COMPARISON OF USING SIMPLE GENETIC ALGORITHM AND PARALLEL GENETIC ALGORITM IN HEAT TRANSFER MODEL OPTIMIZATION

Michal Oravec, Stanislav Števo, Ivan Sekaj

Institute of Control and Industrial Informatics
Slovak University of Technology, Faculty of Electrical Engineering and Information Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel.: +421 2 60291 621 , +421 2 60291 403
*e-mail: michal.oravec@stuba.sk, stanislav.stevo@stuba.sk, ivan.sekaj@stuba.sk*T

Abstract: In this paper the comparison of using simple genetic algorithm and parallel genetic algorithm is presented. As the optimization problems the parameter setting of the heat transfer model of a building and the building's model calibration were chosen. The model simulation requires huge computing capacity and it is time consuming. Therefore the pressure of simulation evaluations number is concerned and the use of parallelism is desirable. Genetic algorithms and parallelization were implemented in Matlab and the simulation of heat transfer model, which is the part of the fitness function, is performed in Comsol Multiphysics.

Keywords: genetic algorithm, parallel genetic algorithm, building model optimization, heat transfer, Matlab, Comsol Multiphysics

1 INTRODUCTION

Genetic algorithms (GA) are effective stochastic optimization approaches imitating natural evolution process (Sekaj, 2005). Despite the fact, that there has been progress in the area of GA, the premature convergence sometimes occurred and large computing capacity is needed. Especially when more complicated system is to be optimized or a model simulation takes a lot of time. In such cases it's necessary to reduce the number of the cost function (fitness) evaluations (simulations).

There are many options to improve GA's. Most common is to tune GA setting to reach the best algorithm performance. However, it is sometimes not possible to tune the algorithm to be able to achieve a sufficient convergence rate to the global optimum. Therefore another option is to use parallelism. Parallel genetic algorithms (PGA) are able to improve the performance of simple genetic algorithms with a single population (Cantú-Paz, 1995).

This paper presents practical comparison of using simple genetic algorithm (SGA) with a single population and parallel genetic algorithm with population distributed into several interconnected subpopulations.

2 PARALLEL GENETIC ALGORITHM

In parallel genetic algorithms (PGA) the evolution is distributed into many more or less isolated subpopulations, where the transfer of genetic information among these subpopulations has an important influence on the evolution process. In this case we don't consider parallelisation into more processors or more computers respectively, which can extend the computational power of the computer system. Let us consider such parallelisation, wich is realised on a single processor or PC.

In our comparison a single GA with 50 individuals in the population and PGA with 5 subpopulations (nodes) with 10 individuals in each subpopulation are experimentally compared. The migrations are performed by replacing a randomly selected individual in the target node (except of the best one) by a copy of the best individual from the source node (best-random policy). The migration in the PGA according to the defined architecture is realized periodically after 5 generations.

The architecture of the considered PGA is depicted in Fig.1 (Cantú-Paz, 2001).

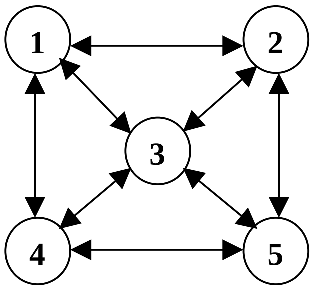

Figure 1: Considered PGA architecture

The genetic algorithm which is used in each node of the PGA and in the SGA is as follows:

1. Random population initialization and fitness calculation.
2. Selection of individuals :
   a. Best individuals which are copied into the new population without any change – Pop1 (2 in PGA and 5 in SGA)
   b. Random selection of a group of individuals which are copied without any change into the new population – Pop2 (4 in PGA and 30 in SGA).
   c. Tournament selection of parents – Pop3 (4 in PGA and 15 in SGA).
3. Mutation and crossover of parents (Pop3) with global mutation rate 0.02, local mutation rate 0.02 and probability of one-point crossover 0.75 – Pop3*
4. Completion of the new population by unification of the groups Pop1, Pop2 and Pop3*.
5. New population fitness calculation.
6. Test of terminating condition, if not fulfilled, then jump to the Step 2.

3 CASE STUDY

In the first experiment the heater proportions (height, width, depth) were optimized. Simulation of heat transfer model of the room was implemented in Comsol Multiphysics using FEM structure (Števo, 2009). The model of the room is shown in Fig.2. SGA and PGA were implemented in Matlab.

The aim of the experiments was to compare the SGA and PGA performance. Each individual in population is represented by a string which contains 3 parameters (height, width, depth) and the fitness function is represented as difference between the mean simulated temperature and the required temperature in the room (294K or 21°C) (Števo, 2009a).

The performance has been measured in a standard way using the convergence rate of the fitness function, which is the graph of the fitness function values of the currently best individual in the SGA population or entire PGA population respectively ("best so far" from all subpopulations).

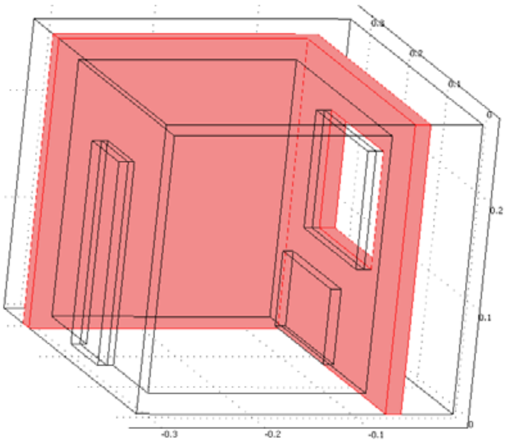

Figure 2: Model of the room in Comsol.

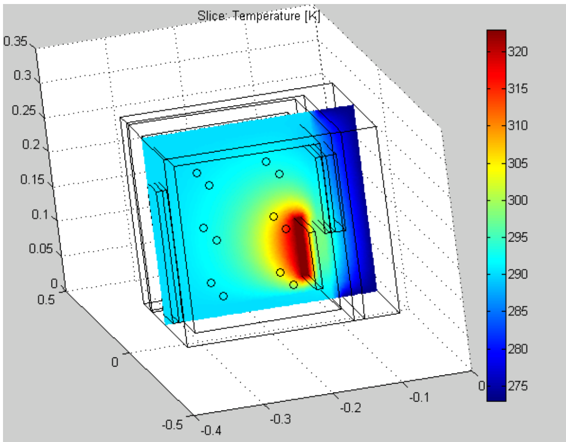

Figure 3: Simulation of the best solution for heater proportion.

In Fig.4 the convergence rate depending on the number of generation and in Fig.5 depending on the number of fitness evaluations is depicted. Number of evaluations is used to show the exact computing effort of the specific algorithm. Each graph represents the mean value of 5 algorithm runs. In Table 1 the number of evaluations required to reach the best solution (approximately the same in SGA and PGA) are presented. Fig. 3 represents the simulation of the solution for heater proportion optimization.

In this case, only 3 parameters were optimized. Therefore using parallelism is not so effective and the number of evaluations required to reach the best solution (in meaning of required computation time) are approximately the same in PGA and SGA.

Table 1: Nr. of evaluations required to reach the best solution in heater proportion optimization

	SGA	PGA	PGA/SGA [%]
run 1	480	476	99.2
run 2	510	532	104.3
run 3	450	448	99.5
run 4	420	448	106.7
run 5	540	504	93.4
average	**480**	**481.6**	**100.4**

In the second experiment a building's model calibration is proposed. The aim is to adjust 14 model parameters. That means the string has 14 genes. Each gene represents thickness of an independent element. The solution properties (fitness function) are considered as the correspondence between measured and simulated data (Števo, 2009b). With a calibrated mode we are able to reduce the maximum error from cca. 2.5°C to 0.3°C (Števo, 2009b).

In Fig.6 the convergence rate depending on the number of generation and in Fig.7 depending on the number of evaluations is depicted. Each graph represents the mean value of 5 algorithm runs. The numbers of evaluations required are presented in Table 2. Fig. 8 shows the well calibrated model of the building.

In this more complex model optimization (14 parameters), the PGA's convergence is much faster than SGA's with saving approximately 50% of computation time (number of fitness evaluations needed), which in our case can save tenth hours of computation time .

Table 2: Nr. of evaluations required to reach the best solution in building's model calibration

	SGA	PGA	PGA/SGA [%]
run 1	1980	1036	52.3
run 2	2250	1288	57.2
run 3	2430	1176	48.4
run 4	2160	1232	57.0
run 5	1710	952	55.7
average	**2106**	**1140.8**	**54.2**

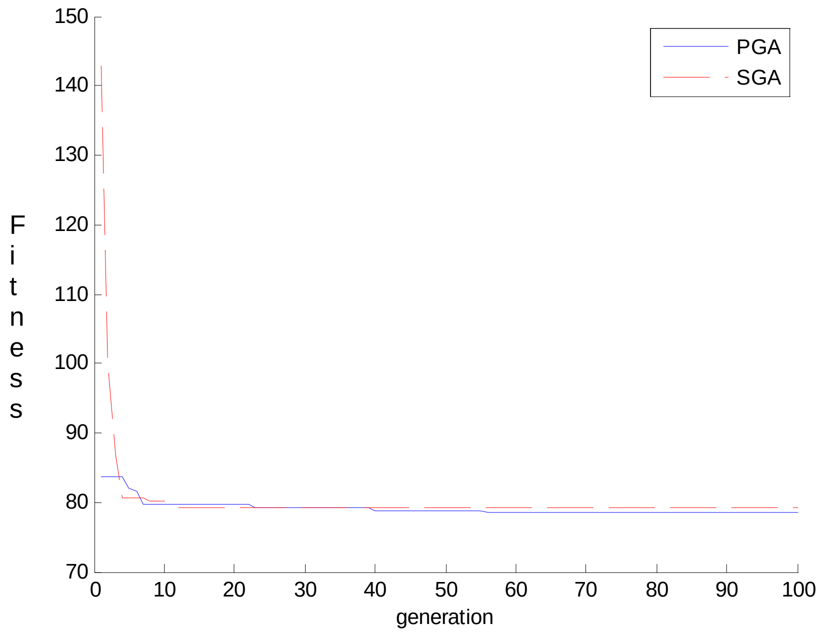

Figure 4: Heater proportion optimization (generations)

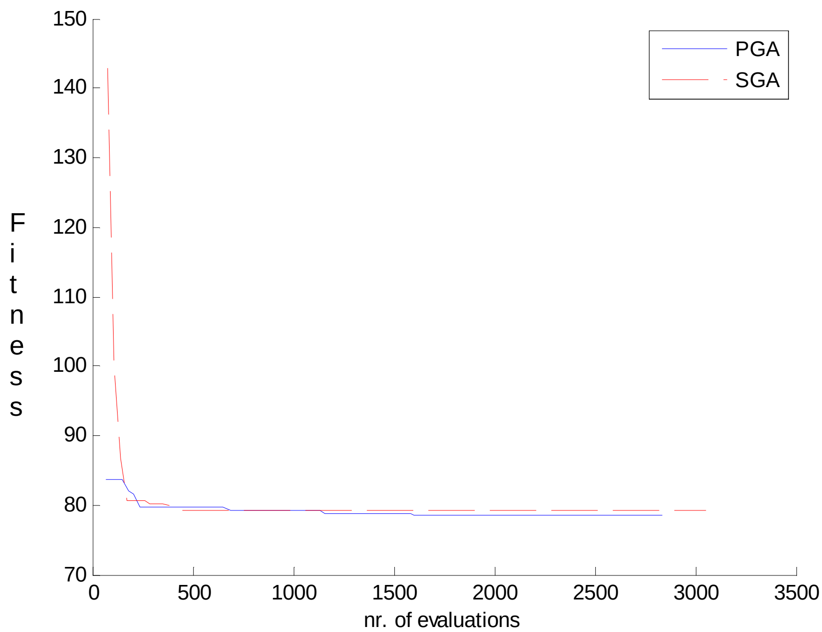

Figure 5: Heater proportion optimization (nr. of evaluations)

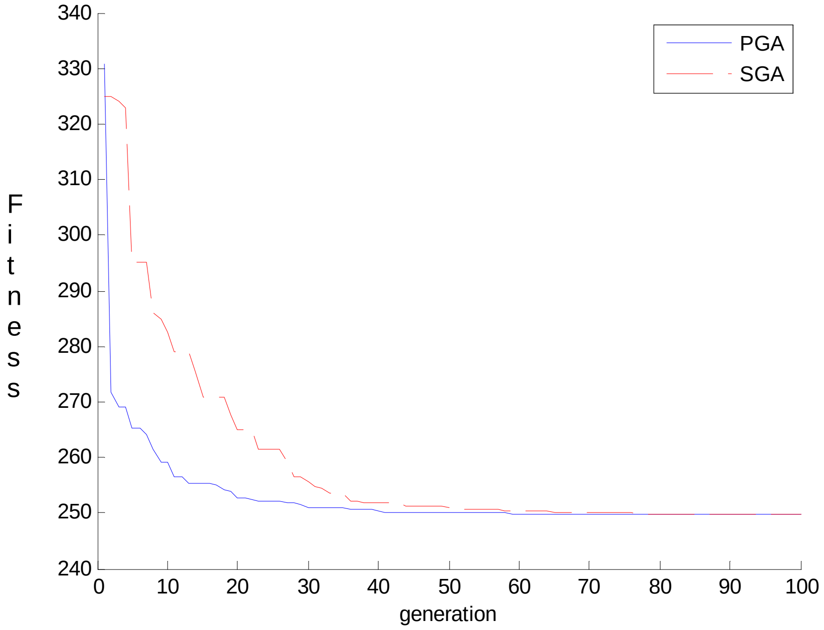

Figure 6: Building's model calibration (generations)

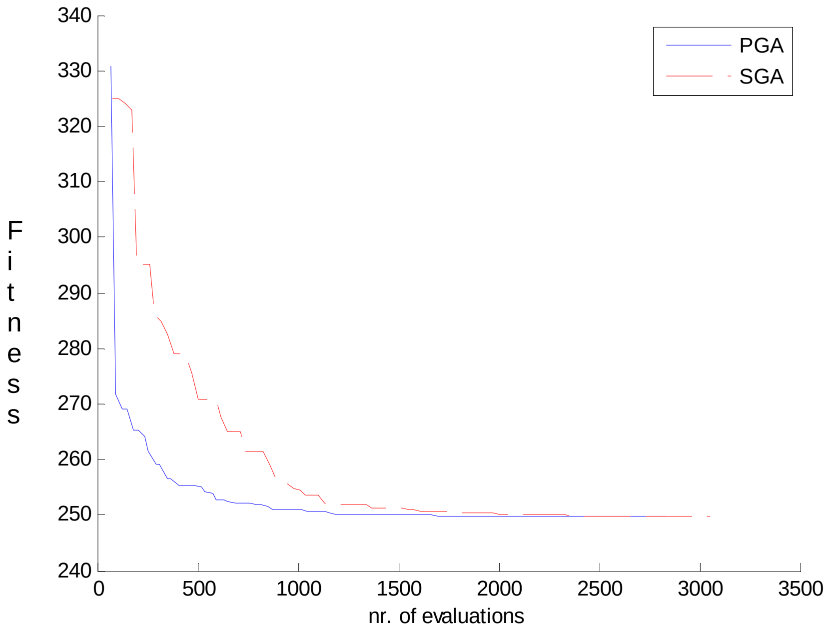

Figure 7: Building's model calibration (nr. of evaluations)

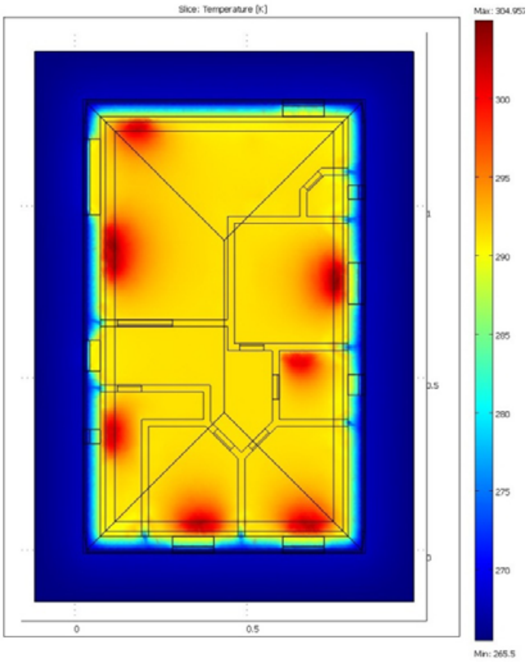

Figure 8: Well calibrated model of the building

4 CONCLUSION

In the paper the use of PGA and SGA for selected heat transfer optimisation problems are compared. Due to migration and information exchange between nodes, the proper PGA configuration brings decrease of computation time in comparison with using simple GA with a single population. This is true mailny in complex and time consuming optimisation/design applications. Next, PGA is able to decrease the measure of premature convergence (local optimum) and to find better solutions (better sub-optimal or global optimum).

REFERENCES

CANTÚ-PAZ, E. (1995): A summary of research on parallel genetic algorithms, *IlliGAL Report No. 95007*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign.

CANTÚ-PAZ, E. (2001): Migration polices, selection pressure, and parallel evolutionary algorithms, *In Journal of heuristics 7(4)*, pp. 311-334.

SEKAJ, I., KAJAN, S., ORAVEC, M. (2008): Realizácia paralelných genetických algoritmov v Matlabe pomocou parallel computing toolboxu, *In Technical Computing Prague '08*, Prague, Czech Republic, pp. 95

SEKAJ, I. (2005). Evolučné výpočty a ich využitie v praxi, IRIS Bratislava

SEKAJ, I., ORAVEC, M. (2009): Comparison of some parallel genetic algorithms with population reinitialization, *In Mendel 2009: 15th International Conference on Soft Computing*, Brno, Czech Republic, pp. 23-28

ŠTEVO, S. (2009a): Optimalizácia vykurovania podľa vonkajších teplotných podmienok, *In Technical Computing Prague '09*, Prague, Czech Republic

ŠTEVO, S., ŠLEZÁROVÁ, A. (2009b): Heat transfer model of building calibrated by genetic algorithm, *In Elitech '09*, Bratislava, Slovak Republic