Kota László; Jármai Károly: Adaptive methods in the optimization of large scale technical inspection and maintenance systems
XXVII. MicroCAD International Scientific Conference J: Material Flow Systems. Logistical Information Technology and Technical Language.
Konferencia helye, ideje: Miskolc, Magyarország, 2013.03.21-2013.03.22. 2013. Paper J17.

ADAPTIVE METHODS IN THE OPTIMIZATION OF LARGE SCALE TECHNICAL INSPECTION AND MAINTENANCE SYSTEMS

László Kota[1], Károly Jármai[2]
[1]PhD, research assistant, [2]DSc, Dr. *habil* university professor

University of Miskolc, Department of Materials Handling and Logistics

ABSTRACT

The aim of this research is developing adaptive optimization algorithms and makes supplementary adaptive methods to existing algorithms. In this article we will introduce a method which adaptively controls the mutation parameters of an evolutionary programming algorithm. The developed method we present here is able to work in the area of genetic methods also.

INTRODUCTION

The network like technical inspection and maintenance systems are getting more common nowadays. They can cover a city, a region, a country or even a continent or worldwide. Their task is to do systematic inspections, supervisions according to the actual regulations in the defined nodes of the network. And the other task is not just doing inspections but do maintenance and/or refurbishment on the equipment in the network. Supporting this tasks there are several part and equipment warehouse scattered all over the network (Figure 1).

EVOLUTIONARY PROGRAMMING

The evolutionary programming handles a population which consist of the possible solutions of the problem. In the evolutionary programming there are no restrictions of the representation of the problem. The solutions were stores according to the problem and there aren't necessary to code the solution to bitvector or numeric vector like the genetic algorithm [1]. But here like at the GA the algorithm starts with a random generated population. At the first step the algorithm copies all the individuals and then the copy goes thru the mutation process. The mutation could be various degree but that's common that the "lesser" mutations have bigger probability like the "greater" mutations. After the calculation of the fitness values of the mutated individuals there is only one decision left. Which individuals will survive? The selection of the surviving individuals is made by a tournament. The simplest means of the tournament selection when we choose two random individuals from each of the populations and the better (with better fitness value) goes into the new population. This process repeats until the new population filled. The evolutionary programming in most of the cases doesn't use crossing, however it is meaningless in some cases. So in biological means it is like the progress of different races because there is no crossing between the races. The evolutionary programming used to solve difficult multi constrained problems with problem specified data structures and functions like the other modern optimisation algorithms as PSO [4] and harmony search [5, 6]. We aren't dealing with the optimization of the technical supervision and maintenance systems due to the article limitations but the [2, 3] is cover the theme in details.

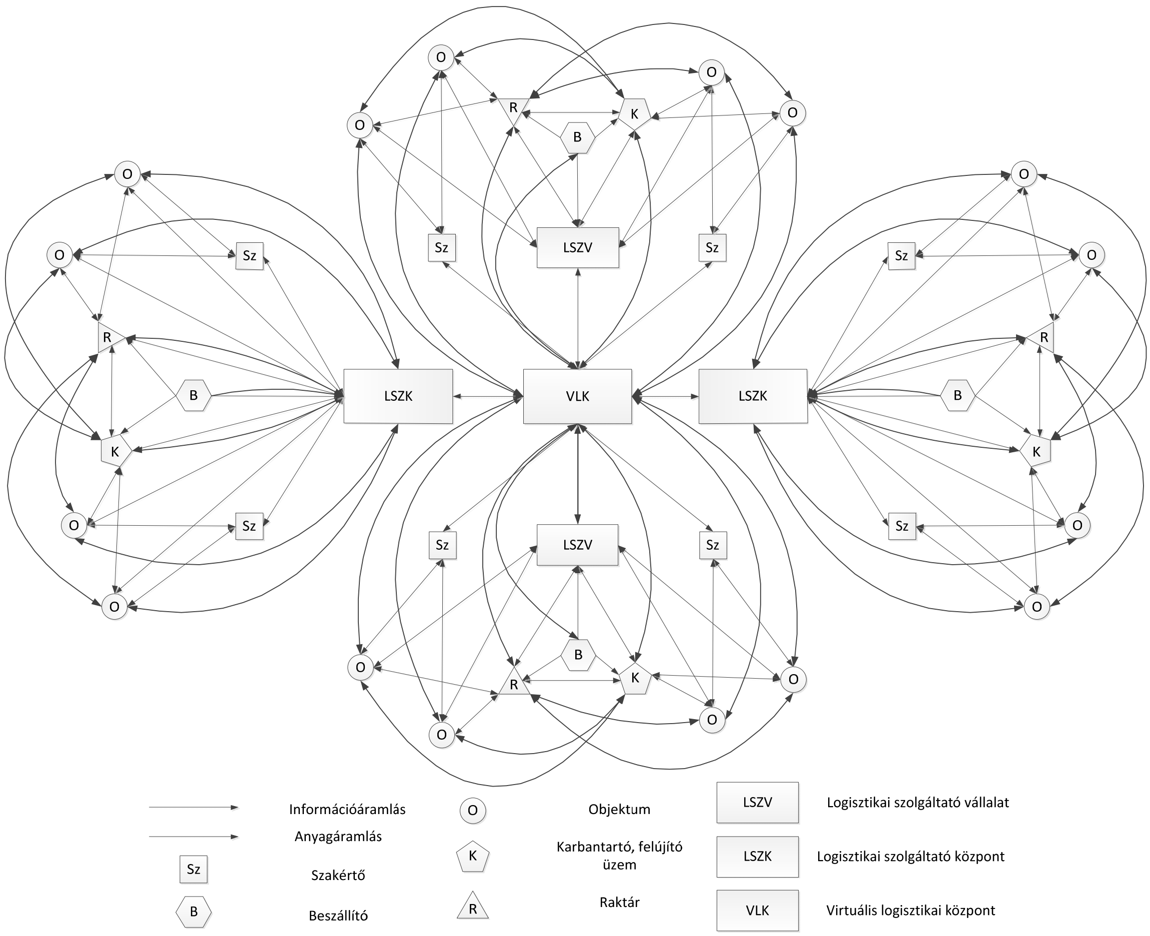

Figure 1.
General structure of the network like technical inspection and maintenance system with regional decenters

ADAPTIVE OPTIMIZATION

The evolution programming method we developed using the following parameters to choose the mutation operator to run:

- Probability of the mutation operators
  - Local mutation
  - Global mutation
- Probability of the local operators
  - Probability of the gene swap
  - Probability of the gene insertion
  - Probability of the gene sequence swap
- Probability of the global operators
  - Probability of the gene swap
  - Probability of the gene insertion
  - Probability of the gene sequence swap

During the adaptive optimization the process controls the parameters by itself according to if target function gets better to the next iteration or not. The notation of the algorithm variants is the following:

- R500: (Randomization) in every 500 iterations random parameters will be generated to all the individuals.
- WB100: (Write back) in every 100 cycle the parameters of the best fitness individual will be written back to all the individuals.
- Re: (Reinforced) if the given mutation is improving the fitness then the probability range of the given mutation will be increased at the expense of the other parameters.
- Ne: (Negative feedback) if the target function isn't improved in the given iteration then the mutation probability will decrease while the others will increase.

We have examined the following methods (mind the notation) apart from the normal non-adaptive algorithm:

- Adaptive WB100R500; in every 100 cycle the parameters of the best fitness individual will be written back to all the individuals. In every 500 iterations random parameters will be generated all the individuals.
- Adaptive Re; if the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased.
- Adaptive ReR500; If the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased. In every 500 iterations random parameters will be generated all the individuals.
- Adaptive ReR500Wb100; If the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased. In every 500 iterations random parameters will be generated all the individuals. In every 100 cycle the parameters of the best fitness individual will be written back to all the individuals.
- Adaptive Ne; if the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase.
- Adaptive NeR500; if the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase. In every 500 iterations random parameters will be generated all the individuals.
- Adaptive NeR500WB100; if the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase. In every 500 iterations random parameters will be generated all the individuals. In every 100 cycle the parameters of the best fitness individual will be written back to all the individuals.
- Adaptive combined: (Com): ReNe; If the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased. If the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase.

- Adaptive combined: (ComR500): ReNeR500; If the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased. If the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase. In every 500 iterations random parameters will be generated all the individuals.
- Adaptive combined: (ComR500WB100): ReNeR500WB100; If the given mutation improves the fitness of the individual then the mutation probability range will increase while the probability range of the other mutations will be decreased. If the target function didn't improve in the given iteration then the mutation probability will decrease while the others will increase. In every 500 iterations random parameters will be generated all the individuals. In every 100 cycle the parameters of the best fitness individual will be written back to all the individuals.

RESULTS

The examination was made on the test examples mentioned in the [2]. The results are shown on the Figure 2, 3, 4.
First we examined a simple problem with 3 experts and 48 nodes and single inspection on each node. The best optimization method was the ReR500 and the ReR500Wb100 algorithm variants gave the best value.

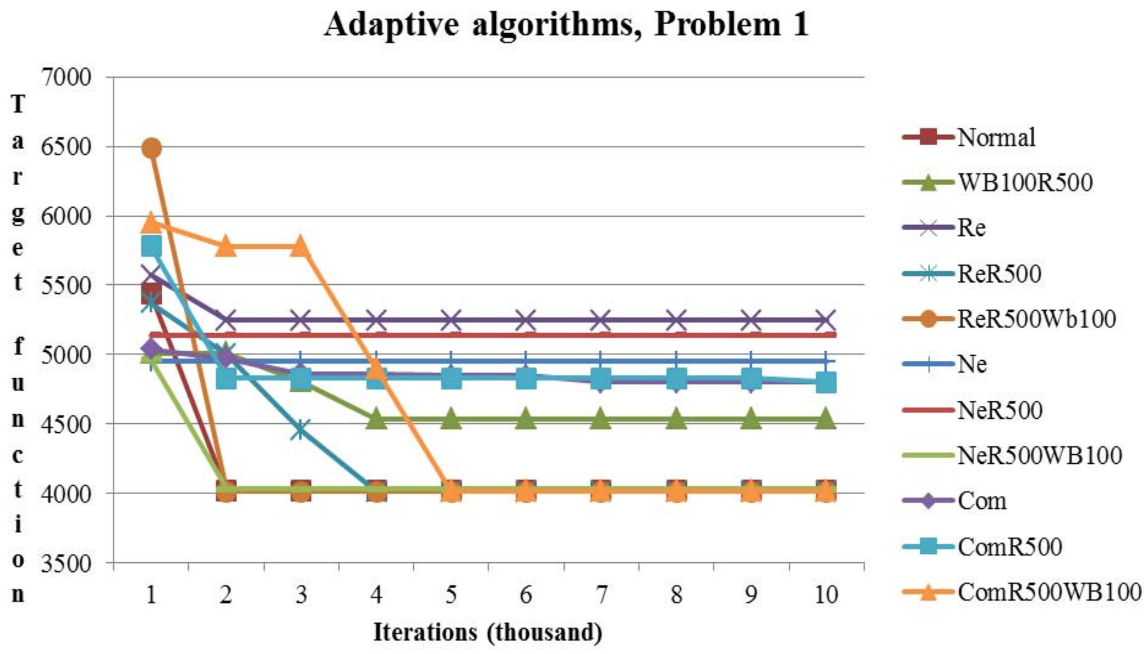

Figure 2.
Comparison of the developed adaptive methods, test problem 1

The second test problem was a three experts 48 node problem but with multiple inspections on each object. The best method was the NeR500WB100 algorithm variant which gives almost 31.5 percent of the solution of the normal, non-adaptive algorithm in the same iteration range (Figure 3.). Even the second one, the ReR500 optimization variant gives a bit over 50 percent of the solution of the normal algorithm. As the results shows, the optimization was a minimization task so the lesser values are the better.

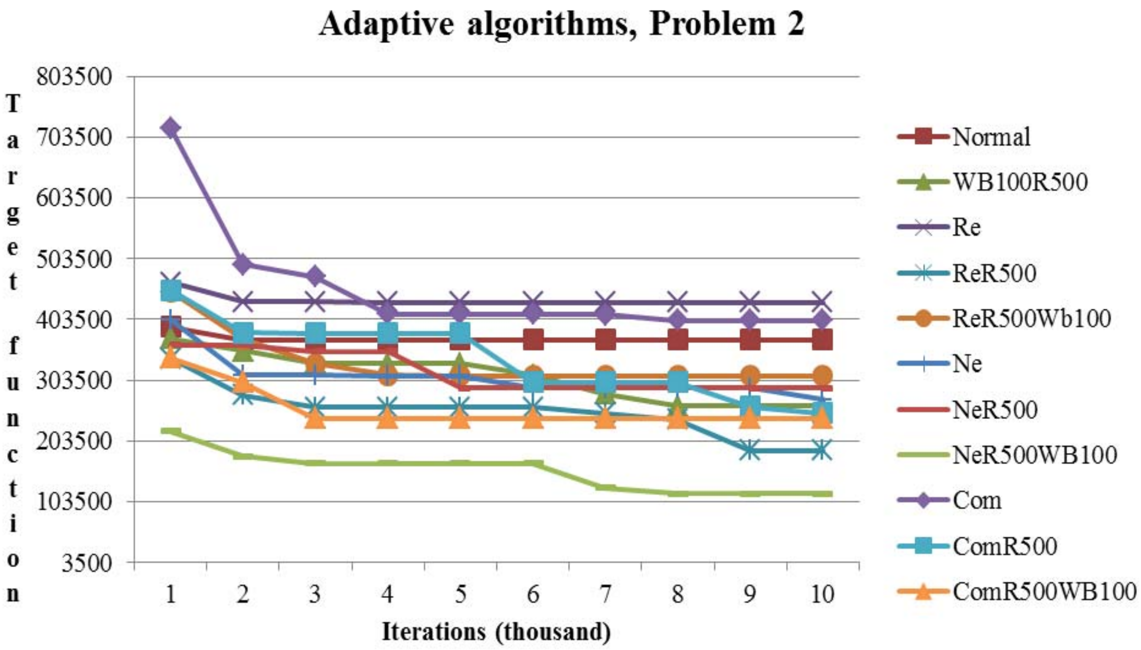

Figure 3. Comparison of the developed adaptive methods, test problem 2

The third test problem was a big problem. It includes three experts and one thousand nodes and random 2-4 inspections on each node. The best method was the ReR500 algorithm and the second one was the ReR500Wb100 (Figure 4.). Here the advantage of the adaptive algorithms isn’t that big like the previous problems, it is around 1 percent. Because of the iteration count wasn’t so large comparing to the iteration count needed to solve the problem.

SUMMARY AND CONCLUSIONS

Examining the developed adaptive algorithm in summary they give better results than the normal non-adaptive algorithm in all the cases. The normal algorithm was only once in the best four. But the efficiency of the adaptive algorithms is clearly visible on difficult problems. It happened at one test example that the improvement of the target function exceeded 30%. But the performance of the adaptive algorithm isn’t uniform. On different test examples different algorithm variant gave the best result so before solving the problem test runs are necessary. So our future goals are the refinement and test of the algorithms and develop new adaptive methods.

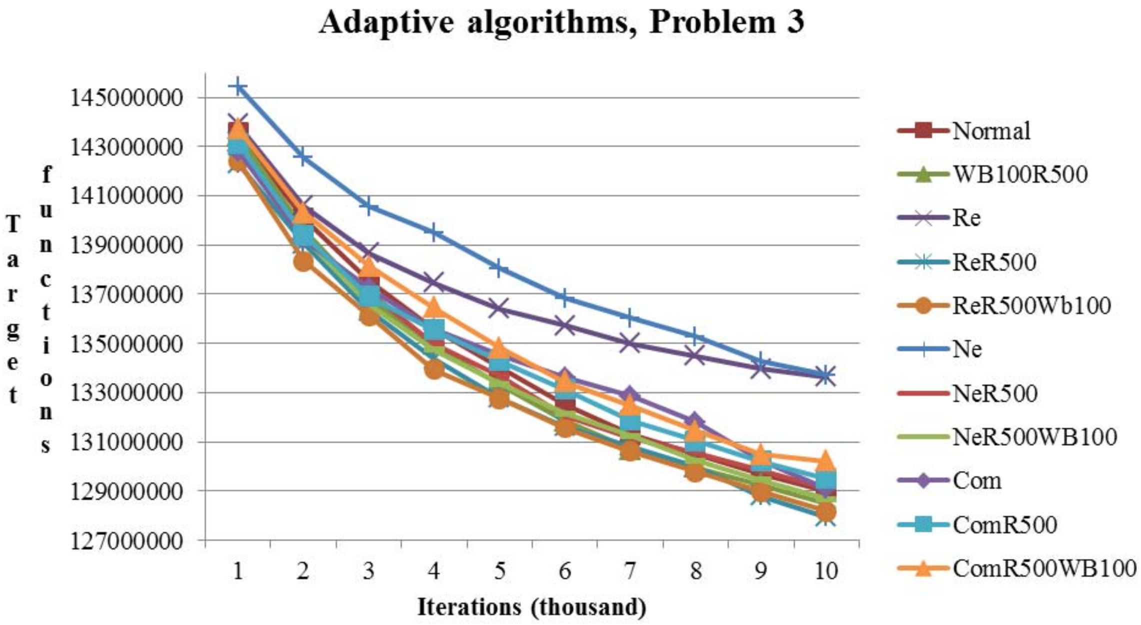

Figure 4. Comparison of the developed adaptive methods, test problem 3

ACKNOWLEDGEMENTS

The research was supported by the TÁMOP 4.2.1.B-10/2/KONV-2010-0001 entitled "Increasing the quality of higher education through the development of research - development and innovation program at the University of Miskolc supported by the European Union, co-financed by the European Social Fund."

LITERATURE

[1.] Barrie M. Baker, M.A. Ayechew: **A genetic algorithm for the vehicle routing problem**, Computers & Operations Research 30 (5), pp. 787–800. (2003), ISSN: 0305-0548, doi:10.1016/S0305-0548(02)00051-5

[2.] Kota L.: **Optimization of the assignment tasks in network like technical inspection and maintenance systems based on logistic costs** (in Hungarian), 121 p., Miskolc 2012., PhD thesis

[3.] Kota L., Jármai K.: **Optimization of technical inspection and maintenance systems** (in Hungarian), GÉP 62:(7-8) pp. 75-78. (2011), ISSN 0016-8572

[4.] J. Farkas, K. Jármai: **Design and Optimization of Metal Structures, Horwood Publishing Limited 2008**, ISBN 978-1-904275-29-9, 300 p

[5.] T. Bányai: **Optimisation of multi level supply chain of automatised production systems with harmony search algorithm**, Proceedings of the II. European conference on logistics, pp.: 65-71., ISBN 978-83-61118-67-1

[6.] T. Bányai: **Optimisation of a Multi-Product Green Supply Chain Model with Harmony Search,** DAAAM International Scientific Book 2011, pp. 15-30., ISBN 978-3-901509-84-1