



Migration Model of Adaptive Differential Evolution Applied to Real-World Problems

Petr Bujok^(✉)

Department of Informatics and Computers, University of Ostrava,
30. dubna 22, 70103 Ostrava, Czech Republic

petr.bujok@osu.cz

<http://prf.osu.eu/kip/>

Abstract. Ten variants of migration model are compared with six adaptive differential evolution (DE) algorithms on real-world problems. Two parameters of migration model are studied experimentally. The results of experiments demonstrate the superiority of the migration models in first stages of the search process. A success of adaptive DE algorithms employed by migration model is systematically influenced by the studied parameters. The most efficient algorithm in the comparison is proposed migration model P15x50. The worst performing algorithm is adaptive DE.

Keywords: Differential evolution · Migration model
Migration frequency · Sub-population size · Experimental study
Real-world problems

1 Introduction

In many areas of research and industry arises necessity to solve various problems. Each problem to be solved is represented by an objective function. These functions should be very simple, but in prevalent cases, the computational costs to solve more complex problems are non-negligibly high. An essential feature of the real problems is their dimensionality that substantially influences the total computational costs of the solution process.

In last decades, a favorite kind of optimization techniques called evolutionary algorithms (EAs) is rapidly developed. An essence of this fact is that EAs are very effective even in the solution of complex problems with higher dimensionality. Popularity of EAs is given also by its simplicity because the main idea is to use randomness to reduce time demands and keep the quality of the provided solution.

One of the most widely studied and applied representatives of EAs is differential evolution (DE). Differential evolution is very popular optimization technique used to search the solution of problems in many areas of research and industry. It was introduced by Storn and Price in 1995, and its popularity could be also measured by more than 7500 citations of the original paper [14].

Although DE algorithm is very useful optimizer there are even real complex problems which DE is not able to solve. In last decades, many researchers of optimization tasks developed adaptation mechanisms of DE control parameters to increase the possibility of finding the global solution of the tasks. Beside the adaptive DE variants, many other principles increasing algorithm efficiency are included in original DE. A comprehensive review of state-of-the-art in a case of DE algorithms are in [7, 8].

The main motivation for this experimental study is applied the existing parallel migration model [4] to real-world problems. These problems are defined by various dimensionality, therefore it is necessary to study two main control parameters of migration model to find out the proper settings. Although the experiments of parallel migration model are performed on single CPU computer, the efficiency measured by the number of the objective function evaluations should detect some good and poor migration settings compare to the original adaptive DE variants. Given results and conclusions of this experiment could provide potential ways how to set parameters of migration model and solve complex problems in physical parallel computational grids.

There are several existing works focused on parallel DE algorithm and real problems. In a first, a multi-objective maximization problem of an amount of ethylene and propylene in a petrochemical industry process is solved by parallel DE in [16]. Several settings A short-term hydro scheduling problem of power plants generator is solved by the parallel model of self-adaptive DE [9]. Authors of [12] study many parameters of migration model of DE. The state-of-the-art of parallel evolutionary algorithms containing deep analysis of various parallel models, several parallel programming frameworks and hardware for distributed computing is summed in [10].

In this paper, a migration model of adaptive DE variants is applied to real-world optimization problems CEC 2011. The main goal is to show how several various settings of migration of individuals influence the efficiency of the proposed model. The performance of the migration model is also compared with the original adaptive DE variants. The remaining part of the paper is organized as follows. A brief review of adaptive state-of-the-art DE variants used in this paper is explained in Sect. 2. Details of the migration model are described in Sect. 3. A test suite of real-world problems CEC 2011 with the experiment settings is introduced in Sect. 4. Results of the experimental comparison of the migration model are presented in Sects. 5 and some remarks are provided in Sect. 6.

2 Adaptive Variants of Differential Evolution

Although DE is efficient optimizer there exist many various adaptive variants of DE which are widely used to solve optimization problems. In this experimental study, four state-of-the-art adaptive DE algorithms [1, 11, 13, 18] have been compared experimentally with DE algorithm using composite trial vector generation strategies and control parameters (CoDE) [17] and with a variant of competitive DE [15]. The main purpose to select this sixth of adaptive DE is to tie up to

the previous successful experimental study [4] and show that simple idea should provide significant increasing of the performance. Necessary to note that more sophisticated adaptive DE variants arisen latter should be also implemented in presented migration model. A very brief familiarization of used adaptive DE follows. More details are provided in the original references.

Self-adaptive *jDE* algorithm proposed by Brest et al. [1] uses the DE/rand-1/bin strategy with an evolutionary self-adaptation of F and CR . Differential Evolution with Strategy adaptation (*SaDE*) [13] uses four strategies which are preferred according to its success rate in the previous LP generations. *JADE* variant of adaptive differential evolution [18] extends the original DE concept with three different improvements – current-to-pbest mutation, a new adaptive control of parameters F and CR , and archive. Next adaptive DE variant using Ensemble of Parameter values and mutation Strategies (*EPSDE*), has mutation strategies and the values of control parameters F and CR stored in pools and was proposed in [11]. In competitive DE H strategies (in this study *b6e6rl* uses $H = 12$ strategies) are used with their control-parameter values held in the pool and each strategy is preferred according to its success rate in the preceding step. DE algorithm with three well-studied composite trial vector generation strategies simultaneously applied on each point in population and control parameters (*CoDE*) has been recently presented [17].

3 Migration Model of DE

There are several scenarios to distribute operations of DE algorithm to achieve better results. A migration model used in this experiment is widely-used because of its simplicity and possibility of exchange information between parallel processes [2, 3, 5]. Migration model is controlled by several parameters which setting is provided as follows. A comprehensive state-of-the-art of distributed evolutionary algorithms is available in [10].

In general migration model there are k islands and each contains one sub-population, P_j , $j = 1, 2, \dots, k$. Each island is linked only to a special island called *mainland* and individuals can migrate only between the linked islands. This topology is called *star*. A pseudocode of experimentally compared parallel migration model is shown in Algorithm 1. Sub-populations has the same size, N_p is an input parameter. Each sub-population is developed independently by one of the six adaptive DE algorithm described above until the moment to migration is reached. Necessary to note that in one studied model only *b6e6rl* variant is performed in all sub-populations and also mainland population (the name of this variants is in an index). The migration from the islands to the mainland occurs after performing a given number of generations nde (studied input parameter of the algorithm). In the migration model used here, the individual with the least function value of the i th sub-population ($\mathbf{x}_{best,i}$) replaces the i th individual (\mathbf{x}_i^m) of the mainland population, and N_{rnd} other randomly chosen points of the sub-population (except $\mathbf{x}_{best,i}$) overwrite N_{rnd} individuals of mainland population on places corresponding to k th sub-population, N_{rnd} is also an input parameter.

In this study, this parameter is not deeply studied, and it is set to $N_{\text{rnd}} = 4$. Thus, $N_{\text{rnd}} + 1$ individuals from each island are copied to central mainland sub-population. It is obvious the size of the mainland sub-population N_m should be set up to $N_m \geq k \times (N_{\text{rnd}} + 1)$. If $N_m = k \times (N_{\text{rnd}} + 1)$, the mainland sub-population is refreshed completely in each epoch, and the elitism of the migration model is ensured. In order to satisfy this condition, the input parameter N_{rnd} was set up to $N_{\text{rnd}} = 4$ and $N_m = 6 \times 5 = 30$ in all the experiments.

After finishing the migration of the selected individuals from the islands to the mainland, the search process continues applying a DE variant on the mainland until the stopping condition for the current epoch (1) is reached. In the proposed migration model, competitive *b6e6rl* as the most reliable in preliminary experiments was chosen for the mainland sub-population. Only in the one case of migration model setting, the fastest JADE variant was performed on mainland sub-population (the name of JADE algorithm is in the index of the model). The stopping condition for the mainland, and the current epoch was formed as follows:

$$f_{\max} - f_{\min} < 1 \times 10^{-6} \quad \text{OR} \quad FES_m > 10^{(\text{epoch}-1)} \times 2 \times nde \times N_m, \quad (1)$$

where f_{\max} and f_{\min} are the worst and the best objective function values of the mainland sub-population, respectively, and FES_m is the number of function evaluations in the mainland population during this epoch. Notice that in early epochs the evolution on the mainland tends to stop due to the given limit of allowed function evaluations (after $2 \times nde$ generations in the first epoch) while in late epochs due to the small difference of the function values in the mainland population. The whole search process of the algorithm is stopped after the pre-defined number of objective function evaluations given by used benchmark set of CEC 2011.

Algorithm 1. Migration Model of Adaptive Differential Evolution

```

initialize mainland population, and sub-populations  $P_i, i = 1, 2, \dots, k$ 
epoch = 1
while stopping condition not reached do
  for  $i = 1, 2, \dots, k$  do
    perform  $nde$  generations of  $i$ th sub-population by  $i$ th adaptive DE
    migrate the best point and  $N_{\text{rnd}}$  points randomly chosen from  $P_i$  to mainland
  end for
  while stopping condition (1) not reached do
    develop mainland sub-population by a adaptive DE variant
  end while
  for  $i = 1, 2, \dots, k$  do
    migrate  $1 + N_{\text{rnd}}$  points from the mainland to  $i$ th island
  end for
  epoch = epoch + 1
end while

```

4 Experimental Settings

The test suite of 22 real-world problems selected for CEC 2011 competition in Special Session on Real-Parameter Numerical Optimization [6] is used as one benchmark in the experimental comparison. The functions in the benchmark differ in the computational complexity, and in the dimension of the search space which varies from $D = 1$ to $D = 240$, the dimensionality of most problems exceeds $D = 20$. For each algorithm and problem, 25 independent runs were carried out. The run of the algorithm stops if the prescribed number of objective function evaluations $MaxFES = 150000$ is reached. The partial results of the algorithms after reaching one third and two-thirds of $MaxFES$ were also recorded. The point in the terminal population with the smallest function value is the solution of the problem found in the run. The population size of original adaptive DE variants $N = 100$ was used in all CEC 2011 problems. The sub-populations size of the proposed migration models is studied parameter set to $N_p = 10, 15, 45$ and 90 . The number of generations performed on all k islands before migration of individuals is second analysed parameter and its values are $nde = 5, 10, 20$, and 50 . The migration models are labelled based on using explicit N_p , and nde settings ‘P+ N_p +x+ nde ’, i.e. ‘P15x5’. The other control parameters are set up according to a recommendation of authors in their original papers. All the algorithms are implemented in Matlab 2010a, and all computations were carried out on a standard PC with Windows 7, Intel(R) Core(TM)i7-4790 CPU 3.6 GHz, 16 GB RAM.

5 Results and Discussion

In this experimental study six original adaptive DE variants are compared with ten newly proposed migration models. Because the lack of place, detailed results are not provided. In Table 2 medians of all algorithms on all real-world problems are presented. The best value is underlined and printed bold, algorithms on second place has median printed bold, and median of algorithm on third place is only underlined. We can see that the best performing algorithms are among the original DE variants, and also migration models. The least median value is not the best testified statistics. Therefore in the last four rows of this table numbers of significant best, second, third and last positions based on Kruskal-Wallis test are printed. Each migration model in study is able to be competitive with the original DE variants. The most migration models are never the worst performing except P10x5 and P15x5. Combination of small sub-population size and frequent migration causes worse results.

Kruskal-Wallis non-parametric one-way ANOVA test was applied to each test problem. It was found that the performance of the algorithms in comparison differs significantly, the null hypothesis on the same performance is rejected in all the problems with achieved significance level $p < 0.000005$. Multiple comparison was then applied using Kruskal-Wallis z method (Dunn’s test). The details of the significant positions of the Kruskal-Wallis test for each problem are in Table 1.

Table 1. Significantly best, second, third and worst performing algorithms from Kruskal-Wallis test.

| Fun | D | 1st | 2nd | 3rd | last |
|--------|-----|---------------------------|------------------------|--------|--------|
| T01 | 6 | p45x5 | p15x50 _{b6e6} | b6e6rl | CoDE |
| T02 | 30 | p15x50 _{jade} | p15x50 | p15x20 | CoDE |
| T03 | 1 | No significant difference | | | |
| T04 | 1 | No significant difference | | | |
| T05 | 30 | p15x50 _{jade} | p15x10 | p15x20 | EPSDE |
| T06 | 30 | JADE | SaDE | p15x50 | EPSDE |
| T07 | 20 | p15x10 | p15x5 | p10x5 | EPSDE |
| T08 | 7 | No significant difference | | | |
| T09 | 126 | SaDE | b6e6rl | jDE | CoDE |
| T10 | 12 | b6e6rl | EPSDE | SaDE | p10x5 |
| T11.1 | 120 | p15x5 | p15x50 _{b6e6} | p45x5 | b6e6rl |
| T11.2 | 240 | p45x5 | p15x50 _{b6e6} | p15x20 | CoDE |
| T11.3 | 6 | b6e6rl | b6e6rl | b6e6rl | p15x5 |
| T11.4 | 13 | b6e6rl | SaDE | p90x50 | p10x5 |
| T11.5 | 15 | b6e6rl | EPSDE | p90x5 | JADE |
| T11.6 | 40 | CoDE | EPSDE | p90x50 | p10x5 |
| T11.7 | 140 | CoDE | EPSDE | p90x50 | jDE |
| T11.8 | 96 | CoDE | EPSDE | b6e6rl | jDE |
| T11.9 | 96 | JADE | EPSDE | b6e6rl | jDE |
| T11.10 | 96 | CoDE | EPSDE | b6e6rl | jDE |
| T13 | 26 | p15x50 | p10x5 | p15x5 | CoDE |
| T14 | 22 | p90x5 | p15x10 | p45x5 | CoDE |

We can see that in three problems all algorithms perform similarly. The best performing original algorithms are b6e6rl, JADE, and CoDE. The best performing migration models are P15x50_{jade} and P45x5. The worst performing algorithms are CoDE, jDE, EPSDE, and P10x5. The CoDE variant provides good performance only for problems T11.1–T11.10. These problems are very similar, and therefore CoDE is able to solve 'one kind' of CEC 2011 problems.

For better comparison of 16 algorithms on 22 problems in three equidistant stages the Friedman test on medians is applied, and results are depicted in Fig. 1. The test was carried out on medians of minimal function values at three stages of the search, namely after $FES = 50,000$, $100,000$, and $150,000$. The null hypothesis on the equivalent efficiency of the algorithms was rejected at the all stages of the search with $p < 5 \times 10^{-6}$. The algorithms in this table are ordered from left to right with respect to their mean rank from Friedman test at the finish of the search, i.e. after reaching $MaxFES = 150,000$.

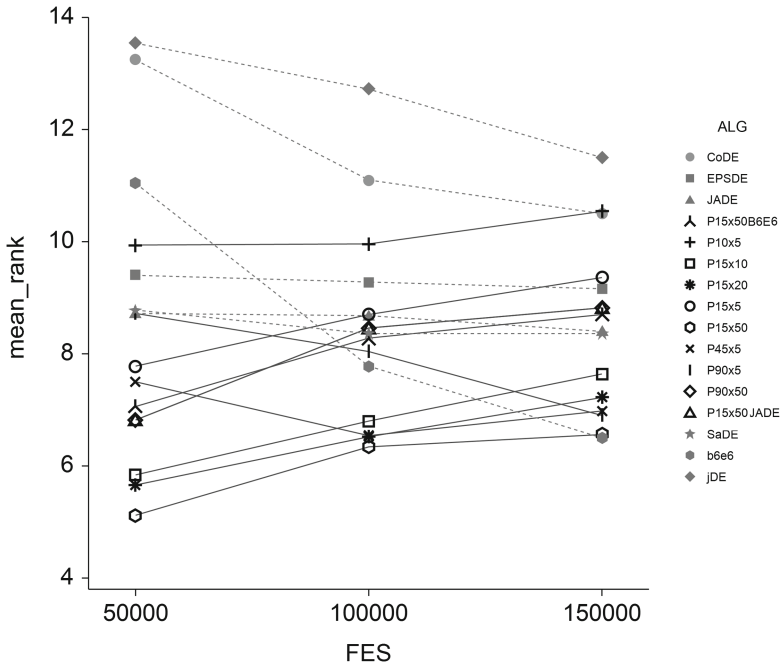


Fig. 1. Mean ranks from Friedman tests at three stages of the search, CEC 2011 test suite, $FES = 50000, 100000, 150000$.

In Fig. 1 the mean rank of all the compared algorithms are provided. We can observe a good performance of most migration models in the first stage. This result is important when a restrict number of evaluations is provided. The worst performing migration model is P10x5, the small sub-populations size ($N_p = 10$), and frequent migration ($nde = 5$) decreases the efficiency. The best performing model is P15x50, therefore the model with JADE on mainland, and model with b6e6rl on all sub-populations employs these settings. The performance of P90x5 variant is increased with increasing number of function evaluations. This conclusion means better performance for algorithms with ‘casual’ population size in DE. Although the best population size, and the good number of generations before migration is combined, unfortunately the performance of P90x50 is rather worse. When the number of function evaluations increases, the performance of migration models decreases whereas the performance of the original DE is rather increasing. The best performing P15x50 is slightly outperformed in the last stage by b6e6rl.

Further success of adaptive DE variants in migration model is analyzed. Each sub-population is developed by one DE, and the number of successfully generated individuals of each sub-population is weighted to % and depicted in Fig. 2. The most successful original algorithm is EPSDE, but the efficiency decreases with increasing sub-populations size N_p . The higher efficiency of EPSDE variant in migration model in most experiments could be caused by a huge number of small positive changes. Further research will be focused on an analysis of the diversity of the sub-populations during the search process.

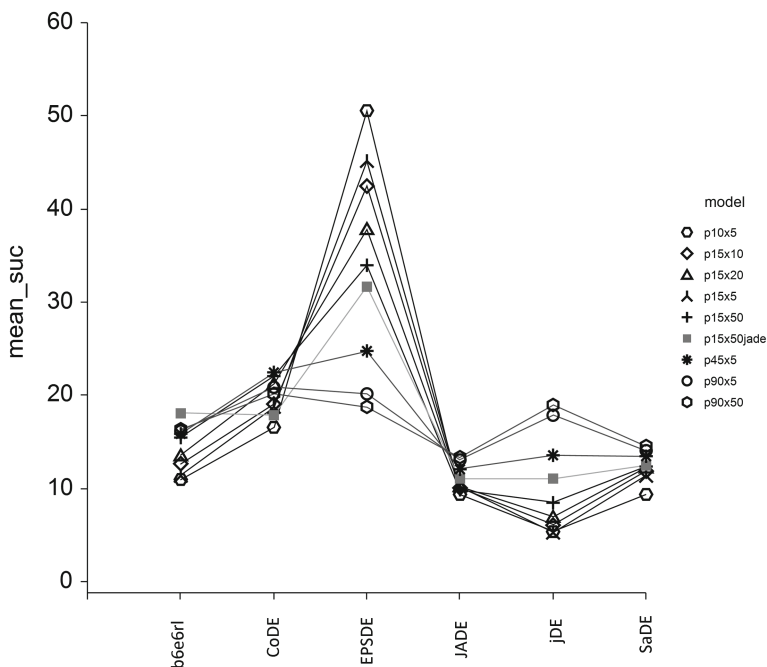


Fig. 2. Success of adaptive DE variants in migration models (%).

6 Conclusion

The results of the experimental comparison of ten various migration adaptive DE models with six popular adaptive DE algorithms demonstrate clearly very good efficiency of most of the migration models. The best performing algorithm in comparison is migration model P15x50, worst performing is adaptive jDE. Most of the migration models perform better than the original DE in the first stage of CEC 2011. This fact is important for real problems restricted by small function evaluation amount. The most successful adaptive DE in migration models is EPSDE, the efficiency of this variant is decreased with increasing subpopulations size. Although migration model of adaptive DE provide very good performance, further study of another migration parameters is fundamental for next research.

References

1. Brest, J., Greiner, S., Boškovič, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**, 646–657 (2006)
2. Bujok, P.: Synchronous and asynchronous migration in adaptive differential evolution algorithms. *Neural Netw. World* **23**(1), 17–30 (2013)

3. Bujok, P.: Hierarchical topology in parallel differential evolution. In: Dimov, I., Fidanova, S., Lirkov, I. (eds.) NMA 2014. LNCS, vol. 8962, pp. 62–69. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15585-2_7
4. Bujok, P., Tvrdík, J.: Parallel migration model employing various adaptive variants of differential evolution. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC/SIDE -2012. LNCS, vol. 7269, pp. 39–47. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29353-5_5
5. Bujok, P., Tvrdík, J.: New variants of adaptive differential evolution algorithm with competing strategies. *Acta Electronica at Informatica* **15**(2), 49–56 (2015)
6. Das, S., Suganthan, P.N.: Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, India and Nanyang Technological University, Singapore, Technical report (2010)
7. Das, S., Mullick, S.S., Suganthan, P.: Recent advances in differential evolution - an updated survey. *Swarm Evol. Comput.* **27**, 1–30 (2016)
8. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**, 27–54 (2011)
9. Glotic, A., Glotic, A., Kitak, P., Pihler, J., Tigar, I.: Parallel self-adaptive differential evolution algorithm for solving short-term hydro scheduling problem. *IEEE Trans. Power Syst.* **29**(5), 2347–2358 (2014)
10. Gong, Y.J., Chen, W.N., Zhan, Z.H., Zhang, J., Li, Y., Zhang, Q., Li, J.J.: Distributed evolutionary algorithms and their models: a survey of the state-of-the-art. *Appl. Soft Comput.* **34**, 286–300 (2015)
11. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **11**, 1679–1696 (2011)
12. Penas, D., Banga, J., González, P., Doallo, R.: Enhanced parallel differential evolution algorithm for problems in computational systems biology. *Appl. Soft Comput.* **33**, 86–99 (2015)
13. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
14. Storn, R., Price, K.V.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997)
15. Tvrdík, J.: Self-adaptive variants of differential evolution with exponential crossover. *Analele West Univ. Timisoara Ser. Math.-Inform.* **47**, 151–168 (2009). <http://www1.osu.cz/~tvrdik/>
16. Wang, X., Tang, L.: Multiobjective operation optimization of naphtha pyrolysis process using parallel differential evolution. *Ind. Eng. Chem. Res.* **52**(40), 14415–14428 (2013)
17. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evol. Comput.* **15**, 55–66 (2011)
18. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**, 945–958 (2009)