MULTI-CHAOTIC DIFFERENTIAL EVOLUTION FOR VEHICLE ROUTING PROBLEM WITH PROFITS

Adam Viktorin
Dusan Hrabec
Michal Pluhacek
Tomas Bata University in Zlin, Faculty of Applied Informatics
Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{aviktorin, hrabec, pluhacek}@fai.utb.cz

KEYWORDS
Vehicle Routing Problem with Profits, Differential Evolution, Optimization, Chaos

ABSTRACT

In this paper a new multi-chaotic variant of differential evolution is used to solve a model of vehicle routing problem with profits. The main goal was to achieve exceptional reliability (success rate) and low time demands in comparison with deterministic solvers. The method will be applied in the future on solving real-world transportation network problems.

INTRODUCTION

Various vehicle routing problem (VRP) variants are still actual category of optimization problems in these days and their solving is challenging for many optimization methods (Laporte 1992; Boussier et al 2007; Avci, Topaloglu 2016). However, in this paper we deal with a modification of VRP (or travelling salesman problem (TSP) alternatively) that is the so-called VRP with profits (the VRP with profits on a non-complete graph or selective VRP, alternatively), where not all customers have to be visited, see (Boussier et al, 2007).

We present initial results of evolutionary optimization method called multi-chaotic success-history based adaptive differential evolution that is being developed for future application on real vehicle routing problems (Pavlas et.al, 2015; Stodola et.al., 2014) and transportation network problems (Roupec et. al., 2013). Approach that we present in the paper is considered to be further developed to follow ideas in network design problems (Roupec et. al., 2013) and in waste management (Somplak et. al., 2013). Other modifications of the problem as well as algorithm are also considered (Stodola et.al., 2014).

The differential evolution (DE) (Storn, Price, 1997) is a foundation for some of the best performing evolutionary optimizers. In recent years, various successful applications of DE enhanced with chaotic pseudo-random number generators (PRNGs) were presented (Senkerik et.al., 2013, Liang et.al., 2011).

In the following section the basics of chaotic systems (maps) and their use as PRNGs are presented. The next section describes proposed Multi-chaotic differential evolution algorithm. In the following section the problem is defined following with the experiment setup. The results are presented and discussed in the following section. The paper closes with a conclusion section.

CHAOTIC MAPS

Chaotic maps are systems generated continuously by simple equations from a single initial position. The current position is used for generation of a new position thus creating a sequence which is extremely sensitive to the initial position, which is also known as the “butterfly effect.” Sequences generated by chaotic maps have characteristics which are not common in classical random number generation. Therefore, their application in evolutionary algorithm (EA) can change its behavior and improve the performance.

The multi-chaotic system presented in this paper uses five different chaotic maps – Burgers, Delayed Logistic, Dissipative, Lozi and Tinkerbell.

The process of acquiring *i*-th random integer $rndInt_i$ from chaotic map is depicted in (1).

$$rndInt_i = \text{round}\left(\frac{\text{abs}(X_i)}{\max(\text{abs}(X_{i\in N}))} * (maxRndInt - 1)\right) + 1 \quad (1)$$

Where abs(X_i) is an absolute value of *i*-th X of a chaotic sequence with length of N, max(abs($X_{i\in N}$)) is a maximum of absolute values of X in chaotic sequence and round() is common rounding function. The generated number $rndInt_i$ is from interval [1, *maxRndInt*].

Lozi Chaotic Map

The Lozi map is a simple discrete two-dimensional chaotic map. The map equations are given in (2). The typical parameter values are: a = 1.7 and b = 0.5 with respect to (Sprott, 2013). For these values, the system exhibits typical chaotic behavior and with this parameter setting it is used in the most research papers and other literature sources. The x,y plot of Lozi map with the typical setting is depicted in Figure 1.

$$\begin{aligned} X_{n+1} &= 1 - a\left|X_n\right| + bY_n \\ Y_{n+1} &= X_n \end{aligned} \quad (2)$$

Proceedings 30th European Conference on Modelling and Simulation ©ECMS Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose (Editors)
ISBN: 978-0-9932440-2-5 / ISBN: 978-0-9932440-3-2 (CD)

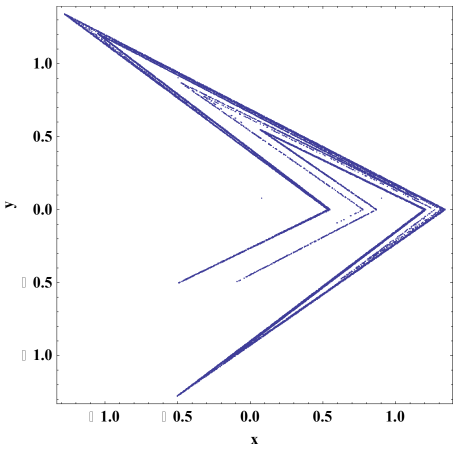

Figure 1: x,y plot of Lozi map

Dissipative Standard Map

The Dissipative standard map is a two-dimensional chaotic map. The parameters used in this work are b = 0.6 and k = 8.8 based on previous experiments [15, 16] and suggestions in literature (Sprott, 2013). The x,y plot of Dissipative standard map is given in Figure 2.The map equations are given in (3).

$$\begin{aligned} X_{n+1} &= X_n + Y_{n+1} (\mathrm{mod}\ 2\pi) \\ Y_{n+1} &= bY_n + k \sin X_n (\mathrm{mod}\ 2\pi) \end{aligned} \tag{3}$$

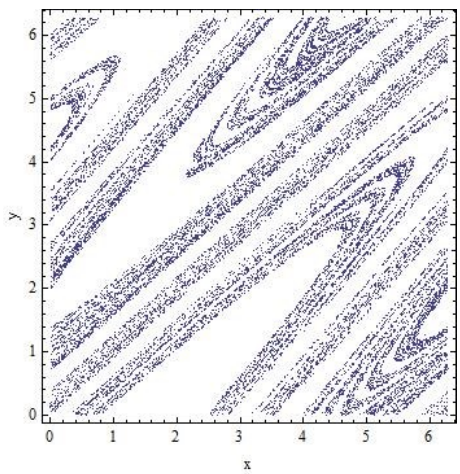

Figure 2: x,y plot of Dissipative standard map

Burgers Chaotic Map

The Burgers map (See Fig. 3) is a discretization of a pair of coupled differential equations The map equations are given in (4) with control parameters a = 0.75 and b = 1.75 as suggested in (Sprott, 2013).

$$\begin{aligned} X_{n+1} &= aX_n - Y_n^2 \\ Y_{n+1} &= bY_n + X_n Y_n \end{aligned} \tag{4}$$

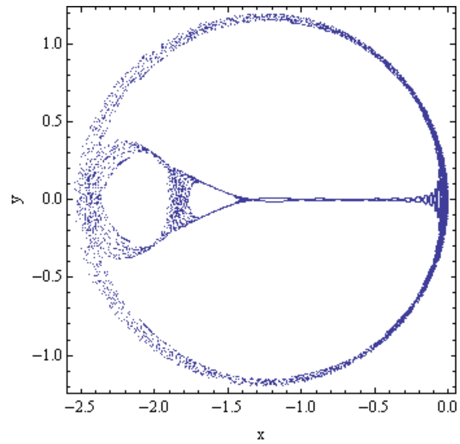

Figure 3: x,y plot of Burgers map

Tinkerbell Map

The Tinkerbell map is a two-dimensional complex discrete-time dynamical system given by (5) with following control parameters: a = 0.9, b = -0.6, c = 2 and d = 0.5 (Sprott, 2013). The x,y plot of the Tinkerbell map is given in Figure 4.

$$\begin{aligned} X_{n+1} &= X_n^2 - Y_n^2 + aX_n + bY_n \\ Y_{n+1} &= 2X_n Y_n + cX_n + dY_n \end{aligned} \tag{5}$$

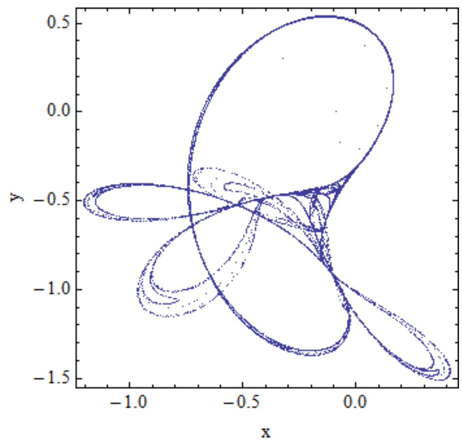

Figure 4: x,y plot of Tinkerbell map

Delayed Logistic Map

The map equations are given in (6). The control parameter A=2.27 (Sprott, 2013). The x,y plot of the Delayed Logistic map is given in Figure 5.

$$\begin{aligned} X_{n+1} &= AX_n(1 - Y_n) \\ Y_{n+1} &= X_n \end{aligned} \tag{6}$$

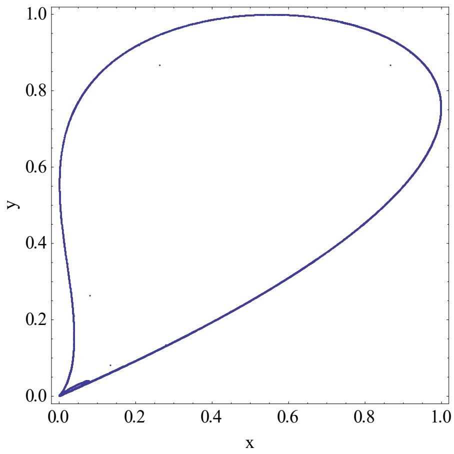

Figure 5: x,y plot of Delayed Logistic map

DIFFERENTIAL EVOLUTION, SUCCESS-HISTORY BASED ADAPTIVE DIFFERENTIAL EVOLUTION AND MULTI-CHAOTIC PARENT SELECTION

DE algorithm (Storn, Price, 1997) has four control parameters – population size *NP*, maximum number of generations G_{max}, crossover rate *CR* and scaling factor *F*. In the canonical form of DE, those four parameters are static and depend on the user setting. Other important features of DE algorithm are mutation strategy and crossover strategy. This work uses "rand/1/bin" mutation strategy (7) and binomial crossover (10). The success-history based adaptive differential evolution (SHADE) algorithm, on the other hand, uses only three control parameters – population size *NP*, maximum number of generations G_{max} and new parameter *H* - size of historical memories. *F* and *CR* parameters are automatically adapted based on the evolutionary process. Values of *F* and *CR* for each individual are generated according to (9) and (11) respectively. Also, the mutation strategy is different than that of canonical DE. Novel mutation strategy used in SHADE is called "current-to-*p*best/1" and it is depicted in (8). The concept of basic operations in DE and SHADE algorithms is shown in following sections. For a detailed description on feature constraint correction, update of historical memories and external archive handling in SHADE see (Tanabe, Fukunaga, 2013).

Initialization

The initial population is generated randomly from objective space and has *NP* individuals in both algorithms. In SHADE algorithm, the external archive *A* is initially empty with a maximum size of *NP* and historical memories M_{CR} and M_F are both set to the size *H* where $M_{CR,i} = M_{F,i} = 0.5$ for ($i = 1, \ldots, H$).

Mutation Strategies and Parent Selection

In canonical forms of both algorithms, parent vectors are selected by classic PRNG with uniform distribution. Mutation strategy "rand/1/bin" uses three random parent vectors with indexes *r1*, *r2* and *r3*, where $r1 = U[1, NP]$, $r2 = U[1, NP]$, $r3 = U[1, NP]$ and $r1 \neq r2 \neq r3$. Mutated vector $\boldsymbol{v}_{i,G}$ is obtained from three different vectors $\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$, $\boldsymbol{x}_{r3}$ from current generation *G* with help of static scaling factor $F_i = F$ as follows:

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1,G} + F_i(\boldsymbol{x}_{r2,G} - \boldsymbol{x}_{r3,G}) \quad (7)$$

Contrarily, SHADEs mutation strategy "current-to-*p*best/1" uses four parent vectors – current *i*-th vector $\boldsymbol{x}_{i,G}$, vector $\boldsymbol{x}_{pbest,G}$ randomly selected from $NP \times p$ ($p = U[p_{min}, 0.2]$, $p_{min} = 2/NP$) best vectors (in terms of objective function value) from *G*, randomly selected vector $\boldsymbol{x}_{r1,G}$ from *G* and randomly selected vector $\boldsymbol{x}_{r2,G}$ from the union of *G* and external archive *A*. Where $\boldsymbol{x}_{i,G} \neq \boldsymbol{x}_{r1,G} \neq \boldsymbol{x}_{r2,G}$. (8)

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{i,G} + F_i(\boldsymbol{x}_{pbest,G} - \boldsymbol{x}_{i,G}) + F_i(\boldsymbol{x}_{r1,G} - \boldsymbol{x}_{r2,G}) \quad (8)$$

The scaling factor F_i is generated from Cauchy distribution with location parameter value of $M_{F,r}$ which is randomly selected value from scale factor historical memory, and scale parameter value of 0.1 (9).

$$F_i = C[M_{F,r}, 0.1] \quad (9)$$

Crossover and Elitism

The trial vector $\boldsymbol{u}_{i,G}$ which is compared with original vector $\boldsymbol{x}_{i,G}$ is completed by crossover operation (5) and this operation is the same for both DE and SHADE algorithms. CR_i value in DE algorithm is again static $CR_i = CR$ whereas with SHADE algorithm its value is generated from a normal distribution with a mean parameter value of $M_{CR,r}$ which is randomly selected value from crossover rate historical memory and with standard deviation value of 0.1 (10).

$$\boldsymbol{u}_{j,i,G} = \begin{cases} \boldsymbol{v}_{j,i,G} & \text{if } U[0,1] \leq CR_i \text{ or } j = j_{rand} \\ \boldsymbol{x}_{j,i,G} & \text{otherwise} \end{cases} \quad (10)$$

Where j_{rand} is randomly selected index of a feature, which has to be updated ($j_{rand} = U[1, D]$), *D* is the dimensionality of the problem. (11)

$$CR_i = N[M_{CR,r}, 0.1] \quad (11)$$

Vector which will be in next generation *G*+1 is selected by elitism. When the objective function value of trial vector $\boldsymbol{u}_{i,G}$ is better than that of the original vector $\boldsymbol{x}_{i,G}$, the trial vector will be selected for the next population and the original will be placed into the external archive *A*. Otherwise, the original will survive and the content of *A* remains unchanged (12).

$$\boldsymbol{x}_{i,G+1} = \begin{cases} \boldsymbol{u}_{i,G} & \text{if } f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & \text{otherwise} \end{cases} \quad (12)$$

Multi-Chaotic Parent Selection

Multi-chaotic framework for parent selection process is based on ranking selection of chaotic map based PRNGs (CPRNGs). A pool of CPRNGs *Cpool* has to be added to the EA and each CPRNG is initialized with the same probability pc_{init} = 1/*Csize*, where *Csize* is the size of *Cpool*. For example, for five CPRNGs *Csize* = 5 and each of them will have the probability of selection pc_{init} = 1/5 = 0.2 = 20%.

For each individual vector $\boldsymbol{x}_{i,G}$ in generation *G*, the chaotic generator $CPRNG_k$ is selected from the *Cpool* according to its probability pc_k, where *k* is the index of selected CPRNG. This selected generator is then used to replace classic PRNG for selection of parent vectors and if the generated trial vector succeeds in elitism, the probabilities are adjusted. There is an upper boundary for the probability of selection pc_{max} = 0.6 = 60%, if the selected CPRNG reached this probability, then no adjustment takes place. Whole process is depicted in (8).

$$\text{if } f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G}) \quad pc_j = \begin{cases} \frac{pc_j + 0.01}{1.01} & \text{if } j = k \\ \frac{pc_j}{1.01} & \text{otherwise} \end{cases} \quad (13)$$

$$\text{otherwise} \quad pc_j = pc_j$$

PROBLEM DEFINITION

The following model presents a modified open VRP with profits (see, e.g., Boussier et al, 2007 for similar problems). In order to make/test our (experimental) computations/algorithm, we consider one vehicle in the model (which corresponds to travelling salesman problem modification of VRP) that does not have to return into the initial node (depot). The basic goal is to deliver cargo from source (production facility) to multiple customers at lowest cost (with maximal profit). In our setting, not every customer must be served (if it is not profitable according to objective function). There are only few links for each node meaning the network is not complete.

The network was designed as is presented in Table 1. In Table 1 each node is given alongside with its demand and neighboring nodes and their distance. The node no. 1 is the source (production facility) therefore its demand is negative. The network is depicted in Figure 6.

The objective function (14) maximizes the total profit, i.e. the revenue minus transportation cost. Equations, or in equations alternatively, (15) - (20) present a set of constraints, where: (15) and (16) guarantee that we can neither come nor leave one node more than once and, moreover (17) guarantee that we have to come and leave every node either once or not at all; (18) sets quantities *q(i)* from the first point of the tour; (19) means that if a node/customer is visited then the customer must be served; (20) is a capacity constraint of the quantity q(i).

Table 1: Experiment setup

No.	Demand	Neighbor No. (distance)
1	-283	2 (18.39), 6 (22.39), 10 (24.48), 15 (27.57), 19 (3.18)
2	11	1 (18.39), 3 (7.24), 7 (2.18), 8 (18.81), 10 (8.18)
3	19	2 (7.24), 19 (8.36)
4	16	6 (8.95), 9 (6.61), 13 (6.18), 20 (4.48)
5	13	9 (12.74), 10 (2.14), 11 (13.76)
6	13	1 (22.39), 4 (8.95), 9 (2.95), 10 (9.67), 14 (4.88), 17 (13.62), 18 (6.9)
7	12	2 (2.18), 9 (18.68)
8	13	2 (18.81), 13 (16.23), 14 (5.22), 18 (9.)
9	19	4 (6.61), 5 (12.74), 6 (2.95), 7 (18.68), 11 (11.14), 14 (1.99)
10	10	1 (24.48), 2 (8.18), 5 (2.14), 6 (9.67), 11 (15.1), 13 (15.28), 17 (17.8), 19 (21.63)
11	15	5 (13.76), 9 (11.14), 10 (15.1), 13 (0.76), 14 (11.59), 16 (4.31)
12	20	13 (11.15), 15 (5.16), 16 (15.07), 20 (3.36)
13	16	4 (6.18), 8 (16.23), 10 (15.28), 11 (0.76), 12 (11.15), 14 (11.01)
14	20	6 (4.88), 8 (5.22), 9 (1.99), 11 (11.59), 13 (11.01), 15 (5.2), 18 (10.25), 20 (9.6)
15	18	1 (27.57), 12 (5.16), 14 (5.2)
16	11	11 (4.31), 12 (15.07)
17	17	6 (13.62), 10 (17.8), 18 (7.47), 19 (7.93)
18	11	6 (6.9), 8 (9.), 14 (10.25), 17 (7.47)
19	12	1 (3.18), 3 (8.36), 10 (21.63), 17 (7.93)
20	17	4 (4.48), 12 (3.36), 14 (9.6)

$$\max \sum_{i,i_1 \in I_C, i \neq i_1} d(i) M_p(i_1, i) p - \sum_{i,i_1 \in I_C} [q(i)c + 1] M_d(i, i_1) M_p(i, i_1) \quad (14)$$

s. t.

$$\sum_{i_1 \in I_C, i_1 \neq i} M_p(i_1, i) \leq 1, \quad \forall i \in I, \quad (15)$$

$$\sum_{i_1 \in I_C, i_1 \neq i} M_p(i, i_1) \leq 1, \quad \forall i \in I, \quad (16)$$

$$\sum_{i_1 \in I_C, i_1 \neq i} M_p(i_1, i) = \sum_{i_1 \in I_C, i_1 \neq i} M_p(i, i_1), \quad \forall i \in I_C, \quad (17)$$

$$q(i) \leq W + [d(i) - W] M_p(1, i), \quad \forall i \in I_C, \quad (18)$$

$$\sum_{i_1 \in I} d(i) M_p(i_1, i) \leq \sum_{i_1 \in I} q(i) M_p(i_1, i), \quad \forall i \in I_C, \quad (19)$$

$$q(i) \leq \sum_{i_1 \in I} M_p(i, i_1) W, \quad \forall i \in I \quad (20)$$

with the decision variables:

q(i) : quantity delivered up to *i*;

$M_p(i; i_1)$: 1 if *i* immediately precedes i_1; 0 otherwise;

the sets of indices:

I : set of all nodes in the network,

I_C : set of customers,

and parameters:

$M_d(i; i_1)$: distance matrix presenting also a cost for using a path,

W : vehicle (e.g. a lorry) capacity or maximal possible production

capacity in a production node, i.e. in $i \in I - I_C$

d(i) : demand in a (customer) node, $i \in I$

p : unit selling price;
c : unit transportation cost per unit of length:

EXPERIMENT SETUP

To generate experimental network (see Figure 6) that approximates real situations, we use a network generator presented in (Pavlas et.al, 2015).
In the experiment three variants of DE were compared, the original DE rand/1/bin, SHADE and proposed MC-SHADE.
The goal for the optimizing algorithm is to find the best possible route and amount of cargo with respect to profit (14).

The maximal number of objective function evaluations Max_FEs was set to 200 000. *Cpool* contained Burgers, Dissipative, Lozi, Tinkerbell and Delayed Logistic CPRNGs. Other parameters were set as follows:
DE:
Dim: 20; *NP*: 100; G_{max}: 2 000; *F*: 0.5; *CR*: 0.8;

SHADE:
Dim: 20; *NP*: 100; G_{max}: 2 000; $H = 10$;

MC-SHADE:
Dim: 20; *NP*: 100; G_{max}: 2 000; $H = 10$;

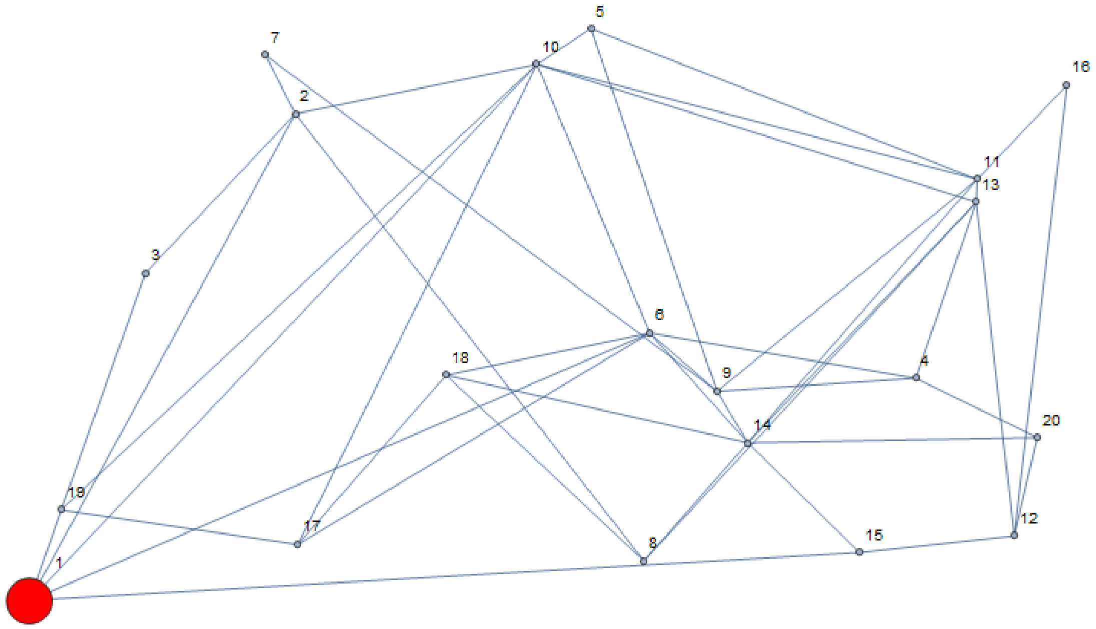

Figure 6: Experiment network setup with highlighted source of cargo

RESULTS AND DISSCUSION

In this section the results of above described experiment are presented and discussed. The statistical overview of the results is presented in Table 2. 50 independent runs were performed for each algorithm. The success rate refers to number of runs in which the best possible solution was found. The best possible solution (confirmed by deterministic commercial solver) is visualized in Figure 7.

Table 2: Results comparison

	DE	**SHADE**	**MC-SHADE**
Time for 50 runs (m:s)	1:42	8:48	9:06
Min:	13176.45	13839.60	13839.60
Max:	13865.72	13865.72	13865.72
Mean:	13588.97	13863.63	**13865.20**
Median:	13623.61	13865.72	13865.72
Std. Dev.:	173.35	7.16	3.69
Success Rate:	8%	92%	**98%**

Best solution details:
Route sequence: 1, 19, 17, 18, 6, 9, 14, 15, 12, 20, 4, 13, 11, 5, 10, 2, 3
Not-visited nodes: 7, 8, 16
Cargo picked-up in node No. 1. : 247

It is clear that the original DE is much faster than SHADE or MC-SHADE however the success rate is unacceptable for practical use. The canonical SHADE achieved satisfactory success rate with higher time demands. With comparable time demands the proposed MC-SHADE achieved exceptional success rate failing to find the optimum only in 1 run from 50. The Wilcoxon signed-rank test between SHADE and MC-SHADE results with alternative hypothesis that mean rank value of SHADE is lower than that of MC-SHADE provided p-value of 0.0745. Based on this result the reliability of the proposed method seems very satisfactory even in comparison to deterministic solvers. The time demand is significantly lower than those of comparable deterministic solvers where a single run for this model can easily take over 1 hour.

The presented evidence strongly supports the feasibility of the proposed multi-chaotic method for solving the modified VRP and its superiority to canonical SHADE.

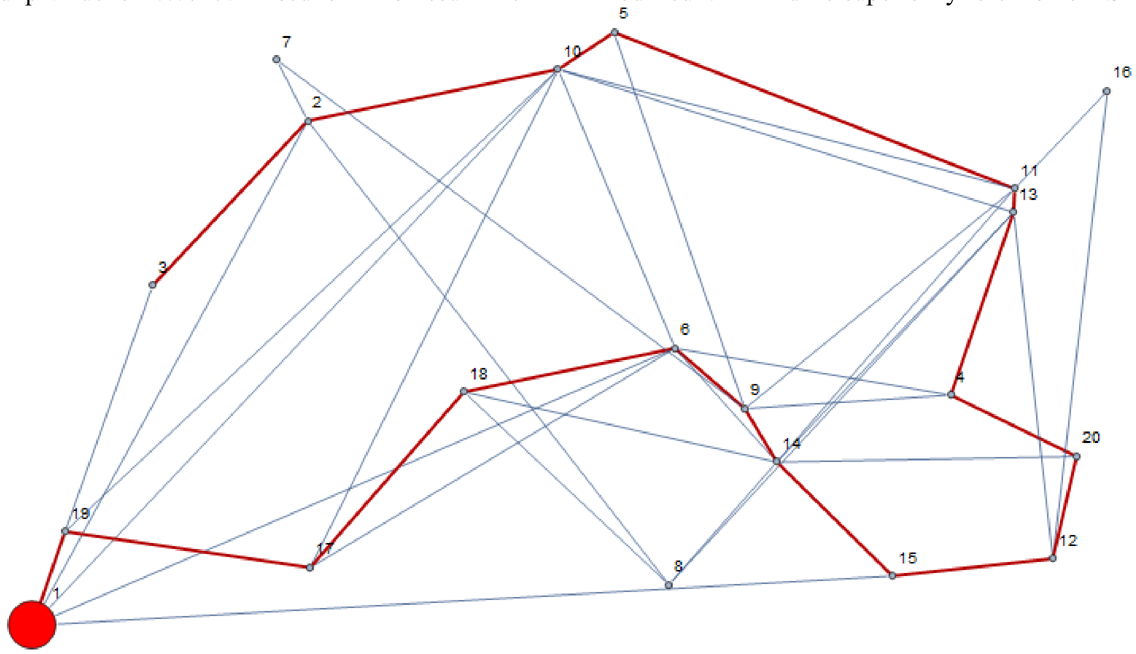

Figure 7: Final solution visualization

CONCLUSION

In this paper a Multi-chaotic Success-history based adaptive differential evolution was applied to a model of open vehicle problem with profits. The performance was compared with canonical version of the algorithm and also with the original differential evolution. The performance of proposed method is superior to both algorithms and the reliability is almost as good as a deterministic solver with significantly lower time demands. This supports the claim that the proposed method can be used as a fast and reliable transportation network problem optimizer. The future research will focus on applying these findings on similar real-world problems.

ACKNOWLEDGEMENT

This work was supported by the Programme EEA and Norway Grants for funding via grant on Institutional cooperation project nr. NF-CZ07-ICP-4-345-2016, by Grant Agency of the Czech Republic GACR P103/15/06700S, , further by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014. Also by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Project no. IGA/CebiaTech/2016/007.

REFERENCES

Avci, M. and Topaloglu, S.: A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. Expert Systems with Applications, Vol. 53, pp. 160-171, 2016

Boussier, S., Feillet, D. and Gendreau, M.: An exact algorithm for team orienteering problems. 4OR, Vol. 5, pp. 211-230, 2007

Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research, Vol. 59 (3), pp. 345-358,1992

Liang W., Zhang L. and Wang M.: The chaos differential evolution optimization algorithm and its application to support vector regression machine. Journal of Software, vol. 6 (7), pp. 1297-1304, 2011

Pavlas, M., Nevrly, V., Popela, P. and Somplak, R.: Heuristic for generation of waste transportation test networks. In: 21st International Conference on Soft Computing, MENDEL 2015, Brno, Czech Republic, 23-25 June, pp. 189-194, 2015

Roupec, J., Popela, P., Hrabec, D., Novotny, J., Olstad, A., and Haugen, K.K.: Hybrid Algorithm for

Network Design Problem with Uncertain Demands. In: Proceedings of the World Congress on Engineering and Computer Science, WCECS 2013, pp. 554-559. San Francisco, USA, 2013

Senkerik, R.; Pluhacek, M.; Oplatkova, Z.K.; Davendra, D.; Zelinka, I.: Investigation on the Differential Evolution driven by selected six chaotic systems in the task of reactor geometry optimization, Evolutionary Computation (CEC), 2013 IEEE Congress on, pp.3087-3094, 20-23 June 2013 doi: 10.1109/CEC.2013.6557946

Somplak, R., Prochazka, V., Pavlas, M., Popela, P.: The logistic model for decision making in waste management. Chemical Engineering Transactions, Vol. 35, pp. 817-822, 2013

Sprott J. C.: Chaos and Time-Series Analysis. Oxford University Press, 2003

Stodola, P., Mazal, J., Podhorec, M., and Litvaj, O.: Using the Ant Colony Optimization Algorithm for the Capacitated Vehicle Routing Problem. 16th International Conference on Mechatronics - Mechatronika (ME), pp.503-510, 2014

Storn R., Price K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, Vol. 11, pp.341–359, 1997

Tanabe, R., Fukunaga, A. Success-history based parameter adaptation for differential evolution. In Evolutionary Computation (CEC), 2013 IEEE Congress on, pp. 71-78, 2013

AUTHOR BIOGRAPHIES

ADAM VIKTORIN was born in the Czech Republic, and went to the Faculty of Applied Informatics at Tomas Bata University in Zlín, where he studied Computer and Communication Systems and obtained his MSc degree in 2015. He is studying his Ph.D. at the same university and the field of his studies are: Artificial intelligence, data mining and evolutionary algorithms. His email address is: aviktorin@fai.utb.cz

DUSAN HRABEC is with Faculty of Applied Informatics at Tomas Bata University in Zlín. He received MSc degree in mathematical engineering at Brno University of Technology in 2011 and, recently, he works on his PhD topic in the field of applied mathematics at the same university. His research and developments efforts centre on operations research, stochastic optimization and its applications. His email address is: hrabec@fai.utb.cz

MICHAL PLUHACEK was born in the Czech Republic, and went to the Faculty of Applied Informatics at Tomas Bata University in Zlín, where he studied Information Technologies and obtained his MSc degree in 2011 and Ph.D. in 2016 with the dissertation topic: Modern method of development and modifications of evolutionary computational techniques. He now works as a researcher at the same university. His email address is: pluhacek@fai.utb.cz