



DEPENDENCE OF EVOLUTIONARY-PROGRAMMING THE TURING MACHINE ON SETTINGS OF DIFFERENTIAL EVOLUTION

KOURIL, L[ukas] & JASEK, R[oman]

Abstract: This paper is focused on examination of dependence of evolutionary-programming the Turing machine (or evolutionary optimization of the rules of the Turing machine's transition function) on custom settings of Differential Evolution. This is one of key issues of the evolutionary optimization as it is considered in relation to the problem of estimating the transition function's rules. It is because the settings influence accuracy and execution time of optimization process and also other aspects of this problem. In the paper, there is brought the analysis of mentioned dependencies.

Key words: differential evolution, turing machine, transition function, evolutionary programming, optimization

1. INTRODUCTION

Evolutionary programming (or evolutionary optimization of the transition function's rules) of the Turing machine is not trivial problem at all. It can be basically imagined as a problem when artificial intelligence programs machine. The former is an evolutionary algorithm (in this case it is Differential Evolution) and the latter is the Turing machine. As it was recently proven (Kouril & Zelinka, 2010a; Kouril & Zelinka, 2010b), this approach to designing the program of the Turing machine by evolutionary optimization is possible and can be even used for programming the Turing machine in order to processing complex tasks. Because the evolutionary optimization especially utilizes Differential Evolution, its settings considerably interfere with reachable results. The described research proposes the analysis which answers to the question "How custom settings of Differential Evolution influence the evolutionary optimization of the Turing machine's rules".

2. METHODS

Because the analysis is aimed to settings of the Differential Evolution, it is essential to briefly deal with evolutionary algorithm used and outline importance of single parameters of the Differential Evolution. It is also necessary to sketch principles of analysis out as well.

2.1 Essentials of Differential Evolution

The Differential Evolution (Lampinen & Zelinka, 1999; Zelinka et al., 2008; Zelinka, 2002) represents one of evolutionary algorithms. These algorithms are based on elements of artificial intelligence which enables to flexible approach to solving problems. The Differential Evolution (DE) is inspired with principles of evolution in accordance with Darwin's theory. It means that DE utilizes partial solutions of the problem which solves and considers them as individuals. These partial solutions are subsequently optimized as if the life-being individuals (organisms) passed nature evolution. In the nature, each individual has own "quality value" which expresses the ability to survive. This value is influenced by properties (e.g. strength, quickness etc.) of the individual. The individuals containing higher "quality value" are capable to

survive thus it can be considered as a form of optimization. In the case of DE, the "quality value" is termed as cost value which is calculated by the cost function considering given problem and related specified conditions. The properties of individuals which influence calculation of the cost value have a form of parameters of partial solutions related to the problem processed by DE. The appearance of individuals is given by specimen that provides information on number of parameters and their boundaries. In accordance with specimen the individuals are encoded.

When DE is used, there is necessary to specify following parameters which represent settings of the algorithm:

- Number of population (NP)
- Mutation constant (F)
- Cross-over value (CR)
- Number of generations (G)

Above mentioned parameters directly influence optimization process by DE. How the DE processes individuals it can be easily understood. At first, there is set initial generation. This generation involves population of individuals which have pre-generated parameters in accordance with the specimen. The number of the individuals matches NP parameter. These individuals are subsequently one by one evaluated. The current one and other three randomly-selected individuals are processed in a following way. Two individuals which were randomly selected are subtracted (parameters of individuals). The result is the differential vector. Then the differential vector is multiplied by mutation constant F . This is a form of mutation which proceeds in the nature alike. The mutation produces the weighted differential vector. The third randomly-selected individual is added to the weighted differential vector thus the noise vector (3) is originated. Now the test vector (1) is created by cross-over of the current individual and the noise vector. The operation of cross-over is influenced by cross-over value CR . Finally, the current individual and the test vector are evaluated (2) by cost function. The individual (or vector) which contains "better" cost value is advanced to the next generation where the optimization process repeats until reaching stated number of generations G . In that way there are processed all individuals within current generation and the population of the next generation is created. All of that what was explained can be mathematically expressed (Zelinka et al., 2008) as:

$$x_{i,j}^{test} = \begin{cases} [x_{r_3,j}^G + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)]^* \\ x_{r_1,j}^G \end{cases} \quad (1)$$

*if $rand_j(0,1) \leq CR \vee j = k$

where

$i = 1, \dots, NP, j = 1, \dots, D$ (dimension of individual)

$k \in \{1, \dots, NP\}$, k – random index

$r_1, r_2, r_3 \in \{1, \dots, NP\}$ – random selection of individuals

$r_1 \neq r_2 \neq r_3 \neq i$

$R \in \langle 0,1 \rangle, F \in \langle 0,1 \rangle$

$$X_i^{G+1} = \begin{cases} x^{test} & \text{if } f_{cost}(x^{test}) \leq f_{cost}(X_i^G) \\ X_i^G & \end{cases} \quad (2)$$

There are several variants of DE which utilize different forms of equation of the noise vector. Within described analysis the DE states the noise vector as:

$$v_{noise} = x_{r_3,j}^G + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G) \quad (3)$$

2.2 Backgrounds of analysis

The analysis of the dependence of the evolutionary-programming the Turing machine on custom settings of the DE is based on related research described in (Kouril & Zelinka, 2010a) where the possibilities of evolutionary-estimated programming the Turing machine by DE are shown and proven by processing several examples. One of those examples – the problem of unary addition – was selected as a suitable sample for processing within the analysis.

The unary addition which was used satisfies the same conditions as in (Kouril & Zelinka, 2010a). It means that the problem used in analysis consists of addition of two numbers (3 + 5) realized in unary number system (111 + 1111). Because that problem is processed by the Turing machine, the example is encoded to the form of data tape as can be seen in (Kouril & Zelinka, 2010a). It is requested to estimate a program of the Turing machine which solves that problem by employing DE as it is explained in (Kouril & Zelinka, 2010a; Kouril & Zelinka, 2010b).

The settings of the DE are also inspired by (Kouril & Zelinka, 2010a). These parameters were stated as initial for analysis and subsequently changed in accordance with recommendations in (Zelinka et al., 2008). The summary of the initial and recommended values is following:

| Parameter | Period | Values | Initial |
|-----------|-----------------------------|------------|---------|
| <i>NP</i> | $\langle 10D, 100D \rangle$ | 200 – 2000 | 100 |
| <i>F</i> | $\langle 0,2 \rangle$ | 0 – 2 | 0.9 |
| <i>CR</i> | $\langle 0,1 \rangle$ | 0 – 1 | 0.2 |
| <i>G</i> | customizable | 100 – 1000 | 100 |

Tab. 1. Summary of settings of DE

3. RESULTS

Within analysis, there were mainly observed influences of variations in settings of DE on estimating the rules and the execution time of optimization process. The estimation rates are expressed as percentage values which mean ratios of successfully processed data tape symbols by the Turing machine initialized by estimated rules. The results in the form of graphs are following (each test of analysis was several times repeated thus results are average values):

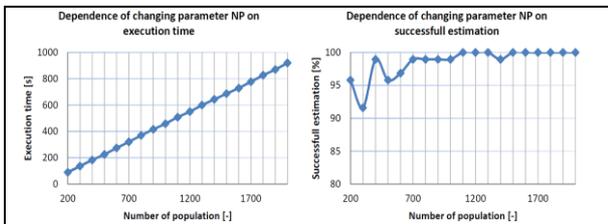


Fig. 1. Dependence of *NP* on evolutionary process

As can be seen in Fig 1 and Fig 2, the most important parameters which significantly influence the successful optimization are *NP* and *G*. These parameters are also most impacted to increasing of execution time contrary to *F* and *CR*.

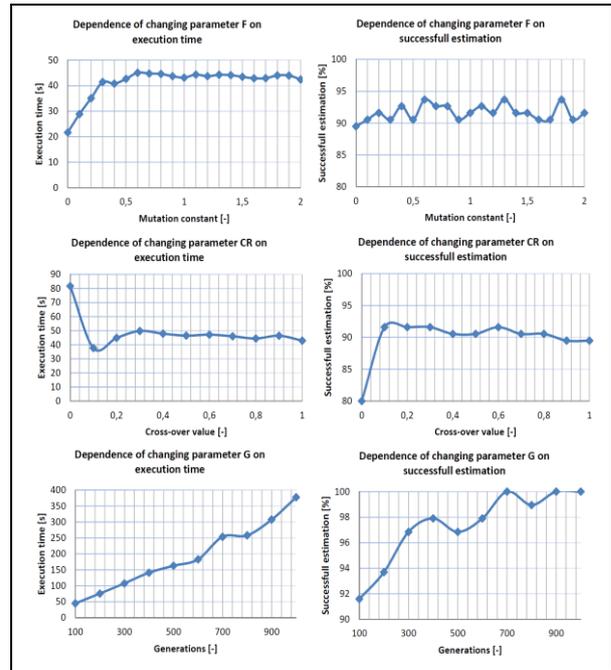


Fig. 2. Dependence of *F*, *CR* and *G* on evolutionary process

4. CONCLUSION

The settings of the DE are important for evolutionary-estimated programming the Turing machine as it was proven by provided analysis. When optimizing the rules by approaches described in (Kouril & Zelinka, 2010a; Kouril & Zelinka, 2010b) the parameters of DE which considerably influence the estimation of the rules are *NP* and *G* above all. The provided analysis can help to correctly set the values of DE parameters. This should ensure the optimal progress of estimation process.

5. ACKNOWLEDGEMENTS

This paper is a part of research No. IGA/42/FAI/10/D supported by the Internal Grant Agency of Tomas Bata University and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

6. REFERENCES

Kouril, L. & Zelinka, I. (2010a). Evolutionary-Estimated Programming the Turing Machine by Differential Evolution, *Proceedings of 16th International Conference on Soft Computing MENDEL 2010*, Brno, Czech Republic, ISBN 978-80-214-4120-0, pp. 41-48

Kouril, L. & Zelinka, I. (2010b). Evolutionary Synthesis of Rules for Programming the Turing Machine. *Odborný vedecký časopis Trilobit*, No. 2, 2010, ISSN 1804-1795, <http://trilobit.fai.utb.cz/evolutionary-synthesis-of-rules-for-programming-a-turing-machine>

Lampinen, J. & Zelinka, I. (1999). Mechanical Engineering Design Optimization by Differential Evolution, In: *New Ideas of Optimization*, 1st London, McGraw-Hill, ISBN 007-709506-5

Zelinka, I. et al. (2008). *Evolucni vypocetni techniky – principy a aplikace*, BEN – technicka literatura, ISBN 80-7300-218-3, Praha

Zelinka, I (2002). *Umela inteligencia v problemech globalni optimalizace*, BEN – technicka literature, ISBN 80-7300-069-5, Praha