Journal of Experimental & Theoretical Artificial Intelligence

ISSN: 0952-813X (Print) 1362-3079 (Online) Journal homepage: www.tandfonline.com/journals/teta20

Empirical-type simulated annealing for solving the capacitated vehicle routing problem

Bochra Rabbouch, Foued Saâdaoui & Rafaa Mraihi

To cite this article: Bochra Rabbouch, Foued Saâdaoui & Rafaa Mraihi (2020) Empirical-type simulated annealing for solving the capacitated vehicle routing problem, Journal of Experimental & Theoretical Artificial Intelligence, 32:3, 437-452, DOI: 10.1080/0952813X.2019.1652356

To link to this article: https://doi.org/10.1080/0952813X.2019.1652356

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE
2020, VOL. 32, NO. 3, 437–452
https://doi.org/10.1080/0952813X.2019.1652356

Taylor & Francis
Taylor & Francis Group

Check for updates

ORIGINAL ARTICLE

Empirical-type simulated annealing for solving the capacitated vehicle routing problem

Bochra Rabbouch[a,b], Foued Saâdaoui[b,c] and Rafaa Mraihi[d]

[a]Institut Supérieur de Gestion de Tunis, Université de Tunis, Tunis, Tunisia; [b]Laboratoire d'Algèbre, Théorie de Nombres et Analyse Non-linéaire, Faculté des Sciences, Université de Monastir, Monastir, Tunisia; [c]Department of Statistics, Faculty of Sciences, King Abdulaziz University, Jeddah, Saudi Arabia; [d]Ecole Supérieure de Commerce de Tunis, Campus Universitaire de La Manouba, Université de Manouba, Tunis, Tunisia

ABSTRACT

The Capacitated Vehicle Routing Problem (CVRP) is a well-known combinatorial optimisation problem used to design an optimal route for a fleet of capacitated vehicles based at a single depot, to serve a set of customers. Over the few past years, the interest in solving real-world applications of the CVRP, especially in transportation and logistics, has grown tremendously. The Simulated Annealing (SA) algorithm is among the most effective employed techniques for finding the CVRP's global optimums. However, because of its lack of flexibility, the SA algorithm may have some weakness, like its slowness and its wandering near the global minimum in the final stage of the search. For this reason, we define in this paper the Empirical-Type Simulated Annealing (ETSA) as a new dynamic version of the SA for effectively solving the CVRP and any other vehicle routing problem. The method operates incrementally by exploiting the last portion of worse feasible solutions, which are fitted using a parametric density function, to update the SA's Boltzmann acceptance criterion. This leads to a more accurate decision within the searching process, and consequently, optimums are more rapidly reached. A comparison to state-of-the-art approaches has proven that the new algorithm is capable of locating all optimums while improving the convergence of the SA algorithm.

ARTICLE HISTORY
Received 25 December 2018
Accepted 1 August 2019

KEYWORDS
Capacitated vehicle routing problem; empirical-type simulated annealing; combinatorial optimisation; stochastic methods; convergence acceleration

Introduction

Meta-heuristics are effective techniques for approximating the global optimum of a given function in a fairly reasonable computation time (Rabbouch, Mraihi, & Saâdaoui, 2017). They especially include evolutionary algorithms, greedy search procedures, and the simulated annealing (SA). The SA is a probabilistic search method inspired by the annealing process in metallurgy. The technique involves heating and then cooling a metal very slowly to obtain an appropriate crystalline structure. In the context of global optimisation, this simulation-based technique is illustrated by an analogy between the physical system and the optimisation problem: the material state corresponds to problem solutions, the energy corresponds to the objective function, the ground state corresponds to the global optimum solution, and the temperature corresponds to a control parameter. At each iteration of the SA, a new solution is randomly generated. The extent of the new solution from the current solution is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all-new solutions that lower the objective, but

CONTACT Foued Saâdaoui foued.saadaoui@gmail.com Department of Statistics, Faculty of Sciences, King Abdulaziz University, P.O BOX 80203, Jeddah 21589, Saudi Arabia

also, with a certain probability, solutions that raise the objective. The SA algorithm is based on a Monte-Carlo iterative strategy. Thus, it has the power to intensify the search from global to local level, which helps escaping local optimums. The solution space is well explored, and consequently, there is an increased probability to find the global optimum. Nevertheless, the SA is considered as a stochastic memoryless approach, since it applies purely random rules at each one of its iterations. In other words, the SA does not exploit information during the search process. These two facts can enable the degradation of the obtained solution. Despite all that, the SA performs well in discrete search spaces, such as for the VRP and its variants.

Nowadays, the SA continues to draw attention as one of the most successful methods for solving vehicle routing problems. The SA was firstly used by Osman (1993), who proposed a hybridisation with the well-known tabu search for solving the Capacitated VRP (CVRP). Breedam (1995) proposed an enhanced heuristic based on the SA to solve the standard VRP. Chiang and Russell (1996) applied the SA for solving a VRP with time-window constraints, where two different neighbourhood structures were investigated and the annealing process was enhanced with a short-term memory function via tabu list. Sigauke and Talukder (2003) proposed a modified version of the hybrid method of Osman for optimising the CVRP using an heterogenous fleet of vehicles. Tavakkoli-Moghaddam, Safaei, and Gholipour (2006) proposed a hybrid SA based on the nearest neighbourhood principle to solve the CVRP with an independent route length, where objectives were to minimise the heterogeneous fleet cost and maximise the capacity utilisation. Lin, Yu, and Chou (2009) applied a SA heuristic to the truck and trailer routing problem. Leung, Zheng, Zhang, and Zhou (2010) presented an SA approach to solve the CVRP with two-dimensional loading constraints, where the loading component of the problem was solved through a collection of packing heuristics. Afifi, Dang, and Moukrim (2013) proposed an SA algorithm incorporating different local search techniques to treat a VRP with time windows and synchronisation constraints. Recently, Yu and Lin (2015) applied an SA-based heuristic technique for solving the open location-routing problem (a CVRP variant). Finally, Yu, Perwira Redi, Hidayat, and Wibowoa (2017) used an SA approach with a restart strategy to solve the hybrid VRP, which is an extension of the Green VRP. The restart strategy was implemented with the Boltzmann and Cauchy functions as acceptance criteria for a worse solution.

In this article, we develop a new SA-type algorithm for solving the CVRP and any multivariate nonlinear problem. The proposed strategy is simple and only requires little calculation effort. In fact, it is acknowledged that the SA is a random algorithm, which suffers from slow convergence speed, especially for complex problems (Johnson & Sachin, 2009). The SA is memory-less, i.e., it captures neither histories nor close structures during the search process. Moreover, it is often time-consuming to escape from a local optimum. In previous research, the tabu search (Glover, 1986) was among the most appealing alternative techniques exploiting a set of historical solutions as short-term memory during the search process. Other techniques have also extended the tabu search to incorporate long-term memory functions for intensifying and diversifying the search. Our technique is founded under the same principles, while it exploits the heuristic memory in some other way mainly involving the SA algorithm. The methodology consists, within a SA algorithm, to exploit the fresh part of worse feasible solutions to empirically update the Boltzmann acceptance rule. For the old SA, this rule is known to use a nonparametric exponential-type function, which does not take into account the histories during the search process. In our approach, we extend this criterion by choosing a gamma (parametric) function, which is continuously adjusted according to the empirical distribution of the worse solutions. The gamma distribution is a well-know probability function characterised by its flexibility with simplicity. In fact, exponential, Erlang, and chi-squared distributions, all are special cases of the gamma distribution. Moreover, only two parameters are sufficient to handle the gamma distribution in both shape and scale. Given its empirical nature, the method will be referred to as Empirical-Type Simulated Annealing (ETSA). On this basis, this new procedure is supposed to reduce the randomness of the SA, and consequently, it is intended to improve its convergence. Indeed, the heuristic memory learns the search path in the SA in such a way as to accelerate escaping from local optimums. As such, this strategy can be thought of as incorporation of machine learning into optimisation processing.

The remaining of the paper is organised as follows. Section 2 reviews the relevant CVRP literature and summarises its main technical background. Then, section 3 introduces the principle of the new proposed extension of the SA for the CVRP and explains its practical implementation. Section 4 reports the computational results, followed by the conclusion in section 5.

Capacitated vehicle routing problem

We present a mixed-integer linear problem for the CVRP. Let us consider a vehicle routing network defined as a directed graph $G = (V, A)$ and a homogenous fleet of L vehicles, where each vehicle has the same transportation capacity. The sets, parameters, and decision variables are firstly summarised (see also above in the list of notations), before we present the mathematical formulation. The graph G includes a set of vertices $V = \{v_0, \ldots, v_n\}$ and a set of arcs $A = \{(v_i, v_j) : v_i, v_j \in V, \forall i \neq j\}$. The customer set $C = \{v_1, \ldots, v_n\}$ regroups the n nodes to visit, while v_0 corresponds to the unique depot, where L vehicles are parked. A constant c denotes the capacity of each one of the L vehicles. Each vertex $v_i \in V$ has a non-negative fixed loads d_i (with $d_0 = 0$ and $v_i \leq c$). The distance between points i and j for all $i, j \in V$ is denoted D_{ij}. The CVRP variables are defined as follows: **1)** assignment variable: $X_{ijl} = 1$ if a vehicle l travelling along an arc (i, j) $(i \in V; j \in V)$ and $X_{ijl} = 0$, otherwise; and **2)** load variable Y_{ij}, which is a non-negative continuous variable denoting the total load remaining in the vehicle before reaching the node j while travelling the arc (i, j), $i \in V, j \in V$.

The CVRP aims at minimising the following objective function:

$$f(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{L} D_{ij} X_{ijl}, \tag{1}$$

subject to the following constraints (whose interpretations are detailed below):

$$\textstyle\sum_{i=1}^{n} \sum_{l=1}^{L} X_{ijl} = 1, \forall j = 2, \ldots, n, \forall i \neq j, \tag{2}$$

$$\textstyle\sum_{j=1}^{n} \sum_{l=1}^{L} X_{ijl} = 1, \forall i = 2, \ldots, n, \forall i \neq j, \tag{3}$$

$$\textstyle\sum_{i=1}^{n} X_{ijl} = \sum_{i=1}^{n} X_{jil}, \forall l = 1, \ldots, L, \forall j = 1, \ldots, n, \forall i \neq j, \tag{4}$$

$$\textstyle\sum_{j=2}^{n} Y_{1j} \geq \sum_{j=2}^{n} d_j, \tag{5}$$

$$\textstyle\sum_{i=1}^{n} Y_{ij} - \sum_{i=1}^{n} Y_{ji} = d_j, \forall j = 2, \ldots, n, \tag{6}$$

$$\textstyle Y_{ij} \leq \sum_{l=1}^{L} cX_{ijl}, \forall i = 1, \ldots, n, \forall j = 2, \ldots, n, \tag{7}$$

$$X_{ijl} \in \{0, 1\}, \forall i, j = 1, \ldots, n, \forall l = 1, \ldots, L, \tag{8}$$

$$Y_{ij} \geq 0, \forall i, j = 1, \ldots, n. \tag{9}$$

The objective function Equation (1) aims to minimise the total distance, which is the sum of all consecutive points serviced by L vehicles starting and ending at a central depot. Constraints Equations (2) and (3) ensure that each customer is visited and serviced only once by a given vehicle. Constraint Equation (4) states that servicing an arc (i, j) implies servicing the arc (j, i) (symmetric VRP). Constraint Equation (5) indicates that the total vehicle's load, when leaving the depot, is superior or equal to the total customer's demand. Constraint Equation (6) shows that the quantity remaining after visiting a customer j is exactly the load before visiting the same customer minus his demand. Constraint Equation (7) indicates that the load needed to be picked up to a customer j is inferior or equal to the capacity of a given vehicle originating from the depot. This constraint guarantees that the vehicle capacity may not be violated. Constraint Equation (8) refers to the binary of the assignment decision variable. Finally, constraints Equation (9) refers to the non-negativity the load decision variable.

Empirical-type simulated annealing (ETSA)

Sa's principle

The SA is a stochastic technique defined by Kirkpatrick, Gelatt, and Vecchi (1983) to approximate the global optimum of a given function. The SA algorithm employs a random search, which not only accepts changes that decrease the objective function f (assuming a minimisation problem), but also admits some solutions that increase it. The process finds its origins from statistical thermodynamics, where p_α, the probability of a physical system being in state a with energy E_α at temperature T, satisfies the Boltzmann distribution (known as the Boltzmann-Gibbs distribution).

$$p_\alpha = \frac{\exp(-E_\alpha T/\kappa)}{\sum_{\alpha'} \exp(-E_{\alpha'} T/\kappa)}, \tag{10}$$

where κ is the Boltzmann's constant (often omitted), T is the absolute temperature, with the summation being taken over all states α' with energy $E_{\alpha'}$ at temperature T. At high T, the system ignores small changes in the energy and approaches thermal equilibrium rapidly, that is, it performs a coarse search of the space of global states and finds a good minimum. As T is lowered, the system responds to small changes in the energy, and performs a fine search in the neighbourhood of the already determined minimum and finds a better minimum. At $T = 0$, any change in the system states does not lead to an increase in the energy, and thus, the system must reach equilibrium if $T = 0$.

When solving a minimisation problem (such as the CVRP) using the SA algorithm, a global minimum is guaranteed to be reached with high probability. In fact, it proceeds as a hill-climbing algorithm with the supplement's ability to escape from local optima in the search space by including some randomisation in the move selection process. The artificial thermal noise is gradually decreased in time. T is a control parameter called computational temperature, which controls the magnitude of the perturbations of the energy function $E(X)$. The probability of a state change is determined by the Boltzmann distribution of the energy difference of the two states:

$$p(X) = \exp\left(\frac{-\Delta E(X)}{T}\right), \tag{11}$$

where $\Delta E(X)$ is the increase in E and T is a control parameter, which by analogy with the original application is known as the system temperature irrespective of the objective function involved.

Practically, the algorithm starts from a feasible solution X_0 of the CVRP. This solution is tried to be optimised by generating randomly a neighbouring solution $X_0 + \delta X$ and the cost of the new solution is calculated. The move from X_0 to $X_0 + \delta X$ is an improving one, if $\Delta = E(X_0 + \delta X) - E(X_0) < 0$. Furthermore, the algorithm starts at a high temperature T and it decreases gradually. As high temperatures allow an improved exploration of the search space and low temperatures permit the fine-tuning of a so obtained good solution, the probability of accepting a low-quality solution is very small. Consequently, the temperature function will be updated using a constant variable α on $T_{i+1} = \alpha T_i$, where i is the current iteration and the typical values of α vary between 0.8 and 0.99. These values can provide a very small diminution of the temperature. For each temperature, a number of moves according to the Metropolis algorithm is executed to simulate getting to the thermal equilibrium. Finally, the algorithm will be stopped when the condition will be reached.

Main approach: ETSA

The principle of the approach is based on the belief that the distribution of worse solutions could have a different distribution than that of the exponential density as expressed in Equation (11) ($f(z;\theta) = \theta \exp(-\theta z)$ with $z = \frac{\Delta E(X)}{T}$ and the parameter θ supposed equal to 1). Indeed, assuming such a probability is considered too restrictive, by imposing to the SA search to be rigid and memoryless. The probability of the occurrence of a new worse solution z_i is naturally related to the distribution of previous ones z_{i-1}, $i = 1, \ldots, n$. Therefore, we extend the SA to an empirical strategy, rather employing a flexible density function whose main characteristics are controlled by a couple of parameters. Thus, a gamma probability density function is employed. Accordingly, the Boltzmann criterion within the SA will be based on the empirically adjusted density function, i.e., the density whose parameters have been empirically estimated on the basis of the previous worse solutions. Such a strategy allows, iteration by iteration, to correct the distribution of the worse changes, and consequently, accepting new worse solutions is intended to be more effective.

Let us consider the random variable $Z = \frac{\Delta E(X)}{T}$. Z is said to follow a gamma distribution if its probability density function is expressed as

$$f(z;\theta,k) = \begin{cases} \frac{1}{\theta^k \Gamma(k)} z^{k-1} \exp\{-(z/\theta)\} & z > 0, \\ 0 & z \leq 0, \end{cases} \tag{12}$$

where $k > 0$ is the shape parameter and $\theta > 0$ is the scale parameter of the distribution. $\Gamma(k)$ is the gamma function evaluated at k. The gamma distribution is related to a number of other probability distributions; in particular, it generalises the exponential distribution ($k = 1$) and the chi-squared distribution (with v degree of freedom when $k = v/2$ and $\theta = 2$). The Maximum Likelihood (ML) method is commonly used for estimating the parameters of this model (θ and k) for a given set of observations. The ML estimation is described in section 6 (Appendix). At the evaluation stage of the SA, if a new solution does not decrease the objective function (i.e., $\Delta E(x) > 0$), it is accepted with a gamma probability whose parameters have been ML-estimated on the basis of the previous outcomes. Accordingly, the extended Empirical-Type SA (ETSA) operates much more flexibly in search of a global minimum.

Remark 1. The SA algorithm can be thought of as the particular case of the ETSA, with shape and scale parameters respectively fixed to $k = 1$ and $\theta = 1$.

In fact, considering $f(\Delta E;\theta,k) = [\theta^k \Gamma(k)]^{-1} (\Delta E/T)^{k-1} \exp\{-(\Delta E/T\theta)\} > \varepsilon$ the new decision criterion, with $\Delta E > 0$, $T > 0$, and $\varepsilon \sim \mathcal{U}_{(0,1)}$. Fixing the shape and scale parameters to $k = \theta = 1$ is exactly the Boltzmann criterion of the old SA algorithm.

Definition 3.1. Let us denote the general case as $\gamma -$ ETSA algorithm. Two main particular schemes can be derived from the $\gamma -$ ETSA algorithm by imposing restrictions on shape and scale parameters:

i Exponential ETSA, denoted $\mathrm{e} -$ ETSA, when the shape parameter takes $k = 1$.
ii Chi-square ETSA, denoted $\chi^2 -$ ETSA, when the scale parameter takes $\theta = 2$.

Implementation

The optimisation procedure begins with a good solution obtained by a constructive algorithm or by a randomly generated solution. An interesting alternative is to make several runs of local search loops (Multi-restart), starting from different initial solutions, and select the best among the obtained final solutions. These strategies have varying computational requirements, thus they would usually result in final solutions of varying quality. Random solutions are known to be quickly generated, but the iterative search may take a large number of iterations to converge to either a local or global optimum solution. On the other hand, a constructive heuristic takes up time, nevertheless the iterative improvement phase converges rapidly if started off with a constructive solution (Sait & Youssef, 1999). In our implementation, we choose to randomly generate a starting point X_0 satisfying all the problem constraints (Equations (2–9)). The procedure consists of a random selection stage at the beginning, and a feasibility checking at the end.

It is then important to define an efficient method for finding neighbours to improve an existing solution. An effective random neighbourhood structure has been recently proposed by Yu and Lin (2016), and used by many other studies, such as those of Mirmohammadsadeghi and Ahmed (2015), and also Birim (2016). At each iteration, the neighbourhood-search algorithm generates a new solution $X_n + \delta X$ from the current solution X_n, by using swap, reversion and insertion mechanisms. The principle of each mechanism is explained as follows: The swap is carried out by selecting two customers randomly and swapping them to generate a new solution from the current solution. In the reversion mechanism, two numbers (customers) from the string of numbers representing the current solution then reversing the route from bigger number to smaller one. The insertion is performed by selecting two customers randomly and inserting the first customer immediately after the second one (Mirmohammadsadeghi & Ahmed, 2015). In our implementation, each one of these three mechanisms is used with equiprobability.

Moreover, we choose to use the Exponential Cooling Scheme (ECS). Indeed, the ECS is the simplest and most common temperature decrement rule, with $T_{i+1} \leftarrow \alpha T_i$, where T_{i+1} and T_i denote the new and old temperature values, respectively, and α is a constant close to, but smaller than, one. This scheme was first proposed by Kirkpatrick et al. (1983) with a value of 0.95. van Laarhoven and Aarts (1987) proposed a geometric cooling scheme described by the temperature-update $T_{i+1} \leftarrow \alpha^i T_i$, where α assumed to be a fixed within the interval $[0.8, 0.99]$. However, several comparisons using finite-length cooling rates of exponential and geometric schemes have found no significant difference in performance. Below, suggested by Kirkpatrick et al. (1983), a value of 0.95 is fixed through all experiments.

It is noticeable that the ETSA approach only requires little calculation work compared to the SA algorithm. Only one extra stage, involving the ML-estimates, is added to the SA to obtain the new strategy. Besides, the method needs the storage of a reduced number of fresh history among worse solutions, which does not exceed few hundreds of observations during all the search process. The overall procedure is summarised by Figure 1 and Table 1 provides a pseudo-code. The main tasks of the ETSA algorithm are detailed as follows:

(1) **Generating a starting solution**: Choice of a solution X_0 for the CVRP to then calculate the objective function at X_0.
(2) **Initialising the temperature**: The initial value of temperature T is an important parameter for an efficient implementation of the SA algorithm. As explained above, the choice of T should be determined according to a trade-off between the speed and accuracy of the procedure.
(3) **Selecting a new solution in the neighbourhood of** X_0: A new solution $X_0 + \delta X$ is processed as a new recent solution depending on T. The objective function for $X_0 + \delta X$ and X_0 is represented by $E(X_0 + \delta X)$ and $E(X_0)$, respectively.
(4) **Calculating the variation of the objective function**: The change of the objective function $\Delta E(X) = E(X_0 + \delta X) - E(X_0)$ is assessed at each iteration of the SA.
(5) **Dealing with negative changes**: If $E(X_0 + \delta X) \leq E(X_0)$, then $X_0 + \delta X$ is accepted and it replaces X_0, update the existing optimal solution and go to step 6.

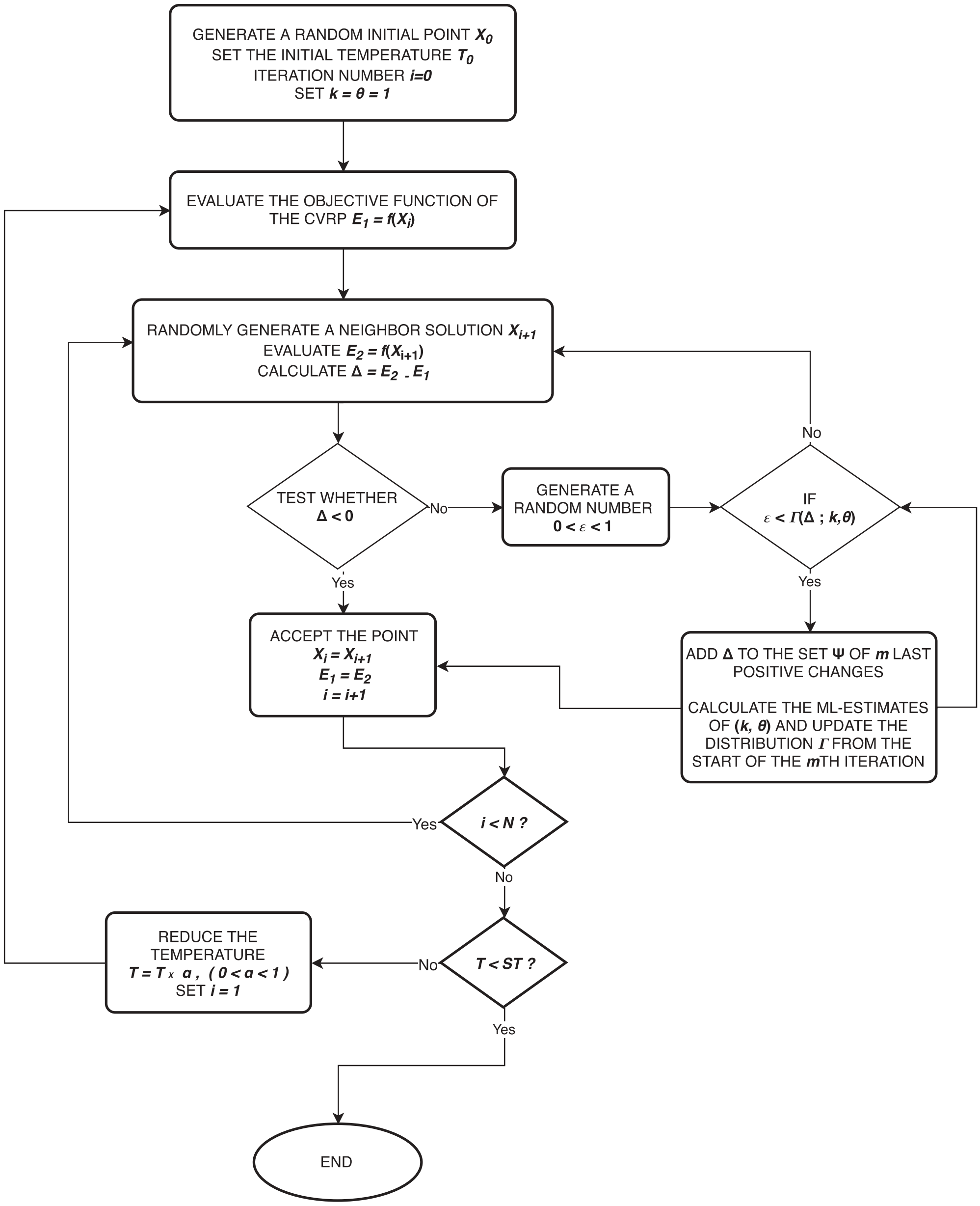


Figure 1. Diagram of the Empirical-Type Simulated Annealing (ETSA) used for solving the capacitated vehicle routing problem.

(6) **Dealing with positive changes**: If $E(X_0 + \delta X) > E(X_0)$, accumulate a sample, whose elements are $z = \frac{\Delta E(X)}{T}$, for estimating the parameters of Equations (14) and (15). Then, the solution $X_0 + \delta X$ is accepted with a probability as expressed in Equation (12) with updated parameters. The ordinary SA is used at m first iterations for starting the new procedure.

(7) **Reducing the temperature T gradually**: The probability of accepting deteriorating solutions at the beginning of the search is high, and gradually decreases from one iteration to another.

(8) **Repeating steps 2–7 until a stopping criterion is met**: The computation is terminated when the termination criterion is satisfied. Otherwise, step 2 until 7 are repeated. The implementation of the parametric SA is presented in the next section.

Table 1. Pseudo-code of the Empirical-Type Simulated Annealing (ETSA) algorithm for solving the CVRP.

Algorithm : Empirical-Type Simulated Annealing (ETSA) algorithm for the CVRP

Output : X_{best} The best solution found by the algorithm
Begin
Generate a CVRP initial solution X_0.
$X_{best} \leftarrow X_0$.
$T \leftarrow T_0$ (Initialise current temperature).
Repeat
For ($i = 0$; $i <$ NumIter; $i++$) Do
Generate a new solution $X_{new} = X_0 + \delta X$ within the neighbourhood of X_0 ($X_{new} \in N(X)$)
$\Delta \leftarrow E(X_{new}) - E(X_0)$
If $\Delta < 0$ Then $X_0 \leftarrow X_{new}$, Else
Compose a set of m last positive changes $\Psi = \{\Delta_1, \ldots, \Delta_m\}$
Use Ψ to the calculate ML estimates $\hat{k}$ and $\hat{\theta}$ (Equations (14) and (15))
Generate a random number $\varepsilon \sim \mathcal{U}_{(0,1)}$ in the interval (0,1)
If $1 < \frac{1}{\epsilon \hat{\theta}^{\hat{k}} \Gamma(\hat{k})} (\Delta/T)^{\hat{k}-1} \exp\{-(\Delta/T\hat{\theta})\}$, Then $X_0 \leftarrow X_{new}$
$T \leftarrow \alpha \times T$, $\alpha \in (0,1)$ (Reduce current temperature)
If $E(X_0) < E(X_{best})$, Then $X_{best} \leftarrow X_0$
Until (Done) (Stopping condition)
Return solution X_{best}
End

Numerical results

The proposed approach was firstly compared to the simple SA algorithm in terms of accuracy, stability and convergence. Two experiments (subsections 4.1 and 4.2) were devoted to this first comparison, where the capacity of vehicles was assumed homogenous in experiment 1, and heterogenous in experiment 2. Then, the ETSA approach is compared to two well-known benchmark methods: **1)** the Genetic Algorithm (GA) with 2-opt heuristic function proposed by Wang, Cheng, Fang, and Qian (2004) (also see Baker and Ayechew, 2003), and **2)** the Discrete Particle Swarm Optimisation (DPSO) coupled with SA defined by Chen, Yang, and Wu (2006). In the third experiment (subsection 4.3), the comparison is mainly based on accuracy and complexity criteria. Altogether, more than 30 benchmark instances were used in all experiments. Through these experiments, an ECS with a value of 0.95 (as in Kirkpatrick et al., 1983) was fixed for all SA variants. The ETSA algorithm and the state-of-the-art algorithms used in the experiments were coded under MATLAB R2018a. The experiments were conducted on a computer with specifications encompassing Intel(R) Core (TM) i7 at 2.30 GHz, 8 Gb of RAM, and running on a 64-bit platform under Windows 8 Operating System.

Experiment 1

In order to test the performance of the ETSA algorithm, it is necessary to first compare it with the ordinary SA for some benchmarking purposes. The compared algorithms are used to solve the CVRP model for a set of benchmark instances. The first set of instances consists of the datasets that have been incrementally used for evaluating many VRP solvers. The main instances were proposed by Augerat et al. (1995) (datasets A, B, and P), Christofides and Eilon (1969) (dataset E), Fisher (1994) (dataset F), and Christofides, Mingozzi, and Toth (1979) (dataset M). The input data and optimal solutions are available online at http://www.coin-or.org/ (last access 3/2018). A summary of the datasets is given in Table 2. Only a part of the data of this table was used in the first experiment, in particular, instances No. 1, 2, 6, 7, 9, 13, 17, and 21. The remaining data were used in the below experiment.

The aim of this experiment is to compare the new generation of SA with their old homologous. Two variants, γ-ETSA and e-ETSA, among the ETSA family were compared to the SA algorithm. As

Table 2. List of benchmark instances used in experiments 1 and 3.

No.	Instances	Exact Solution	Number of Tours	Capacity	No.	Instances	Exact Solution	Number of Tours	Capacity
1.	A-n32-k5	784	5	100	12.	B-n78-k10	1221	10	100
2.	A-n33-k5	661	5	100	13.	E-n23-k3	569	3	4500
3.	A-n34-k5	778	5	100	14.	E-n30-k3	534	3	4500
4.	A-n46-k7	914	7	100	15.	E-n51-k5	521	5	160
5.	A-n60-k9	1354	9	100	16.	E-n76-k7	682	7	220
6.	B-n31-k5	672	5	100	17.	F-n45-k4	724	4	2010
7.	B-n34-k5	788	5	100	18.	F-n72-k4	237	4	30,000
8.	B-n35-k5	955	5	100	19.	F-n135-k7	1162	7	2210
9.	B-n43-k6	742	6	100	20.	M-n101-k10	820	10	200
10.	B-n45-k5	751	5	100	21.	P-n40-k5	458	5	140
11.	B-n68-k9	1272	9	100	22.	P-n101-k4	681	4	400

discussed in the last section, γ-ETSA possesses two parameters k and θ, while e-ETSA involves only one parameter. In both cases, the Maximum Likelihood method was used for estimating the parameters. The set of historical solutions for fitting the likelihood function consisted of the $m = 100$ last worse solutions. The results of the comparison, reported in Table 3, represent the statistics of the best solutions (minimal cost) and their averages after a number of 100 runs. As it can be observed, the proposed approaches provide the most accurate solutions for all tested instances. The γ-ETSA has the best performance for seven instances from a total of eight tested instances. The e-ETSA also shows appealing features with a best average for one instance. It is notable that the SA fails to find the optimal solution for two instances, but for each ETSA algorithm, the failure is only for one instance.

Experiment 2

The second experiment aims to further evaluate the ETSA, but with a special focus on the convergence and stability of the procedure. In fact, being one of the randomised algorithms, the convergence and stability of this SA variant need also to be checked. Thus, in this experiment, the ETSA was compared to the ordinary SA not only in terms of speed but also in terms of stability. The two compared schemes will therefore be applied to a slightly modified version of the CVRP, which assumes that the capacities of the vehicles are different. This problem, known as heterogeneous fleet VRP (HVRP), is obtained by reinforcing constraint Equation (7) in the CVRP. For more details on the definition and mathematical formulation of this problem, the reader can consult basic works such as in Salhi, Sari, Sadi, and Touati (1992), and also Taillard (1999). A new dataset of benchmark instances is used in this subsection. This set of benchmarks, proposed by Kalami (2015), is relatively new, since it has only been recently used for testing a MATLAB code for solving a CVRP using SA (http://yarpiz.com/372/ypap108-vehicle-routing-problem (last access 3/2018)). The main properties of these instances are detailed in Table 4. It is noticeable that the owners of the 10

Table 3. Computational results on the first set of benchmark problems. Instances No. 1, 2, 6, 7, 9, 13, 17, and 21 from Table 2 are used.

					γ-ETSA		e-ETSA	
	Cap.	Exact Sol.	SA Best	Avg.	Best	Avg.	Best	Avg.
A-n32-k5	100	784	784	823.03	784	818.50	784	821.77
A-n34-k5	100	778	778	812.11	778	807.02	778	805.49
B-n31-k5	100	672	673	753.04	672	684.09	673	686.72
B-n34-k5	100	788	788	812.42	788	798.91	788	798.12
B-n43-k6	100	742	745	756.79	742	752.83	744	759.59
E-n23-k3	4500	569	569	569.97	569	569.44	569	574.23
F-n45-k4	2010	724	725	750.59	724	751.88	724	753.51
P-n40-k5	140	458	459	476.61	459	475.33	459	478.70

Table 4. List of benchmark instances used in experiment 3.

No.	Instances	Number of Customers	Number of Vehicles	Capacities	Convergence* Bound
1.	hvrp-n8-k3	8	3	[54; 46; 49]	221
2.	hvrp-n10-k3	10	3	[66; 67; 73]	286
3.	hvrp-n14-k4	14	4	[74; 88; 79; 76]	276
4.	hvrp-n20-k4	20	4	[74; 88; 79; 76]	352
5.	hvrp-n25-k5	25	5	[92; 98; 87; 93]	331
6.	hvrp-n30-k5	30	5	[116; 111; 106; 109; 118]	350
7.	hvrp-n40-k6	40	6	[125; 117; 140; 137; 131; 133]	368
8.	hvrp-n50-k7	50	7	[130; 139; 134; 138; 135; 138; 150]	363
9.	hvrp-n60-k7	60	7	[157; 163; 159; 184; 158; 166; 172]	386
10.	hvrp-n70-k8	70	8	[167; 160; 173; 162; 172; 165; 166; 189]	390

instances do not provide their optimal bounds. Thus, since in this experiment we are only going to compare convergence speed and stability of ETSA and SA, we instead provided a convergence bound (threshold) to the cost function below which the procedure is considered to converge. Moreover, contrary to the above experiment, the capacity of the vehicles in this experiment is assumed heterogenous.

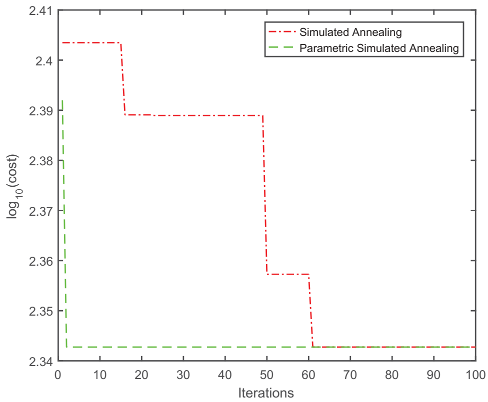
In the first part of this second experiment, we used instances from 1 to 8 to graphically analyse and compare the convergence of the $\gamma -$ ETSA and SA algorithms. The empirical results correspond to the minimal cost ($\log_{10}$ scale) reached at a preliminarily fixed number of iterations. Corresponding line charts are depicted in Figures 2 and Figure 3.. The plots clearly show the better convergence properties of the ETSA algorithm. For instances 1, 3, 5 and 6, ETSA spectacularly reaches the solution in a few iterations, which proves its appealing convergence characteristics, especially at the first steps. The second part of this experiment is essentially based on instances 9 and 10 of Table 4 and aims at further testing the convergence of the ETSA in comparison with the SA algorithm. The aim of the study is to consider a sufficiently high number of runs (we chose 100) and compare the minima of the objective function that are reached by each method. To better appreciate the convergence evolution, this process was repeated 5 times, with a number of iterations fixed each time to a multiple of hundred. The results of Tables 5 and 6 report the descriptive statistics. It is noticeable that the stability of a strategy can be assessed by the minimal standard deviation or range (maximal minus minimal values). The overall efficiency of a method is obviously based on the average minimum cost as well as its minimal dispersion. The results confirm the good performance of the new method, which is very promising.

Experiment 3

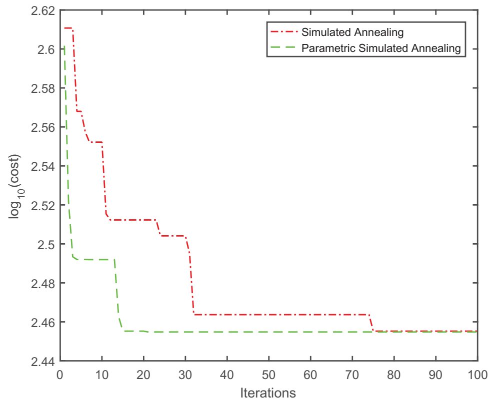
To further illustrate the effectiveness and good performance of the proposed strategy, we compared it to some of its homologous algorithms. A new set of benchmark instances with different sizes was also selected for the computation. Instances No. 3, 4, 5, 8, 10, 11, 12, 14, 15, 16, 18, 19, 20 and 22 from Table 2 were used in this experiment. The three chosen benchmark models are:

- Ordinary Simulated Annealing (SA)
- Genetic Algorithm (GA) with 2-opt heuristic function proposed by Wang et al. (2004) (also see Baker & Ayechew, 2003).
- Discrete Particle Swarm Optimisation (DPSO) coupled with SA by Chen et al. (2006).

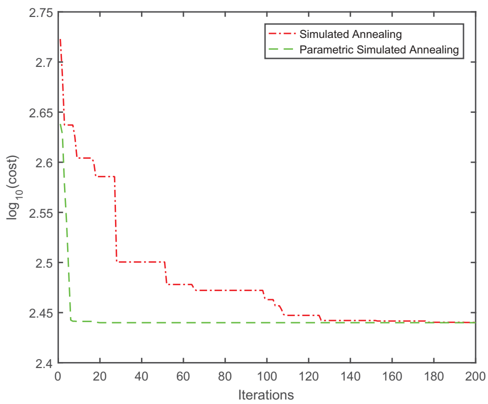
The results of this experiment are presented in Table 7. For each algorithm, the statistics correspond to the best objective value (minimal) and CPU time (in seconds) found over 5 runs. At the bottom of the table, averages of CPU are written in bold. Although it is considered to be an elementary method the ETSA outperforms hybrid approaches in terms of speed of convergence. Even when considering



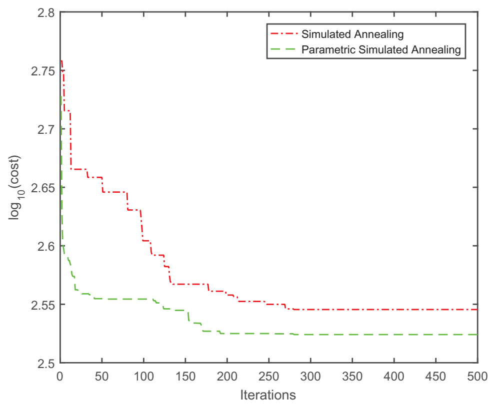
(a) Instance No. 1



(b) Instance No. 2



(c) Instance No. 3



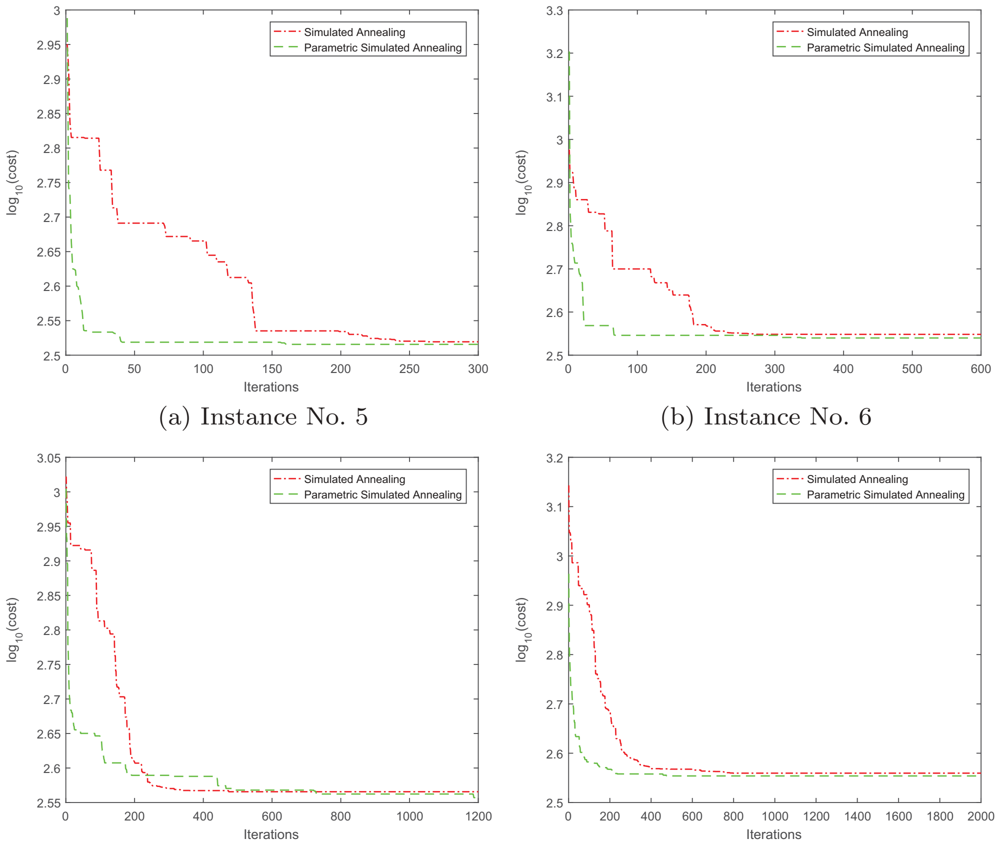
(d) Instance No. 4

Figure 2. Convergence analysis: The numerical results correspond to the minimal cost ($\log_{10}$ scale) reached by SA and ETSA (parametric) methods at a preliminarily fixed number of iterations.

accuracy, it provides the best performance, together with DPSO-SA, with 7 exact solutions from a total of 14 tested instances. It is noticeable that the ETSA could be straightforwardly implemented within any hybrid procedure instead of the SA, such as with the DPSO. The conclusion drawn from this experiment is that, despite its simplicity, the ETSA has proven able to provide accurate results for various instances. In future researches, the model revealed in this paper could be further extended with more flexible probabilistic models, where ML estimates could be computed using an accelerated version of the Expectation Maximisation (EM) algorithm (Rabbouch, Saâdaoui, & Mraihi, 2016, 2018; Saâdaoui, 2010, 2012, 2016).

Conclusion

We studied a new algorithmic optimisation approach extending the simulated annealing to solve the capacitated vehicle routing problem with homogenous fleets of vehicles. The technique exploits, within an SA algorithm, the fresh portion of the worst feasible solutions to update its Metropolis decision rule. In fact, SA usually uses a nonparametric exponential function that does not take into account histories during the search process. In this paper, we extended this criterion by choosing a new parametric density function, which is incrementally adjusted according to the



(a) Instance No. 5 (b) Instance No. 6

(c) Instance No. 7 (d) Instance No. 8

Figure 3. Convergence analysis: The numerical results correspond to the minimal cost ($\log_{10}$ scale) reached by SA and ETSA (parametric) methods at a preliminarily fixed number of iterations.

Table 5. Stability analysis: The empirical results correspond to the minimal cost reached by each one of the compared methods at a preliminarily fixed number of iterations. Each experiment consists of a number of 100 trials.

	# Iterations	Mean	Std Deviation	Minimum	Maximum
SA					
	100	407.0463	13.2542	387.6370	446.2373
	200	389.7896	9.3428	377.1103	406.5970
	300	386.4811	6.9329	372.6674	394.5048
	400	382.0818	5.1122	373.9193	395.2519
	500	382.8549	4.7477	373.7025	398.9479
ETSA					
	100	398.2610	10.1810	384.9517	433.9208
	200	386.6161	6.5407	376.3567	405.8942
	300	383.4539	5.6593	371.3648	393.7225
	400	380.6920	4.4922	371.3644	392.1754
	500	379.7474	3.8251	372.6115	386.6073

distribution of the rejection solutions. This new procedure helps to reduce the randomness of the SA and is important for improving its convergence. The proposed approach was compared to a number of state-of-the-art metaheuristic methods for the resolution of the CVRP for a set of benchmark instances. The results showed that the proposed algorithm can reach a great accuracy in comparison with its homologous. Considered as an elementary procedure, the new approach

Table 6. Stability analysis: The empirical results correspond to the minimal cost reached by each one of the compared methods at a preliminarily fixed number of iterations. Each experiment consists of a number of 100 trials.

	# Iterations	Mean	Std Deviation	Minimum	Maximum
SA					
	100	427.9562	16.7116	400.7943	488.2732
	200	401.6930	13.9535	372.6733	440.9981
	300	393.1250	11.6521	373.1680	432.8431
	400	388.7070	9.6061	370.8303	412.1850
	500	385.7596	10.4657	368.1426	413.7035
ETSA					
	100	424.8148	16.6206	399.3501	482.2155
	200	396.9267	12.2620	371.7579	436.1957
	300	392.0087	11.2373	371.6521	422.6790
	400	386.9257	9.3117	367.9390	410.3227
	500	384.2821	9.4752	367.4562	406.7424

Table 7. Comparative study: Computational results of SA, GA with 2-opt, DPSO-SA (Chen et al., 2006) and γ-ETSA. Instances No. 3, 4, 5, 8, 10, 11, 12, 14, 15, 16, 18, 19, 20 and 22 from Table 2 are used.

	Ordinary SA		DPSO-SA		GA with 2-opt		γ-ETSA	
Instances	Best	CPU (s)	Best	CPU (s)	Best	CPU (s)	Best	CPU (s)
A-n33-k5	661	38.2	661	32.3	661	39.6	661	3.016
A-n46-k7	931	143.8	914	128.9	928	136.4	914	19.52
A-n60-k9	1363	286.3	1354	308.8	1360	295.5	1354	49.16
B-n35-k5	960	58.4	955	37.6	955	46.9	955	9.219
B-n45-k5	760	123.5	751	134.2	762	129.3	751	25.02
B-n68-k9	1298	409.2	1272	344.3	1296	396.2	1272	77.03
B-n78-k10	1256	483.3	1239	429.4	1248	568.4	1236	138.9
E-n30-k3	534	69.3	534	28.4	534	30.5	534	4.53
E-n51-k5	541	362.4	528	300.5	531	289.6	523	14.78
E-n76-k7	704	619.3	688	526.5	697	498.7	696	27.88
F-n72-k4	253	604.6	244	398.3	246	468.5	238	37.67
F-n135-k7	1243	2533.9	1215	1526.3	1246	1894.2	1181	153.6
M-n101-k10	848	986.6	824	874.2	836	992.1	836	84.89
P-n101-k4	715	1964.9	694	977.5	706	1213.2	699	94.22
Average		**620.26**		**431.94**		**499.93**		**52.816**

could be easily integrated within a hybrid scheme involving another metaheuristic algorithm. In future work, the method could be extended to tackle other VRP problems, such as the CVRP with time window or the multi-depot VRP.

List of notations

$G = (V, A)$: Directed graph
$A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$: Set of arcs
L: Number of vehicles ($l = 1, \ldots, L$)
c: Capacity of vehicles
$X_{ijl} = \{0, 1\}$: Vehicle l travelling (1) or not (0) along arc (i, j)
D_{ij}: Distance between points i and j, for all $i, j \in V$
T: Computational temperature
Δ: First forward difference operator (measuring change)
$p(X)$: Probability of a state change
$f(X; \theta, k)$: Gamma density probability with parameters θ and k
θ: Scale parameter of the Gamma distribution
$V = \{v_0, \ldots, v_n\}$: Set of vertices (v_0 is the depot)
$C = \{v_1, \ldots, v_n\}$: Set of customers
v_0: Depot where are parked the L vehicles
d_i: Demand at vertex $v_i \in V$ ($d_0 = 0$ and $d_i \leq c$)
Y_{ij}: Load in the vehicle before reaching j coming from i
$E(X)$: Energy function of a solution X
ε: Uniform random variable
Ψ: Set of last positive changes
α: Cooling rate ($\alpha \in (0, 1)$)
$\Gamma(k)$: Gamma function evaluated at k
k: Shape parameter of the Gamma distribution

Acknowledgments

We would like to thank the anonymous reviewers, for their insightful and constructive comments that have greatly contributed to improving the paper, and to the editorial staff for their generous support and assistance during the review process.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Foued Saâdaoui http://orcid.org/0000-0002-0574-2922

References

Afifi, S., Dang, D., & Moukrim, A. (2013). A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints. *7th International Conference, Learning and Intelligent Optimization (LION7)* (pp. 259–265), Catania, Italy.

Augerat, P., Belenguer, J., Benavent, E., Corbern, A., Naddef, D., & Rinaldi, G., (1995). *Computational results with a branch and cut code for the capacitated vehicle routing problem* (Research Report 949-M). Grenoble, France: Universite Joseph Fourier.

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, *30*(5), 787–800.

Birim, S. (2016). Vehicle routing problem with cross docking: A simulated annealing approach. *Procedia – Social & Behavioral Sciences*, *235*, 149–158.

Breedam, A. V. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operations Research*, *86*(3), 480–490.

Chen, A. L., Yang, G. K., & Wu, Z. M. (2006). Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University Science A*, *7*(4), 607–614.

Chiang, W.-C., & Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, *63*(1), 3–27.

Choi, S. C., & Wette, R. (1969). Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, *11*(4), 683–690.

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, *20*, 309–318.

Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), *Combinatorial optimization* (pp. 313–318). Chichester: Wiley.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum K-trees. *Operations Research*, *42*(4), 626–642.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Johnson, R. K., & Sachin, F., (2009). Particle swarm optimization methods for data clustering. *IEEE fifth international conference soft computing, computing with words and perceptions in system analysis, decision and control*, Famagusta, Cyprus. (pp. 1–6).

Kalami, S. M., (2015). *Capacitated Vehicle Routing Problem (CVRP) using SA*. Yarpiz Project. Retrieved from www.yarpiz.com

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Leung, S. C. H., Zheng, J., Zhang, D., & Zhou, X. (2010). Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services and Manufacturing Journal*, *22*(1), 61–82.

Lin, S.-W., Yu, V. F., & Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, *36*, 1683–1692.

Minka, T. P., (2002). *Estimating a gamma distribution* (Technical Report). Cambridge, UK: Microsoft Research.

Mirmohammadsadeghi, S., & Ahmed, S. (2015). Metaheuristic approaches for solving truck and trailer routing problems with stochastic semands: a case study in dairy industry. *Mathematical Problems in Engineering*, vol 2025. Article ID 494019.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, *41*, 421–451.

Rabbouch, B., Mraihi, R., & Saâdaoui, F., (2017). A recent brief survey for the multi-depot heterogenous vehicle routing problem with time windows. *17th International Conference on Hybrid Intelligent Systems (HIS'2017)*, Delhi, India.
Rabbouch, H., Saâdaoui, F., & Mraihi, R. (2016). Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing. *Neurocomputing*, *260*, 157–173.
Rabbouch, H., Saâdaoui, F., & Mraihi, R. (2018). A vision-based statistical methodology for automatically modeling continuous urban traffic flows. *Advanced Engineering Informatics*, *38*, 392–403.
Saâdaoui, F. (2010). Acceleration of the EM algorithm via extrapolation methods. *Review, Comparison and New Methods, Computational Statistics and Data Analysis*, *54*(3), 750–766.
Saâdaoui, F., (2012). Technical iterative strategies for extreme risk modeling in finance (Thèse de Doctorat de l'Université de Sousse, Avril 2012).
Saâdaoui, F., (2016). *Selected topics on wavelet analysis and statistical applications* (Rapport de Synthèse pour l'Habilitation Universitaire). Université de Sousse, Juillet.
Sait, S. M., & Youssef, H., (1999). Iterative computer algorithms with applications in engineering: solving combinatorial optimization problems. *IEEE Computer Society*, Los Alamitos, CA.
Salhi, S., Sari, M., Sadi, D., & Touati. (1992). Adaptation of some vehicle fleet mix heuristics. *Omega*, *20*(5–6), 653–660.
Sigauke, C., & Talukder, H. M. (2003). A modified Osman's simulated annealing and tabu search algorithm for the vehicle routing problem. *Australian Society For Operations Research*, *22*(3), 9–14.
Taillard, E. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO Operations Research*, *33*(1), 1–14.
Tavakkoli-Moghaddam, R., Safaei, N., & Gholipour, Y. (2006). A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, *176*(2), 445–454.
van Laarhoven, P. J. M., & Aarts, E. H. L. (1987). *Simulated annealing: Theory and applications, D. Reidel*. Dordrecht: Kluwer Academic Publishers.
Wang, Z. Z., Cheng, J. X., Fang, H. G., & Qian, F. L. (2004). A hybrid optimization algorithm solving vehicle routing problems. *Operations Research & Management Science*, *13*, 48–52. (in Chinese).
Yu, V. F., & Lin, S. Y. (2015). A simulated annealing heuristic for the open location-routing problem. *Computers & Operations Research*, *62*, 184–196.
Yu, V. F., & Lin, S.-Y. (2016). Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. *International Journal of Production Research*, *54*(2), 526–549.
Yu, V. F., Perwira Redi, A. A. N., Hidayat, Y. A., & Wibowoa, O. J. (2017). A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, *53*, 119–132.

Appendix ML Estimates of a Gamma Distribution

The likelihood function of a gamma distribution for N $i.i.d.$ observations $(z_1, \ldots, z_N)$ is

$$L(k,\theta) = \prod_{i=1}^{N} f(z_i; k, \theta),$$

which leads to the log-likelihood function

$$\ell(k,\theta) = (k-1)\sum_{i=1}^{N} \ln(z_i) - \sum_{i=1}^{N} \frac{z_i}{\theta} - Nk\ln(\theta) - N\ln(\Gamma(k)). \tag{13}$$

The Maximum Likelihood Estimator (MLE) of θ is:

$$\hat{\theta} = \frac{1}{kN}\sum_{i=1}^{N} z_i. \tag{14}$$

Substituting Equation (14) into the log-likelihood function and maximising with respect to k yields

$$\ln(k) - \psi(k) = \ln\left(\frac{1}{N}\sum_{i=1}^{N} z_i\right) - \frac{1}{N}\sum_{i=1}^{N} \ln(z_i),$$

where $\psi(.)$ the digamma function. An initial value of k can be found either using the method of moments, or using the approximation

$$\ln(k) - \psi(k) \approx \frac{1}{2k}\left(1 + \frac{1}{6k+1}\right).$$

If we consider that

$$q = \ln\left(\frac{1}{N}\sum_{i=1}^{N} z_i\right) - \frac{1}{N}\sum_{i=1}^{N} \ln(z_i)$$

then k is approximately

$$\hat{k} \approx \frac{3 - q + \sqrt{(q-3)^2 + 24q}}{12q}, \tag{15}$$

which is within 1.5% of the actual value (Minka, 2002). Another form from the Newton–Raphson procedure is given by Choi and Wette (1969):

$$\hat{k} := k - \frac{\ln(k) - \psi(k) - s}{\frac{1}{k} - \psi'(k)}. \tag{16}$$