Self-organization in Evolution for the Solving of Distributed Terrestrial Transportation Problems

Jean-Charles Créput and Abderrafiaâ Koukam

Systems and Transportation Laboratory
University of Technology of Belfort-Montbeliard, 90010 Belfort, France
{Jean-Charles.Creput, Abder.Koukam}@utbm.fr

1 Introduction

The method presented in this chapter has its origin in adaptive meshing, using planar honeycomb structures as a tool to dimension radio-cellular network according to mobile traffic (Créput et al. 2000, 2005; Créput and Koukam 2006). Here, the approach has been transferred and generalized to a terrestrial transportation context. The transport mesh is a geometric structure, in the plane, that adapts and modifies its shape according to traffic demands. By separating the transportation network from the underlying demands, the approach theoretically allows to deal with noisy or incomplete data as well as with fluctuating demand. Furthermore, continuous visual feedback during simulations is naturally allowed.

Here, a natural tool which is exploited to implement adaptive meshing is the self-organizing map (SOM) (Kohonen 2001) algorithm, a neural network approach dealing, when applied in the plane, with visual patterns moving and adapting to brut distributed data. Its main emergent property is to allow adaptation by density and topology preservation of a planar graph (transport mesh) to an underlying data distribution (demand set). Then, the positioning of distributed transportation facilities and infrastructures of the network will reflect the data density distribution and preserve a given topology for the connected network components. Hence, the SOM naturally deals with incomplete or noisy data as well as with stochastic demands. We exploit also the natural property of the SOM of being a center based clustering algorithm which adds topologic relationships between cluster centers. The topologic relationships naturally represent routes and integrate clustering and vehicle routing in a unified way.

This paper generalizes the adaptive meshing concept and summarizes several applications already developed for distributed terrestrial optimization problems (Créput and Koukam 2007a, 2007b; Créput et al. 2007; Hayari et al. 2005). A general clustering and routing optimization problem encompassing the different applications is given. It is called unified clustering and routing problem (UCRP).

To solve the optimization problem, following hybridization of meta-heuristics as done for example in (Boese et al. 1994; Gambardella et al. 1999; Mühlenbein 1991), we present an evolutionary framework which incorporates self-organizing maps as internal operators. It is called memetic SOM by reference to memetic algorithms (Moscato 1999), which are hybrid evolutionary algorithms (EA) incorporating a neighborhood search.

B. Prasad (Ed.): Soft Computing Applications in Industry, STUDFUZZ 226, pp. 189–205, 2008.
springerlink.com

The evolutionary dynamics consists of interleaving the SOM execution with specific operators, such as fitness evaluation, selection operators and greedy insertion/move operators. Operators have a similar structure based on closest point findings and distributed local moves performed in the plane. Theoretically, self-organization is intended to confer robustness according to demand fluctuation and noisy data. Evolution through selection is intended to guide a population based search toward near-optimal solutions.

We show that the approach lets extend self-organizing neural networks to a large class of applications and improve its performance. We present applications to transportation network dimensioning, clustering problems as clustering k-median problem (Arora 1998), combined clustering and routing problems (UCRP), and classical vehicle routing problems such as travelling salesman problem (TSP), capacitated and time duration vehicle routing problems (VRP, DVRP) (Christofides et al. 1979), vehicle routing problem with time windows (VRPTW) (Solomon 1987).

The chapter is organized as follows. Section 2 presents the combined vehicle clustering and routing problem. Objectives and constraints are given. Section 3 describes the optimization framework, which allows to configure and execute evolutionary algorithms embedding neural networks. Section 4 presents applications to concrete problems of clustering and routing. Evaluation against SOM based approaches and against state of the art heuristics are reported. Finally, the last section is devoted to the conclusion and further research.

2 Problem Statement

2.1 Basic Concepts

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.

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.

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 $R_i = (n_0^i, ..., n_j^i, ..., n_{k_i}^i)$, $n_j^i \in B$, of k_i+1 successive cluster centers. To each route is associated a single vehicle. We then denote and identify a vehicle with its route R_i.

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. Each vehicle has a load L_i defined as the sum of its assigned request quantities. A vehicle has a travel time T_i defined as the time for the vehicle starting from transport point n_0^i, and following intermediate straight line paths $\left(n_j^i, n_{j+1}^i\right)$, $j \in \{0, \ldots, k_i\text{-}1\}$, to reach its final transport point $n_{k_i}^i$ at a given constant speed, adding service time of its assigned requests. Vehicles have capacity C and maximum time duration D. 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 $R = \{R_1, \ldots, R_m\}$ 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. The induced graph, or network, is an underlying intermediate structure between demands and vehicles. It can be used to compute shortest paths from an origin point to a destination point. Here, its main interest is of being the visual pattern on which the self-organizing map algorithm directly operates.

2.2 Unified Clustering and Routing Problem

The general problem considered is a unified clustering and routing problem. It is stated as follows:

Unified Clustering and Routing Problem (UCRP). The problem input is given by a set of requests $V = \{r_1, \ldots, r_n\}$ and a set of interconnected routes $R = \{R_1, \ldots, R_m\}$. Using notations and definitions of previous section, 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:

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

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

subject to the capacity constraint:

$$L_i \leq C\ ,\ i \in \{1,\ldots,m\},\tag{3}$$

and time duration constraint:

$$T_i \leq D\ ,\ i \in \{1,\ldots,m\},\tag{4}$$

and time-window constraint:

$$\underset{r_i \in V}{Min}\left(b_i - t_{arr}\left(n_{r_i}\right)\right) \geq 0\ .\tag{5}$$

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 (Arora 1998), using a transport mesh and adding time windows. 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, and that the cluster size is adjusted depending on the application under consideration.

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 VRPTW. Classical TSP and VRP are subclass problems of it. VRP is obtained by removing the time window constraint, TSP by considering a single route and no constraint. Solutions with minimum *distortion* (discarding *length*) are solutions of the Euclidean k-median problem. Whereas, considering a bound stated on one or two objectives yields to non comparable solutions which are possibly not a VRP nor a k-median optimal solution. Since objectives are possibly conflicting, these intermediate solutions are the compromises that are useful and interesting for application of combined clustering and routing. Discarding constraints and considering a complex graph of interconnected routes adapted according to (1) and (2), leads to what we call the network dimensioning problem. The problem can also be seen as a clustering version of the VRPTW (Solomon 1987).

3 Evolutionary Approach Embedding Self-organizing Map

3.1 Self-organizing Map

The self organizing map is a non supervised learning procedure performing a non parametric regression that reflect topological information of the input data set (Kohonen 2001). The standard algorithm operates on a non directed graph $G = (A, E)$, called the network, where each vertex $n \in A$ is a neuron having a location $w_n = (x, y)$ in the plane. The set of neurons A 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 $d(n, n')$.

```
Repeat niter times.
     1.  Randomly extract a point p from the demand set.
     2.  Perform competition to select the winner vertex n* ac-
     cording to p.
     3.  Apply learning law to move the neurons of a neighbor-
     hood of n*.
     4.  Slightly  decrease  learning  rate  α  and  radius  σ  of
     neighborhood.
End Repeat.
```

Fig. 1. Self-organizing map algorithm

The training procedure, summarized in , 2008.
springerlink.com © Springer-Verlag Berlin Heidelberg 2008, applies a given number of iterations *niter* to a graph network. The data set is the set of demands, or customers. Vertex coordinates are randomly initialized into an area delimiting the data set. Each iteration follows four basic steps. At each iteration t, a point $p(t)$ is randomly extracted from the data set (extraction step). Then, a competition between neurons against the input point $p(t)$ is performed to select the winner neuron n^* (competition step). Usually, it is the nearest neuron to $p(t)$ according to Euclidean distance.

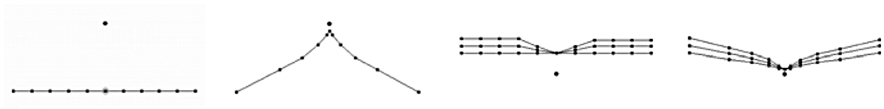
$$w_n(t+1) = w_n(t) + \alpha(t).h_t(n^*, n).(p - w_n(t)) \tag{6}$$

Then, the learning law (6) (triggering step) is applied to n^* and to all neurons within a finite neighborhood of n^* of radius σ_t, in the sense of the topological distance d_G, using learning rate $\alpha(t)$ and function profile h_t. The function profile is given by the Gaussian in (7).

$$h_t(n^*, n) = \exp\left(-d_G(n^*, n)^2 / \sigma_t^2\right) \tag{7}$$

Finally, the learning rate $\alpha(t)$ and radius σ_t are slightly decreased as geometric functions of time (decreasing step). To perform a decreasing run within t_{max} iterations, at each iteration t coefficients $\alpha(t)$ and σ_t are multiplied by $\exp(\ln(x_{final} / x_{init}) / t_{\max})$, with respectively $x = \alpha$ and $x = \sigma$, x_{init} and x_{final} being respectively the values at starting and final iteration.

Examples of a basic iteration are shown in Fig. **2**. We can see in (a)-(b), a basic iteration on a route, and in (c)-(d), how neighborhood influence can be subdivided



(a) (b) (c) (d)

Fig. 2. A single SOM iteration with learning rate α and radius σ. (a)(c) Initial configuration. (b) $\alpha = 0.9$, $\sigma = 4$. (d) $\alpha = 0.75$, $\sigma = 4$.

along possibly interconnected routes. Application of SOM to a set of routes $R = \{R_1, ..., R_m\}$ consists of applying iterations to the undirected graph $G_R = (N, E)$ induced by R, with N the cluster center set, and E the set of edges from routes.

3.2 Evolutionary Embedding Strategy

A construction loop starts its execution with solutions having randomly generated vertex coordinates into a rectangle area containing demands. The improvement loop starts with the single best previously constructed solution, which is duplicated in the new population. The main difference between the construction and improvement loops is that the former is responsible for creating an initial ordering from random initialization. It follows that SOM processes embedded in the construction loop will have a larger initial neighborhood, proportional to n, to deploy the initial network. The improvement loop, however, is intended for local improvements using SOM processes with small and

```
Initialize population with Pop randomly generated individuals.
Do while not Gen generations are performed.
      1. Apply one or more standard SOM processes (denoted
      SOM), with their own parameter settings, to each individ-
      ual in population separately. /* Each SOM operator is ap-
      plied with probability prob, performing niter iterations
      at each generation, for each individual */
      2. Apply mapping operator MAPPING to each individual in
      population to assign each demand to a closest vertex.
      3. Apply fitness evaluation operator FITNESS to each in-
      dividual in population. /* The fitness value depends on
      the problem under consideration */
      4. Save the best individual encountered.
      5. Apply selection operator SELECT.
      6. Apply elitist selection operator SELECT_ELIT.
      7. Apply operators from set {SOMVRP, SOMDVRP, SOMTW},
      that are derived from the self-organizing map algorithm
      structure, to optimize individual routes, or to perform
      greedy insertion moves of the residual (not inserted) de-
      mands.
 End do.
 Report best individual encountered.
```

Fig. 3. The generic evolutionary loop embedding self-organizing map

constant initial neighborhood sizes. Thus, SOM processes can play different roles within their embedding loop. Details of operators are the followings:

1) Self-organizing map operator. It is the standard SOM applied to the graph network. It is denoted by its name and its internal parameters, as $SOM\left(\alpha_{init}, \alpha_{final}, \sigma_{init}, \sigma_{final}, t_{\max}\right)$. One or more instances of the operator, with their own parameter values, can be combined. A SOM operator executes *niter* basic iterations by individual, at each generation. Parameter $t_{\max}$ is the number of iterations defining a long decreasing run performed in the stated generation number *Gen*, for each individual. Other parameters define the initial and final values of learning rate and neighborhood size. The operator can be used to deploy the network toward customers in construction phase, or to introduce punctual moves to exit from local minima in improvement phase.

2) SOM derived operators. Three operators are derived from the SOM algorithm structure for dealing with the VRP and VRPTW. The first operator, denoted *SOMVRP,* is a standard SOM restricted to be applied on a randomly chosen vehicle, using customers already inserted into the vehicle/route. It helps eliminate remaining crossing edges in routes. While capacity constraint is greedily tackled by the mapping/assignment operator below, two operators, denoted *SOMDVRP* and *SOMTW*, deal respectively with the time duration and time window constraints. They performs greedy insertion moves of cluster centers toward customers not already inserted in routes, a vehicle cluster center being selected if it leads to the smallest time increase.

3) Mapping/assignment operator. This operator, denoted *MAPPING*, generates solutions by inserting customers into routes and possibly modifying the shape of the network accordingly. The operator greedily maps customers to their nearest vertex, not already assigned, for which vehicle capacity constraint is satisfied. Then, specifically for classical vehicle routing problems for which *distortion* = 0, the operator moves the vertices to the location of their assigned customer (if exist) and dispatches regularly (by translation) other vertices along edges formed by two consecutive customers in a route.

4) Fitness operator, denoted *FITNESS*. Once the assignment of customers to routes has been performed, this operator evaluates a scalar fitness value for each individual that has to be maximized and which is used by the selection operator. The value returned is $fitness = sat - \alpha \times length - \beta \times distortion$, where α and β are weighting coefficients depending on the problem under consideration, and *sat* is the number of customers that are successfully assigned to routes. Admissible solutions are the ones for which $sat = n$, n being the number of customers.

5) Selection operators. Based on fitness maximization, at each generation the operator denoted *SELECT* replaces *Pop*/5 worst individuals, which have the lowest fitness values in the population, by the same number of bests individuals, which have the highest fitness values in the population. An elitist version *SELECT_ELIT* replaces *Pop*/10 worst individuals by the single best individual encountered during the current run.

4 Applications to Terrestrial Transportation Problems

4.1 Network Dimensioning Problems

The following example illustrates the visual specificity of SOM and the adaptive meshing concept for radio-cellular networks (Créput et al. 2005; Créput and Koukam 2006). The goal is to adjust an intermediate structure (the network) to an underlying distribution of demands, shown in Fig. **4**(a), subject to topology constraints. This is illustrated for cellular network dimensioning in Fig. **4**(b), with a honeycomb mesh representing cell transceiver base stations covering a territory. The terrestrial transportation case is illustrated in Fig. **4**(c), where interconnected lines stand for interconnected routes.

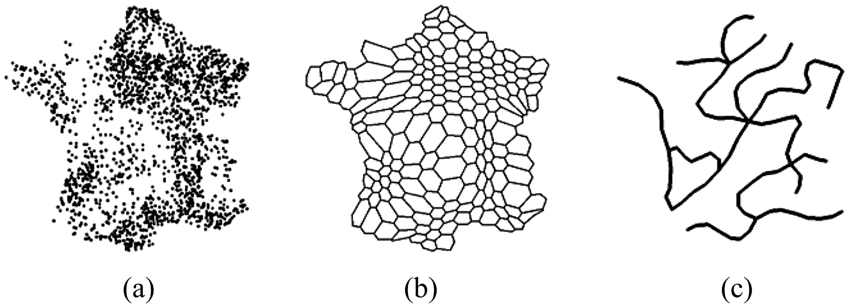


Fig. 4. Meshing of a territory. (a) Sampling of the demand (1000 dots) on a territory. (b) Adapted honeycomb mesh representing a radio-cellular network. (c) Adapted graph of interconnected routes representing a transportation network.

The demands and territory are globally cover, the distances of customers to transportation infrastructures are minimized and the density of the network infrastructures mirrors the underlying density distribution. As well, the network topology is preserved. Then, a customer would easily find a closest facility to communicate through the underlying cell base stations, or transport himself from an origin point to a destination point, following the interconnected routes which globally cover the territory.

Using visual patterns as intermediate structures that adapt and distort according to demands has several positive aspects. 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.

4.2 Combined Clustering and Routing Problems

4.2.1 Median Cycle Problem

From our knowing, the unified clustering and routing problem has never been studied previously. The closest problem encountered, called median-cycle problem (MCP), has been investigated very recently (Labbé et al. 2004, 2005; Renaud et al. 2005). It consists of finding a ring passing through a subset of the demands, minimizing objective length (1), subject to a bound on distortion (2). Such problems arise in the design of ring shape infrastructures such as circular metro lines or motorways, or in a telecommunication context to interconnect terminals via a set of concentrators.

The UCRP is Euclidean, whereas MCP and other related problems presented in (Labbé and Laporte 1986; Volgenant and Jonker 1987; Balas 1989) are defined on graphs, cluster centers being located at request locations. Hierarchical combinations of clustering and routing are studied in location-routing problems (LRP) (Min et al. 1998), but here the hierarchical order of clustering and routing is different. Here, routes visit cluster centers, whereas in LRP routes are built on separate clusters.

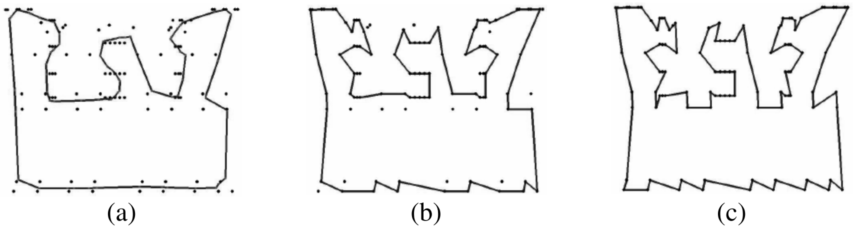


Fig. 5. Median cycle problems (MCP) using the lin105 TSPLIB problem. (a) Euclidean MCP. (b) Classical MCP. (c) Euclidean TSP.

The MCP is illustrated in Fig. 5, on the lin105 instance of TSPLIB (Reinelt 1991), considering three versions of it solved with the memetic SOM approach (Hayari 2005). In (a), is shown a Euclidean, or continuous, version of MCP, where the ring has cluster centers free to position anywhere in the plane. In (b), cluster centers are forced to be located on request locations, as it the case for the classical and discrete MCP. In (c), with a same number of centers and requests, MCP becomes a TSP.

4.2.2 Application to Bus Transportation

Here, the approach is illustrated on a real life case of combined clustering and vehicle routing (Créput and Koukam 2007a, 2007b). The goal is to locate bus-stops on roads and define the regional routes of buses to transport the 780 employees of a great enterprise in city of Belfort, east of France. Three configurations of the algorithm were built in order to solve the problem in a unified way, as well as sequentially by clustering first and routing second. We first apply a memetic SOM configuration to address the unified clustering and routing problem, thus in a unified way. Then, we apply two other configurations to tackle the problem sequentially. One addresses a k-median problem while the other addresses a VRP, using as input the obtained k-median cluster centers and their assigned quantities.

The real life case problem consists of a set of 780 employees dispatched over a geographic area of 73×51 km around the towns of Belfort and Montbeliard in the East of France. Their locations are shown by dots in Fig. 6(a), on a simulator written in Java, using a geographic information system. Each request represents a worker with quantity of one unit. Routes share a single common arrival point fixed at the enterprise location in city of Belfort. This common and fixed arrival point has some importance here, since it participates to transmit the SOM neighborhood influence. Each vehicle has a capacity of 45 units and a speed of 50 km / h.

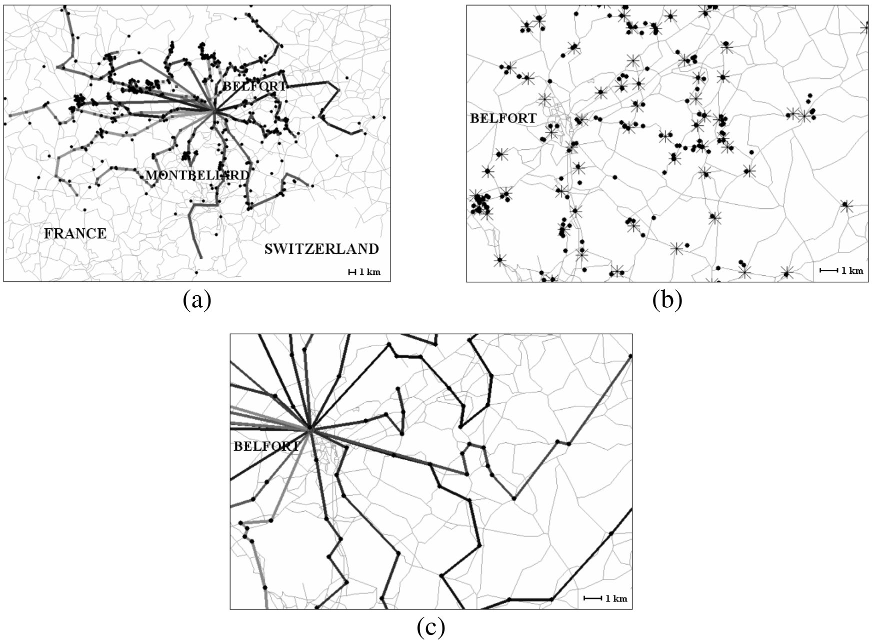


Fig. 6. Clustering and routing for the transportation of 780 customers (dots) of a great enterprise. (a) Unified clustering and routing problem solution. (b) Clustering k-median problem first (crosses are cluster centers). (c) Capacitated vehicle routing problem second (routes pass among cluster centers).

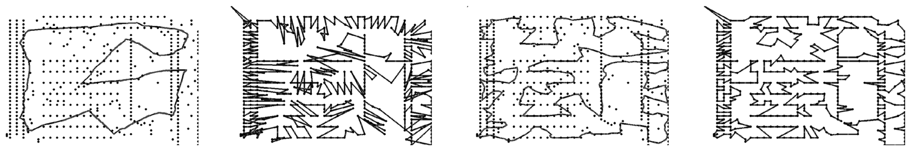
In Fig. 6(a-c), are shown visual patterns of solutions for the three types of memetic SOM applications. A transport mesh obtained for the unified clustering and routing problem is shown in Fig. 6(a), juxtaposed over requests and underlying roads. It illustrates the visual shape of a typical solution, where bus stops, located on roads, reflect demand distribution and constitute routes. In Fig. 6(b-c), a zoom is performed on the right side of the area to illustrate the two main steps for solving the problem sequentially. Fig. 6(b) presents cluster centers, symbolized by crosses, obtained for the k-median problem. **Fig. 6**(c) presents the VRP solution obtained subsequently, with routes exactly passing by the crosses. The numerical results obtained for the problem indicate that the (simple) unified approach competes with the more complex sequential method, yielding diverse non dominated solutions.

4.3 Classical Vehicle Routing Problems

4.3.1 Traveling Salesman Problem

In the literature, many applications of neural networks have addressed the traveling salesman problem. For more information on this subject, we refer the reader to the extensive survey of (Cochrane and Beasley 2003).

Here, we evaluate the memetic SOM performance on the Euclidean TSP. We compare it against the Co-Adaptive Net of (Cochrane and Beasley 2003), which is



(a) (b) (c) (d)

Fig. 7. Traveling salesman problem using the pcb442 instance

considered at the date of writing as the best performing neural network application to the TSP. Experiments are conducted on large size TSPLIB instances from 1000 to up 85900 cities. Mainly, numerical results tend to show that memetic SOM competes with the Co-Adaptive Net, with respect to solution quality and/or computation time.

The memetic SOM principle is illustrated in Fig. 7(a-d) on the pcb442 instance of TSPLIB (Reinelt 1991). Two consecutive pictures respectively show the ring as it is distorted by a SOM operator (a, c), followed immediately by the ring as modified by the mapping operator (b, d), at a given generation. In (a-b) is shown the network at the first generation and in (c-d) at the middle of construction phase. While large neighborhood allows for solution diversification at the beginning of the simulation, decreasing neighborhoods allows for intensification (local improvements) at the final steps of the simulation.

Results are illustrated in Fig. 8, showing the performance of the memetic SOM for each TSPLIB problems of size larger than 1000 cities, with up to 85900 cities. Here, performance is the percentage deviation (%PDM) from optimum of the mean solution value over 10 runs. From the graphic, we can conclude that the memetic SOM yields better quality results on average than the Co-Adaptive Net. The memetic SOM yields 8.83 % of average deviation for the 33 test problems, whereas Co-Adaptive Net yields 9.49 %.

The experiments were conducted within similar (estimated) computation time. The approaches however scale differently. Co-Adaptive Net has a O(n^2.log(n)) time complexity, whereas memetic SOM a O(n^2) time complexity, n being the city number.

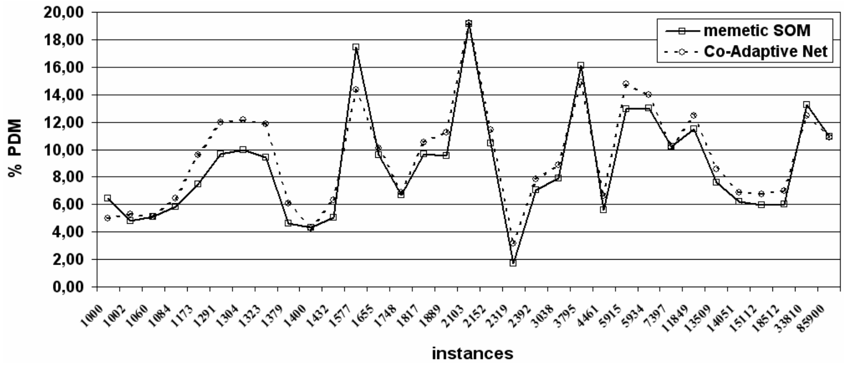


Fig. 8. Percentage deviation to optimum of the mean solution (over 10 runs) for the 33 great size TSPLIB problems with up to 85900 cities

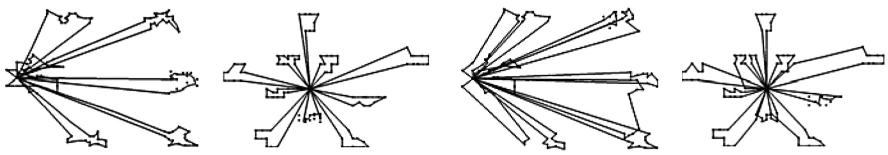
Then, since the gain on computation time comes from largest instances, gain on solution quality is provided by almost all other instances. Considering the very large instance of size 85900, quality solutions are very similar.

However, neural networks do not compete with state of the art powerful heuristics of Operations Research for the TSP. But this domain benefits from the considerable effort spent over more than thirty years, still lacking in the neural community to the TSP. To be competitive with the best performing Lin and Kernighan heuristic implemented recently by (Helsgaun 2000), solution quality produced by neural networks, as well as computation time, would have to be improved both by a factor ten.

4.3.2 Vehicle Routing Problem

Extending SOM to the more complex vehicle routing problem remains a difficult task. Few works were carried out trying to extend SOM, or elastic nets, to the VRP. As far as we know, the most recent approaches are (Ghaziri 1996; Gomes and Von Zuben 2002; Matsuyama 1991; Modares et al. 1999; Schumann and Retzko 1995; Schwardt and Dethloff 2005; Vakhutinsky and Golden 1994). They are generally based on a complex modification of the internal learning law, altered by problem dependant penalties. Here, the SOM execution interleaves with other processes, or operators, following the evolutionary method. Then, operators can be designed independently and then combined.

Example of local improvement moves performed during the improvement optimization phase are visualized in Fig. **9**(a-d) on the clustered instances c11-14, from the publicly available Christofides, Mingozzi, and Toth (CMT) test problems (Christofides et al. 1979). The two instances on the right are time duration versions of the two on the left, hence they need more vehicles.



(a) (b) (c) (d)

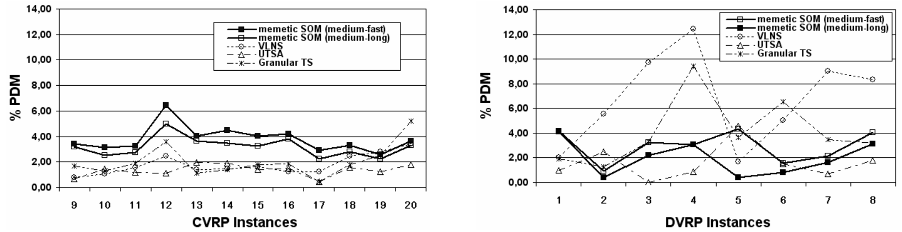
Fig. 9. VRP using c11 (a) and c12 (b) instances. VRP with time duration constraint using c13 (c) and c14 (d) CMT instances.

Evaluation of the proposed approach was performed against neural networks and Operations Research heuristics (Créput et al. 2007b). In the former case, we compared memetic SOM to the three representative approaches of (Ghaziri 1996; Modares et al. 1999; Schwardt and Dethloff 2005). Only these authors have made significant use of the CMT test problems. Other approaches typically used just a few and specific test problems and are hard to evaluate. We evaluated the percentage deviation (%PDM) to best known solution of the mean solution value over 10 runs, on CMT instances. Memetic SOM yields a gain of accuracy, from roughly 5 % for previous SOM based applications, to 3.3 % for short running times, and 1.20 % for long running times.

Considering Operations Research heuristics, we used the large test problems of (Golden et al. 1998), with up to 483 requests, and compared the approach to the Active Guided Evolution Strategy (AGES) (Mester and Bräysy 2005), Granular Tabu

Search (GTS) (Toth and Vigo 2003), Unified Tabu Search Algorithm (UTSA) (Cordeau et al. 2001) and Very Large Neighborhood Search (VLNS) (Ergun et al. 2003), which from our view point cover the range of heuristic performances. For example, as stated in (Cordeau et al. 2005), the AGES approach, which is considered complicated, has displayed the best performance according to the literature. The UTSA and VLNS are considered more simple and flexible but less performing, whereas GTS exhibits an intermediate tradeoff between solution quality and computation time.

Results are illustrated graphically in Fig. 10, showing %PDM for each test problem. We used two configurations of the memetic SOM "medium-fast" and "medium-long", for adjusting computation time at intermediate levels according to the compared approaches.



(a) (b)

Fig. 10. Comparison with Operations Research heuristics on Golden et al. (1998) large size instances.

Memetic SOM reduces the gap from neural networks to classical heuristics. In Fig. 10(a), memetic SOM appears less accurate than other approaches. The instances have no time duration constraint (CVRP). In Fig. 10(b), memetic SOM appears more accurate. In that case, the instances have time duration constraint (DVRP). Average deviation is closed to UTSA deviation. Quality solution is better than for GTS and VLNS. Computation times were comparable to UTSA and lesser than for VLNS.

On the 20 test cases, Memetic SOM yields 2.70 % of average deviation in 39 minutes, whereas UTSA yields 1.45 % in 51 minutes. Minutes are normalized to our AMD Athlon 2000MHz computer. On average, the proposed memetic SOM performs better than VLNS, considering both quality solution and computation time. It yields 3.45 % of average deviation in 10 minutes, whereas VLNS 3,76 % in 22 minutes. For such instances, GTS computes very quickly yielding 2.87 % in 0.79 minutes. Referring to (Cordeau et al. 2005), the very powerful AGES yields a deviation of 0 % to best known value in 66 minutes, and 0.93 % in 0.58 minutes.

4.3.2 Vehicle Routing Problem with Time Windows

Self-Organizing Map has been extended to address the vehicle routing problem (VRP) but, as far as we know, it has not been applied yet to the vehicle routing problem with time windows (VRPTW). Here, a memetic SOM application allows to generate solutions to a clustering version of the classical VRPTW (Créput et al. 2007). The number of vehicles is an input and the generated solutions present walking distance from customers to their closest bus stops. Whereas, travel time are shorter. Solutions for the classical VRPTW are derived from such clustered solutions.

This is illustrated on Solomon's standard test problems (Solomon 1987). Problems are 100-customer Euclidean problems. The tests are embedded into a 100 km × 100 km geographic area, using vehicle speed of 60 km/h, time-windows being given in minutes. Visual patterns in Fig. 11(a-c) illustrate, on the rc201 test case, the different steps to generate a classical VRPTW solution from an obtained solution with non null distortion. The deformable pattern generated by the algorithm is shown in (a). An intermediate result obtained by removing empty clusters is shown in (b). Then, a VRPTW solution, as drawn in (c) is derived by projecting each cluster center to the location of its single assigned request.

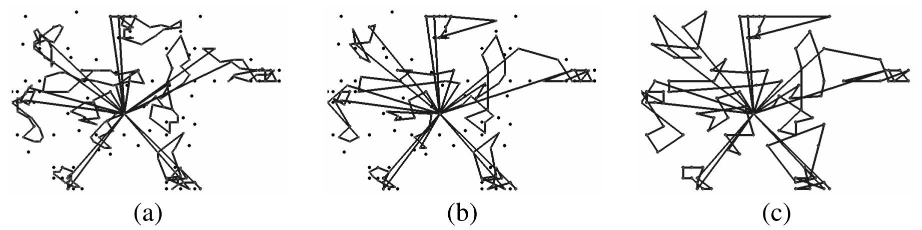


Fig. 11. Clustering Vehicle Routing Problem with Time Windows, using the rc201 Solomon test case. (a) Obtained solution. (b) Same solution after removing empty clusters. (c) A classical VRPTW solution obtained after projecting cluster centers to their (single) assigned requests.

Since SOM application to the VRPTW is new, and to allow further comparisons with memetic SOM, the Table 1 reports the average results on all the 56 Solomon instances, which are divided into six problem sets. Results are given for a single run, performed in roughly 5 minutes on average on our AMD Athlon 2000 MHz computer. The number of routes was set to the one of the best solution reported in the literature, for each problem. The first column indicates the instance name and the number of vehicles. Second column presents the best known length value. Then, the number of satisfied requests, the total route length and the average distortion (*distortion / n)* are given in columns "sat", "length" and "avg. dist.", respectively for the obtained clustering VRPTW solutions, and the derived standard VRPTW solutions. Percentage deviation to the best-known value is given within parenthesis.

The approach performs the best on clustered instances of classes C1 and C2, and mixed uniform/clustered instances of set RC1. Whereas, quality solution slightly decreases with uniformly distributed instances of set R1. It particularly diminishes with the problems which have a large horizon and need a small number of vehicles, of classes R2 and RC2.

Considering clustering VRPTW results, lengths that are obtained are lower than the best known lengths for the classical VRPTW, while average distortion is maintained within a narrow interval of less than 2 km by customer. We can appreciate visually on Fig. 11, how bus stop assignment takes place. Assignment of a request to its closest cluster center is an important characteristic of the solutions generated, since a customer would like to walk specifically to the closest bus stop, and since otherwise, finding the right assignment would become a difficult problem by itself. We think that the approach leads to a new way of thinking about the VRPTW by generating underlying patterns that dispatch through the requests, letting some place for dynamic adaptation to local modifications.

Table 1. Results for the six Solomon classes with 56 problems

	Best known	Clustering VRPTW solution			Derived VRPTW solution		
Problem set – vehicle number	length	sat	length	avg. dist.	sat	length	avg. dist.
C1–10	826.7	101	778.41 (–5.84%)	1.25	100.56	854.08 (+3.32%)	0.0
R1–11.92	1209.89	101	1109.58 (–8.52%)	2.07	99.83	1311.84 (+8.49%)	0.0
RC1–11.50	1384.16	100.75	1260.05 (–8.64%)	1.82	97.63	1424.22 (+3.45%)	0.0
C2–3	587.69	101	482.89 (–17.84%)	1.78	98.88	606.43 (+3.19%)	0.0
R2–2.75	867.54	100.83	1067.35 (+10.35%)	1.54	99.17	1187.97 (+23.88%)	0.0
RC2–3.25	1119.35	100	1340.46 (+18.05%)	1.35	99.38	1428.92 (+26.8%)	0.0

As usual with neural networks, when applied to vehicle routing problems, the approach is not yet competitive with regards to the complex and powerful Operations Research heuristics, specifically dedicated for the VRPTW. Such approaches are presented in the large survey of (Bräysy et al. 2004). For example, the tabu search adaptive memory of (Taillard et al. 1997) produces a solution of less than 1 % deviation to the best known within roughly less than 10 minutes by run, using a Sun Sparc 10 workstation. We are far from obtaining the same accuracy and execution time, except on clustered instances of class C. But, simplicity (easy to understand) and flexibility (easy to extend) are key points that are addressed, using neural networks combined in evolutionary algorithms. As well, intrinsic parallelism is an implicit advantage.

5 Conclusion

By incorporating SOM into an evolutionary algorithm, the approach extends and improves SOM based neural network applications. Operators can be interpreted as performing parallel and massive insertions, simulating spatially distributed agents which interact, having localized and limited abilities. The evolutionary framework helps directing the search toward problem goals.

This massive and natural parallelism at different levels differentiates the approach from classical Operations Research heuristics which operate on graphs (sequentially) and are often considered complex and difficult to implement. Since the communication times at the level of selection is relatively small, the long running times of independent SOM processes favor parallel execution of the method.

The approach has been applied to many spatially distributed transportation optimization problems, thus illustrating its flexibility and accuracy in regard to classical heuristics. Applications to dynamic and stochastic problems are questions to be addressed in the future. Exploiting the natural parallelism of the approach for multi-processor implantation is also a key point to address in further work.

References

Arora, S.: Polynomial-time Approximation Schemes for Euclidean TSP and other Geometric Problems. Journal of the ACM 45(5), 753–782 (1998)

Balas, E.: The prize collecting traveling salesman problem. Networks 19, 621–636 (1989)

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

Bräysy, O., Dullaert, W., Gendreau, M.: Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. Journal of Heuristics 10(6), 587–611 (2004)

Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., et al. (eds.) Combinatorial Optimization, pp. 315–338. Wiley, Chichester (1979)

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

Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 52, 928–936 (2001)

Cordeau, J.F., et al.: New Heuristics for the Vehicle Routing Problem. In: Langevin, A., Riopel, D. (eds.) Logistics Systems: Design and Optimization, pp. 279–297. Springer, New York (2005)

Créput, J.C., Koukam, A.: Local search study of honeycomb clustering problem for cellular planning. International Journal of Mobile Network Design and Innovation 1(2), 153–160 (2006)

Créput, J.C., Koukam, A.: Interactive Meshing for the Design and Optimization of Bus Transportation Networks. Journal of Transportation Engineering, ASCE (in press, 2007a)

Créput, J.C., Koukam, A.: Transport Clustering and Routing as a Visual Meshing Process. Journal of Information and Optimization Sciences, Taru Publications (in press, 2007b)

Créput, J.C., Koukam, A., Hajjam, A.: Self-Organizing Maps in Evolutionary Approach for the Vehicle Routing Problem with Time Windows. International Journal of Computer Science and Network Security 7(1), 103–110 (2007)

Créput, J.C., Lissajoux, T., Koukam, A.: A Connexionnist Approach to the Hexagonal Mesh Generation. In: 16th IMACS World Congress, Lausanne (2000)

Créput, J.C., et al.: Automatic Mesh Generation for Mobile Network Dimensioning using Evolutionary Approach. IEEE Transactions on Evolutionary Computation 9(1), 18–30 (2005)

Ergun, Ö., Orlin, J.B., Steele-Feldman, A.: Creating very large scale neighborhoods out of smaller ones by compounding moves: a study on the vehicle routing problem. MIT Sloan Working Paper No. 4393-02, USA (2003)

Gambardella, L.M., Taillard, E., Agazzi, G.: MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 63–76. McGraw-Hill, UK (1999)

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

Golden, B.L., et al.: Metaheuristics in vehicle routing. In: Crainic, T.G., Laporte, G. (eds.) Fleet Management and Logistics, pp. 33–56. Kluwer, Boston (1998)

Gomes, L.C.T., Von Zuben, F.J.A.: Vehicle Routing Based on Self-Organization with and without Fuzzy Inference. In: Proc. of the IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1310–1315 (2002)

Hayari, N., Créput, J.C., Koukam, A.: A Neural Evolutionary Algorithm for Geometric Median Cycle Problem. In: European Simulation and Modeling Conference, ESM 2005, EUROSIS, Porto, Portugal (2005)

Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. European Journal of Operational Research 126, 106–130 (2000)

Kohonen, T.: Self-Organization Maps and associative memory, 3rd edn. Springer, Berlin (2001)

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

Labbé, M., et al.: The Ring Star Problem: Polyhedral Analysis and Exact Algorithm. Networks 43, 177–189 (2004)

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

Matsuyama, Y.: Self-organization via competition, cooperation and categorization applied to extended vehicle routing problems. In: Proc. of the International Joint Conference on Neural Networks, Seatle, WA, pp. 385–390 (1991)

Mester, D., Bräysy, O.: Active Guided Evolution Strategies for Large Scale Vehicle Routing Problems with Time Windows. Computers & Operations Research 32, 1593–1614 (2005)

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

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

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

Mühlenbein, H.: Evolution in Time and Space – The Parallel Genetic Algorithm. In: Rawlins, G. (ed.) Foundations of Genetic Algorithms, Morgan Kaufmann, Los Altos, CA (1991)

Reinelt, G.: TSPLIB-A traveling salesman problem library. ORSA Journal on Computing 3, 376–384 (1991)

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

Schumann, M., Retzko, R.: Self-organizing maps for vehicle routing problems minimizing an explicit cost function. In: Proc. of the International Conference on Artificial Neural Networks, Paris, pp. 401–406 (1995)

Schwardt, M., Dethloff, J.: Solving a continuous location-routing problem by use of a self-organizing map. Int J. Physical Distribution & Logistics Management 35(6), 390–408 (2005)

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

Taillard, E.D., et al.: A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation science 31, 170–186 (1997)

Toth, P., Vigo, D.: The granular tabu search and its application to the vehicle routing problem. INFORMS Journal on Computing 15, 333–348 (2003)

Vakhutinsky, A.I., Golden, B.L.: Solving vehicle routing problems using elastic net. In: IEEE International Conference on Neural Network, pp. 4535–4540 (1994)

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