of various urbanization types. Summarizing, the transformation operations carried out were:

1. Given a grid network of size $m \cdot n$, we determined a number of changes equals to $\sqrt{\min(m,n)}$.
2. For each change to be done, we set their type: “edge change” or “node change”.
 If the kind of change is “edge change”, we select a random edge from the network and we eliminate it.
 If the king of change is a “node change”, we select a random node from the network and we eliminate it and all the edges attached to it.

An example of a network construction can be seen in Fig. 13.4.

On the other hand, once it is defined the traffic network, it is necessary to generate the traffic demand associated to the situation previous to the population increase to be dealt with (every Pop_{ijp}). To do that, starting from the traffic networks previously created, an initial number of vehicles circulating during an hour of a simulated time in a random value between $100 \cdot \max(m,n)$ and $10 \cdot m \cdot n$ was fixed. Having the total

number of vehicles, the origin and destiny of each one are randomly raffled from the network nodes through a non-repository sampling

Regarding the maximum capacity and cost of each node, the first one was fixes in a value similar to its current population assignation increased by a random factor contained between 1.01 and 2.00. The unitary costs of increased capacity were randomly fixed too, taking some range value [10 : 100].

In total, we worked with 3 initial traffic networks, sized $10 \cdot 10$, $25 \cdot 50$ and $50 \cdot 75$, from which we generated 3 other daughter networks, according to the previously described technique. Afterwards, for each of these 9 networks, 3 random flows were created, leaving a total of 27 traffic scenarios. To every scenario a maximum capacity of every node and increase costs were randomly assigned. Every traffic scenario was evaluated with a population growth of 1.05 and 1.15, leaving 54 cases of study defined in total.

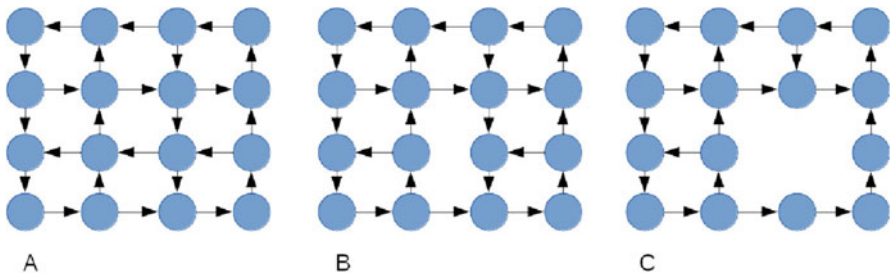


Fig. 13.4 (a) Grid net. (b) Elimination of one arc. (c) Elimination of one node

13.6.2 Tests

To evaluate how well our algorithm approximates to the problem Pareto frontier, for each scenario 10 runs were carried out. To create the Pareto frontier associated to the scenario, all generated solutions by each 10 run were consolidated and the Pareto frontier of this new data was calculated. This gave us a Global Estimated Pareto frontier for the total scenario, made up by the non-dominance solutions found in every run. That global estimated Pareto frontier is the one used as a pattern to measure the quality of the found solutions en each individual run.

Two solution quality indicators were measured: “Convergence” and “Uniformity” “Convergence” is calculated by averaging the minimum Euclidean distance of each point of the Pareto frontier obtained in one run of the algorithm with the Global Estimated Pareto frontier. This method of estimating the convergence is called “Generational Distance”. The complete index is shown in Eq. (13.11):

$$\text{generational distance} = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{N} \quad (13.11)$$

Where N is the number of solutions to be compared with the Pareto Frontier and d_i is the minimum distance from solution i to the Pareto Frontier.

Moreover, the “uniformity” is calculated by Eq. (13.12).

$$\text{uniformity} = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + \bar{d}(N-1)} \quad (13.12)$$

Where d_f and d_l are the minimum Euclidean distances between the ends of the Global Estimated Pareto Frontier and the estimated Pareto Frontier in current run, d_i is the distance between the solution i and the solution $i+1$ from the estimated Pareto Frontier in current run, \bar{d} is the average of $N-1$ distances d_i and N is the number of points in the estimated Pareto Frontier in current run.

For a further comparison, we have studied the performance of the NSGA-II [12] for the same experimental procedure as our proposal.

13.6.3 Algorithm Parameters

The parameters under which the SMPSO was executed are the following:

- *Probability of Mutation* = 0.5
- $w = 0.4$
- $\varepsilon = 0.05$
- *Swarm Size* = 50
- *Max Number of Generations* = 100

On the other hand, the simulation time to evaluate each solution was set in 1.200 s.

13.7 Results

In this section, we present the results of the experiments done.

13.7.1 Convergence

The average of generational distance between found solutions by the algorithm and the global estimated Pareto frontier is kept in low values, with a deviation coefficient

oscillating between 0.27 and 0.41. In Table 13.1 it can be observe convergence values for one of the scenarios of size $10 \cdot 10$ with an increase in vehicle population of 1.15. The convergence average of this group is of 0.40384 and the desviation coefficient is 0.38759. It can be noticed that the majority of the runs were kept in the same generational distance range, except two atypical observations in one of which we can see the effect of the presence of local optimum (item 4). The lower generational distance in item 3 is because a group of solutions kept a low mean travel time with a low infrastructure cost. For this specific scenario, the algorithm was run once more with a swarm size of 250 individuals. The convergence results are shown in Table 13.2. An improvement in the indicators can be noticed.

Table 13.1 Convergence and Uniformity of grid $10 \cdot 10$ and population increase of 1.15

Run	Convergence	Uniformity
1	0.3533009	0.2291444
2	0.4108350	0.4742433
3	0.0837602	0.4537517
4	0.7143926	0.3012058
5	0.3307537	0.1526166
6	0.4584290	0.1768969
7	0.3969564	0.2108491
8	0.4987557	0.3343977
9	0.4229219	0.2606873
10	0.3682840	0.1414944

Table 13.2 Convergence and Uniformity of grid $10 \cdot 10$ and population increase of 1.15 (swarm size = 250)

Run	Convergence	Uniformity
1	0.2106560	0.1374553
2	0.3108199	0.3622892
3	0.1789973	0.2721308
4	0.2022944	0.2191117
5	0.2454660	0.1283510
6	0.3290329	0.3134578
7	0.4945728	0.2041503
8	0.0927502	0.3940818
9	0.2933584	0.1594452
10	0.0771648	0.3538501

13.7.2 Uniformity

Uniformity finds its average in low values, but with a higher desviation coefficient than in convergence (oscillates between 0.33 and 0.46). Table 13.1 shows uniformity values for the presented case in Sect. 13.7.1. It can be observed that the behavior here is different, with a significant number of solutions with good uniformity values, and some solutions with a high uniformity. Once it was run again the algorithm with a swarm size of 250, it was impossible to observe significant improvements in this indicator.

13.7.3 Evolution of the Solutions Generated by the Algorithm

In Figs. 13.5 and 13.6 you can see the solutions from iterations 25 and 100 for one run of the algorithm, compared with the Pareto Frontier. The adjustment in the iteration 100 is very good, with almost all solutions belonging to the Pareto Frontier. It further notes that while there is not a great diversity of solutions, the Pareto frontier is swept almost entirely. At iteration 25 can see that while convergence of the solutions is very low, the algorithm found several individuals who belong to the frontier. Moreover, in Figs. 13.7 and 13.8 you can see the evolution of the population and the Estimated Pareto Frontier in iterations 25, 50, 75 and 100. In the earlier iterations, the dispersion of solutions is big and there are few solution in the estimated Pareto Frontier. In the last iterations, the dispersion is lesser and the majority of solutions are near to the estimated Pareto Frontier.

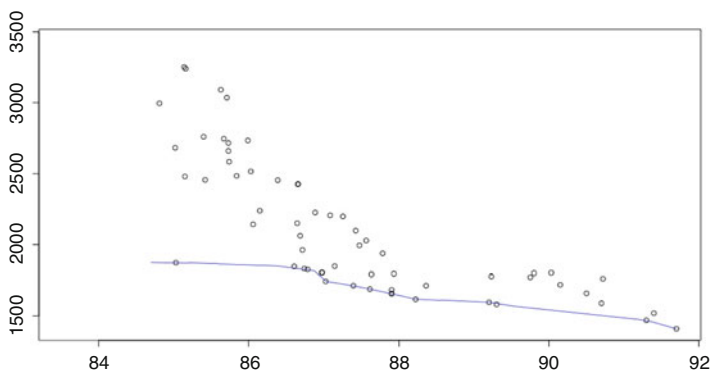


Fig. 13.5 Iteration 25. The line is the Pareto Frontier, points are the individual of the iteration

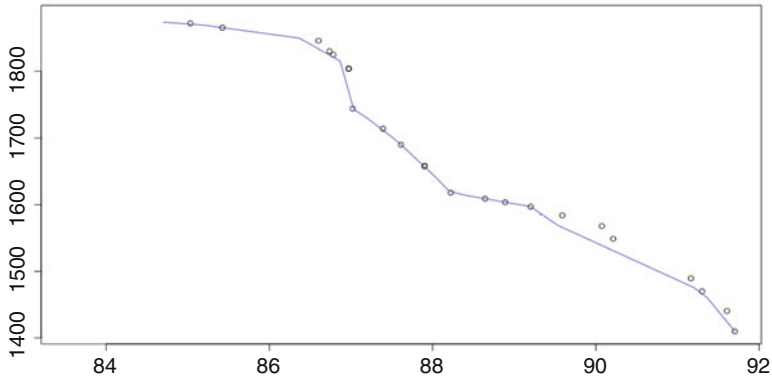


Fig. 13.6 Iteration 100. The line is the Pareto Frontier, points are the individual of the iteration

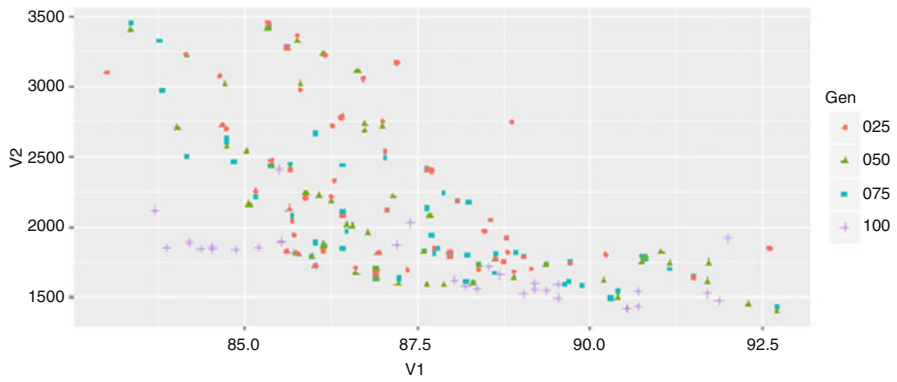


Fig. 13.7 Evolution of population

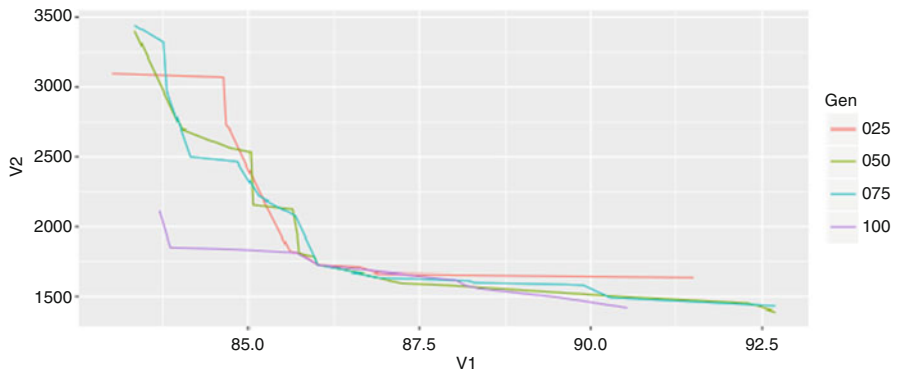


Fig. 13.8 Evolution of estimated Pareto Frontier

It's interesting see that the convergence to the Pareto Frontier is faster for the combination of "low cost-high travel time" than the "high cost-low travel time". This is a consequence of the structure of the problem: the infrastructure costs are constant, but the mean speed is a dynamic function of the vehicles assignment in the origins and destination of traffic flows.

13.7.4 Comparison with NSGA-II Metaheuristic

The same set of experiments were running under a NSGA-II metaheuristics, in order to check the quality of SMPSO solution. In Table 13.3 are shown the result of 10 runs for the same scenario that is shown in Table 13.2.

Table 13.3 Convergence and Uniformity of grid 10 · 10 and population increase of 1.15 (swarm size = 250)

Run	Convergence	Uniformity
1	0.2451904	0.2159275
2	0.4593215	0.3635322
3	0.4765746	0.1010504
4	0.2518910	0.3037142
5	0.2027211	0.3484107
6	0.4149186	0.1499699
7	0.1445920	0.1737689
8	0.1791007	0.2756582
9	0.3145456	0.4711506
10	0.3351634	0.3871142

According to the "Wilcoxon Signed Rank" the convergence of solution founded by the SMPSO algorithm are better than the NSGA-II, but for a low margin (p -value $\cong 0.03$). The uniformity is similar for both algorithms.

13.8 Conclusions

In this chapter, we presented an algorithm that combines SMPSO with a traffic simulation to solve a multi-objective version of the ODTAP. The obtained results show a good performance, being able to find a great approximation to the problem Pareto frontier with a swarm size not so big. The results get better when the swarm size increases, scarifying an increase in the computation time. The SMPSO also showed a slightly better performance than the NSGA-II for this problem.

Using simulation as an evaluating function of the optimization algorithm lets us faithfully represent the effect of policies to be implemented in the system, without the distorting effects of an analytical simplification. The algorithm has showed a fast convergence rate for the "low cost-high travel time" side of the Pareto Frontier.

This is consequence of the dynamic nature of the speed of cars in the system, which varies in function of the assignments in the origins and destinations. The use of simulations allows us to evaluate this dynamic in a easy way.

Acknowledgements The work of Baquela E. G. was supported by the Universidad Tecnologica Nacional, under PID TVUTNSN0003605. Olivera A. C. thanks to ANPCyT for grant PICT 2014-0430, CONICET (PCB-I), and Universidad Nacional de la Patagonia Austral for PI 29/B168.

References

1. S.K. Azad, O. Hasancebi, S.K. Azad, Upper bound strategy for metaheuristic based design optimization of steel frames. *Adv. Eng. Soft.* **57**, 19–32 (2013)
2. A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part I: background and development. *Nat. Comput.* **6**(4), 467–484 (2007)
3. A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. Comput.* **7**(1), 109–124 (2008)
4. E.G. Baquela, A.C. Olivera, Combining genetic algorithms with traffic simulations for restructuring traffic networks subject to increases in population, in *International Conference on Metaheuristics and Nature Inspired Computing 2014* (2014)
5. M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, Sumo – simulation of urban mobility: an overview, in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, 2011, pp. 63–68
6. J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes* (2011). <https://cs.gmu.edu/~sean/book/metaheuristics/>
7. L. Caggiani, M. Ottomanelli, Traffic equilibrium network design problem under uncertain constraints. *Procedia Soc. Behav. Sci.* **20**, 372–380 (2011)
8. P. Carrasqueira, M.J. Alves, C.H. Antunes, A bi-level multiobjective pso algorithm, in *Evolutionary Multi-Criterion Optimization – 8th International Conference*, Apr 2015
9. H.W. Casey, Simulation optimization of traffic light signal timings via perturbation analysis, Ph.D. thesis, Faculty of the Graduate School of the University of Maryland, College Park, 2006
10. H. Ceylan, M. Bell, Genetic algorithm solution for the stochastic equilibrium transportation networks under congestion. *Transp. Res. B* **39**, 169–185 (2005)
11. D. Chowdhury, L. Santen, A. Schadschneider, Statistical physics of vehicular traffic and some related systems. *Physics Report* 329 (2000), pp. 199–329
12. K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in *Parallel Problem Solving from Nature PPSN VI. Lecture Notes in Computer Science*, vol. 1917 (Springer, Berlin, 2000), pp. 849–858
13. S. Dinu, G. Bordea, A new genetic approach for transport network design and optimization. *Bull. Pol. Acad. Sci. Tech. Sci.* **59**(3), 263–272 (2011)
14. J.J. Durillo, J. Garcia-Nieto, A.J. Nebro, C.A.C. Coello, F. Luna, E. Alba, Multi-objective particle swarm optimizers: an experimental comparison, in *Evolutionary Multi-Criterion Optimization 2009* (2009)
15. R.C. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources, in *Proceedings of the 2001 Congress on Evolutionary Computation, 2001*, vol. 1 (2001), pp. 81–86
16. R.Z. Farahani, E. Miandoabchi, W. Szeto, H. Rashidi, A review of urban transportation network design problems. *Eur. J. Oper. Res.* **229**, 281–302 (2013)

17. J.A. Ferreira, B. Condessa, Defining expansion areas in small urban settlements an application to the municipality of Tomar (Portugal). *Landsc. Urban Plan.* **107**, 281–302 (2012)
18. J.A. Ferreira, B. Condessa, J.C. e Almeida, P. Pinto, Urban settlements delimitation in low-density areas an application to the municipality of Tomar (Portugal). *Landsc. Urban Plan.* **97**(3), 156–167 (2010)
19. H. Fredrik, Towards the solution of large-scale and stochastic traffic network design problems. Master's thesis, Uppsala Universitet, 2010
20. T.L. Friesz, H.-J. Cho, N.J. Mehta, R.L. Tobin, G. Anandalingam, A simulated annealing approach to the network design problem with variational inequality constraints. *Transp. Sci.* **26**(1), 18–26 (1992)
21. M. Frutos, A.C. Olivera, F. Tohmé, A memetic algorithm based on a NSGAI scheme for the flexible job-shop scheduling problem. *Ann. Oper. Res.* **181**, 745–765 (2010)
22. M.C. Fu, Optimization via simulation: a review. *Ann. Oper. Res.* **53**, 199–247 (1994)
23. M.C. Fu, Optimization for simulation: theory vs. practice. *INFORMS J. Comput.* **14**(3), 192–215 (2002)
24. M. Gallo, L. D'Acerno, B. Montella, A meta-heuristic algorithm for solving the road network design problem in regional contexts. *Procedia Soc. Behav. Sci.* **54**, 84–95 (2012). Proceedings of EWGT2012 – 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris
25. A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi, 1 – metaheuristic algorithms in modeling and optimization, in *Metaheuristic Applications in Structures and Infrastructures*, ed. by A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi (Elsevier, Oxford, 2013), pp. 1–24
26. J. Garcia-Nieto, A. Olivera, E. Alba, Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Trans. Evol. Comput.* **17**, 823–839 (2013)
27. H.C. Gomes, F. de Assis das Neves, M.J.F. Souza, Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations. *Comput. Oper. Res.* **44**, 92–104 (2014)
28. N. Haregeweyn, G. Fikadu, A. Tsunekawa, M. Tsubo, D.T. Meshesha, The dynamics of urban expansion and its impacts on land use land cover change and small-scale farmers living near the urban fringe: a case study of Bahir Dar, Ethiopia. *Landsc. Urban Plan.* **106**(2), 149–157 (2012)
29. C. He, N. Okada, Q. Zhang, P. Shi, J. Li, Modelling dynamic urban expansion processes incorporating a potential model with cellular automata. *Landsc. Urban Plan.* **86**(1), 79–91 (2008)
30. C. He, J. Tian, P. Shi, D. Hu, Simulation of the spatial stress due to urban expansion on the wetlands in Beijing, China using a GIS-based assessment model. *Landsc. Urban Plan.* **101**(3), 269–277 (2011)
31. A. Horvat, A. Tomic, Optimization of traffic networks by using genetic algorithms. *Elektrotehnicki Vestnik* **79**, 197–200 (2012)
32. M.K. Jha, M. Head, S.P. Gar, 23 – metaheuristic applications in bridge infrastructure maintenance scheduling considering stochastic aspects of deterioration, in *Metaheuristic Applications in Structures and Infrastructures*, ed. by A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi (Elsevier, Oxford, 2013), pp. 539–556
33. J. Jin, T.G. Crainic, A. Lokketangen, A cooperative parallel metaheuristic for the capacitated vehicle routing problem. *Comput. Oper. Res.* **44**, 33–41 (2014)
34. T. Kalganova, G. Russell, A. Cumming, Multiple traffic signal control using a genetic algorithm, in *Artificial Neural Nets and Genetic Algorithms* (Springer, Vienna, 1999), pp. 220–228
35. J. Kennedy, R. Eberhart, Particle swarm optimization, in *International Conference on Neural Networks* (IEEE Service Center, Piscataway, NJ, 1995), pp. 1942–1948
36. J. Kratica, An electromagnetism-like metaheuristic for the uncapacitated multiple allocation p-hub median problem. *Comput. Ind. Eng.* **66**(4), 1015–1024 (2013)
37. S. Krauss, Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. Hauptabteilung Mobilität und Systemtechnik des DLR Köln, 1998

38. V. Kumar, S. Minz, Multi-objective particle swarm optimization: an introduction. *Smart Comput. Rev.* **4**, 335–353 (2014)
39. D.T. Lang, *XML: Tools for parsing and generating XML within R and S-Plus*, R package version 3.95-0.1 (2012)
40. Y. Li, D. Sun, Microscopic car-following model for the traffic flow: the state of the art. *J. Control Theory Appl.* **10**(2), 133–143 (2012)
41. Y. Li, X. Zhu, X. Sun, F. Wang, Landscape effects of environmental impact on bay-area wetlands under rapid urban expansion and development policy: a case study of Lianyungang, China. *Landsc. Urban Plan.* **94**(3,4), 218–227 (2010)
42. A. Liefvooghe, S. Verel, J.-K. Hao, A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming. *Appl. Soft Comput.* **16**, 10–19 (2014)
43. S. Luke, *Essentials of Metaheuristics*, 2nd edn. (Lulu, 2013). <https://cs.gmu.edu/~sean/book/metaheuristics/>
44. E. Miandoabchi, R.Z. Farahani, Optimizing reserve capacity of urban road networks in a discrete network design problem. *Adv. Eng. Softw.* **42**(12), 1041–1050 (2011)
45. K. Nagel, M. Schreckenberg, A cellular automaton model for freeway traffic. *J. Phys. I* **2**, 2221–2229 (1992)
46. N. Nedjah, L. de Macedo Mourelle, Evolutionary multi objective optimisation: a survey. *Int. J. Bio-Inspired Comput.* **7**(1), 1–25 (2015)
47. E. Olivares-Benitez, R.Z. Rios-Mercado, J.L. Gonzalez-Velarde, A metaheuristic algorithm to solve the selection of transportation channels in supply chain design. *Int. J. Prod. Econ.* **145**(1), 161–172 (2013)
48. M.J. Reddy, D.N. Kumar, Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrol. Process.* **21**, 2897–2909 (2007)
49. M. Reyes-Sierra, C.A.C. Coello, Improving pso-based multi-objective optimization using crowding, mutation and e-dominance, in *Evolutionary Multi-Criterion Optimization – Third International Conference* (2005)
50. M. Reyes-Sierra, C.A.C. Coello, Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* **2**, 287–308 (2006)
51. T.M. Sands, D. Tayal, M.E. Morris, S.T. Monteiro, Robust stock value prediction using support vector machines with particle swarm optimization, in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 3327–3331
52. M.-P. Song, G.C. Gu. Research on particle swarm optimization: a review, in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 2004*, vol. 4, Aug 2004, pp. 2236–2241
53. K. Stanilov, M. Batty, Exploring the historical determinants of urban growth patterns through cellular automata. *Trans. GIS* **15**(3), 253–271 (2011)
54. S. Talatahari, 17 – optimum performance-based seismic design of frames using metaheuristic optimization algorithms, in *Metaheuristic Applications in Structures and Infrastructures*, ed. by A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi (Elsevier, Oxford, 2013), pp. 419–437
55. R.C. Team, *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, 2012). ISBN 3-900051-07-0
56. S.T. Waller, A.K. Ziliaskopoulos, A chance-constrained based stochastic dynamic traffic assignment model: analysis, formulation and solution algorithms. *Transp. Res.* **14**, 418–427 (2006)
57. S.T. Waller, K.C. Mouskos, D. Kamaryiannis, A.K. Ziliaskopoulos, A linear model for the continuous network design problem. *Comput. Aided Civ. Inf. Eng.* **21**, 334–345 (2006)
58. J. Xiao, Y. Shen, J. Ge, R. Tateishi, C. Tang, Y. Liang, Z. Huang, Evaluating urban expansion and land use change in Shijiazhuang, China, by using GIS and remote sensing. *Landsc. Urban Plan.* **75**(1,2), 69–80 (2006)
59. X.-S. Yang, 1 – optimization and metaheuristic algorithms in engineering, in *Metaheuristics in Water, Geotechnical and Transport Engineering*, ed. by X.-S. Yang, A.H. Gandomi, S. Talatahari, A.H. Alavi (Elsevier, Oxford, 2013), pp. 1–23