



A cross entropy multiagent learning algorithm for solving vehicle routing problems with time windows

Tai-Yu Ma

► To cite this version:

Tai-Yu Ma. A cross entropy multiagent learning algorithm for solving vehicle routing problems with time windows. J.W. Böse et al. ICCL 2011, Lecture Notes in Computer Science (LNCS) 6971, Springer-Verlag Berlin Heidelberg, pp.59-73, 2011. halshs-00592118v2

HAL Id: halshs-00592118

<https://shs.hal.science/halshs-00592118v2>

Submitted on 11 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A cross entropy multiagent learning algorithm for solving vehicle routing problems with time windows

Tai-Yu Ma

LET-ISH, 14, Avenue Berthelot
F-69363 Lyon Cedex 07

`tai-yu.ma@let.ish-lyon.cnrs.fr`

Abstract. The vehicle routing problem with time windows (VRPTW) has been the subject of intensive study because of its importance in real applications. In this paper, we propose a cross entropy multiagent learning algorithm, which considers an optimum solution as a rare event to be learned. The routing policy is node-distributed, controlled by a set of parameterized probability distribution functions. Based on the performance of experienced tours of vehicle agents, these parameters are updated iteratively by minimizing Kullback-Leibler cross entropy in order to generate better solutions in next iterations. When applying the proposed algorithm on Solomon's 100-customer problem set, it shows outperforming results in comparison with the classical cross entropy approach. Moreover, this method needs only very small number of parameter settings. Its implementation is also relatively simple and flexible to solve other vehicle routing problems under various dynamic scenarios.

Keywords: Vehicle routing problem, heuristic, cross entropy

1 Introduction

The vehicle routing problem with time windows (VRPTW) has been known as one of the NP-hard problems in combinatorial optimization. The problem consists of delivering goods to a set of customers, which must be visited within given time windows, at a minimum cost under available capacitated vehicle constraints. The standard VRPTW has been the subject of intensive study because of its importance in real applications, such as pickup and delivery problems in transportation of goods and fleet operation management. The exact and heuristic algorithms have been proposed and applied to Solomon's test instances and many real-life situations. Recent reviews in these solution techniques for the VRPTW can be found in [1][2][3].

The state of the art of exact methods has successfully solved most of Solomon's 100-customer benchmark instances. However, only very limited large instances have been solved to optimality [4]. Hence, an intensive effort has been engaged in proposing efficient heuristics for solving the VRPTW and related vehicle routing problems. The state of the art of heuristics includes: local search [5][6], adaptive large

neighborhood search [7] and evolutionary mechanism [8][9]. Basically, these heuristics apply two main procedures to construct and improve the solutions: the route construction procedure and the route improving procedure. The former consists of generating a set of feasible routes based on deterministic or stochastic route-building heuristics. This procedure provides initial points in order to apply some improving local search algorithms for finding high-quality solutions on the neighborhoods. In general, the neighborhood search space is very large; one needs to develop efficient searching strategies to find local optima. To this end, the local search algorithms are widely used, which consist of replacing a subset of initial solutions such that better solutions can be found. Numerous variants based on the local search framework have been proposed such as iterative local search [10][11], multistart local search [12][13] and adaptive multistart local search [5]. Basically, these local search algorithms generate initial solutions by the route construction heuristics or some random mechanisms, and then apply the local search heuristics such as k-opt, i.e. replacing k edges on a route traveled by a vehicle with k edges not on this route, or swap move, exchanging the position of two nodes on the same route, to improve solution quality. To avoid being trapped into local optima, related perturbation or randomization procedures need to be applied. The performance of the local search heuristics depends not only on the interaction between the route construction and the improvement procedures but also on the strength of the perturbation or randomization of the two procedures. To this issue, an adaptive large neighborhood search method has been proposed [7]. The authors proposed a flexible framework aiming to adaptively choose a set of local search techniques. This method provided an auto-adjusted mechanism to intensify or diversify searching neighborhood according to the performance of the set of heuristics. However, it needs to implement a set of heuristics, which is more time-consuming and complicated than single-heuristic-based algorithm.

Different from the aforementioned heuristics, the agent-based distributed solution techniques have been proposed recently. Vokrinek et al. [17] proposed a vehicle routing problem solver based on multiagent framework. The solver is composed of three types of agents in order to collect demand (task agent), allocate demand (allocation agent) and find routes (vehicle agent). The route construction algorithm is based on greedy search heuristics. The experiments were conducted for capacitated vehicle routing problem. The average solution quality within 91.3% optimality was reported. Barbucha and Jedrzejowicz [16] developed a multiagent platform for simulating dynamic vehicle routing system, i.e. customer requests arrive when the vehicles are running. The system is composed of a set of agents with different functionalities for executing different tasks such as initialization, customer request generation and route assignments for requests. Simple insertion rules are applied for customer request assignment to the vehicles. Other applications based on the multiagent framework in logistics can be found in [18]. In summary, these studies have developed simulation tools based on the multiagent approach to static or dynamic vehicle routing problems. However, the solutions found by these multiagent approaches are still far from the optima.

In this work, a multiagent learning algorithm based on cross entropy (CE) method [14] is proposed. We associate a set of routing probabilities with nodes (customers) on the network, iteratively guiding the vehicles to find optimal routes, which minimize total cost, and satisfying capacity and time window constraints. Based on

the performance of the routes traveled by the vehicles in each iteration, the choice probability distributions of next outgoing nodes are iteratively updated. We consider the optimum solution as a rare event to be estimated based on the importance sampling theory. More precisely, we specify a random mechanism to generate feasible solutions (samples), controlled by a set of parameterized probability distribution functions (pdf). Based on the performance of the samples, the parameters are updated iteratively by minimizing Kullback-Leibler cross-entropy in order to generate better solutions in next iterations. A set of vehicle-specific transition matrices is associated with the nodes of the network to construct subsequently a feasible route for each vehicle. As the capacity and time window constraints need to be satisfied during the route construction process, a sequential importance sampling technique is utilized by constructing the solution sequentially, conditional on vehicle's capacity and customer's time windows constraints. The stochastic route construction procedure is repeated until all customers are serviced. As the classical CE method may be trapped on the local optima at its early stages, some local search techniques are combined with the CE method to avoid this problem and to improve significantly the convergence speed of the classical CE method.

The rest of this paper is organized as follows. In section 2, we define the VRPTW problem and provide its mathematical formulation. Section 3 introduces the concept of the classical CE method and the proposed multiagent system for the VRPTW problem. Based on the performance of routes travelled by the vehicles, a hybrid scheme combining the agent-based CE algorithm and the local search procedure is proposed. This scheme enables local search performed only on a small subset of good solutions. Section 4 provides the computational results for Solomon's 100-customer VRPTW instances. Finally, the conclusions and future extensions are discussed.

2 Problem formulation

The mathematical formulation of the VRPTW problem can be stated as follows. Let $G(V, E)$ be a directed graph with a vertex set V and an arc set E . The vertex set is composed of one depot (node 0) and n customers 1 to n , denoted as $V = \{0, 1, \dots, n\}$. The arc set is $E = \{(i, j) | i \neq j, \forall i, j \in V\}$. Each customer is associated with an amount of goods d_i to be delivered. Let $M = \{1, 2, \dots, |M|\}$ be a set of homogeneous/heterogeneous vehicles with $|M|$ being the total number of vehicles. Each vehicle m has a fixed capacity q_m . Each customer is associated with a time window $[a_i, b_i]$, for which the customer cannot be serviced before a_i and after b_i . Let s_i be the arrival time for customer i . The service time for customer i is denoted as u_i with $u_i > 0$, but no service time at the depot, i.e. $u_0 = 0$. Let the depot be associated with a scheduling time window $[a_0, b_0]$, for which any vehicle cannot depart the depot before a_0 and return to it after b_0 . Each arc $(i, j), \forall i \neq j$ has an asymmetric travel time t_{ij} and an operation cost c_{ij} .

The problem is to determine a set of routes, originating and terminating at the depot, such that total cost is minimal by satisfying: (1) each customer is serviced exactly once; (2) vehicle capacity cannot be violated; (3) all customers must be serviced within the service time windows. Let r_m denote a route, starting and terminating at the depot, composed of a sequence of customers visited by the vehicle m . The k -th visited customer of vehicle m is denoted as $r_m(k)$. Let $\delta_m = \{0, 1, 2, \dots, n_m, n_m + 1\}$ be a sequence of visiting order of the customers in r_m with $r_m(0) = r_m(n_m + 1) = 0$. We define an indicator y_{im} as 1 if customer i is serviced by vehicle m , and 0 otherwise. The set of routes for all vehicles is denoted as $\mathbf{r} = \{r_1, r_2, \dots, r_M\}$. We formulate the VRPTW problem mathematically as follows:

$$\text{Min } S(\mathbf{r}) = \sum_{m \in M} \sum_{k \in \delta_m \setminus \{0\}} c_{r_m(k-1)r_m(k)} \quad (1)$$

subject to

$$\sum_{i \in r_m \setminus \{0\}} d_i \leq q_m, \forall m \in M \quad (2)$$

$$\sum_{m \in M} y_{im} = 1, \forall i \in V \setminus \{0\} \quad (3)$$

$$s_{r_m(k-1)} + u_{r_m(k-1)} + t_{r_m(k-1)r_m(k)} \leq s_{r_m(k)}, \forall k \in \delta_m \setminus \{0\} \quad (4)$$

$$a_i \leq s_i \leq b_i, \forall i \in V \quad (5)$$

$$y_{im} \in \{0, 1\}, \forall i \in V \setminus \{0\}, \forall m \in M \quad (6)$$

The objective function Eq. (1) minimizes the total travel cost. The constraints (2) mean that each vehicle cannot load goods exceeding its capacity restriction. The constraints Eq. (3) ensure that each customer is serviced exactly once. The constraints Eq. (4) ensure the consistency of service time for next visiting customer. The time windows are imposed by Eq. (5). The constraints Eq. (6) are the integrality constraints. Note that the above formulation is route-based, convenient for constructing routes for each vehicle based on the multiagent framework.

3. Cross entropy learning algorithm

The main concept of the CE method is to associate the optimization problem with an estimation problem throughout the route-searching process. The search process is characterized by a set of density functions associated with the nodes of the network. These density functions are iteratively updated based on the minimization of the

Kullback-Leibler distance (cross entropy) between density functions and optimal density functions. In the following, we present first the multiagent model for general vehicle routing problem simulation. Then we describe the cross entropy multiagent learning algorithm to solve the VRPTW problem.

3.1 Multiagent model for vehicle routing problems with time windows

The multiagent simulation framework is very convenient for modeling and simulating the general vehicle routing problems since it captures the system behavior dynamics resulting from the interactions of supply and demand. Basically, the system is composed of a set of heterogeneous agents with predefined behavior. The advantage of the multiagent approach resides on its flexibility in capturing complex interactions between different components of the system.

The simulation of the VRPTW problem by the multiagent approach describes vehicle's optimal route search process under its capacity and service time windows constraints. In current application, the system is specified as:

- **Customer agent:** a set of customers with fixed demand is known *a priori*. Each customer agent needs to be visited exactly once.
- **Vehicle agent:** a set of vehicle agents with fixed capacity is available in the system. Vehicle agents depart from and return to the depot by picking up customer requests under its capacity and service time windows constraints. The violation of these constraints is not allowed.
- **Environment (network):** it is represented by a directed graph on which a set of vehicles operate on it. The network is characterized by a set of nodes representing the customers and a set of links associated with related characteristics. In static case, the travel time of link is fixed. In dynamic case, one can replace it with time-dependent travel cost function.

The present multiagent framework is convenient in dynamic situations with time-dependent travel cost or stochastic customer demand.

3.2 Cross entropy learning algorithm

The main idea of the proposed cross entropy learning algorithm is that we specify a parameterized random mechanism for vehicle agents to generate feasible routes. Based on the performance of these "samples" (a sample is a *feasible solution* of the VRPTW), the parameters of the random mechanism are updated towards the optimal solutions. This stochastic search algorithm is originated from the importance sampling techniques aiming to increase the accuracy of rare-event probability estimation. Basically, the CE learning algorithm is composed of two steps [14][19]: (1) generate a set of samples according to some stochastic mechanism; (2) update the parameters of stochastic mechanism based on the performance of the samples in order to generate better solutions at next iteration. As the classical CE method may be trapped in local

optima, we propose a hybrid scheme by combining local search algorithms in order to overcome this problem and increase the solution quality.

The random feasible route generation mechanism moves vehicle agents from current customer/depot to next unvisited customers, one at each step, based on Markov chain on graph $G(V, E)$, starting and ending at the depot. A set of vehicle-specific transition matrix (a routing probability matrix), $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{|M|}\}$ are associated with each node of the graph to construct subsequently a feasible route for one vehicle agent. The stochastic route construction procedure is repeated until all customers are serviced. Different from the classical CE method, the idea of the proposed method is based on the sequential importance sampling techniques [21], i.e. to construct a solution sequentially, conditional on capacity, while time windows constraints and partial solutions are already constructed before the current state. Note that the proposed CE learning algorithm is quite intuitive. It needs only to rescale the transition probabilities at current visiting node by eliminating *infeasible candidates* (next infeasible not yet visiting customers) in order to generate a feasible route at each step. If there is no feasible candidate to visit, the vehicle agent returns to the depot and next vehicle starts its route construction.

Let the transition matrix of vehicle m be $\mathbf{p}_m = (p_{ij}^m), \forall m \in M$ with p_{ij}^m being the transition probability for the vehicle m moving from node i to node j . Note that \mathbf{p}_m is a $|V| \times |V|$ matrix with the first column and row being the depot. We set $p_{ij}^m > 0$ if $i \neq j$, and 0 otherwise. Each vehicle constructs a route sequentially based on its transition matrix and a prohibition list (infeasible candidates) until current state.

As mentioned above, we cast the original optimization problem to an estimation problem of rare event probability. To increase the sampling performance at each iteration, a sequence of new sampling densities, called importance sampling densities, need to be chosen. The optimal importance sampling density \mathbf{P}^* can be iteratively derived by minimizing the Kullback-Leibler cross-entropy distance [14][19]. First, a criterion of rare event (an approximate to optimal solution) γ is associated with each iteration and updated according to the performance of independent and identically distributed (i.i.d.) samples. We associate γ with an indicator $I_{\{S(\mathbf{r}) \leq \gamma\}}$ being 1 if the performance of solution is better than γ , i.e. $S(\mathbf{r}) \leq \gamma$, and 0 otherwise. Let $f(\mathbf{r}; \mathbf{P})$ denote the probability distribution function (pdf) of \mathbf{r} (defined around Eq. (1)), parameterized by \mathbf{P} . The probability of a global tour \mathbf{r} is the multiplication of outgoing node choice probabilities in vehicle's route constructing process. Its logarithm can be written as:

$$\ln f(\mathbf{r}; \mathbf{P}) = \sum_{m=1}^{|M(\mathbf{r})|} \sum_{k=1}^{n_m+1} \sum_{i,j} I_{\{r_m \in \mathbf{R}_{ij}(k)\}} \ln p_{ij}^m, \quad (7)$$

where $M(\mathbf{r})$ is the set of vehicles utilized for the solution \mathbf{r} . $\mathbf{R}_{ij}(k)$ denotes the set of feasible routes such that the k -th transition (move) is from node i to node j . The optimal important sampling distribution based on the minimization of the CE distance can then be obtained as:

$$\mathbf{P}^* = \arg \max_{\mathbf{P}} \mathbb{E}_{\mathbf{P}_{w-1}} I_{\{S(\mathbf{r}) \leq \gamma\}} \ln f(\mathbf{r}; \mathbf{P}) \quad (8)$$

Note that the above equation derives the optimal important sampling pdf based on known \mathbf{P}_{w-1} . It is equivalent to the following optimization problem:

$$\text{Max} \quad \mathbb{E}_{\mathbf{P}_{w-1}} I_{\{S(\mathbf{r}) \leq \gamma\}} \ln f(\mathbf{r}; \mathbf{P}) \quad (9)$$

subject to

$$\sum_{j \in \Lambda^+(i)} p_{ij}^m = 1, \forall i \in V, \forall m \in M \quad (10)$$

$$p_{ij}^m \geq 0, \forall (i, j) \in E, \forall m \in M \quad (11)$$

where $\Lambda^+(i)$ denotes the successors of node i .

By applying Karush-Kuhn-Tucker optimality conditions, the optimal solution of the above optimization problem can be obtained by differentiating with respect to p_{ij}^m as:

$$p_{ij}^m = \frac{\mathbb{E}_{\mathbf{P}_{w-1}} \left[I_{\{S(\mathbf{r}) \leq \gamma\}} \sum_{k=1}^{n_m+1} I_{\{r_m \in \mathbf{R}_{ij}(k)\}} \right]}{\mathbb{E}_{\mathbf{P}_{w-1}} \left[I_{\{S(\mathbf{r}) \leq \gamma\}} \sum_{k=1}^{n_m+1} I_{\{r_m \in \mathbf{R}_i(k)\}} \right]} \approx \frac{\sum_{h=1}^N \left[I_{\{S(\mathbf{r}) \leq \gamma\}} \sum_{k=1}^{n_m+1} I_{\{r_m \in \mathbf{R}_{ij}(k)\}} \right]}{\sum_{h=1}^N \left[I_{\{S(\mathbf{r}) \leq \gamma\}} \sum_{k=1}^{n_m+1} I_{\{r_m \in \mathbf{R}_i(k)\}} \right]} \quad (12)$$

where $\mathbf{R}_{ij}(k)$ is the set of routes, for which the k -th transition is from node i to node j . $\mathbf{R}_i(k)$ is the set of routes for which its k -th transition is starting from node i . N is the sample size.

Note that the above updating rule states that p_{ij}^m is updated based on the transition proportion from node i to node j of the vehicle m , conditional on the global tour performance satisfying $S(\mathbf{r}) \leq \gamma$. As the above important sampling pdf could generate some infeasible tours in the sequential route construction process, a rescaling procedure is conducted at each step such that next node to be visited is drawn from the feasible node set. The CE learning algorithm is stated as follows.

Algorithm 1: The CE learning algorithm

Step 1: Initialize \mathbf{P}_0 as a uniform pdf over the node set V . Order the endings of the time windows $b_i, \forall i \in V$ from the smallest to the biggest. Set the iteration index $w = 1$.

Step 2: Generate N i.i.d. samples according to the feasible route generation algorithm (described later).

Step 3: Order the solution performance $S(\mathbf{r})$ as a sequence $S(\mathbf{r}_1) \leq S(\mathbf{r}_2) \leq \dots \leq S(\mathbf{r}_N)$.
 Apply the local search procedure (described later) for the samples in the λN best solutions. We set $0.1 \leq \lambda \leq 0.3$.

Step 4: Order the solution performance after the local search. The ρ -quantile of the performances $S(\tilde{\mathbf{r}})$, i.e. $\lceil \rho N \rceil$ th lowest cost ($\lceil x \rceil$ denotes the smallest integer greater than or equal to x), is denoted as a new “rare-event” criteria:

$$\gamma_w = S_{(\lceil \rho N \rceil)} \quad (13)$$

As for the value of ρ and the sample size N , it is recommended that as the sampling number increases, the ρ value decreases. We set ρ as $\frac{c_1}{|V|}$ and

$N = c_2 |V|^2$ with $1 \leq c_1 \leq 10$ and $0.5 \leq c_2 \leq 1$, where the sample size N is smaller than in the classic CE method, reducing considerably the computational times.

Step 5: Calculate \mathbf{P}_w by Eq. (12) and apply the smoothed updating rule as follows:

$$\mathbf{P}_w := \alpha \mathbf{P}_{w-1} + (1 - \alpha) \mathbf{P}_w, \quad (14)$$

It is recommended that α takes a value in the range of $0.4 \leq \alpha \leq 0.9$ (see [19]).

Step 6: To avoid being trapped into local optimum, the following dynamic parameter adjustment process is applied. If $|S(\mathbf{r}_{w-2}^*) - S(\mathbf{r}_{w-1}^*)| / S(\mathbf{r}_{w-1}^*) \leq \theta$, where $0.02 \leq \theta \leq 0.03$, multiply λ by k_1 and ρ by k_2 with $1.5 < k_1 < 3$, $1 < k_2 < 2$, where $S(\mathbf{r}_{w-1}^*)$ is the cheapest travel distance obtained in iteration $w-1$.

If for some iteration \bar{w} , the value of $\gamma_{\bar{w}}$ stabilizes, i.e. $\gamma_{\bar{w}} = \gamma_{\bar{w}-1} = \dots = \gamma_{\bar{w}-c}$, where c is a constant, or $\bar{w} = w^{\max}$ then stops; otherwise set $w = w + 1$ and go to step 2.

Algorithm 2: Feasible route generation algorithm

This procedure rescales the transition matrix \mathbf{p}_m (eq. 12) for each vehicle in a sequential way such that each customer (node) is visited exactly once and satisfies vehicle’s capacity and time windows constraints. The complexity of the rescaling procedure is $O(n^2)$.

Step 1: Initialization. Set all customers as unvisited and vehicles depart from the depot. Set position index $k=0$ for each vehicle.

Step 2: Rescaling choice probability of next visiting customers. For vehicle m currently located at its node i (customer), rescale the choice probability p_{ij}^m for next visiting node j as follows. First, set the choice probability of infeasible nodes (not satisfying capacity and time windows constraints, eq. (2)-(4)) as 0

and then normalize choice probabilities over all feasible outgoing nodes j to sum up to 1, i.e.

$$p_{ij}^m := \frac{p_{ij}^m}{\sum_{j' \in \Lambda} p_{ij'}^m}, \text{ where } \Lambda \text{ is the set of feasible nodes for vehicle } m \text{ at node } i.$$

Step 3: If next visiting node is the depot, stop for vehicle m and set $m := m + 1$, otherwise repeat step 2.

3.3 Local search procedure

To improve the solution quality for the samples in the ρ -quantile of the best solutions in main algorithm 1, we apply the local search procedure. The procedure contains two phases. First, a greedy local search is applied. Then a route-exchange local search procedure is applied.

- **Greedy search:** this procedure aims to reduce the number of utilized vehicles and exchanges some nodes to get the largest travel cost reduction. It contains two steps as follows:
 - a. Remove the route with the least customers,
 - b. Insert all nodes of the removed route, one node at a time, at the cheapest position of other routes.
- **Route exchange search:** this procedure applies randomly one of the following two local search methods:
 - c. Reverse move: reverse current ordering two nodes of the same route,
 - d. K-Or-opt [15][22]: remove k nodes of current route and insert them in another position of the same route so that a feasible set of routes is preserved. This local search is implemented in the order of 1-Or-opt, 2-Or-opt and 3-Or-opt.

Note that one can also apply some more efficient local search techniques such as LKH local search [12] or sequential search procedures [15] to obtain an improved solution quality.

4 Computational experiments

The algorithm is tested on three datasets C1, R1 and RC1 of Solomon's 100-customer benchmark instances [20]. These datasets reflect different characteristics of customers' positions, tight or loose time windows constraints and vehicles' capacity. The algorithm is programmed in C++.

The results are shown in Table 1. The best known or optimal solutions are listed in the left part of Table 1. The results show that the proposed algorithm finds near-optimal solutions with an average error of 6.24% (N=10000) in comparison with optimal solutions. As the reported results are based on only 1 to 3 tests, better solutions can be found if trying more tests and modifying related parameter settings.

The comparison of running times and solution quality based on different sample size shows that the running time is basically proportional to the sample size. Fig. 1 reports the outperforming results of the proposed hybrid scheme compared with the classical cross entropy method. Note that the hybrid scheme utilizes fewer samples ($0.7n^2$) than the cross entropy method ($\approx 10n^2$) to achieve better solution quality. Fig. 2 presents the average performance of different sample size on 12 instances of C1, R1 and RC1 classes. Fig. 3-5 presents the impact of different parameter settings on solution quality. These tests are performed on the R102 dataset of Solomon's 100 customer instances. Fig. 3 presents the smooth parameter α on the convergence speed. The test experiences suggest that α may be taken between 0.6 and 0.9 to obtain better solution quality. The impact of the size of the elite sample, parameterized by ρ , on solution quality is shown in Fig. 4. It suggests a range of $[0.05, 0.1]$ for ρ . If ρ is too small, say 0.01, the algorithm converges quickly to local optima. The influence of the sample size of local search on the solution quality is not very significant (Fig. 5). However, more efficient local search techniques play an important role in increasing the performance of the proposed algorithm.

Table 1. Application of CE multiagent learning algorithm to Solomon's VRPTW 100 customer instances

Instance	Optimum/best known solution ⁴	CE multiagent learning algorithm							
		N=7000				N=10000			
	Distance	Distance ¹	Relative error (%)	Time ¹ (sec.)	Time ² (sec.)	Distance ¹	Relative error (%)	Time ¹ (sec.)	Time ² (sec.)
C101	827.3	840.1	1.55%	1105	2509	898.7	8.63%	1930	4384
C103	826.3	890.6	7.78%	1479	3359	896.3	8.47%	1942	4411
C105	827.3	867.1	4.82%	1366	3104	869.3	5.08%	2250	5111
C107	827.3	843.1	1.91%	1407	3196	841.1	1.67%	2157	4900
R102	1466.6	1551.0	5.75%	1325	3010	1539.1	4.94%	2151	4886
R104	971.5	1061.0	8.04%	1876	4263	1077.8	9.76%	2643	6003
R106	1234.6	1334.0	8.05%	1292	2936	1323.1	7.17%	2074	4712
R108	932.1	1040.3	8.26%	1557	3537	1044.2	8.67%	2367	5378
RC102	1457.4	1571.8	7.85%	1285	2918	1584.4	8.71%	2354	5347
RC104	1132.3	1248.8	9.98%	1721	3909	1226.6	8.02%	2242	5092
RC106	1372.7	1477.6	3.71%	1311	2978	1465.6	2.87%	2045	4647
RC108	1114.2	1242.0	8.97%	1427	3242	1241.2	8.90%	2126	4829
Average error		6.39%				6.24%			

Remark: 1. The parameter settings are

$$\alpha = 0.7, \rho = 0.05, \lambda = 0.2, \theta = 0.02, k_1 = 2.0, k_2 = 1.2$$

for all instances, except C103 with $\rho = 0.07, \lambda = 0.15$.

2. The above mentioned result is based on 1 to 3 runs for each instance.

3. Time¹ is the total time executed on Dell Latitude E6400 with 2.53GHz

and 3.48G memory. Time² is based on the execution of the same program on Asus Intel Pentium M processor 740 with 512MB memory. The speedup ratio between the two laptops is about 2.27.

4. The best know solution is based on [23].

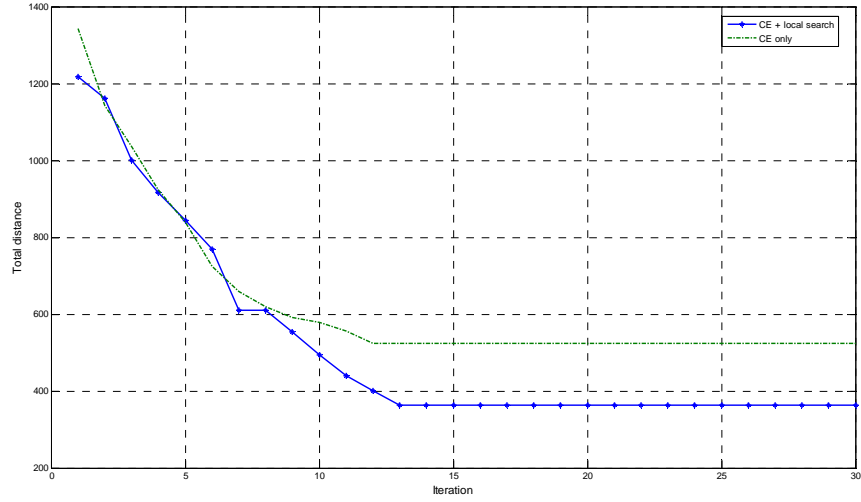


Fig. 1. Performance of CE multiagent learning algorithm compared with the classical CE method (without applying local search and dynamic parameter adjustment of Algorithm 1) (Solomon's 50-customer C101 instance, $\rho=0.05$, $\alpha=0.7$, optimum= 362.4)

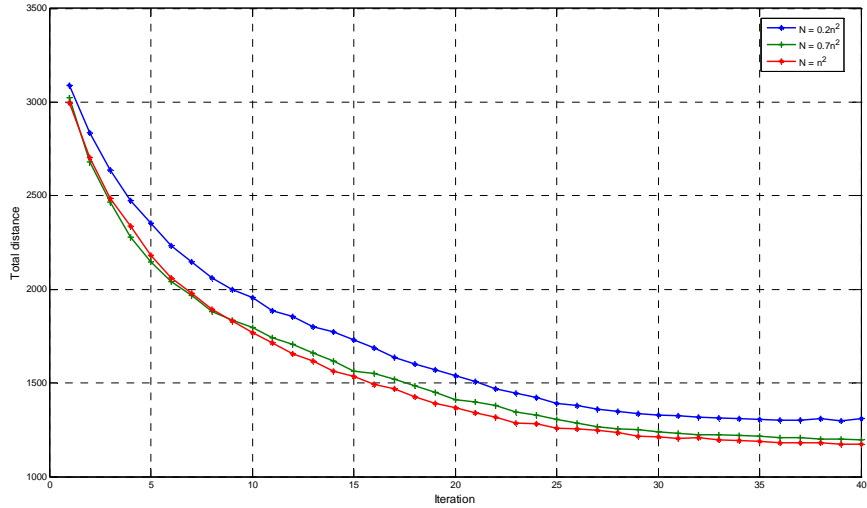


Fig. 2. Influence of sample size on the average performance of the algorithm over 12 instances of C1, R1 and RC1 classes of Solomon's 100-customer instances. The average optimum is 1082.47.

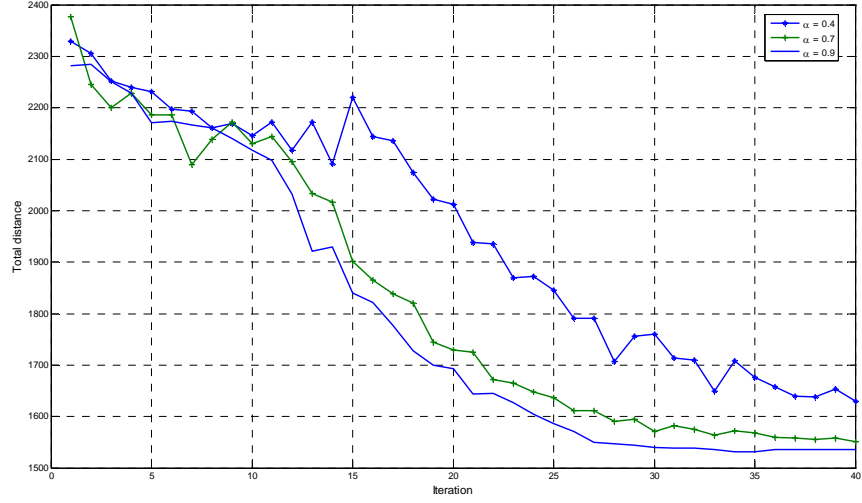


Fig. 3. Influence of the smooth parameter α on the algorithm performance (Solomon's 100-customer R102 instance, $\rho = 0.05$, $\lambda = 0.2$, $\theta = 0.02$, $k_1 = 2.0$, $k_2 = 1.2$ $N=7000$, optimum is 1466.6)

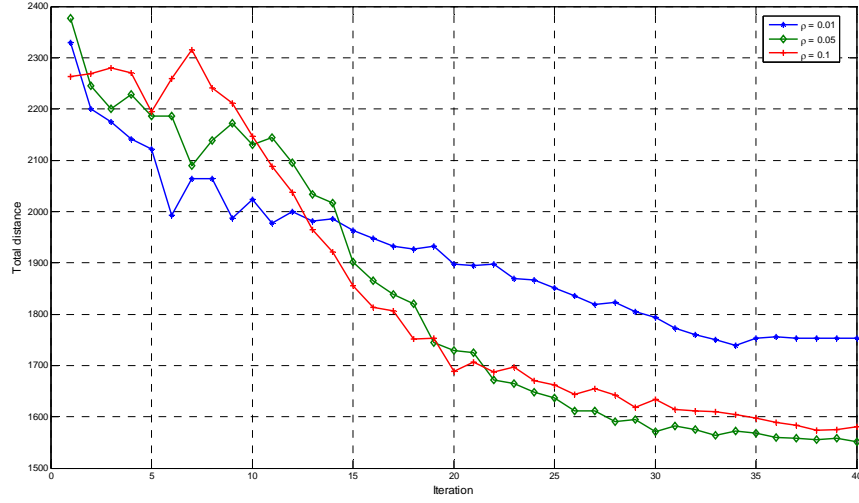


Fig. 4. Influence of the elite sample size on the algorithm performance (Solomon's 100-customer R102 instance, $N=7000$, $\alpha = 0.7$, $\lambda = 0.2$, $\theta = 0.02$, $k_1 = 2.0$, $k_2 = 1.2$)

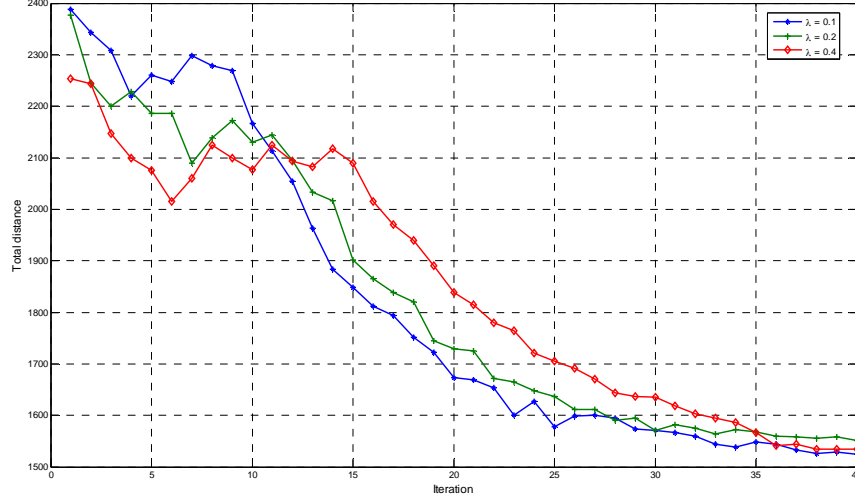


Fig. 5. Influence of the size of local search on the algorithm performance (Solomon's 100-customer R102 instance, $N=7000$, $\rho = 0.05$, $\theta = 0.02$, $k_1 = 2.0$, $k_2 = 1.2$)

5 Conclusions

In this paper, a cross entropy learning algorithm is proposed for solving vehicle routing problems with time windows. The advantage of the proposed approach is that the optimal routing probability is iterative learned based on importance sampling and rare event simulation theory. By combining the local search procedure and introducing the dynamic parameter adjustment procedure, the proposed method can avoid being trapped in the local optimum and find quickly near-optimal solutions. The numerical tests show also that the hybridization of the local search technique and the CE method can improve efficiently the solution quality. As the proposed method is based on an adaptive learning procedure, it provides a general method for solving vehicle routing problems under stochastic environment. Moreover, the proposed method needs only very small number of parameters settings. Its implementation is also relatively simple and flexible for various vehicle routing problems. Currently, the numerical study based on some Solomon's 100-customer VRPTW instances is implemented.

Future extension concerns the application of multiagent framework on general dynamic vehicle routing problems solving. It is also interesting to apply an appropriate data structure and more efficient implementation techniques for large scale instance (200 to 1000 customers). Moreover, other efficient local search techniques can be implemented to improve local search performance.

Acknowledgements

This research has benefited from a grant of ANR (the French Agency for Research) project, Mutualisation et Optimisation de la Distribution Urbaine de Marchandises (MODUM).

References

1. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M. M., Soumis, F.: The VRP with time windows. In: Toth, P., Vigo, D. (eds.) *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. pp. 157--194 (2002)
2. Braysy, O., Gendreau, M.: Vehicle routing problem with time windows. part I: Route construction and local search algorithms. *Transportation Science*. 39, 104--118 (2005)
3. Braysy, O., Gendreau, M.: Vehicle routing problem with time windows. part II: Metaheuristics. *Transportation Science*. 39, 119--139 (2005)
4. Golden B, Raghavan S, Wasil E, (eds). *The vehicle routing problem, latest advances and new challenges*. In: *Operations research/computer science interfaces series*, vol. 43. Berlin: Springer; (2008)
5. Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., Yagiura, M: Effective local search algorithms for routing and scheduling problems with general time window constraints. *Transportation Science*. 39(2), 206--232 (2005)
6. Hashimoto, H., Ibaraki, T., Imahori, S., Yagiura, M.: The vehicle routing problem with exible time windows and traveling times. *Discrete Applied Mathematics*. 154, 2271--2290 (2006)
7. Pisinger, D., Ropke, S. A general heuristic for vehicle routing problems. *Computers & Operations Research*. 34, 2403--2435 (2007)
8. Braysy, O., Dullaert, W. Gendreau, M.: Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*. 10, 587--611 (2004)
9. Homberger, J., Gehring, H.: A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*. 162, 220--238 (2005)
10. Martin, O., Otto, S.W., Felten, E.W.: Large-step Markov chains for the TSP incorporating local search heuristic. *Operation Research Letters*. 11, 219--224 (1992)
11. Hashimoto, H., Yagiura, M., Ibaraki, T.: An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*. 5, 434--456 (2008)
12. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Datalogiske skrifter (Writings on Computer Science)* no. 81. Roskilde University (1999)
13. Braysy, O., Hasle, G., Dullaert, W.: A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*. 159, 586--605 (2004)
14. Rubinstein, R. Y.: *The Cross-Entropy Method for Combinatorial and Continuous Optimization*. *Methodology and Computing in Applied Probability*. 2, 127--190 (1999)
15. Irnich, S., Funke, B., Grunert, T.: Sequential search and its application to vehicle-routing problems. *Computers & Operations Research*. 33, 2405--2429 (2006)
16. Barbucha, D., Jedrzejowicz, P. Multi-agent platform for solving the dynamic vehicle routing problem. In: *Proc.of 11th Int. IEEE Conf. on Intelligent Transportation Systems*, pp.517-522. (2008)

17. Vokrinek, J., Komenda, A., Pechoucek, M.: Agents Towards Vehicle Routing Problems. In: Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems, pp.773-780 (2010)
18. Davidson, P., Henesey, L., Ramstedt, L., Tornquist, J., Wernstedt, F.: An analysis of agent-based approaches to transport logistics, Trans. Res. Part C. 13, 255--271 (2005)
19. De Boer, P.T., Kroese, D. P., Mannor, S., Rubinstein, R.Y.: A Tutorial on the Cross-Entropy Method. Annals of operations research. 134(1), 19--67 (2005)
20. Solomon, M. M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research. 35, 254-265 (1987)
21. Rubinstein, R.Y., Kroese, D.K.: Simulation and the Monte Carlo Method. Wiley Series in Probability and Statistics (2008)
22. Croes G.: A method for solving traveling-salesman problems. Operations Research. 6, 791--812 (1958).
23. Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D. : Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows. Operations Research. 56(2), 497--511 (2008)