Efficiency of Migration in Parallel Differential Evolution

Petr Bujok

University of Ostrava
Department of Computer Science
30. dubna 22, 70103 Ostrava
Czech Republic
petr.bujok@osu.cz

International Student Conference on Applied Mathematics and Informatics 2013

Outline

UNIVERSITAS OSTRAVIENSIS

Differential evolution algorithm

Description

- population-based stochastic optimization algorithm which heuristically explores the area of possible solutions
- possible solutions are represented as vectors: $\boldsymbol{x_i} = \{x_{i,1},\ x_{i,2}, \ldots x_{i,D}\}$ and create population of individuals $P = \{\boldsymbol{x_1},\ \boldsymbol{x_2}, \ldots \boldsymbol{x_N}\}$
- population of possible solutions is developed by evolutionary operators in order to find solution of task
- evolutionary operators used in DE are *mutation*, *crossover*, *selection* and *migration*

UNIVERSITAS OSTRAVIENSIS

Adaptive differential evolution

DE algorithm has four control parameters

- type of mutation with variant of crossover is called *strategy*
  - eg. $DE/rand/1/bin$
- control parameter of mutation $F \in (0, 2\rangle$
  - determines length of shift of parent in mutation $\longrightarrow$ *mutant vector*
- control parameter of crossover $CR \in \langle 0, 1\rangle$
  - determines probability of selection elements from parent or mutant vector $\longrightarrow$ *offspring*
- population size N
  - bigger population brings more detailed exploring of solution area
  - smaller population brings faster searching process

adaptive DE enables to adapt the values of the control parameters during search process in order to increase efficiency of DE

Parallelism in evolutionary algorithms

Motivation for using parallelism in optimization

- **problem:** solving of difficult tasks (*CEC'05*) requires large time demands
- **solution:** distribution time demands of algorithm among several parallel computational units
  - length of whole parallel algorithm is equivalent to length of the slowest parallel unit
- *master-slave*, *neighbourhood*, ***migration***, *hybrid*, *hierarchic*
- migration model distributes computations among k independent processes and enables to migrate information (by individuals)

UNIVERSITAS OSTRAVIENSIS

Parallel migration model

Details of migration model

- population P is distributed onto k sub-populations of equal sizes and located on isolated islands
- sub-populations are independently developed for several generations - *epoch*
- *migration* - selected individuals are moved between islands in order to exchange information, at the end of epoch
- parameters of migration model - *topology*, *sub-population size*, *migration rate*, *migration policy*, *migration frequency*
- migration frequency - *synchronous* and **asynchronous**

UNIVERSITAS OSTRAVIENSIS

Asynchronous migration

Description of proposed asynchronous migration model

- migration in the tested model occurs asynchronously by criterion:

$$worst_{\text{new}} - best_{\text{new}} < max(\frac{1}{exp(epoch_j - 1)}, 1 \times 10^{-4})$$
$$\text{AND}$$
$$best_{\text{old}} - best_{\text{new}} > \frac{\varepsilon}{100}$$
$$\text{OR}$$
$$newgen_j - oldgen_j \geq N,$$

  - $worst_{\text{new}}$, $best_{\text{new}}$ - worst and best function values in the current population
  - $epoch_j$ is epoch counter
  - $best_{\text{old}}$ - best function value in the preceding generation of population
  - $(newgen_j - oldgen_j)$ - number of generations completed till current epoch
  - ε - input parameter dependent on problem, $\varepsilon \in (1e-6,\ 1e-1)$

- best solution of one sub-population migrates to second sub-population where replaces worst solution

Experiments

Settings of the experiments

- three state-of-the-art adaptive DE algorithms are paralleled by proposed asynchronous migration model
  - b6e6rl, CoDE0, EPSDE
- population of individuals ($N = 60$) is equidistantly distributed among two islands with sub-populations sizes $N_p = 30$
- both islands are developed by one adaptive DE
- algorithms end when reached 300000 evaluations of individuals
- parallel algorithms are implemented in the pseudo-parallel regime – parallel parts are performed sequentially

UNIVERSITAS OSTRAVIENSIS

Experiments

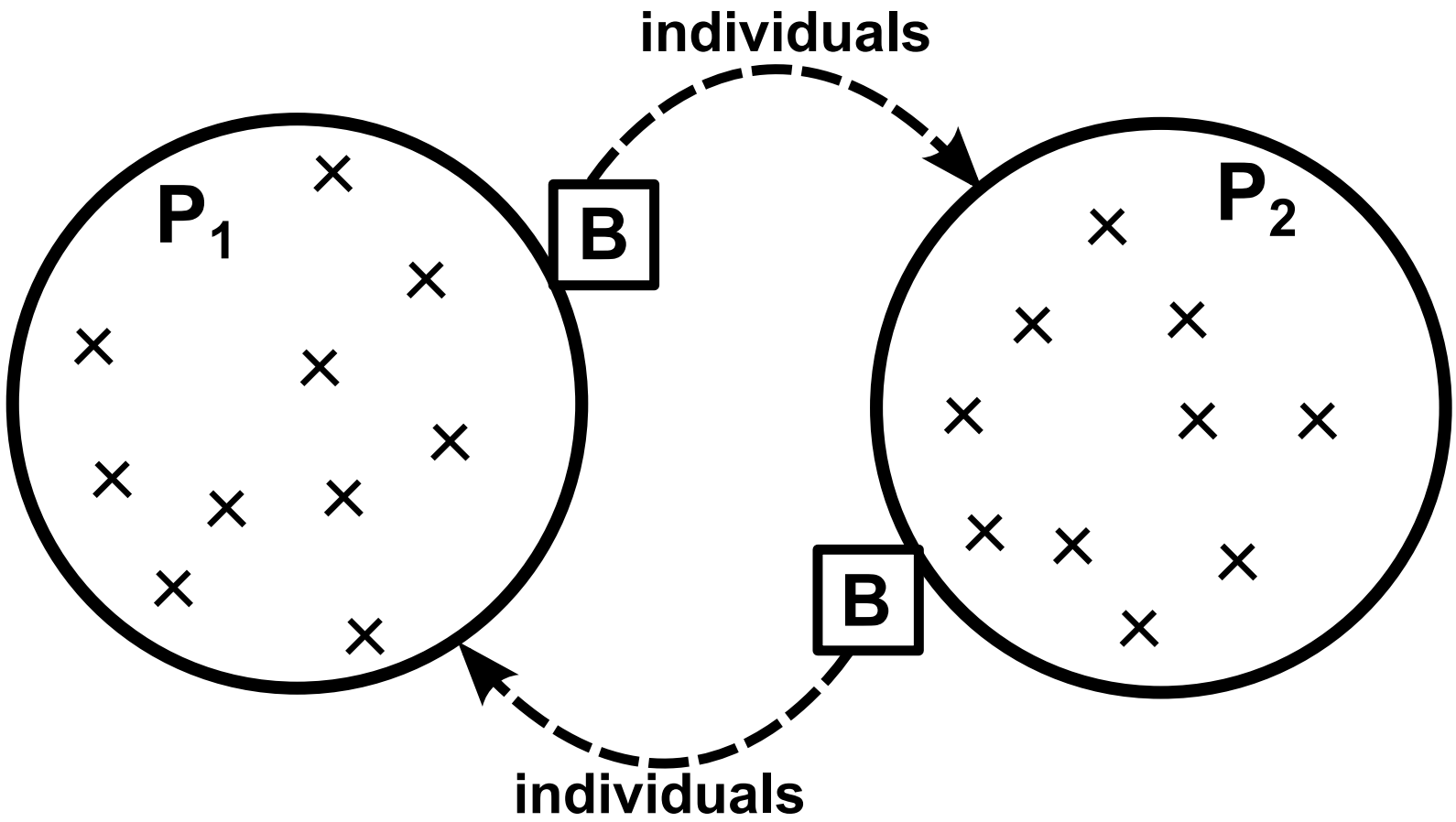

Figure: Topology

UNIVERSITAS OSTRAVIENSIS

Experiments

Benchmark functions *CEC'05*

- algorithms run on 25 functions of various hardness called *CEC'05*
- F_1-F_5 are unimodal problems
- F_6-F_{12} are multimodal problems
- F_{13}, F_{14} are expanded problems
- F_{15}-F_{25} are hybrid composition problems
- 25 runs was done for each algorithm and each problem
- best and worst solution, median, mean and standard deviation was calculated (of 25 runs)

UNIVERSITAS OSTRAVIENSIS

Results

Table: Algorithms better in functions

Algorithm	non-parallel	paralel asynchronous	"$\approx$"
b6e6rl	F_2-F_7, F_{10}, F_{16}, F_{18}, F_{19}, F_{21}-F_{24}	F_8, F_{11}-F_{15}, F_{17}, F_{20}, F_{25}	F_1, F_9
CoDE0	F_2-F_7, F_{18}, F_{19}, F_{21}-F_{25}	F_8-F_{17}, F_{20}	F_1
EPSDE	F_2-F_7, F_{25}	F_8-F_{17}, F_{19}, F_{21}, F_{23}, F_{24}	F_1

UNIVERSITAS OSTRAVIENSIS

Conclusion

Summary

- asynchronous migration enables to achieve better solutions in some of *CEC'05* problems
- parallel migration model is more efficient than non-parallel variants mostly in harder functions
- migration can increase efficiency of all tested non-parallel algorithms at least in several cases
- best efficiency of asynchronous migration is achieved with algorithm EPSDE
- best efficiency among all six algorithms is achieved in non-parallel b6e6rl

Future research will be focused on migration models of adaptive DE variants using HPC.

Thank you for attention!