# Solving Stochastic Vehicle Routing Problem with Real Simultaneous Pickup and Delivery Using Differential Evolution

Eshetie Berhan[1], Pavel Krömer[2,3],
Daniel Kitaw[1], Ajith Abraham[3], and Václav Snášel[2,3]

[1] Addis Ababa University
Addis Ababa Institute of Technology
School of Mechanical and Industrial Engineering
Addis Ababa, Ethiopia
{eshetie_ethio,danielkitaw}@yahoo.com
[2] Faculty of Electrical Engineering and Computer Science
VŠB Technical University of Ostrava
Ostrava, Czech Republic
[3] IT4 Innovations
VŠB Technical University of Ostrava
Ostrava, Czech Republic
{pavel.kromer,vaclav.snasel}@vsb.cz

**Abstract.** In this study, Stochastic VRP with Real Simultaneous Pickup and Delivery (SVRPSPD) is attempted the first time and fitted to a public transportation system in Anbessa City Bus Service Enterprise (ACBSE), Addis Ababa, Ethiopia. It is modeled and fitted with real data obtained from the enterprise. Due to its complexity, large instances of VRP and/or SVRPSPD are hard to solve using exact methods. Instead, various heuristic and metaheuristic algorithms are used to find feasible VRP solutions. In this work the Differential Evolution (DE) is used to optimize bus routes of ACBSE. The findings of the study shows that, DE algorithm is stable and able to reduce the estimated number of vehicles significantly. As compared to the traditional and exact algorithms it has exhibited better used fitness function.

**Keywords:** vehicle routing problem, pickup and delivery, machine learning, differential evolution, real-world application.

## 1 Introduction

The problem of designing a minimum cost set of routes to serve a collection of customers with a fleet of vehicles is a fundamental challenge in the field of logistics, distribution and transportation [1]. This is because transportation and distribution contribute approximately 20% to the total costs of a product [2]. The task of designing delivery or pickup routes to service customers in the transport and supply chain is known in the literature as a Vehicle Routing Problem [1]. It

A. Abraham et al. (eds.), *Innovations in Bio-inspired Computing and Applications*,
Advances in Intelligent Systems and Computing 237,
DOI: 10.1007/978-3-319-01781-5_18, © Springer International Publishing Switzerland 2014

was the first time proposed by [3] under the title "Truck dispatching problem" with the objective to design optimum routing of a fleet of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied by the terminal. Often the context is that of delivering goods located at a central depot to customers who have placed orders for such goods, but the area of application of VRP is also versatile and are used in many areas in real world life.

This paper is trying to develop a VRP model that addresses the stochastic nature of passengers' pickup and delivery services in Anbessa City Bus Service Enterprise (ACBSE), Addis Ababa, Ethiopia. ACBSE is a urban public bus transport enterprise which provides a public transport service in the city of Addis Ababa, Ethiopia. The model developed is called Stochastic VRP with Real Simultaneous Pickup and Delivery (SVRPSPD). It is the first of its kind in the literature of VRP due to the fact that it considered stochastic pickup and delivery of commuters', and both the pickup and delivery services are also performed simultaneously during the transportation services. The SVRPSPD model is simulated and solved using Differential Evolution with real data collected from ACBSE.

## 2 Literature Review

In general, VRPs are represented in a graph theory. The general model is defined as: let $G = (V, A)$ be a directed or asymmetric graph where $V = \{0, ...n\}$ is a set of vertices representing *cities* with *depot* located at vertex 0, and $A$ is the set of arcs. With every arc $(i, j), i \neq j$ is associated a non-negative distance matrix $C = (c_{ij})$. In some context, $c_{ij}$ can be interpreted as a *travel cost* or as a *travel time* [4]. When $C$ is symmetrical, it is often convenient to replace $A$ by a set $E$ of undirected edges, the graph being called symmetric or undirected graph. In many practical cases, the cost or the distance matrix satisfies the triangular inequality such that $c_{ik} + c_{kj} \geq c_{ij}$, $\forall_{i,j,k} \in V$ [5]. The general or classical VRP consists of designing a set of at most $K$ delivery or collection routes such that each route starts and ends at the depot, each customer is visited exactly once by exactly one vehicle, the total demand of each route does not exceed the vehicle capacity and the total routing cost is minimized [5].

Moreover, because of the complexity and the practical relevance of VRPs, vast literature is devoted to the Bus Scheduling Problem (BSP) and many optimization models have been proposed [6]. The models tried to achieve several near optimal solutions with a reasonable amount of computational effort [1]. Various extensions for the Vehicle Schedule Problem (VSP) or VRP with different additional requirements were also covered in the literature over the last fifty years [5]. Among others the existence of one depot [7] or more than one depot [8], a heterogeneous fleet with multiple vehicle types [7] the permission of variable departure times of trips, VRP with Stochastic Demand (VRPSD) [1,9,10] is also one of the major variants of VRP in the literature and the VRP with incapacitated vehicle [11,12,10] are the very few examples in the VRP literature.

The basic version of VRP, Stochastic VRP (SVRP) and/or VRP with stochastic Customer Demand (VRPSD), which are stated above are either a pure pickup or a pure delivery problems [13,14]. The pure pickup or pure delivery types of VRP has been extensively studied in the literature with many application areas [15].

The VRP with delivery and pickup (VRPDP) is also a well studied in the literature but with the assumption of deterministic demand or with modifications of as classical VRP models separately for the pickup and for the delivery [16]. As it is evident in [17] and the surveying of [18], the problem can be divided into two independent CVRPs [16]; one for the delivery (linehaul) customers and one for the pickup (backhaul) customers, such that some vehicles would be designated to linehaul customers and others to backhaul customers.

A similar models and approach was also studied VRPSDP by [19] with consideration of first delivery and then followed by pickup service but named as simultaneous delivery and pickup VRP problem [20,21,22]. This assumption is more clearly illustrated by [18] and [19] with a symbol representation and mathematical model. The model illustrated first with symbols ▼ as delivery and second with symbols ▲ as pickup along each route [19]. Where as in the work of [18,20,21,23] it considered $P$ as a set of backhauls or pickup vertices, $P = \{1, \ldots, n\}$ and $D$ as a set of linehauls or delivery vertices, $D = \{n + 1, \ldots, n + \tilde{n}\}$. As it can be seen from this consideration, the assumption is that, in VRPSPD all delivered goods must be originated from the depot and served to n nodes ( $\{1, \ldots, n\}$ ) and all pickup goods must be transported back to the depot $\{n + 1, \ldots, n + \tilde{n}\}$ or even viceversa.

A real simultaneous delivery and pick up with deterministic demand was noted on the work of [24,25,26] bust with deterministic demand. In addition to this drawback, previous works on VRPSDP has also limitation on the consideration of demand. In other work of the same authors [27], studied simultaneous pickup and delivery service but deliveries are supplied from a single depot at the beginning of the service followed by pickup loads to be taken to the same depot at the conclusion of the service.

### 2.1 Differential Evolution

The DE is a versatile and easy to use stochastic evolutionary optimization algorithm [28]. It is a population-based optimizer that evolves a population of real encoded vectors representing the solutions to given problem. The DE was introduced by Storn and Price in 1995 [29,30] and it quickly became a popular alternative to the more traditional types of evolutionary algorithms. It evolves a population of candidate solutions by iterative modification of candidate solutions by the application of the differential mutation and crossover [28]. In each iteration, so called trial vectors are created from current population by the differential mutation and further modified by various types of crossover operator. At the end, the trial vectors compete with existing candidate solutions for survival in the population.

**The DE Algorithm.** The DE starts with an initial population of $N$ real-valued vectors. The vectors are initialized with real values either randomly or so, that they are evenly spread over the problem space. The latter initialization leads to better results of the optimization [28].

During the optimization, the DE generates new vectors that are scaled perturbations of existing population vectors. The algorithm perturbs selected base vectors with the scaled difference of two (or more) other population vectors in order to produce the trial vectors. The trial vectors compete with members of the current population with the same index called the target vectors. If a trial vector represents a better solution than the corresponding target vector, it takes its place in the population [28].

There are two most significant parameters of the DE [28]. The scaling factor $F \in [0, \infty]$ controls the rate at which the population evolves and the crossover probability $C \in [0, 1]$ determines the ratio of bits that are transferred to the trial vector from its opponent. The size of the population and the choice of operators are another important parameters of the optimization process.

The basic operations of the classic DE can be summarized using the following formulas [28]: the random initialization of the $i$th vector with $N$ parameters is defined by

$$x_i[j] = rand(b_j^L, b_j^U), \quad j \in \{0, \ldots, N-1\} \tag{1}$$

where $b_j^L$ is the lower bound of $j$th parameter, $b_j^U$ is the upper bound of $j$th parameter and $rand(a, b)$ is a function generating a random number from the range $[a, b]$. A simple form of the differential mutation is given by

$$v_i^t = v_{r1} + F(v_{r2} - v_{r3}) \tag{2}$$

where $F$ is the scaling factor and $v_{r1}$, $v_{r2}$ and $v_{r3}$ are three random vectors from the population. The vector $v_{r1}$ is the base vector, $v_{r2}$ and $v_{r3}$ are the difference vectors, and the $i$th vector in the population is the target vector. It is required that $i \neq r1 \neq r2 \neq r3$. The uniform crossover that combines the target vector with the trial vector is given by

$$l = rand(0, N-1) \tag{3}$$

$$v_i^t[m] = \begin{cases} v_i^t[m] & \text{if } (rand(0,1) < C) \text{ or } m = l \\ x_i[m] & \end{cases} \tag{4}$$

for each $m \in \{1, \ldots, N\}$. The uniform crossover replaces with probability $1 - C$ the parameters in $v_i^t$ by the parameters from the target vector $x_i$.

There are also many other modifications to the classic DE. Mostly, they differ in the implementation of particular DE steps such as the initialization strategy, the vector selection, the type of differential mutation, the recombination operator, and control parameter selection and usage [28].

**Recent Applications of DE to the Vehicle Routing Problem.** Large VRP instances are due to the NP-hardness of the problem hard to solve using exact methods. Instead, various heuristic and metaheuristic algorithms are employed to find approximate VRP solutions in reasonable time [31,32]. A categorized bibliography of different metaheuristic methods applied to VRP variants can be found in [32].

The DE has proved to be an excellent method for both, continuous and discrete optimization problems. This section provides a short overview of recent applications of DE to different variants of the VRP published in 2012. Hou et al. introduced in [33] a new discrete differential evolution algorithm for stochastic vehicle routing problems with simultaneous pickups and deliveries. The proposed algorithm used natural (integer) encoding with the symbol 0 as sub-route separator and fitness function incorporating routing objective and constraints. Besides the traditional DE operators, new bitwise mutation was proposed. The algorithm also utilized an additional *revise* operator to eliminate illegal chromosomes that might have been created during the evolution. The experiments conducted by the authors have shown that the proposed algorithm delivers better solutions and converges faster than other DE-based and GA-based VRP solvers.

Liu et al. [34] used a memetic differential evolution algorithm to solve vehicle routing problem with time windows. The algorithm used a real-valued source space and discrete solution space. A source vector was translated into an solution vector by modifications of the source vector (e.g. insertion of sub-route separator '0' in feasible locations) and optimized by three local search algorithms. The fitness of the best routing found by the local searches was called generalized fitness of the source vector. The experiments performed by the authors have shown that the proposed modifications improve the quality of solutions found by the DE and that the new algorithm is especially suitable for solving VRP instances with clustered locations.

Xu and Wen [35] used differential evolution for unidirectional logistics distribution vehicle routing problem with no time windows. The authors approached the task as an multi-objective optimization problem (although none of the traditional multi-objective DE variants was used) and established an encoding scheme that mapped the real-valued candidate vector to a routing of $K$ vehicles.

## 3 Model Formulation

The presented model is the first of its kind considering VRP with real simultaneous pickup and delivery at each bus stop where the pickup and delivery demand at each bus stop is treated as stochastic and random. The presented model assumes:

1. The number of passengers expected to be picked up and dropped is uncertain, but follows a poisson probability distribution.
2. The cumulative number of passengers picked up along the route will not exceed the vehicle capacity. Moreover, split delivery is not allowed.
3. The fleet consists of homogeneous vehicles with limited capacity operating from a single depot.
4. Each vehicle can be used repeatedly within the planning horizon.

The mathematical model for SVRP which considered a simultaneous pickup and delivery techniques is formulated as a mixed integer LP problem. Consider a fleet of $K$ vehicles with $k = \{1, 2, \ldots, K\}$ with identical vehicle capacity of $Q$ serving a set of passengers with demand (to be picked or dropped) in passenger's location $V \backslash \{0\}; v_0 =$ depot, each passenger must be completely served by a single vehicle. Each vehicle starts from the depot $v_0$ and picks and/or drops passengers on each visited node $v_i$ except the depot. Moreover, the first node $v_1$ is treated as pickup-only node and the last node before the depot $v_n$ is treated as drop-only node.

Suppose a vehicle starts from the depot $v_0$ and travels along a certain path until it reaches node $v_n$. Along the path $(v_1, v_2, \ldots, v_n)$, the vehicle will pick and/or drop passengers up to the last node $v_n$. The cumulative number of passengers picked by vehicle $k$ denoted as $C_p$ and the cumulative number of passengers dropped along the path $v_n$ $(v_1, v_2, \ldots, v_n)$ denoted as $C_d$ are given by:

$$C_p(V_n) = \sum_{i \in V(0, V_n)} p_i \tag{5}$$

$$C_d(V_n) = \sum_{i \in V(0, V_n)} d_i \tag{6}$$

where $p_i$ is the number of passengers picked up at node $i$ and $d_i$ is the number of passengers dropped at the same node $i$. At the depot, $C_p = C_d = 0$ and the vehicle capacity equals to $Q$. The path becomes infeasible if the cumulative load exceeds the vehicle capacity $Q$, that is when: $C_d \geq Q$ and $C_p \geq Q$.

Each feasible route will be formed when: $C_d(v_n) \leq Q$ and $C_d(v_{n+1}) \geq Q$ and $C_p(v_n) \leq Q$ and $C_p(v_{n+1}) \geq Q$. The other consideration checks whether the net load of a bus for any consecutive nodes will not exceed the bus capacity after the bus visiting node $v_n$. Let the net load picked is $L_p(v_n)$ it is computed as $L_p(v_n) = C_p(v_n) + L_p(v_{n-1}) - C_d(v_n)$.

The solution will be feasible if a vehicle served all the demand (for pick up or drop off) at each node along the path or route without exceeding its capacity. That is the net load in transit between any consecutive nodes or vertices should not exceed the vehicle capacity $(L_p(v_n) \leq Q)$. The objective is to determine a route for each vehicles that serve a set of nodes $(v_i)$ so that the total distance traveled is minimized. To formulate the model of SVRP with simultaneous pickup and delivery as Mixed Integer LP problem, the following notations and definitions are used:

$V$ is set of nodes or vertices $V = \{v_0, v_1, v_2, \ldots, v_n\}$ treated as bus stops; $v_0 =$ depot
$A$ is set of arcs $(i, j) \in A$
$K$ is number of vehicles $k = \{1, 2, \ldots, K\}$
$c_{ij}$ is the distance traversing from node $i$ to node $j$
$p_i$ is the number of passengers (demand) to be picked at node $i$, which is a random non-negative integer
$d_i$ is the number of passengers (serviced) to be dropped at node $i$ which is a random non-negative integer

$Q$ is the vehicle capacity
$n$ is total number of nodes or vertices or bus stops included in the model

### Decision Variables

$y_i^k$ is the cumulative number of passengers picked by vehicle $k$ when leaving from node $i$.
$z_i^k$ is the number of passengers remaining in vehicle $k$ when leaving from node $i$.
$x_{ij}^k = 1$ if vehicle $k$ travels from node $i$ to node $j$; 0 otherwise

Then the general model is represented as follows:

$$\text{Minimize} \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij}^k \tag{7}$$

Subject to

$$\sum_{j=1}^{n} x_{0j}^k \leq 1; \tag{8}$$

$$\sum_{i=0}^{n} x_{ij}^k = 1; \tag{9}$$

$$\sum_{i=0}^{n} x_{ij}^k - \sum_{i=0}^{n} x_{ji}^k = 0; \tag{10}$$

$$z_i^k + y_i^k \leq Q; \tag{11}$$

$$(z_i^k - d_j - z_j^k) x_{ij}^k = 0; \tag{12}$$

$$(y_i^k + p_j - y_j^k) x_{ij}^k = 0; \tag{13}$$

$$z_0^k = y_0^k = 0; \tag{14}$$

$$z_i^k \geq 0; \tag{15}$$

$$y_i^k \geq 0; \tag{16}$$

$$x_{ij}^k \in \{0, 1\}; \tag{17}$$

$$\forall_k = 1, \ldots, K \quad and \quad i, j = 1, \ldots, n \tag{18}$$

The decision variables are given above and equation 7 is the objective function to be minimized. Equation 8 ensures that each vehicle is used at most once, equation 9 indicates that each node has to be visited exactly by one vehicle, equation 10 shows that the same vehicle arrives and departs from each node it serves, 11 ensures that the load on vehicle $k$ when departing from node $i$ is always less than the vehicle capacity. Equation (12) and eq. (13) are the transit load constraint, which indicate that when arc $(i, j)$ is traversed by vehicle $k$, the number of passengers to be dropped by the vehicle has to be decreased by $d_j$ while the number of passengers picked-up has to be increased by $p_j$. 14 ensures that the remaining and the cumulative number of passengers when a vehicle $k$ departs from the depot is always zero; indicates that the vehicle is empty and available with full capacity. Constraints 15 and 16 are a non-integer and non-negative sign restriction and the last equation 17 is an restriction on the non-negative integer value.

# 4 Model Input Parameters

To run and evaluate the model, different input parameters that have to be substituted to the model are required. These inputs are either collected or generated/computed. The from-to-distance, the demand realization probability, the demand distributions are computed or generated whereas the longitude and latitude value for each location point $v_i$ is collected from the Google Earth. Each of them are briefly explained and presented in section 4.1 and 4.2

## 4.1 From-to-Distance

Origin-destination of 59 locational points that depart and end from Merkato Terminal (lat. 9°1'50"N and long. 38°44'15"E), are used in the model validation. The origin-destination points are modified in such a way that they can fit to the model but without losing its information. The from-to-distance computational input parameters of each location $i$ is computed by taking the longitude and latitude location of each point using Great Circle distance formula that considers the circular nature of earth.

**Table 1.** From-to-distance matrix ($C_{ij}$)

| $v_{ij}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | . | . | . | 57 | 58 | 59 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 8 | 17 | 7 | 7 | 8 | 1 | 1 | . | . | . | 17 | 20 | 19 |
| 2 | 8 | 0 | 11 | 7 | 2 | 8 | 9 | 7 | . | . | . | 22 | 14 | 23 |
| 3 | 17 | 11 | 0 | 11 | 13 | 10 | 18 | 15 | . | . | . | 33 | 20 | 21 |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 58 | 20 | 14 | 20 | 21 | 14 | 22 | 21 | 20 | . | . | . | 27 | 0 | 37 |
| 59 | 19 | 23 | 21 | 15 | 24 | 15 | 20 | 18 | . | . | . | 34 | 37 | 0 |

Each $c_{ij}$ is defined as the distance from $i$ to $j$, which can be directly considered as the cost associate to transport passenger demand including depot 1. Further, it assumes that the distance is symmetric, $c_{ij} = c_{ji}$, and $c_{ii} = 0$. The sample output is shown in Table 1.

## 4.2 Stochastic Passengers Demand

The other input parameter is the number of passengers picked up and dropped off at each location point called passengers demand. The passengers' demand collected in 10 routes of ACBSE are used to fit the demand behavior of passengers in the remaining routes. The snap shot of the demand distribution of passengers picked and dropped along the 10 selected routes were used to fit the demand distribution of the remaining location points.

The volumes of passengers picked up at each vertex $V$ are integer-valued random variables with known probability distributions denoted by vector $\widetilde{p}(i) \in Z$[36]. In the demand based modeling individual demands are estimated by using parameter $\gamma = \{0, 1\}$, which determines the risk preferences [14].

Therefore, the demand used for computing the solutions is evaluated using $\hat{p} = \gamma p_i^{min} + (1-\gamma)p_i^{max}$. Finally, for convenience of notation, it can be defined that $\underline{p}(0) = \overline{p}(0) = 0$. Similarly, $\hat{d}$ is also computed using the same formula. $\gamma = 0$ clearly implies the risk averse (risk free) case where failure can never occur, while $\gamma = 1$ is the other extreme, which is risk seeking. Moreover, $\gamma = 0.5$ corresponds to computing solutions with the expected demand. According to the simulation run, the demand distribution of the number passengers picked up and delivered for some nodal points are given below in Table 2.

**Table 2.** Sample demand distribution and location data

| $v_i$ | Location(decimal) | | Expected Passengers | |
|---|---|---|---|---|
| | Longi. | Lati. | $\hat{p_i}$ | $\hat{d_i}$ |
| 1 | 38.8 | 9.16 | na | na |
| 2 | 38.9 | 9.1 | 8 | 4 |
| 3 | 38.9 | 8.94 | 9 | 3 |
| . | . | . | . | . |
| . | . | . | . | . |
| 58 | 39.0 | 9.13 | 9 | 4 |
| 59 | 38.8 | 9.04 | 10 | 3 |

## 5 Differential Evolution for SVRPSPD

The version of VRP considered in this work can be seen as a combinatorial optimization problem. The goal of the optimization is to find a set of routes connecting selected locations (bus stops) so that each location is visited by a vehicle exactly once, each route starts and terminates in a special location (depot), considered constraints are satisfied, and selected objective function is minimized. In this work we represent a set of routes as a permutation of considered locations (without the depot) and separate each sub-route by a special sub-route separator similarly as e.g. in [31].

The DE proposed in this work uses permutation-based VRP representation, automatically selects the number of vehicles when an upper bound is given, and avoids the creation of illegal candidate solutions.

**Encoding.** There is a variety of possible encoding schemes for modelling permutations for populational metaheuristic algorithms [37]. The DE uses real-encoded candidate solutions so a modified version of the random key (RK) encoding [38] was chosen. An RK encoded permutation is represented as a string of real numbers (random keys), whose position changes after sorting correspond to the permutation gene. The advantage of RK encoding is that it is at a large extent prone to creation of illegal solutions in course of the artificial evolution (e.g. by the crossover operator in Genetic Algorithms). The drawbacks of the RK encoding include computational complexity as it is necessary to perform a sorting of the random keys every time the candidate solution is decoded. Moreover, RK encoding translates a discrete combinatorial optimization problem into a real-valued optimization problem with a larger continuous search space.

The routing of a maximum of $k$ buses for $n$ locations (without the depot) is encoded as $\mathbf{x} = (x_1, x_2, \ldots x_{k+n-1}), x_i \in \mathbb{R}$. Routing $\mathcal{R}$ is from the encoded vector $\mathbf{x}$ created according to Algorithm 1. During the decoding process, $k-1$ largest values of $\mathbf{x}$ are interpreted as route separators. The remaining values are used as random keys and translated into permutation of $n$ locations $\pi$. The values of $\pi$ are split into $k$ routes, each of which starts in the depot and terminates in the depot. Empty routes can be created when the vector $\mathbf{x}$ contains two or more route separators next to each other. The set of non-empty routes defines the routing $\mathcal{R}$.

```
Sort candidate vector: x^s = sort(x);
Use the k − 1 largest values of x as route separators: x_sep = x^s_{n−1};
Create key vector k and vector with route sizes s:
route_size = 0;
for i ∈ {0, . . . , k + n − 1} do
    if x_i > x_sep then
        Append route_size to s;
        route_size = 0;
    else
        Append x_i to k;
        route_size = route_size + 1;
    end
end
Translate key vector k to permutation π;
index = 0;
for i ∈ {0, . . . , k} do
    if s_i > 0 then
        for j ∈ {0, . . . , s_i} do
            Append location π_index to route i;
            index = index + 1;
        end
        Add i to routing R;
    end
end
```

**Algorithm 1.** Decoding of routing $\mathcal{R}$

**Fitness Function.** Fitness function used in this work is based on covered distance, number of routes, and penalty for bus capacity violation.

$$fit(\mathcal{R}) = \frac{\sum_{r \in \mathcal{R}} dist(r)}{|\mathcal{R}|} \tag{19}$$

where $dist(r)$ is distance of route $r$. A penalty is applied (route distance is artificially decreased) when the capacity of the bus is depleted. $|\mathcal{R}|$ represents the number of routes in $\mathcal{R}$.

### 5.1 Experiments

The proposed algorithm was implemented in C++ and used to optimize routings of the ACBSE company. The DE was executed with population size 100, maximum number of vehicles 20, bus capacity 70, number of generations 1000, and parameters $F = 0.9$ and $C = 0.4$ respectively. The parameter values were set on the basis of initial experiments and algorithm tuning. The optimization was repeated 30 times due to the stochastic nature of the algorithm.

The results of the optimization were: average number of routes 5.433, minimum number of routes 5 and maximum number of routes 7. The fitness values in each generation of the 30 independent runs of the algorithm are shown in fig. 1. The results show that the algorithm is stable and able to reduce the estimated number of vehicles significantly. Moreover, the solutions found in the DE algorithm are better (in terms of used fitness function) than a traditional savings algorithm for the VRP [15].

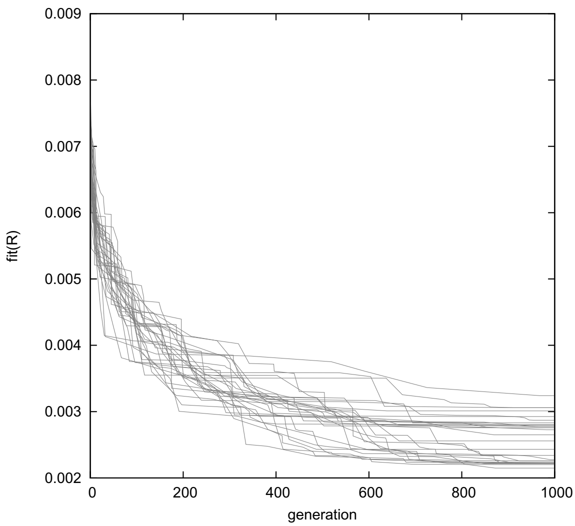


**Fig. 1.** The evolution of fitness function

## 6 Conclusions

This work introduced a new variety of VRP, Stochastic VRP with real Simultaneous Pickup and Delivery and used it to describe the situation of a real-world transportation company operating in Addis Ababa, Ethiopia. A new metaheuristic model based on the Differential Evolution was proposed to optimize vehicle routing in ACBSE network. The DE encoded set of tours as a permutation and automatically minimized the number of buses from an initial upper estimate.

In contrast to a number of previous metaheuristic algorithms, all solutions generated by the proposed algorithm are valid routings through the set of bus stops and computational resources were not wasted on processing of invalid solutions. However, solutions that violate constraints such as bus capacity can be still obtained in course of the evolution.

The routing found by the algorithm was compared to a VRP solution obtained by a traditional (savings) VRP algorithm and it was found better with regard to the used fitness function. The results presented in this study are promising and metaheuristic solvers with various permutation-based representations of candidate solutions [37] will be investigated in the future.

**Acknowledgement.** This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic. For the data used in this work, the researchers would like to also acknowledge ACBSE, Addis Ababa, Ethiopia.

## References

1. Christopher, G.J.: Solutions Methodologies for VRP with Stochastic Demand. Dessirtation, Iowa (2010)
2. Reimann, M., Doerner, K., Hartl, R.: D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research 31(4), 563–591 (2003)
3. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Journal of Management Science, Management Science 6(1), 80–91 (1959)
4. Cordeau, J.F., Laporte, G., Mercier, A.: A unifid tabu search heuristic for vehicle routing problem with time windows. Journal of Operations Research Society 53, 928–936 (2001)
5. Paolo, T., Daniele, V. (eds.): The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia (2002)
6. Dror, M., Trudeau, P.: Savings by split delivery routing. Transportation Science 23, 141–145 (1989)
7. Calvete, H.I., Carmen, G., María, J.O., Belén, S.-V.: Vehicle routing problems with soft time windows: an optimization based approach. Journal of Monografías del Seminario Matemático García de Galdeano 31, 295–304 (2004)
8. Bertsimas, D.: A vehicle routing problem with stochastic demand. Journal of Operations Research 40(3), 554–585 (1991)
9. Secomandi, N.: A rollout policy for the vehicle routing problem with stochastic demands. Operations Research 49, 796–802 (2001)
10. Laporte, G., Louveaux, F., van Hamme, L.: An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. Operations Research 50, 415–423 (2002)

11. Gendreau, M., Laporte, G., Seguin, R.: Stochastic vehicle routing. European Journal of Operational Research 88, 3–12 (1996a)
12. Kenyon, A.S., Morton, D.P.: Stochastic vehicle routing with random travel times. Journal of Transportation Science 37(1), 69–82 (2003)
13. Bertsimas, D.: A vehicle routing problem with stochastic demand. Operations Research 40, 574–585 (1992)
14. Reimann, M.: Analyzing a vehicle routing problem with stochastic demand using ant colony optimization. In: EURO Working Group on Transportation (2005)
15. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581 (1964)
16. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40(4), 455–472 (2006)
17. Linong, C.Y., Wan, R.I., Khairuddin, O., Zirour, M.: Vehicle routing problem: Models and solutions. Journal of Quality Measurement and Anlysis 4(1), 205–218 (2008)
18. Parragh, S., Doerner, K., Hartl, R.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft 58, 81–117 (2008), 10.1007/s11301-008-0036-4
19. Kanthavel, K., Prasad, P.S.S., Vignesh, K.P.: Optimization of vehicle routing problem with simultaneous delivery and pickup using nested particle swarm optimization. European Journal of Scientific Research 73(3), 331–337 (2012)
20. Goksal, F.P., Karaoglan, I., Altiparmak, F.: A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Computers & Industrial Engineering 65(1), 39–53 (2013)
21. Liu, R., Xie, X., Augusto, V., Rodriguez, C.: Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. European Journal of Operational Research (2013)
22. Wang, H.F., Chen, Y.Y.: A genetic algorithm for the simultaneous delivery and pickup problems with time window. Computers & Industrial Engineering 62(1), 84–95 (2012)
23. Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B.: The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. Omega 40(4), 465–477 (2012)
24. Zhang, T., Chaovalitwongse, W.A., Zhang, Y.: Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. Computers & Operations Research 39(10), 2277–2290 (2012)
25. Cruz, R., Silva, T., Souza, M., Coelho, V., Mine, M., Martins, A.: Genvns-ts-cl-pr: A heuristic approach for solving the vehicle routing problem with simultaneous pickup and delivery. Electronic Notes in Discrete Mathematics 39, 217–224 (2012)
26. Fermin, A.T., Roberto, D.G.: Vehicle routing problem with simultaneous pick-up and delivery service. Operational Research Society of India (OPSEARCH) 39(1), 19–34 (2002)
27. Fermin, A.T., Roberto, D.G.: A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Operational Research Society of India (OPSEARCH) 33(1), 595–619 (2006)
28. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)
29. Storn, R., Price, K.: Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical report (1995)
30. Storn, R.: Differential evolution design of an IIR-filter. In: Proceeding of the IEEE Conference on Evolutionary Computation, ICEC, pp. 268–273. IEEE Press (1996)

31. Alba, E., Dorronsoro, B.: Solving the vehicle routing problem by using cellular genetic algorithms. In: Gottlieb, J., Raidl, G.R. (eds.) EvoCOP 2004. LNCS, vol. 3004, pp. 11–20. Springer, Heidelberg (2004)
32. Gendreau, M., Potvin, J.Y., Bräysy, O., Hasle, G., Løkketangen, A.: Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In: Golden, B., Raghavan, S., Wasil, E. (eds.) The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, vol. 43, pp. 143–169. Springer, US (2008)
33. Hou, L., Hou, Z., Zhou, H.: Application of a novel discrete differential evolution algorithm to svrp. In: 2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO), pp. 141–145 (2012)
34. Liu, W., Wang, X., Li, X.: Memetic differential evolution for vehicle routing problem with time windows. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012, Part I. LNCS, vol. 7331, pp. 358–365. Springer, Heidelberg (2012)
35. Xu, H., Wen, J.: Differential evolution algorithm for the optimization of the vehicle routing problem in logistics. In: 2012 Eighth International Conference on Computational Intelligence and Security (CIS), pp. 48–51 (2012)
36. Anbuudayasankar, S., Ganesh, K.: Mixed-integer linear programming for vehicle routing problem with simulatneous delivery and pick-up with maximum routelength. The International Journal of Applied Management and Technology 6(1), 31–52 (2008)
37. Krömer, P., Platoš, J., Snášel, V.: Modeling permutations for genetic algorithms. In: Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR 2009), pp. 100–105. IEEE Computer Society (2009)
38. Snyder, L.V., Daskin, M.S.: A random-key genetic algorithm for the generalized traveling salesman problem. European Journal of Operational Research 174(1), 38–53 (2006)