

# Self-Organizing Maps in Evolutionary Approach meant for Dimensioning Routes to the Demand

J.-C. Créput, A. Koukam, and A. Hajjam

**Abstract**—We present a non standard Euclidean vehicle routing problem adding a level of clustering, and we revisit the use of self-organizing maps as a tool which naturally handles such problems. We present how they can be used as a main operator into an evolutionary algorithm to address two conflicting objectives of route length and distance from customers to bus stops minimization and to deal with capacity constraints. We apply the approach to a real-life case of combined clustering and vehicle routing for the transportation of the 780 employees of an enterprise. Basing upon a geographic information system we discuss the influence of road infrastructures on the solutions generated.

**Keywords**—Evolutionary algorithm, self-organizing map, clustering and vehicle routing.

## I. INTRODUCTION

FOLLOWING works that were carried out for dimensioning radio-cellular networks with the help of meshing techniques as in [9], we are proposing to extend and adapt the concept to terrestrial transport. Vehicle interconnected lines are seen as a transport mesh which adapts its shape to the underlying distribution of demands. Here, the problem consists in positioning bus stops and finding vehicle routes among them, minimizing at the same time both route length and walking distances from customer locations to bus stops, called distortion measure, taking care of capacity constraints. We call it the Euclidean vehicle clustering and routing problem. It can be seen as a combination of the Euclidean k-median problem [3] with the classical vehicle routing problem (VRP) [7].

From our knowledge, the problem has never been studied previously. In Clustered TSP [24], clusters are given as an input. The closest problem encountered, called median-cycle problem (MCP), has been investigated recently [22][23]. It is defined on graphs. It consists of finding a subset of vertices to visit within a single tour minimizing a combination of routing cost and assignment cost. In our case, cluster centers are located anywhere in the plane, many vehicle routes are used and capacity constraints have to be satisfied. Related problems looking at a cycle, path or tree on a subset of vertices use specific function costs or constraints on assignment cost [4][20][21][39]. Hierarchical combinations of clustering and routing are possibly manifold as in location-routing problems (LRP) [29]. 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. Thus, no benchmark and results are available specifically for our new

problem. It is why we will perform experiments on a real life case for which the existing bus routes will serve as a basis for comparison.

Such problems of clustering and vehicle routing are NP-Hard. Then, the use of metaheuristics is encouraged to get satisfactory results for large instances. Here, what we will focus on is on "natural" metaheuristics, because often inspired by metaphors of natural evolution and living systems, such as evolutionary algorithms [14][33] and self-organizing neural networks [17]. A common characteristic lies in their biological models having crude and blind mechanisms depicted through selection theories [10][26]. They are naturally parallel and are often recognized as easy to implement.

In practice, hybridization of methods is often compelling. It is a common and promising practice using a population based metaheuristic incorporating a neighborhood search. While neighborhood search quickly finds solutions in a small region of the search space, the embedding strategy determines interesting regions to visit. Examples of such methods are memetic algorithms [28][31] or genetic local search [32]. Memetic algorithms are special case of evolutionary algorithms, combining advantage of heuristics within a population based search. They take their denomination from cultural evolution, where "memes" are unit of information that evolve during life and replicate through knowledge transmission.

We present the memetic SOM, a population based search embedding self-organizing maps (SOM) as internal operators. Using neural networks to solve the traveling salesman problem (TSP) was done using the Hopfield model [37], the elastic nets [11][36] and the self-organizing maps (SOM) [1][2][8][17][25][38], the latter being the most efficient approach on large size problems. Many papers on SOM application to the TSP have been published since two decades as mentioned by [8], which is one of the most complete review on this subject. Their extension to more complex and abstract problems remains an important task. For example, SOM have been extended to vehicle routing problems [13][30] but, as far as we know, not to combined clustering and vehicle routing problems. We argue that the SOM naturally integrates clustering and routing within a context of stochastic demands. For this reason, we present vehicle routing from the more general viewpoint of adaptive meshing, solutions being interconnected routes adapted to the demands. Furthermore, by using the evolutionary embedding strategy, we extend the potentiality for the approach to address capacity constraints and a scalar combination of the two objectives of both length and distortion minimization.

The paper is organized as follows. Section 2 is a short

discussion about classical self-organizing map applications, and its relation to clustering and routing. Section 3 presents the clustering and routing problem. Objectives and constraints are given. Section 4 presents the standard SOM operator. Section 5 describes the population based approach. Section 6 presents experiments carried out on an artificial test case and on a real life case of urban transportation. The last section is devoted to the conclusion and further research.

## II. FROM PATTERN ANALYSIS TO CLUSTERING AND ROUTING

As it is known in pattern analysis field, the goal when achieving data compression, density estimation or regression is to extract from learning data their main characteristics in order to favor generalization. Two drawbacks have to be avoided during regression: over-fitting and under-fitting [6]. In the former case, noise in data is over-estimated. In the latter case, main characteristics are loose from data. Finding the right balance between these two aspects is an essential challenge that, for example, Kohonen's self organizing maps [17][18] can address.

Kohonen's algorithm is a nonparametric regression, related to kernel smoothing and curve fitting [34], based on "code-books" which are spatially distributed vectors representing data. It can be understood as a center based clustering algorithm, adding topological relationships between cluster centers. Thus, related optimization problems it can address, are clustering k-median, k-center or k-mean problems [16][27]. Advantage of the SOM is to maintain topological relationships between cluster centers, and this property allows to integrate routing between them in a unified approach.

As already known, SOM provides an interesting solution for the Euclidean TSP, but it seems natural also to address a wider class of problems adding a level of clustering on vehicle routing problems. By incorporating SOM into an evolutionary algorithm the aim is to address such harder problems with capacity constraints and improve performances as well.

## III. PROBLEM STATEMENT

We denote by  $V = \{r_1, \dots, r_n\}$ , the finite set of customer demands, called requests. Each request  $r_i \in V$  has a location in the plane. It has a non-negative quantity demand  $q_i$ . Let  $B = \{n_1, \dots, n_k\}$  being a finite set of cluster centers, also called transport points or bus-stops, localized by their coordinates in the plane. A transport mesh, or set of interconnected routes, is a collection  $R = \{R_1, \dots, R_m\}$  of  $m$  routes where each route is a sequence  $R_i = (n_0^i, \dots, n_j^i, \dots, n_{k_i}^i)$ ,  $n_j^i \in B$ , of  $k_i+1$  successive cluster centers. 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. Thus, each request  $r$  must be assigned to a single cluster center, denoted  $n_r$ , in one of the  $m$  routes and several requests are possibly assigned to a same cluster center. To each route is associated a single vehicle. We then denote and identify a vehicle with its route  $R_i$ . The vehicle assigned requests are

assigned to its transport points. Each vehicle has a load  $L_i$  defined as the sum of its assigned request quantities. Vehicles have capacity  $C$ . Let  $d(p, p')$  denotes the Euclidean distance between two points  $p$  and  $p'$  of the plane.

**Euclidean vehicle clustering and routing problem (VCRP).** The problem input is given by a set of  $n$  requests  $V = \{r_1, \dots, r_n\}$ , and a set of  $m$  routes  $R = \{R_1, \dots, R_m\}$ .

The problem output consists of finding cluster center locations in the plane, except for some fixed transport points at a depot location, and an assignment of the requests to cluster centers in routes, in order to minimize the two following objectives:

$$length = \sum_{i=1, \dots, m} \sum_{j=0, \dots, k_i-1} d(n_j^i, n_{j+1}^i), \quad (1)$$

$$distortion = \sum_{i=1, \dots, n} d(r_i, n_{r_i}), \quad (2)$$

subject to the capacity constraint:

$$L_i \leq C, \quad i \in \{1, \dots, m\}. \quad (3)$$

Since the problem has two conflicting objectives of both length (1) and distortion (2), we have to take care of what is called an optimal solution. We say that a first (admissible) solution dominates an other (admissible) solution if the two objectives of the former are inferior to those of the latter, one of them being strictly inferior. Then, an optimal solution is a solution which is dominated by no other solution. The set of such non comparable optimal solutions are called Pareto optimal solutions or optimal non dominated solutions.

The set of Pareto optimal solutions are possibly numerous. Solutions with  $distortion = 0$  and minimum  $length$  are solutions of an Euclidean VRP. Solutions with minimum  $distortion$  (discarding  $length$ ) are solutions of the well-known Euclidean k-median problem. Whereas, considering the two objectives simultaneously yields to non comparable solutions which are possibly not a VRP nor a k-median optimal solution. These intermediate solutions are the compromises that are useful for combined clustering and routing.

## IV. KOHONEN'S SELF-ORGANIZING MAPS

### A. The Standard Self-Organizing Map Algorithm

The self organizing map is a non supervised learning procedure performing a non parametric regression that reflect topological information of the input data [17]. The algorithm has batch versions [34] where all data are supposed to be known in advance. The SOM is sometimes presented as an optimization problem minimizing an explicit cost function [15]. Here, we present the standard on-line algorithm used within a two-dimensional context, using the request set as the input data set. It can be seen as a stochastic approximation algorithm, tackling input points one by one.

The topological map is a non directed graph  $G = (N, E)$  in which each vertex  $n \in N$  is a neuron having a synaptic weight vector  $w_n = (x, y) \in \mathfrak{R}^2$ , where  $\mathfrak{R}^2$  is the two-dimensional Euclidean space. Synaptic weight vector corresponds to the vertex position in the plane. The set of

neurons  $N$  is provided with the  $d_G$  induced canonical metric,  $d_G(n, n') = 1$  if and only if  $(n, n') \in E$ , and with the usual Euclidean distance  $d(n, n')$ . The  $F_n$  influence field of a neuron  $n \in N$  is defined by its Voronoï region within the Euclidean space:

$$F_n = \left\{ w \in \mathcal{R}^2 / \forall n' \in N, n' \neq n, d(w, w_n) < d(w, w_{n'}) \right\}. \quad (4)$$

The training procedure follows three basic steps. At each iteration  $t$ , a point  $p(t) \in \mathcal{R}^2$  is extracted from the set of learning data (extraction step). Then, a competition between

neurons is performed on various mesh configurations with different intensities and neighborhood sizes are shown in Fig. 1. We can see in (a)-(d) how neighborhood influence is subdivided along the interconnected paths, and in (e)-(h) standard iterations on a single path.

A SOM simulation is denoted by its running parameters  $(\alpha_{init}, \alpha_{final}, \sigma_{init}, \sigma_{final}, t_{max})$ . It becomes an operator into the population based algorithm.

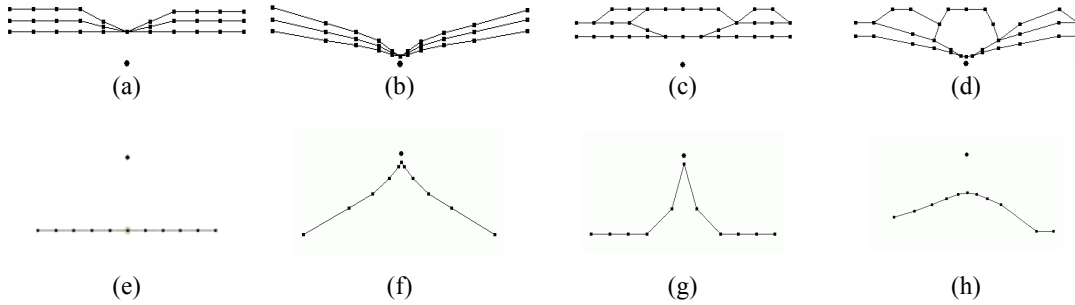


Fig. 1 A single iteration of SOM algorithm with different radius  $r$  and learning rate  $\alpha$ , on interconnected routes. (a)(c)(e) Initial configurations. (b)(d)  $r = 4, \alpha = 0.75$ . (f)  $r = 4, \alpha = 0.9$ . (g)  $r = 1, \alpha = 0.9$ . (h) With asymmetric neighborhood  $r = 5;3, \alpha = 0.5$

neurons against the input point  $p$  is performed in order to select a winner neuron (competition). Usually, it is the neuron  $n^*$  for which  $p \in F_{n^*}$ , i.e. the closest neuron to  $p$ . Finally, the learning rule (triggering step)

$$w_n(t+1) = w_n(t) + \alpha(t) \cdot h_t(n^*, n) \cdot (p(t) - w_n(t)) \quad (5)$$

is applied, with learning rate  $\alpha(t)$  and function profile  $h_t$ , to  $n^*$  and to all neurons into a finite neighborhood of  $n^*$  of radius  $\sigma(t)$  in the sense of the topological distance  $d_G$ .

Function profile  $h_t$  has the shape of a "bell curve". That function models the biological lateral interactions between neurons relative to the  $d_G(n^*, n)$  distance. Both  $\alpha(t)$  and  $h_t$  coefficients are time decreasing functions which play a similar role as the "gains" used in stochastic approximation [19], resembling to temperature. It is usual to consider the process as a succession of two phases: an exploratory or ordering phase followed by an asymptotic smooth gradient descent phase to stable states. Here, we used geometrically decreasing learning rate  $\alpha(t)$  and radius  $\sigma(t)$ , multiplying coefficient at each iteration by

$$\exp\left(\ln(x_{final}/x_{init})/t_{max}\right) \text{ applied with } \alpha_{init}, \alpha_{final},$$

$\sigma_{init}, \sigma_{final}$ , respectively the values at starting and final iteration  $t_{max}$ . The activation profile is given by the Gaussian

$$h_t(n^*, n) = \exp\left(-d_G(n^*, n)^2 / \sigma_t^2\right). \quad (6)$$

Application of SOM to a set of interconnected routes  $R = \{R_1, \dots, R_m\}$  is straightforward. The data distribution is the request set. The standard SOM algorithm is applied to the induced undirected graph  $G_R = (N, E)$  from  $R$ , where the vertex set  $N$  is the set of cluster centers, whereas  $E$  is the set of edges composed of any two successive centers from routes. Examples of a basic

### B. Self-Organizing Maps Properties

Self-organizing maps have two main properties. The first expresses preservation of the data density distribution: the peaks of the grid are the most dense where data are also the most dense. The second one expresses preservation of topological distances: two peaks on the grid that are beside each other using distance  $d_G$ , are also closest one to each other using Euclidean distance. However, SOM analysis in statistical mechanics remains a difficult task and main results presented by [35] show that density of vertices will not reflect proportionally the underlying data density. It is admitted that Kohonen's algorithm tends to under-sample high probability regions and over-sample low probability ones. Generating a topological map can also be viewed as an optimizing problem minimizing an explicit cost function. The cost function is given by

$$E = \frac{1}{2} \sum_i \sum_n h(n_{x_i}, n) \cdot \|x_i - w_n\|^2, \quad (7)$$

where  $x_i$  denotes an input point,  $n_{x_i}$  is the closest vertex to  $x_i$ ,  $h$  denotes the activation profile and where weights  $w_n$  are the parameters to be determined. It has been shown that SOM algorithm performs a gradient descent on  $E$  when applied to a discrete input data set, whereas it only minimizes the function cost approximately in the continuous case [12]. It is admitted that minimizing the cost function  $E$  is a hard task because of numerous local minima that should be bypassed throughout the optimization process.

### V. THE MEMETIC SOM

The first improvement one could introduce using SOM is multi-start or re-start runs from random initializations. Furthermore, one can follow metaphor of biological selection applied to a population of solutions. Here, we use

the standard SOM algorithm as a local search process, embedded into an evolutionary algorithm having fitness evaluation and selection operators. Large learning rates and neighborhood sizes will help exit from local minima and generate diversity, whereas smoothly decreasing parameters will fix the runs toward stable states representing a compromise of route length and distortion minimization. Fitness and selection operators will address specific problem goals.

```

Initialize population with Pop randomly generated
individuals.
Do while not Gen generations are performed.
  For each individual.
    Apply self-organizing map operators SOM.
    Apply request assignment and fitness evaluation operator
    FITNESS.
    Apply selection operator SELECT.
  End for.
End do.
Report best individual encountered.

```

Fig. 2 The memetic loop embedding SOM

The memetic loop is presented in Fig. 2. As usual, one individual represents exactly one solution, that is, a single transport mesh. The loop consists of applying at each iteration, called a generation, a set of operators to a population of solutions. Each operator can be a self-organizing map, denoted *SOM*, a fitness evaluation and requests assignment, denoted *FITNESS*, or a selection operator *SELECT*. Each operator has a probability of application *prob*. Details of operators are the followings:

• Request assignment operator *FITNESS*. It greedily assigns requests to their closest cluster center into a route, vehicle capacity constraint being satisfied. Then, it evaluates a scalar fitness value that has to be maximized and which is used by the selection operator. The capacity constraint (3) is greedily tackled thru the requests assignment. Other objectives (1) and (2) are traduced and combined into the scalar fitness value  $fitness = -\alpha \times length - \beta \times distortion$ , where  $\alpha$  and  $\beta$  are weighting coefficients.

• Selection operator *SELECT*. Based on fitness maximization, the operator denoted *SELECT* replaces worst individuals, which have the lowest fitness values in the population, by the same number of best individuals, which have the highest fitness values in the population.

With capacity constraint, as a bin packing problem, requests assignments to vehicles is by itself a NP-hard problem. Here, we choose to systematically perform a greedy assignment to the closest transport point encountered for which vehicle capacity constraint is satisfied. To perform  $n$  closest point findings in expected  $O(n)$  time for uniform distributions, we have implemented the spiral search algorithm of Bentley, Weide and Yao [5] based on a cell partitioning of the area. Therefore, since the number of transport points is set proportional to the number of requests  $n$ , and assuming that the initial neighborhood size is set to a constant value, the memetic algorithm performs in expected  $O(n)$  time for uniform distributions, and  $O(n)$  space complexity. Worst case computation time is  $O(n^2)$  as usual with SOM.

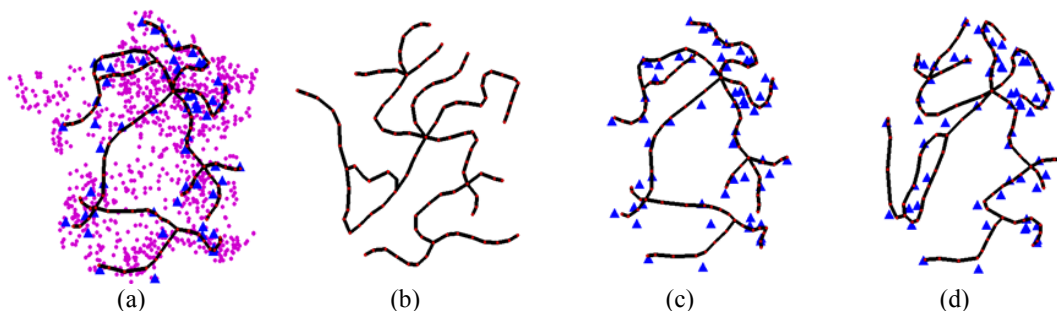


Fig. 3 Meshing of a territory. (a) Sampling of the demand (1000 dots) on a territory with 40 specific places (triangles) and a juxtapose transport mesh result. (b) Transport mesh obtained considering only the demand. (c)(d) Transport mesh derived considering the 40 places

- Self-organizing map operator *SOM*. It is the standard SOM applied to the graph network. It is characterized by its internal parameters presented in section 0 and its name, as  $SOM(\alpha_{init}, \alpha_{final}, \sigma_{init}, \sigma_{final}, t_{max})$ . It is applied running  $niter$  basic iterations by individual at each generation. Parameters are fixed before each run. One or more instances of the operator can be combined with their own parameters. Parameter  $t_{max}$  is the total amount of iterations applied to all individuals. Once  $t_{max}$  are performed, the operator resets to its initial parameters and starts again.

## VI. EXPERIMENTAL RESULTS

### A. Transport Mesh Adaptation to Stochastic Demands

Here, we illustrate the visual specificity of SOM discarding other operators. The simulator was made up with Java and it incorporates a graphical interface to allow user interactive control and visual feedback. Using visual patterns as intermediate structures that adapt and distort according to demands have several advantages. It takes into account the 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.

The Fig. 3 illustrates the meshing of a geographic

territory obtained by executing SOM operators on the graph, *Nonlinear* incorporates a Geographic Information System (GIS) in order to geo-localize requests and to take into account the underlying road infrastructures. The area is defined by the geodesic Lambert II two-points coordinates (897.870, 2324.270) and (972.220, 2272.290). It corresponds to an area of 74 km by 51 km. The goal of the following experiments will be to generate solutions with less vehicles than the actual bus system and having lowest

length (1) and distortion (2). Furthermore, we will evaluate influence of the fitness function choice and show that route projection on road infrastructures globally preserves optimality.

TABLE I  
MEMETIC SOM TYPICAL OPERATOR PARAMETERS

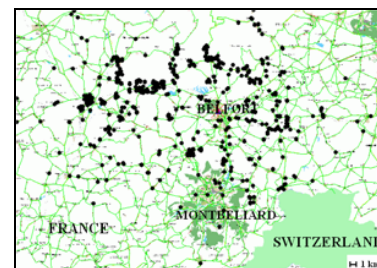
Operators		Internal SOM parameters				Evolutionary parameters		
		$\alpha_{init}$	$\alpha_{final}$	$\sigma_{init}$	$\sigma_{final}$	$t_{max}$	$prob$	$niter$
1	<i>SOM</i>	0.9	0.05	6	2	100000	1.0	1000
2	<i>SOM</i>	0.9	0.9	4	4	-	0.1	10
3	<i>SOM</i>	0.2	0.05	2	2	10000	1.0	1000

randomly into a rectangle area delimiting the set of demands. The transport mesh is composed of few interconnected routes with a total of 118 cluster centers. Its graph network is represented after running a two phases simulation in Fig. 3(a)-(d). It is built taking as input only the demand distribution in (b), and considering the 40 specific places in (c)(d). This example shows that dense zones are more driven through and that the whole territory is globally covered by the mesh.

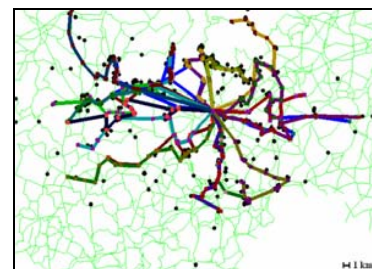
Simulations are done in two phases, using the three operators of Table I into a memetic loop with 10 individuals, running 100 generations. The first phase uses *SOM* no. 1 and *SOM* no. 2 operators with the 1000 demand points as input. Second phase uses *SOM* no. 3 and *SOM* no. 2 operators adding the 40 specific places (triangles in Fig. 3) in the input. The places (triangles) are added by introducing small-size high density zones within the whole data distribution. Second phase starts from solutions obtained by the first phase, as the one shown in Fig. 3(b), and makes the mesh slowly deform and adapt according to places, as in Fig. 3(c)(d). Operator *SOM* no. 2 punctually introduces large moves to help exit from possibly undesirable states, whereas *SOM* no. 1 and *SOM* no. 3 address length and distortion minimization, generating compromises between the two objectives. Note that *SOM* no. 1 has an initial large neighborhood in order to first deploy the initial random transport mesh. It can be imagined that the demand is more or less fluctuating and that the network is constantly adapting to such variations and fluctuations.

#### B. Clustering and Routing for the Customers of a Great Enterprise

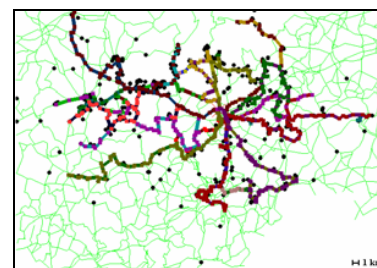
From our knowledge, the vehicle clustering and routing problem has never been studied previously considering both length (1) and distortion (2) objectives simultaneously. Thus, to present a detailed study using a benchmark of reasonable size, we apply the approach to a real life case problem with 780 requests dispatched all over a geographic area around the towns of Belfort and Montbéliard in the East of France. We don't know the optimal value, but have the actual real-life case solution as a best-known value. The



(a)



(b)



(c)

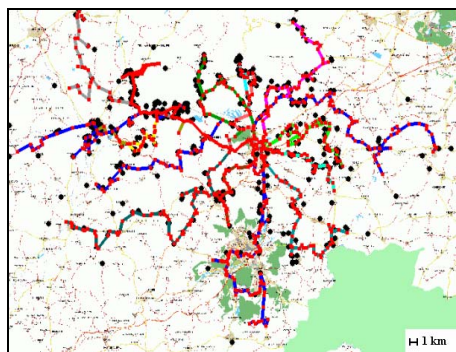
Fig. 4 (a) Distribution of requests on the geographic area. (b) The 23 bus routes of the actual enterprise bus system. (c) The same 23 routes projected on roads



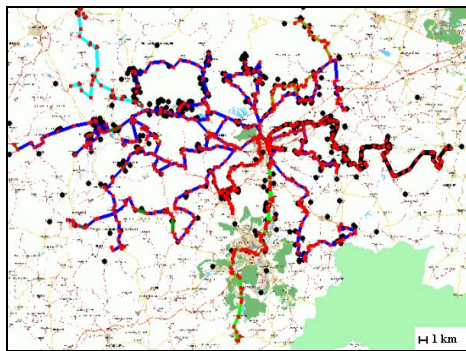
MEMETIC SOM LOOP FOR THE VEHICLE CLUSTERING AND ROUTING PROBLEM

<i>Pop</i> = 10; <i>Gen</i> = 1000									
Operators		Internal SOM parameters				Evolutionary parameters			
Name		$\alpha_{init}$	$\alpha_{final}$	$\sigma_{init}$	$\sigma_{final}$	$t_{max}$	<i>prob</i>	<i>niter</i>	<i>replace</i>
1	<i>SOM</i>	0.9	0.1	6	1	100000	1.0	100	-
2	<i>FITNESS</i>	-	-	-	-	-	1.0	1	-
3	<i>SELECT</i>	-	-	-	-	-	0.1	1	1

The data consist in 780 workers of a great enterprise that are customers of the enterprise bus system. Their locations are illustrated by dots in Fig. 4(a). Fig. 4(b-c) presents the actual bus routes, showing routes as an ordering of bus-stops in (b) and showing the same routes projected on the underlying roads (thin lines on the figure) in (c). All the routes converge to a (fixed) depot point at the enterprise location. The projection on roads is done projecting cluster centers to their closest points in roads, followed by expanding intermediate 2-point paths with a Dijkstra algorithm. The actual solution has 23 vehicles with a capacity of 45 customers each. Here, the memetic SOM is applied to a transport mesh having 20 vehicles with the same capacity of 45 customers each, hence for a total capacity of 900 customers.



(a)

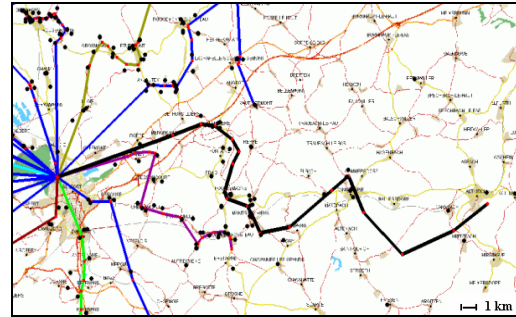


(b)

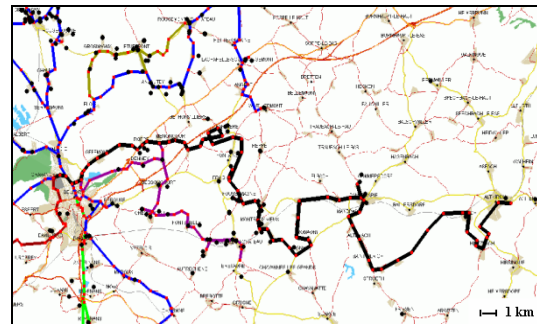
Fig. 5 Two examples of projected solutions with 20 lines

The transport mesh used in simulations has 20 routes with 20 cluster centers each. Routes are connected at a common center (the depot) fixed at the enterprise location, which participates to the neighborhood propagation. The initial vertex coordinates of the transport mesh are generated randomly into a rectangle area delimiting the request set. We fixed the operators and their parameters as in Table II. Two results projected on roads are shown in Fig. 5. They

illustrate how diversity takes place in the generated population.



(a)



(b)

Fig. 6 (a) Viewing routes as an ordering of cluster centers. (b) Viewing the projected routes on roads

The projection process is illustrated in Fig. 6(a-b) applying a zoom on the right part of the Fig. 5(b). The solution obtained by the algorithm, where routes are modeled by an ordering of cluster centers, is shown in Fig. 6(a). The same part of the solution is shown after projection on roads in Fig. 6(b).

To quantitatively evaluate the influence of the fitness function choice and the impact of road projections, we used three fitness functions  $f_1$ ,  $f_2$ , and  $f_3$  using different weighting coefficients for the two objective values of routes length (1), abbreviated by  $l$ , and distortion (2), denoted  $d$ . Function  $f_1$  is an additive aggregation of the two objectives, it is given by  $f_1 = -l - d$ . Function  $f_2$  is the length value ( $f_2 = -l$ ), whereas  $f_3$  is the distortion value ( $f_3 = -d$ ). We performed 10 runs with each fitness function. Each run took approximately 10 mn on a AMD Athlon 2 GHz computer. Results are reported in Table III. Rows present values for the “hand made” actual bus system followed by the results for each fitness function case, respectively for the solutions once obtained and after projection on roads. For each test case, we report the mean and standard deviation within parenthesis and the best result

TABLE III

SOLUTION COMPROMISES OBTAINED WITH 20 VEHICLES

Fitness – nb. of vehicles		Bus stops number	Total length (km)	Average length (km)	Max length (km)	Total distance request-vehicle (km)	Average distance request-vehicle (km)	Maximum distance request-vehicle (km)
Hand made bus system – 23 vehicles (not projected on roads)		251	750.33	32.62	47.87	509.79	0.654	13.08
$f_1$ - 20 vehicles (not projected)	Avg.	283(6.4)	570.25(10.7)	28.51(0.5)	50.39(4.9)	437.57(31.1)	0.561(0.04)	14.75(4.8)
	Best	292	559.57	27.98	44.45	407.61	0.523	19.89
$f_2$ - 20 vehicles (not projected)	Avg.	205.5(14.1)	444.67(7.2)	22.24(0.4)	38.50(1.8)	1345.85(168.5)	1.725(0.2)	23.11 (1.9))
	Best	184	433.65	21.68	36.4	1483.4	1.902	22.28
$f_3$ - 20 vehicles (not projected)	Avg.	281.8(4.3)	591.90(34.5)	29.6(1.7)	51.35(6.5)	446.38(50.6)	0.572(0.06)	15.54(6.4)
	Best	282	634.19	31.71	67.92	376.30	0.482	11.02
Hand made bus system – 23 vehicles (projected on roads)		268	910.62	39.59	57.99	445.16	0.571	11.28
$f_1$ - 20 vehicles (projected)	Avg.	260.7(5.5)	824.05(19.7)	41.20(1.0)	76.75(7.9)	637.11(91.1)	0.817(0.12)	19.45(1.5)
	Best	271	818.85	40.94	70.48	456.95	0.586	17.69
$f_2$ - 20 vehicles (projected)	Avg.	213.6(13.3)	703.07(31.5)	35.15(1.6)	73.39(15.8)	1276.25(178.7)	1.636(0.22)	23.77(2.5)
	Best	198	651.74	32.59	60.06	1512.74	1.939	24.98
$f_3$ - 20 vehicles (projected)	Avg.	258.3(7.8)	866.63(54.2)	43.33(2.7)	85.11(17.1)	661.31(138.9)	0.848(0.18)	21.15(3.8)
	Best	265	831.99	41.6	70.90	382.17	0.490	19.70

found over 10 runs. The first column of Table III indicates the type of network and/or fitness. Second column reports the number of bus stops obtained, that is, the number of non empty clusters. The three following columns report the total length, average length by route and maximum length of a route. The last 3 columns present the total distortion, which is the total distance from request locations to their assigned bus stops, the average distortion by customer, which has a more natural meaning as a average walking distance by customer, and the maximum walking distance of a customer. Numerical values in Table III let us compare new solutions with the “by-hand made” actual bus-system. The formers have only 20 vehicles whereas the latter has 23 vehicles for similar objective values.

To help evaluate spread present into the results and impact of road projections, values for the 30 runs before and after projection are drawn in plots of Fig. 7(a-c). We can observe that the projection process yields to a substantial increase of the route lengths but globally preserves lengths variation. Conversely, it slightly alters the distortion values but introduces a greater variation between them. Nevertheless, all solutions generated are non dominated by the actual hand made bus system, whereas the solution no. 3 with the  $f_3$  fitness function, shown in Fig. 7(b-c), clearly dominates the actual bus system on both the two objectives of length and distortion. This indicates that the approach has the potential to yield a set of competitive new solutions for the enterprise bus system with less vehicles.

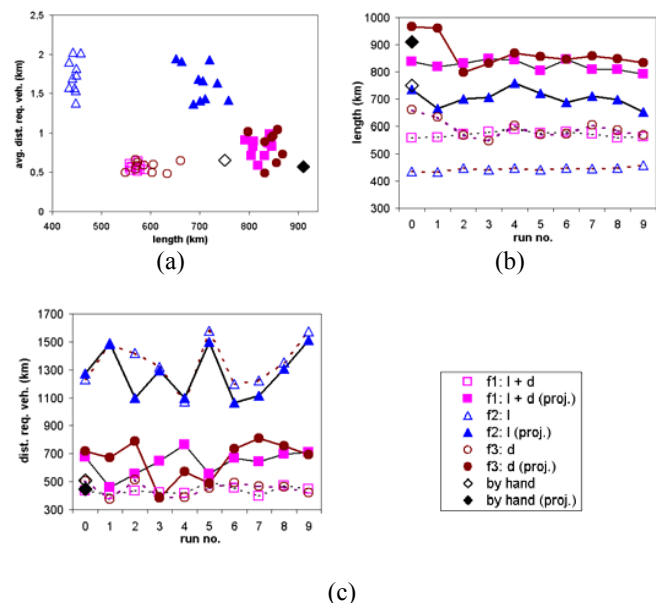


Fig. 7 (a) Correlation between average distance from customers to bus-stops (avg. distortion) and total routes length. (b) Lengths before and after projection for each run and each fitness function choice. (c) Distortion before and after projection for each run and each fitness function choice

We have presented a new combined Euclidean clustering and vehicle routing problem with capacity constraints. To solve the problem, we used the self-organizing map embedded into an evolutionary loop. The approach considers deformable templates of interconnected routes, where cluster centers defining routes are possibly fixed or mobile, or shared between routes. Solutions are visual patterns moving and adapting to the demand distribution. Considering a real life case application with 780 customers, the algorithm generates competitive solutions for the capacitated vehicle clustering and routing problem having less vehicles than the actual real solution. The following steps will consist of applying the approach to other vehicle routing problems. Dynamic and stochastic versions of such routing problems seem to be good candidates for further research, since visual patterns have by their own a potential to address adaptability to a fluctuating demand.

## REFERENCES

- [1] Angéniol, B., G. de La Croix Vaubois, and J.Y. Le Texier. (1988). "Self-Organizing feature Maps and the Travelling Salesman Problem". *Neural Network*, vol. 1, pp. 289-293.
- [2] Aras, N., B.J. Oommen, and I.K. Altinel. (1999). "The Kohonen network Incorporating Explicit Statistics and its Application to the Travelling Salesman Problem". *Neural Networks*, vol. 12, no 9, pp. 1273-1284.
- [3] Arora, Sanjeev. (1998b). "Polynomial Time Approximation Schemes for Euclidean k-medians and related problems". In *ACM STOC'98*.
- [4] Balas, E. (1989). "The prize collecting traveling salesman problem". *Networks*, 19, 621-636.
- [5] J. L. Bentley, B. W. Weide, and A. C. Yao, "Optimal Expected Time Algorithms for Closest Point Problems," *ACM Transactions on Mathematical Software*, vol. 6, no. 4, pp. 563-580, December 1980.
- [6] Bishop, C.M. (1985). "Neural Networks for Pattern Recognition". Oxford University Press, ISBN-0-19-853849-9.
- [7] Christofides, N., A. Mingozzi, and P. Toth. (1979). "The vehicle routing problem". In Christofides N. et al. (eds), *Combinatorial Optimization*, Wiley, 315-338.
- [8] E. M. Cochrane and J. E. Beasley, "The co-adaptive neural network approach to the Euclidean Travelling Salesman Problem," *Neural Networks*, vol. 16, pp.1499-1525, 2003.
- [9] Créput, J.C., A. Koukam, T. Lissajoux, and A. Caminada. (2005). "Automatic Mesh Generation for Mobile Network Dimensioning using Evolutionary Approach". *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 18-30.
- [10] Darden, L. and J. A. Cain. (1989). "Selection type Theories". *Philosophy of Science*, Vol. LVI.
- [11] Durbin, R. and D.J. Willshaw. (1987). "An Analogue Approach to the Traveling Salesman problem using an Elastic Net Method". *Nature* 326, pp 689-691.
- [12] Erwin, E., K. Obermayer and K. Schulten. (1992). "Self-organizing maps: Ordering, convergence properties and energy functions". *Biological Cybernetics*, vol. 67, pp. 47-55.
- [13] 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.
- [14] Goldberg, David E. (1991). "Genetic Algorithms". Addison-Wesley USA.
- [15] Graepel, T., M. Burger and K. Obermayer. (1998). "Self-organizing maps: Generalisations and new optimisation techniques". *Neurocomputing*, vol. 20, pp. 173-190, 1998.
- [16] Kariv, O. and S.L. Hakimi. (1979). "An algorithmic approach to network location problems; part 2. The p-median". *SIAM Journal on Applied Mathematics* 37, 539-560.
- [17] Kohonen, T. (2001). "Self-Organization Maps and associative memory". Springer Verlag, Berlin, 3rd edition.
- [18] Kohonen, T. (1982). "Clustering, Taxonomy, and Topological Maps of Patterns". *Proceedings of the 6th International Conference on Pattern Recognition*.
- [19] Kushner, H.J. and D.S. Clark. (1978). "Stochastic Approximation Methods for Constrained and Unconstrained Systems". New York, Berlin: Springer-Verlag.
- [20] Labbé M. and G. Laporte. (1986). "Maximizing user convenience and postal service efficiency in post box location". *Belgian Journal of Operations Research, Statistics and Computer Science*, 26, 21-35, 1986.
- [21] Labbé, M., G. Laporte and I. Rodríguez Martín. (1998). "Path, tree and cycle location". In *Fleet Management and Logistics*, T. Crainic and G. Laporte (eds), Kluwer, Boston, 187-204.
- [22] Labbé, M., I. Rodríguez Martín, and J.J. Salazar González. (2005). "Locating Median Cycles in Networks". *European Journal of Operational Research*, 160:457-470.
- [23] Labbé, M., G. Laporte, I. Rodríguez Martín, and J.J. Salazar González. (2004). "The Ring Star Problem: Polyhedral Analysis and Exact Algorithm". *Networks*, vol. 43, pp. 177-189.
- [24] Laporte, G. and U. Palekar. (2000). "Some Applications of the Clustered Traveling Salesman Problem". *Les Cahiers du GERAD*, <http://www.gerad.ca>.
- [25] Leung, K.S., H.D. Jin, and Z.B. Xu. (2004). "An expanding self-organizing neural network for the travelling salesman problem". *Neurocomputing*, 62, 267-292.
- [26] Manderick, B. (1992). "Selectionist Systems as Cognitive Systems". *Proceedings of the First European Conference on Artificial Life, ECAL'91*, MIT Press, Cambridge, MA.
- [27] Megiddo N. and K.J. Supowit. (1984). "On the complexity of Some Common Geometric Location Problems". *SIAM Journal on Computing* 13, 182-196.
- [28] Merz, P. and B. Freisleben. (2002). "Fitness Landscape and Memetic Algorithm Design", in F. Glover und G. Kochenberger (Herausgeber), *Handbook of Metaheuristics*, pages 321-353, Kluwer Academic Publishers, Norwell, MA.
- [29] Min, H., V. Jayaraman, and R. Srivastava. (1998). "Combined location-routing problems: A synthesis and future research directions". *European Journal of Operational Research* 108:1-15.
- [30] Modares, A., S. Somhom, and T. Enkawa. (1999). "A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems". *International Transactions in Operational Research*, 6:591-606.
- [31] Moscato, P. (1999). "Memetic Algorithms: A Short Introduction". In *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, eds., McGraw Hill.
- [32] Mühlenbein, H. (1991). "Evolution in Time and Space – The Parallel Genetic Algorithm". In G. Rawlins (Ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, Los Altos, CA.
- [33] Mühlenbein, H., M. Georges-Schleuter, and O. Kramer. (1988). "Evolution algorithms in combinatorial optimization". *Parallel Computation* 7, 65-85.
- [34] Mulier, F. and V. Cherkassky. (1995). "Self-Organization as an iterative kernel smoothing process", *Neural Computation*, 7:1165-1177.
- [35] Ritter, H. and K. Schulten. (1986). "On the stationary state of Kohonen's self-organizing sensory mapping". *Biological Cybernetics*, vol. 54, pp. 99-106.
- [36] Simic, P.D. (1990). "Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimizations". *Network* 1:89-103.
- [37] Smith, K. (1996). "An Argument for Abandoning the Traveling Salesman Problem as a Neural Network Benchmark". *IEEE Transactions on Neural Networks*, Vol. 7, No. 6.
- [38] Vieira, F.C., A.D. Doria Neto, and J.A. Ferreira Costa. (2003). "An efficient approach to the travelling salesman problem using self-organizing maps". *International Journal of Neural Systems*, vol. 2, pp. 59-66.
- [39] Volgenant, T. and R. Jonker. (1987). "On some generalizations of the traveling salesman problem". *Journal of the Operational Research Society*, 38, 1073-1079.