Interactive Meshing for the Design and Optimization of Bus Transportation Networks

Jean-Charles Créput[1], Abder Koukam[2]

Abstract: We propose an intelligent tool for terrestrial transportation design based on the underlying concept of adaptive meshing. Vehicle routes are deformable visual patterns which cover the geographic area adapting shapes to distributed demands or specific places. The system allows dynamic interaction and continuous visual feed-back for the designer constructing and evaluating a transportation bus network. To assist the designer, an optimization framework merging neural networks and evolutionary algorithms permits to position services and create routes among them automatically. The transport system optimizer built on a geographic information system is illustrated by application to a real life case of clustering and routing for the transportation of the 780 employees of an enterprise.

CE Database Subject Headings: transportation networks, clustering and routing, neural networks, evolutionary computation

1 Introduction

The optimal organization of the vehicles canvassing routes for transportation of goods, or customers, is an essential challenge. How to organize the vehicle canvassing routes to collect and deliver goods relative to fluctuating and varying demand, how to organize them by supplementing the existing networks without weighing upon them or hampering the traffic. To give a response, advanced traveler information systems (ATIS) are built to provide the

[1] Systems and Transportation Laboratory (SeT), University of Technology of Belfort-Montbeliard 90010 Belfort Cedex, France. E-mail: Jean-Charles.Creput@utbm.fr

[2] Systems and Transportion Laboratory (SeT), University of Technology of Belfort-Montbeliard 90010 Belfort Cedex, France. E-mail: Abder.Koukam@utbm.fr

right information (Kumar el al. 2005) at every moment and help the user analyze and design transportation infrastructures, as roads and streets. Route planning, finding the closest facility or the origin-destination bus travel are usual functionalities of ATIS. They are generally based on shortest path algorithms (Edelkamp et al. 2005) with polynomial time complexity, operating on a geographic information system (GIS) and communicating with positioning system infrastructures.

Thus, and responding to the needs of reusable tools to deal with hard problems within ATIS and GIS platforms, we propose a generic approach based on the concept of adaptive meshing as for example presented in (Créput et al. 2005) for radio-cellular network. We apply the approach to the context of terrestrial transportation. The transport mesh is a geometric structure, in the plane, that adapts and modifies its shape according to traffic demands. It follows that continuous visual feedback during simulations is a main functionality in the transport system optimizer presented, which merges user by-hand editing functionalities with population based optimization algorithms. Here, optimization techniques are a complement tool to automatically build routes, adapted to specific known demands, and which are possibly the variable part of an already existing transportation network. Using a GIS, the approach will be illustrated on a real life case of urban and regional transportation. The goal is to locate bus-stops and define the regional routes of vehicles to transport the 780 employees of a great enterprise in city of Belfort, France.

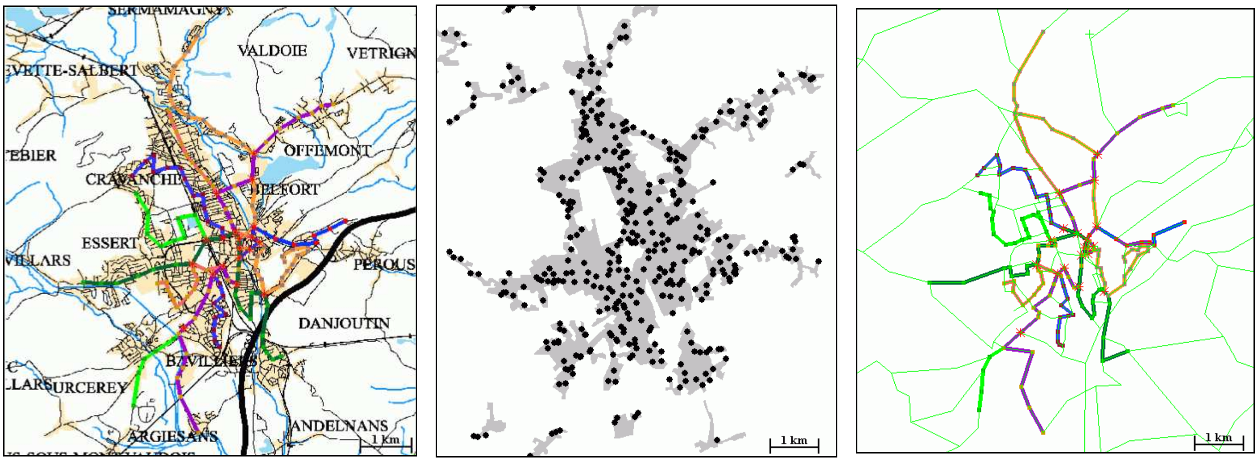
We define a general Euclidean clustering and vehicle routing problem as an extension and combination of Euclidean k-median (Arora 1998) and vehicle routing problem (VRP) (Christofides et al. 1979). The problem is NP-hard. The goal is to minimize the distances from customer demands to cluster centers defining routes, as well as minimize route lengths, subject to capacity and time window constraints. We call it the vehicle clustering and routing problem with time windows (clustering VRP). The problem seems to be never studied previously. In Clustered TSP (Laporte and Palekar 2000), clusters composition is given as an input. The closest problem encountered, called median-cycle problem (MCP), has been investigated very recently (Labbé et al. 2004, 2005; Renaud et al. 2005). The clustering VRP can be seen as a Euclidean version of MCP, whereas MCP and other related problems presented by (Labbé and Laporte 1986; Volgenant and Jonker 1987; Balas 1989) are defined on graphs. Hierarchical combinations of clustering and routing are possibly manifold, as in location-routing problems (LRP) (Min et al. 1998). But in our case, the hierarchical order of clustering and routing is different since here routes visit cluster centers, whereas in LRP routes are built on separate clusters. The problem presented can also be seen as a clustering version of the vehicle routing problem with time windows (VRPTW) (Solomon 1987).

To solve the optimization problem, following hybridization of meta-heuristics as done for example in (Boese et al. 1994; Moscato 1999; Resende and Ribeiro 2002), we present an evolutionary algorithm (EA) framework which incorporates self-organizing maps (SOM) (Kohonen 1982, 2001) as internal operators. Many applications of neural networks have addressed the traveling salesman problem (TSP) (Smith 1999; Cochrane and Beasley 2003), but their extension to more complex and abstract vehicle routing problems remains a difficult task. For example, SOM has been applied to solve the classical VRP (Christofides et al. 1979) but with less success than for the TSP, the most representative applications to VRP being the ones of Ghaziri (1996) and Modares et al. (1999). Here, we exploit the natural property of the SOM of being an Euclidean clustering algorithm which adds topologic relationships between cluster centers (Kohonen 2001). These topologic relationships naturally represent routes and integrate clustering and vehicle routing. To improve performance and to deal with capacity and time window constraints as well, SOM operators are combined into an EA framework. Furthermore, SOM is a basic operator from which specific versions are derived for example to tackle time windows.

The paper is organized as follows. Section 2 presents the basic concepts and the general clustering and routing problem. Objectives and constraints are given. Section 3 presents the user graphical editor of the transportation editing system based on a GIS. Section 4 describes the transport optimization framework, which allows to configure and execute evolutionary algorithms embedding neural networks. Section 5 presents experiments carried out on a real life case of bus transportation. Finally, the last section is devoted to the conclusion and further research.

2 Problem statement

2.1 Basic concepts



(a) (b) (c)

Fig. 1. (a) Transport network on a geographic area. (b) Sampling of transport demands (requests) on habitat zones. (c) Transport network among streets and roads.

1) Unit grid. The unit grid is a rectangular grid of size $X \times Y$ matching some geographical area. It defines the finite set L of possible locations in the plane, called pixels. Pixels are referred by their coordinates in the grid. Using integer coordinates rather than floating point values is intended to allow computational efficiency. The metric is the usual Euclidean distance, denoted *d*(*p*, *p'*) for points *p* and *p'* of the plane. For example, Fig. 1(a) presents the urban center of the city of Belfort, France, with its bus system on the front into a 1086×1245 map, with 7.5 m / pixel.

2) Requests. We denote by $V=\{r_1,...,r_n\}$, the finite set of customer demands, called requests. Each request $r_i \in V$ has a geographic location $l_i \in L$. It has a non-negative demand q_i, a service time s_i and a time window (a_i, b_i). If a vehicle arrives at a location where request r_i is intended to be served, the vehicle can not begin the service before a_i. The vehicle has to arrive before b_i. Service is done with service time s_i. Fig. 1(b) illustrates by dots some distribution of demands on habitat zones.

3) Transport mesh. Let $B=\{n_1,...,n_k\}$ being a finite set of cluster centers, also called transport points or bus-stops, localized by their coordinates in the unit grid. A transport mesh is a set of interconnected routes. Formally, it is a collection $R=\{R_1,...,R_m\}$ of m routes, where each route is a sequence $R_i=\left(n_0^i,...,n_j^i,...,n_{k_i}^i\right)$, $n_j^i \in B$, of k_i+1 successive cluster centers. Fig. 1(c) presents such a transport mesh, that is, the internal city bus system of Belfort visualized among the thinner roads network.

4) Request assignment. In our approach, the main difference with classical vehicle routing modeling is that routes are defined by an ordering of cluster centers, rather than by an ordering of customer requests. It follows that each request r must be assigned to a single cluster center $n_r \in B$ in one of the m routes. C_{R_i} is the capacity of a vehicle associated with route R_i, and L_i its load (sum of request quantities). We denote by $t_{arr}(n_r)$ the time of arrival of the vehicle to point n_r for each request $r \in V$.

5) Induced graph. Routes can share common transport points. They define a graph structure. The induced undirected graph $G_R=(N,E)$ of an interconnected set of routes $R=\{R_1,...,R_m\}$ is defined as follows: N is the set of vertices composed by all cluster centers defining routes, E is the set of edges composed of any two successive centers from routes.

2.2 Vehicle clustering and routing problem with time windows

The problem presented is a clustering version of the VRP adding time windows. It is stated as follows:

Euclidean vehicle clustering and routing problem with time windows (clustering VRP). The problem input is given by a set of requests $V = \{r_1, ..., r_n\}$ and a set of interconnected routes $R = \{R_1, ..., R_m\}$. Using notations and definitions of section 2.1, the problem consists of finding cluster center locations, except for some fixed transport points, and assignment of requests to cluster centers and vehicles, in order to minimize the following objectives:

$$length = \sum_{i=1,..,m,} \sum_{j=0,...,k_i-1} d\left(n_j^i, n_{j+1}^i\right), \tag{1}$$

$$distortion = \sum_{i=1,...,n} d\left(r_i, n_{r_i}\right), \tag{2}$$

subject to the capacity constraint:

$$L_i \le C_{R_i},\ i \in \{1,...,m\}, \tag{3}$$

and time-window constraint:

$$\underset{r_i \in V}{Min}\left(b_i - t_{arr}\left(n_{r_i}\right)\right) \ge 0\,. \tag{4}$$

Objective *length* in (1) is the routes total length. Objective *distortion* in (2) is the sum of distances from request locations to their assigned bus-stops, it is called distortion measure. The problem can be seen as a combination of a standard vehicle routing problem with the well-known Euclidean k-median problem, or Multi-source Weber problem (Hansen et al. 1998), using a transport mesh. Note that if we replace the not squared distances in objective *distortion* (2) by the squared distances, we retrieve the k-mean problem for which fast computational methods are k-mean algorithm and its stochastic version called vector quantization (VQ) algorithm (Kohonen 2001). As stated in (Kohonen 2001) the SOM algorithm itself extends VQ by adding topologic relationships between cluster centers. Replacing squared distances by maximum distance yields to the k-center problem. Here, we assume that requests are served in parallel inside each cluster.

3 The transport mesh edition module

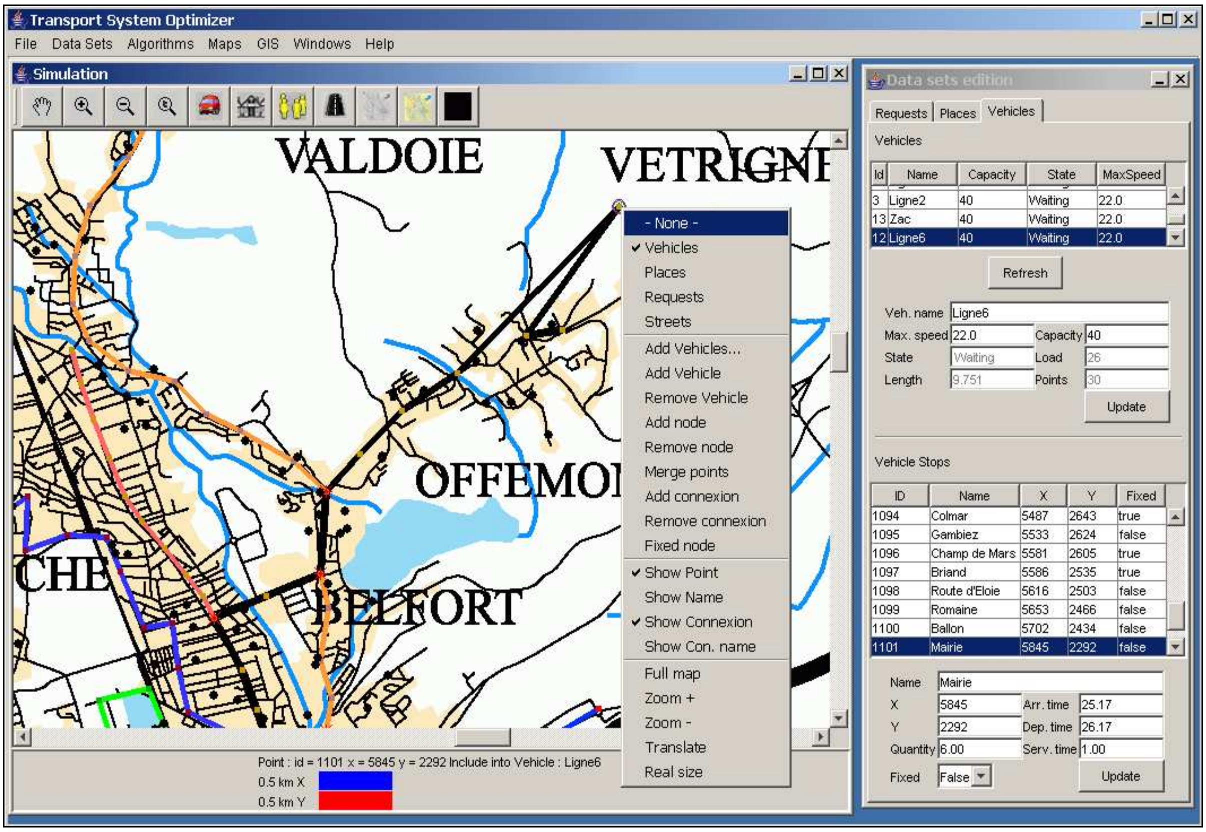


Fig. 2. Editing a transport mesh

The transport system optimizer made up with Java incorporates a geographic information system in order to display and localized geographically referenced objects. Main editing window is shown in Fig. 2. The urban center is the city of Belfort, France, with its bus system on the front. Tools are provided to help the user build at hand the transport network, using the mouse and keyboard interface to point, add or remove basic elements as route transport points. It is possible to create, add and remove routes and their bus-stops, connect and disconnect routes thru common bus-stops. To complete user by-hand editing functionalities, several windows lets the user evaluate adaptation to customer demands. For example, main criteria for clustering and routing are given thru window criteria presented in Fig. 3, beside graphical frame. In order to tackle real-life applications, positioning and editing customer requests using their postal addresses is a main GIS based functionality in the system.

Correspondence between postal addresses and their locations on the map are resumed graphically and semantically, as shown in Fig. 4. We localize enterprise bus customers at the scale of a region.

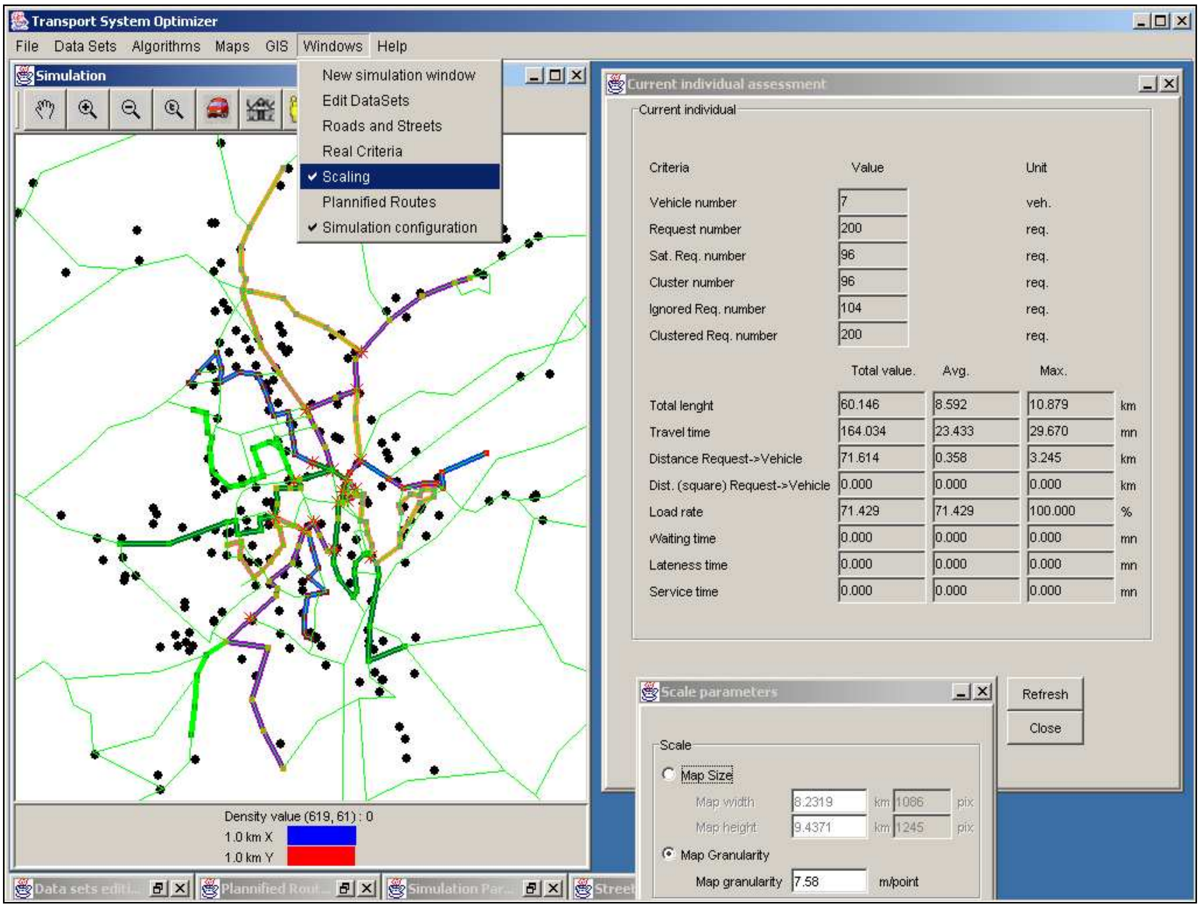


Fig. 3. Evaluation according to transport demands.

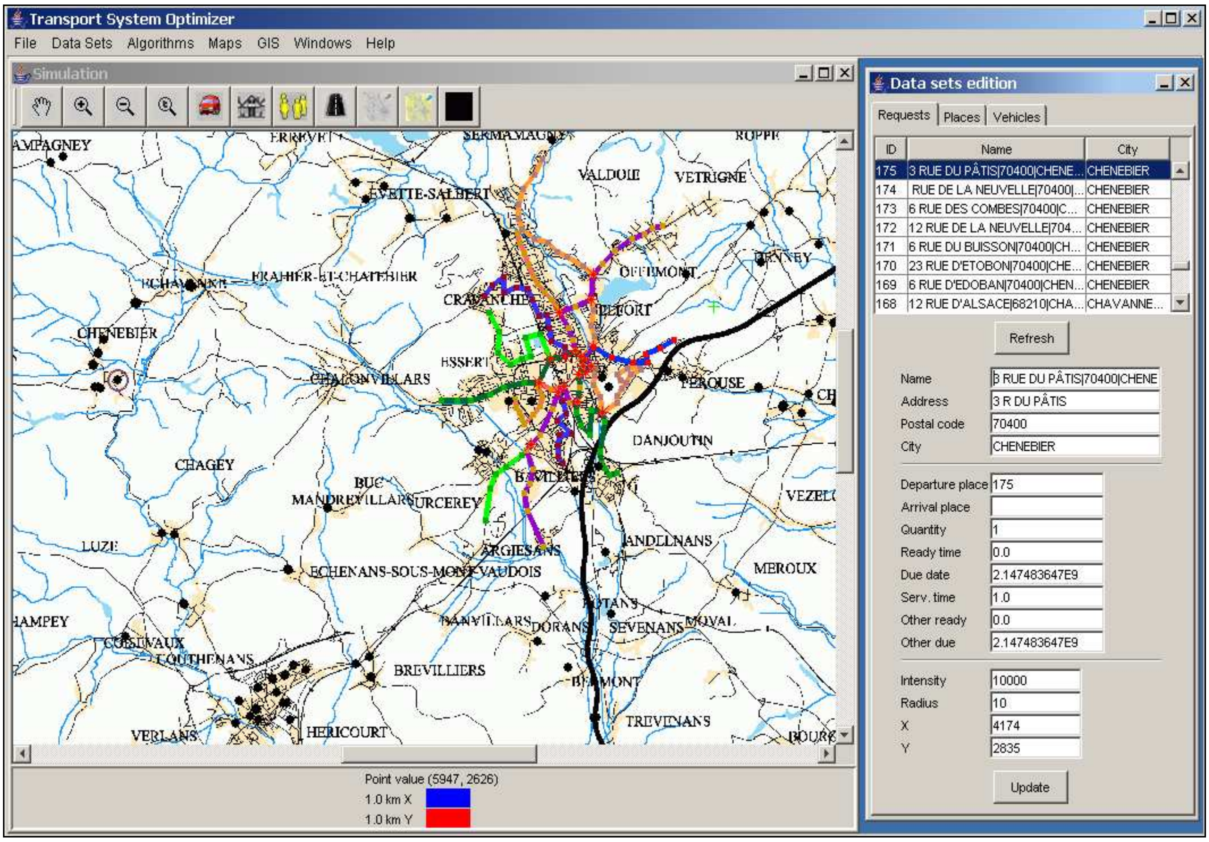


Fig. 4. Positioning and editing customer requests (dots) using postal addresses and GIS.

4 The evolutionary and neural network based transport optimizer

4.1 The optimization framework

The optimization framework includes an island based evolutionary algorithm embedding neural networks. It is a tool intended to complement the user editing system, living the designer free to control the optimization process dynamically. Using visual patterns as intermediate structures that adapt and distort according to demands have several advantages. It takes into account geometric nature of transportation routing, and lets the user quickly and visually evaluate solutions as they evolve. In turn, the designer adjusts optimization parameters to direct the search toward useful compromises.

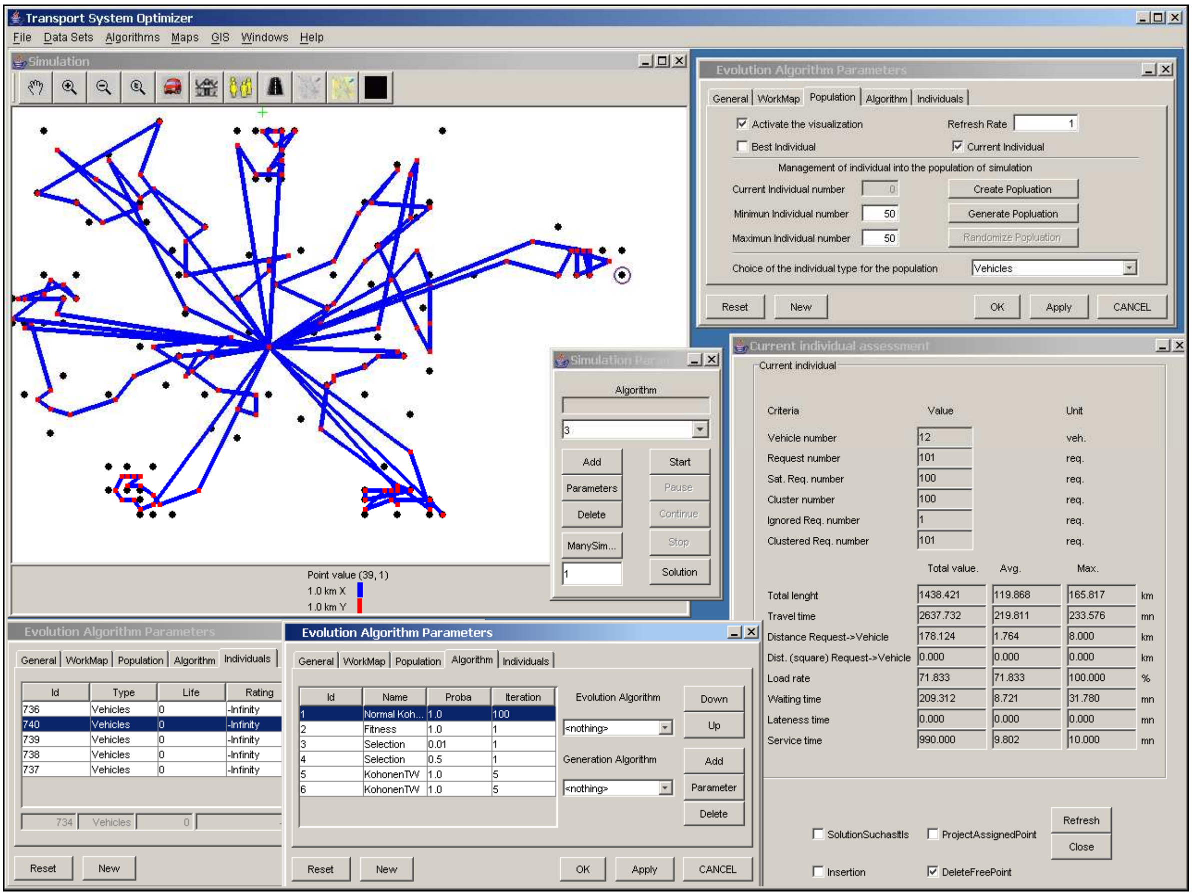


Fig. 5. Evolutionary algorithms into the interactive optimizer system.

As illustrated in Fig. 5, user gets immediate feedback from the system while running evolutionary algorithms that communicate by exchange of solutions. As shown in object model of Fig. 6, the optimization framework is based on a recursive definition of evolutionary algorithm and operators. An EA object contains a population of individuals, each individual being associated with a single solution. It contains as well an object called WorkMap that contains requests and useful input data, and a set of operators that are applied sequentially to the solutions. Main operators are standard SOM algorithms (Kohonen 2001) encapsulated into a standard evolutionary or memetic loop (Moscato 1999), memetic algorithms being hybrid evolutionary algorithms incorporating a neighborhood search process. We call the approach “memetic SOM”. Operators are possibly self-organizing map algorithms, request assignment and fitness evaluation, and selection operators. To implement island based mechanisms, EA are themselves possibly encapsulated as operators, called a

ConfigurationAlgorithm, population movement operators permitting interaction between islands by solution exchanges.

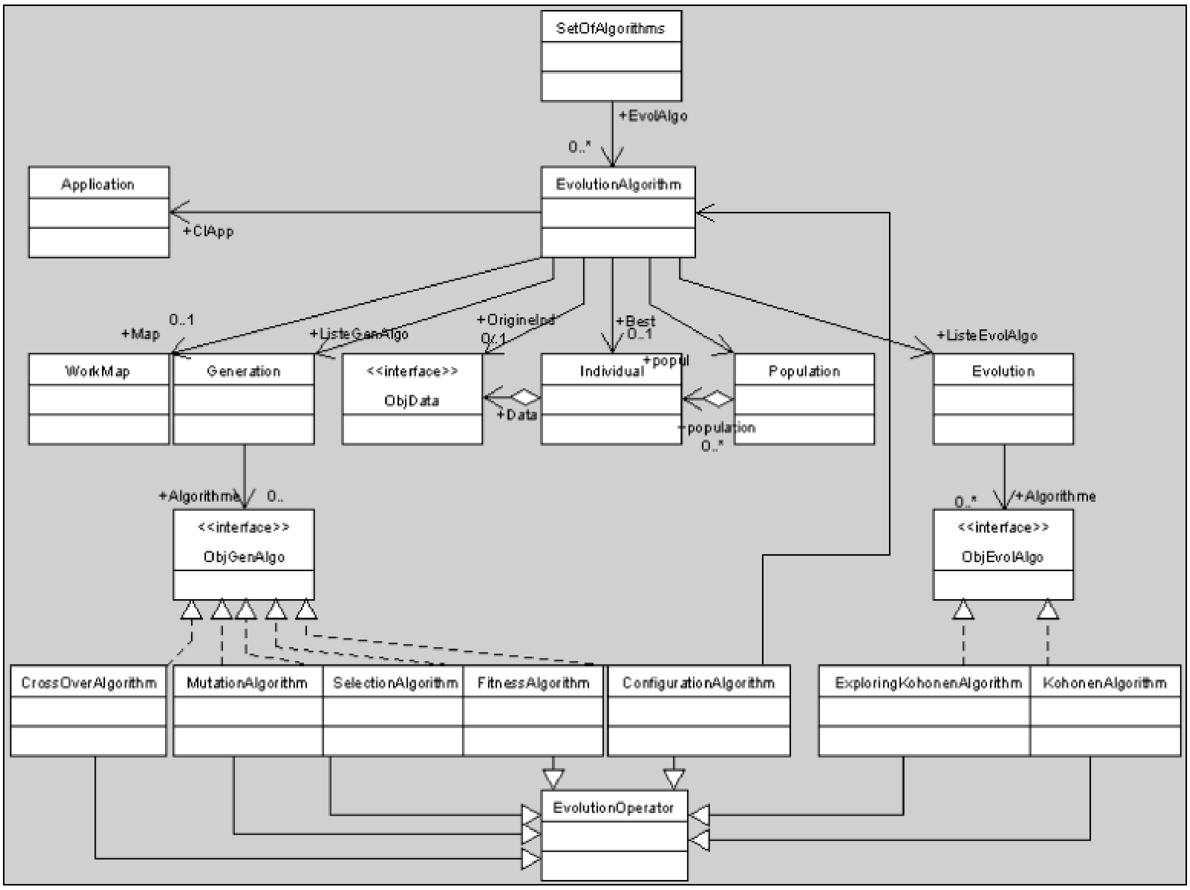


Fig. 6. Object model of the island based evolutionary framework.

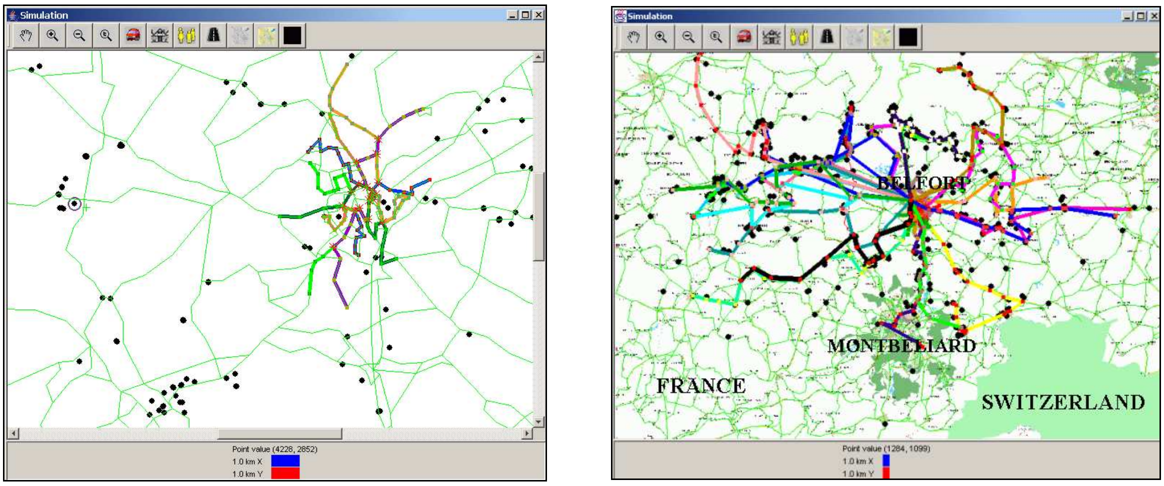
4.2 Solving a clustering VRP test case

For example, to solve the VRPTW rc102 Solomon test case (Solomon 1987) with 101 requests, illustrated within Fig. 5 and where cluster size is reduced to a single request, three island EA are built. A master EA controls two other EA and activates population movements. It regularly gets best solutions generated. The first EA quickly generates solutions which are continuously introduced as new materials on the second EA. Second EA shown in the bottom and middle of Fig. 5 applies to a 50 individuals population, a fitness/assignment operator, named *Fitness* in the figure, which assigns requests to routes and evaluates quality of solutions, two selection operators (elitist and not elitist), named *Selection*, which replace few worst (with low fitness) individuals by best ones (with high fitness), and two SOM based operators which compete one against each other. A problem specific SOM operator, named *KohonenTW*, realizes greedy insertion moves, whereas the standard SOM, named *Normal Kohonen*, acts on the transport mesh maintaining route topology and addressing length reduction. In *KohonenTW*, given an unsatisfied request (not already assigned), an insertion move simply selects the closest route transport point encountered for which request constraints are satisfied, letting other inserted requests in the route also satisfied, then it moves the point on the request location. A second *KohonenTW* version selects the route with shortest length. We used a scalar aggregative fitness function, adding two main objectives of travel length (1) and distortion (2). On the rc102 test case with 12 vehicles, length of our solution shown in Fig. 5 is 1438 for 100 satisfied requests. Length is 7.5 % lower than the length of best known rc102 solution since distortion objective (2) is introduced in our approach. A residual request was not satisfied in this example. Nevertheless, this experiment illustrates the evolutionary approach potential to solve a clustering version of the classical vehicle routing problem with time-windows.

5 Dimensioning a real life enterprise bus system

5.1 Actual bus system and goal of experiments

We now focus on a real life case problem with capacity constraints and no time windows, addressing generation of solution compromises between the two main objectives of length (1) and distortion (2). The enterprise bus routes are within a geographic area around the towns of Belfort and Montbeliard in the East of France. The area is defined by the geodesic Lambert II two-points coordinates (897990, 2324046) and (971518, 2272510). It is bit-mapped on a 9807 × 6867 map and it corresponds to an area of 73.528 km × 51.536 km, with 7.5 m by pixel.



(a) (b)

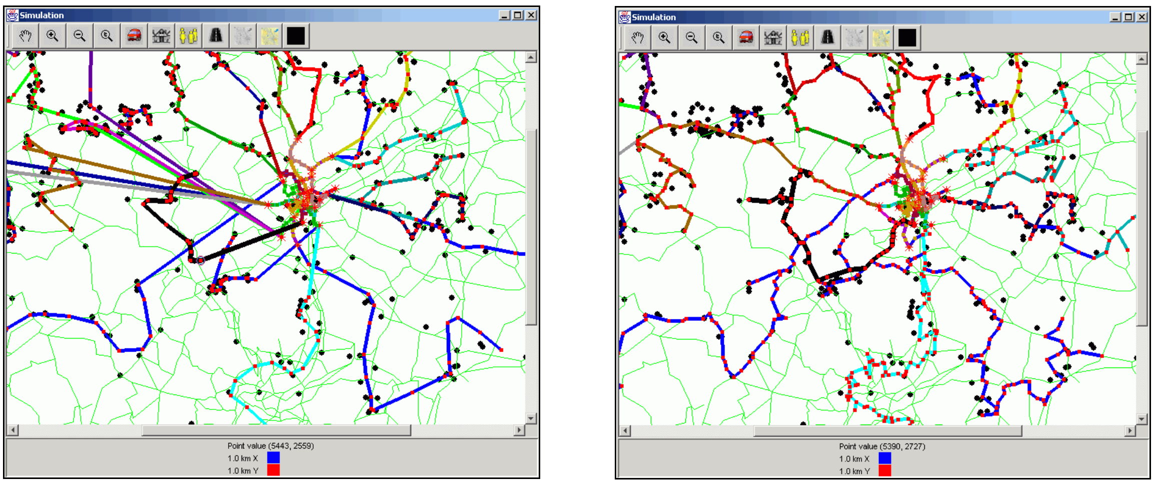
Fig. 7. (a) Zoom around internal Belfort city bus system. (b) Enterprise bus system around city of Belfort at the scale of a region.

The data consist in 780 employees of a great enterprise that are customers of the enterprise bus system. Their localization is illustrated by dots in Fig. 7(a-b). Fig. 7(a) presents the internal city bus system (i.c.b.s.) of Belfort having 7 interconnected routes. Whereas, Fig. 7(b) shows the enterprise actual bus system (a.b.s.) inside the geographic area and juxtaposed to the smallest i.c.b.s.. The a.b.s. has 23 inter-city routes to transport the 780 customers. Actual routes are connected at the enterprise location in Belfort. It is possibly the best empirically built bus system. Here, we investigate how to automatically generate new inter-city routes. We perform experiments with a varying number of routes to evaluate when length and distortion are maintained into adequate ranges as the route number diminishes. To illustrate reuse of existing infrastructures, a.b.s. and i.c.b.s. are merged into a single transport mesh, routes of the former being connected to entrance points of the latter which is the fixed part of the mesh.

But, what's about the influence of road infrastructures visualized by thin lines in Fig. 7(a-b). Certainly, road infrastructures have influence in the case of metropolitan networks. But as shown by many works, as for example (Love and Morris 1972; Tobler 1993), Euclidean nature of transportation problems remains predominant in many cases. In our case roads infrastructure are considered at a subsequent step, the final routes being projected on roads and evaluated. Letting routes simply defined by few cluster centers, or bus-stops, rather than by a complete path fitting roads, also addresses lowest computational costs. We will evaluate how road projections affect optimality. Projection on roads is done projecting cluster centers to the closest road, followed by expanding intermediate 2-point paths using a Dijsktra algorithm.

5.2 *Simulations results*

The transport meshes used in simulations have a varying number of inter-city routes, from 11 to 23 routes interconnected to entrance points of the internal city bus system of Belfort, which is defined by 7 fixed routes. Each inter-city route has 20 cluster centers. We used a single EA with a standard SOM (Kohonen 2001) operator applied to a population of 50 individuals, a fitness/assignment operator and a not-elitist selection.



(a) (b)

Fig. 8. (a) A transport mesh result with routes connected to the internal city bus system. (b) The same result expanded on streets and roads.

A simulation needed approximately 30 minutes on a Pentium workstation 1537 MHz, integrating visual feedback performed at each generation. To evaluate impact of road infrastructures, we report solutions once obtained and after being expanded on roads. The projection process is illustrated in Fig. 8(a-b). Only obtained cluster centers in routes are shown in Fig. 8(a), whereas cluster centers are projected on roads and intermediate transport points generated to match the roads in Fig. 8(b).

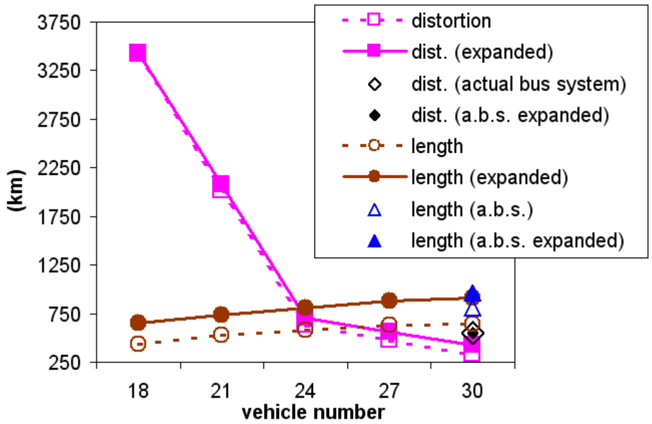
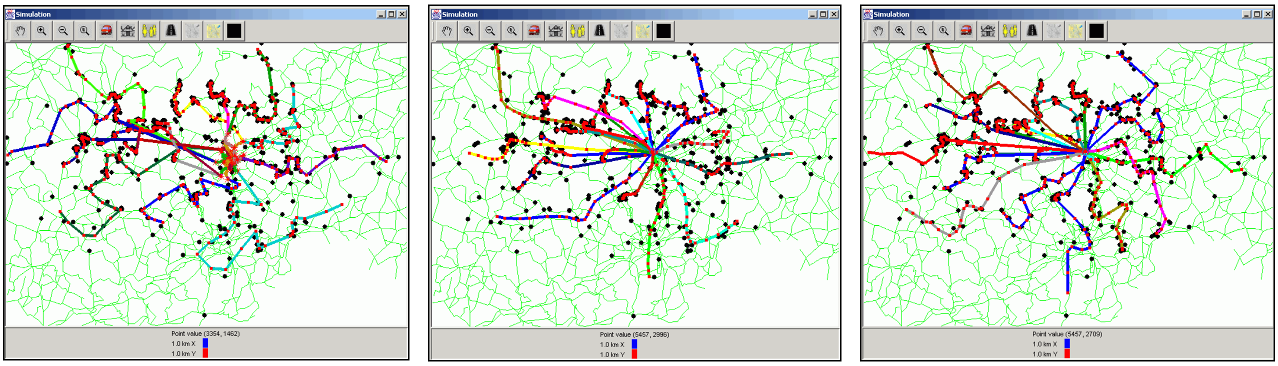


Fig. 9. Distortion and length as vehicle number increases, as obtained and once expanded on roads, compared to the actual bus system with 30 (23+7) routes.

Table 1 gives best results obtained over 3 runs for each vehicle number case. Numerical values in Table 1 present new solution values with comparison to the actual bus system and internal city bus system. First column describes the bus system configuration, second column gives the number of non empty clusters obtained. The three following columns present the length, average length and maximum length of the obtained routes. The three following columns present the distortion, average distortion by customer and maximum distance from a customer to its assigned bus stop. The last column presents the load in percentage of the total vehicle capacity. Taking into account the two main objectives of length (1) and distortion (2), main results are summarized in plots in Fig. 9. Note that a threshold of 24 vehicles (17 variable regional routes connected to the 7 fixed routes of the i.c.b.s.) is found for the new system objective values to become very similar with those of the actual bus system. Results with 30 routes have lower objective values than the ones of the existing system of i.c.b.s. plus a.b.s., in the two cases where routes are expanded or not expanded on roads. We can observe that projection and expansion on roads slightly alters length and distortion, leading to a set of potential and diverse new solutions for the enterprise bus system. As illustrated in Fig. 10, other bus system configurations can be defined to reach other compromises. Augmenting the cluster center number by routes yields to a result as in Fig. 10(a). As well, a higher priority can be given to the length objective (1) using a higher coefficient weight into the scalar fitness additive combination of length and distortion, as shown in Fig. 10(b). Conversely, a higher priority can be given to the distortion objective (2) as illustrated in Fig. 10(c).



(a) (b) (c)

Fig. 10. Other compromises between length and distortion. (a) With 14+7 routes and a higher initial number of cluster centers (27 points by route). (b) A 19 routes (20 points by route) transport mesh, with routes interconnected at the enterprise location, with length as first objective. (c) The same transport mesh with distortion as first objective.

Fig. 11. Variantes de configuration de compromis longueur/distorsion. (a) Lignes reliées au réseau urbain. (b) Longueur en tant que premier objectif. (c) Distorsion en tant que premier objectif.

Table 1. Solution compromises with a varying route number

Transp. mesh – nb. of routes	Number of clusters	Total length (km)	Average length (km)	Max length (km)	Total dist. request-vehicle (km)	Average dist. request-vehicle (km)	Max dist. request-vehicle (km)	Load (%)
Internal city bus system – 7 routes (not expanded on roads)	13	54.486	7.784	9.648	4188.623	14.959	38.606	100
Actual bus system – 23 routes (not expanded on roads)	194	754.727	32.814	48.333	554.450	0.711	12.942	75.362
a.b.s. + i.c.b.s. – 30 routes (not expanded on roads)	198	810.245	27.008	48.333	544.349	0.698	12.942	59.316
11+7 routes (not expanded)	145	438.984	24.388	54.389	3416.617	4.409	38.394	100
14+7 routes (not expanded)	204	529.471	25.213	60.752	2012.414	2.580	33.630	85.714
17+7 routes (not expanded)	259	584.696	24.362	67.784	633.270	0.812	17.333	74.641
20+7 routes (not expanded)	279	636.563	23.576	47.296	474.806	0.609	6.509	66.102
23+7 routes (not expanded)	314	638.258	21.275	50.618	331.227	0.425	5.388	59.316
Internal city bus system – 7 routes (expanded on roads)	13	59.511	8.502	10.764	4188.623	14.959	38.606	100.000
Actual bus system – 23 routes (expanded on roads)	194	905.574	39.373	57.857	554.450	0.711	12.942	75.362
a.b.s. + i.c.b.s. – 30 routes (expanded on roads)	198	965.085	32.169	57.857	544.349	0.698	12.942	59.316
11+7 routes (expanded)	128	648.587	36.033	100.733	3428.784	4.424	38.349	100
14+7 routes (expanded)	162	737.716	35.129	90.031	2078.459	2.665	33.660	85.714
17+7 routes (expanded)	188	806.383	33.599	99.697	702.749	0.901	17.295	74.641
20+7 routes (expanded)	203	874.915	32.404	66.320	554.770	0.711	6.492	66.102
23+7 routes (expanded)	214	909.886	30.330	83.040	421.948	0.541	6.002	59.316

6 Conclusion

We have presented a transport system optimizer merging user editing functionalities, to build at hand a transportation network, and optimization functionalities based on a evolutionary framework embedding self-organizing maps. The underlying concept is adaptive meshing used as an interactive tool allowing continuous user graphical feedback. Transport networks are visual patterns adapted to the demands. Here, automatic optimization concerns a part of the network that is modeled by a vehicle clustering and routing problem with capacity and time window constraints. The approach has been applied to a real life case for the transportation of the 780 employees of a great enterprise, the generated regional routes being connected to an existing urban bus system. Since the self-organizing map is a center based clustering algorithm, adding topology between cluster centers, we tried to show how it allows to naturally explore a combined minimization of route length and of distance from customers to bus stops. Next steps will consist of extending the optimization functionalities to tackle other transportation network problems as, for example, bus frequency allocation according to origin-destination demands. As well, studying application to stochastic and dynamic versions of vehicle routing problems are questions to address.

References

Arora, S. (1998). “Polynomial Time Approximation Schemes for Euclidean k-medians and related problems.” *In ACM STOC'98*.

Balas, E. (1989). “The prize collecting traveling salesman problem.” *Networks*, 19, 621-636.

Boese, K. D., Kahng, A.B., Muddu, S. (1994). “Adaptive Multi-Start Technique for Combinatorial Global Optimization.” *In Journal of Operations Research Letters*, 16, 101-113.

Christofides, N., Mingozzi, A., Toth, P. (1979). “The vehicle routing problem.” *Combinatorial Optimization*, Christofides N. et al. (eds), Wiley, 315-338.

Cochrane, E.M., and Beasley, J.E. (2003). “The co-adaptive neural network approach to the Euclidean Travelling Salesman Problem.” *Neural Networks*, 16, 1499-1525, 2003.

Créput, J.C., Koukam, A., Lissajoux, T., Caminada, A. (2005). “Automatic Mesh Generation for Mobile Network Dimensioning using Evolutionary Approach.” *IEEE Transactions on Evolutionary Computation*, 9(1), 18-30.

Edelkamp, S., Jabbar, S., Willhalm, T. (2005). “Geometric Travel Planning.” *IEEE Transactions on Intelligent Transportation Systems*, 6(1), 5-16.

Ghaziri, H. (1996). "Supervision in the Self-Organizing Feature Map: Application to the Vehicle Routing Problem." *In Meta-Heuristics: Theory & Applications*, I.H. Osman and J.P. Kelly (eds), Kluwer, Boston, 651-660.

Hansen, P., Mladenovic, N., Taillard, E.D. (1998). "Heuristic solution of the multisource Weber problem as a p-median problem." *O.R. Letters*, 22, 55-62.

Kohonen, T. (1982). "Clustering, Taxonomy, and Topological Maps of Patterns." *Proceedings of the 6th International Conference on Pattern Recognition*.

Kohonen, T. (2001). *Self-Organization Maps and associative memory*, Springer Verlag, Berlin, 3rd edition.

Kumar, P., Singh, V., Reddy, D. (2005). "Advanced Traveler Information System for Hyderabad City." *IEEE Transactions on Intelligent Transportation Systems*, 6(1), 26-37.

Labbé M., and Laporte, G. (1986). "Maximizing user convenience and postal service efficiency in post box location." *Belgian Journal of Operations Research, Statistics and Computer Science*, 26, 21-35.

Labbé, M., Laporte, G., Rodríguez Martín, I., Salazar González, J.J. (2004). "The Ring Star Problem: Polyhedral Analysis and Exact Algorithm". *Networks*, 43, 177-189.

Labbé, M., Rodríguez Martín, I., Salazar González, J.J. (2005). "Locating Median Cycles in Networks." *European Journal of Operational Research*, 160, 457-470.

Laporte, G., and Palekar, U. (2000). "Some Applications of the Clustered Traveling Salesman Problem." *Les Cahiers du GERAD*, <http://www.gerad.ca>.

Love, R.F., and Morris, J.G. (1972). "Modelling intercity road distances by mathematical functions." *Oper. Res. Quart.*, 23, 61-72.

Min, H., Jayaraman, V., Srivastava, R. (1998). "Combined location-routing problems: A synthesis and future research directions." *European Journal of Operational Research*, 108, 1-15.

Modares, A., Somhom, S., Enkawa, T. (1999). "A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems." *International Transactions in Operational Research*, 6:591-606.

Moscato, P. (1999). "Memetic Algorithms: A Short Introduction." *In New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, eds., McGraw Hill.

Renaud J., Boctor, F., Laporte, G. (2004). "Efficient heuristics for median cycle problems." *Journal of the Operational Research Society*. 55, 179-186.

Resende, M.G.C., and Ribeiro, C.C. (2002). "Greedy randomized adaptive search procedures." *In State of the Art Handbook in Metaheuristics*, F. Glover and G. Kochenberger, eds., Kluwer.

Smith, K.A. (1999). "Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research." *INFORMS Journal on Computing*, 11(1).

Solomon, M.M. (1987). "Algorithms for the vehicle routing, and scheduling problems with time window constrains." *Operations Research* 35, 254–264.

Tobler, W.R. (1993). "Speculations on the geometry of geography, in Three presentations on geographical analysis and modelling." *National Center for Geographic Information and Analysis (NCGIA) Technical Rpt 93-1*, Univ. of California, Santa Barbara, 1993.

Volgenant, T., and Jonker, R. (1987). "On some generalizations of the traveling salesman problem." *Journal of the Operational Research Society*, 38, 1073-1079.