Distributed evolutionary algorithm and grids in optimization of structures

Wacław Kuś, Tadeusz Burczyński

Department for Strength of Materials and Computational Mechanics
Silesian Univeristy of Technology
Gliwice, Poland
{waclaw.kus, tadeusz.burczynski}@polsl.pl

Abstract

The paper is devoted to computational grids applications in distributed evolutionary optimization of mechanical structures. The LCG2 grid middleware is used. The optimization is performed using distributed evolutionary algorithm. The fitness function is computed using the finite element method. The numerical example is presented in the paper.

Keywords: Distributed evolutionary algorithm, Optimization, FEM, Grid.

1. Introduction

The shape optimization of structures can be solved using methods based on sensitivity analysis information or non-gradient methods based on genetic algorithms [Michalewicz96]. Applications of evolutionary algorithms in optimization need only information about values of an objective (fitness) function. The fitness function is calculated for each chromosome in each generation by solving the boundary - value problem by means of the Finite Element Method (FEM) [Kleiber98] [Zienkiewicz00]. This approach does not need information about the gradient of the fitness function and gives the great probability of finding the global optimum. The main drawback of this approach is the long time of calculations. The applications of the distributed evolutionary algorithms [Tanese89] can shorten the time of calculations [Kus00][Burczynski02][Burczynski03] [Burczynski04a][Burczynski04b].
The computational grids allows to use distributed computational resources. The authorization is one of the most important elements of grids. The Public Key Infrastructure is used in most grid projects. The Virtual Organizations (VO) created by people with similar interests or working on similar projects allows to create grids and share resources.
The use of grid techniques in optimizations can lead to improvements in hardware and software utilization. The other advantages of grids are simple and uniform end user communication portals and programs. The first evolutionary optimization tests [Kus04a] were performed using Condor package [Condor]. The plugins and programs for evolutionary optimization of structures using UNICORE environment [Unicore] were presented in [Kus04b]. The use of LCG middleware[LCG] and Crossgrid [Crossgrid] project resources is presented in the paper.

2. Optimization of structures using the distributed evolutionary algorithm

Sequential genetic and evolutionary algorithms are well known and applied in many areas of optimization problems. The main disadvantage of these algorithms is the long time needed for computation. The distributed evolutionary algorithms (DEA) works similarly to many evolutionary algorithms operating on subpopulations. The evolutionary algorithms exchange chromosomes during a migration phase

ISSN: 1137-3601.

between subpopulations. When DEA is used the number of fitness function evaluations can be lower in comparison with sequential and parallel evolutionary algorithms. DEA works in the parallel manner, usually. Each of the evolutionary algorithms in DEA work on a different processing unit. The reduction of time could be bigger then the number of processing units when compared to sequential evolutionary algorithm. The flowchart of the distributed evolutionary algorithm for one subpopulation is presented in Fig. 1. The sample DEA with four subpopulations is shown in Fig. 2.

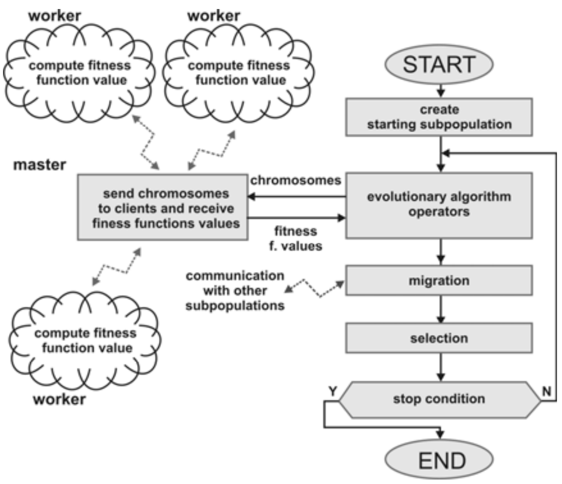

Figure 1. The flowchart of the distributed evolutionary algorithm for one subpopulation

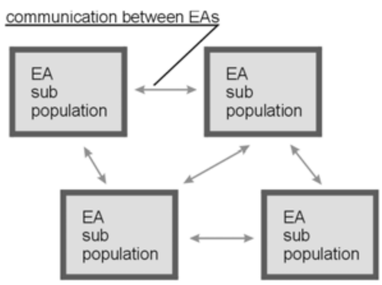

Figure 2. The sample DEA with four subpopulations

The starting subpopulation of chromosomes is created randomly. The evolutionary operators change chromosomes and the fitness function value for each chromosome is computed. The migration exchanges a part of chromosomes between subpopulations. The selection decides which chromosomes will be in the new population. The selection is done randomly, but the fitter chromosomes have bigger probability to be in the new population. The selection is performed on chromosomes changed by operators and immigrants. The next iteration is performed if the stop condition is not fulfilled. The stop condition can be expressed as a maximum number of iterations.

The computation of fitness function in optimization problems is performed using results of the FEM analysis. The genes describe the shape, material properties, topology of the structure. The structure is meshed and proper boundary conditions are applied before FEM analysis. The flowchart of fitness function evaluation is presented in Fig. 3.

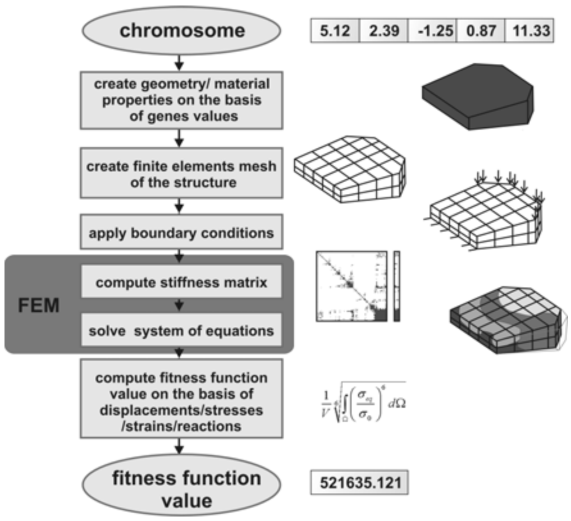

Figure 3. Fitness function evaluation using FEM

3. Evolutionary optimization using grid based on LCG middleware

The goal of the LCG project [LCG] is to create middleware (based on Globus Toolkit) which allows to create big grids. The LCG project is connected with Large Hadron Collider project realized in CERN. Many European grid projects uses LCG as software basis, for example Crossgird [Crossgrid], EGEE [EGEE]. The grids consists of user interface (computer for submitting, monitoring jobs), resource broker (computer which authorizes users, transfers files across grid, decides which resources will be used by user), gatekeepers (computers which translate the resource brokers job requests into working nodes job requests), working nodes (computers executing jobs) and storage elements (computers allowing to high performance access to storage data). The computer elements of the grid are distributed in many sites. The user perform action using user interface. The job submition is presented in Fig. 4.

The resource broker decides on the basis of job descripion provided by the user, current sites load and virtual organization policies, which computing elements should be used. The communication between computational sites and user are performed using resource broker, also jobs monitoring and fetching jobs results.

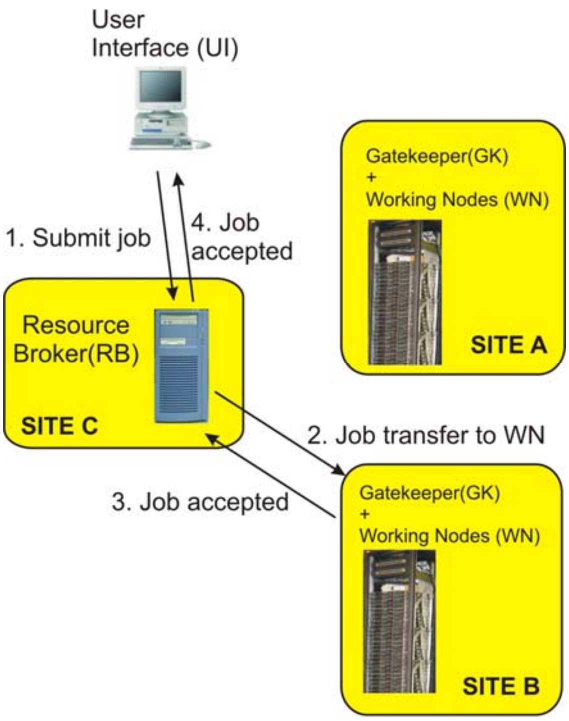

Figure 4. The job submition

The simplest way to use such grids is to submit evolutionary optimization job. The Crossgrid project testbed allows to use MPICH [MPICH] jobs. The distributed evolutionary algorithm can be implemented using MPICH library. There is submition of one job for one optimization problem.

4. Numerical test

The shape optimization of mechanical structure is considered. The elastic, static problem is analyzed for each chromosome. The minimization of the mean equivalent stresses in a plate is considered. The plate is loaded using two load schemas. The fitness function is expressed as a sum of mean equivalent stresses for two load cases:

$$F = \frac{1}{V}\int_{\Omega_1} \sigma_{eq} \mathrm{d}\Omega_1 + \frac{1}{V}\int_{\Omega_2} \sigma_{eq} \mathrm{d}\Omega_2 \qquad (1)$$

where V is volume of the plate, σ_{eq} means equivalent Huber-Mises stresses, Ω_1 is the plates area for first load case, Ω_1 is the plates area for the second load case. The chromosome contains 6 genes (g_0-g_6) and describes shape of the plate as shown in Fig. 5a,b).

a)

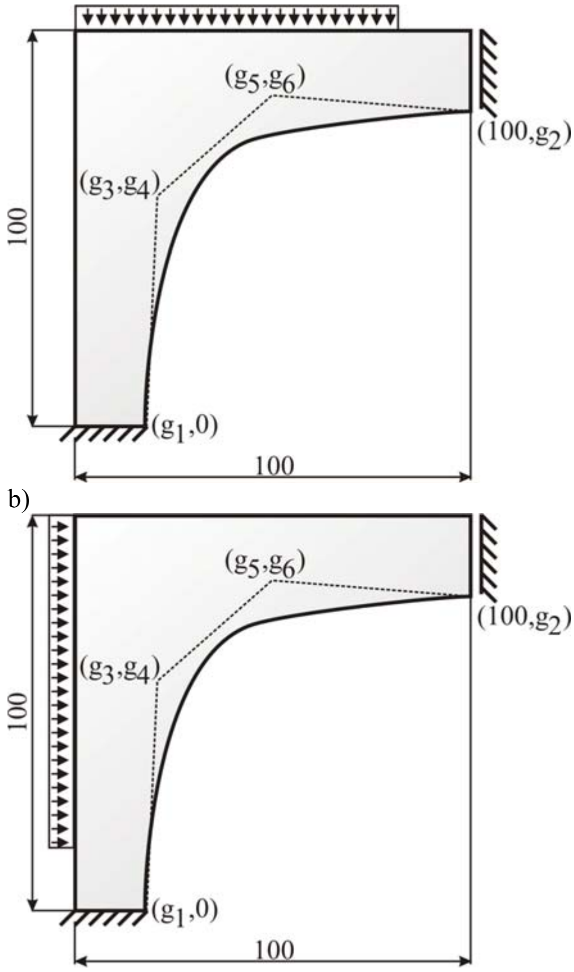

Figure 5. The geometry of the plate, a) first load case, b) second load case

The gaussian mutation, simple crossover and ranking selection were used in the evolutionary algoritm. Every chromosomes in subpopulations are crossovered and than mutated. The number of subpopulations is equal to 2, number of chromosomes is equal to 10 in each subpopulation.
The LCG grid environment and CrossGrid testbed were used during evolutionary optimization. The MPICH version of the coevolutionary algorithm was used (the tests were performed using two processors, the location of the used clusters were chosen by the resource broker). The parallelization of evolutionary algorithms can achieve good efficiency when low number of processors are used. The speedup near 2 was obtained using two processors. The best found result is shown in Fig. 6a,b). The distribution of equivalent stresses for both load cases is presented. The *Eq. stresses* means equivalent stress value for contour number *No*.

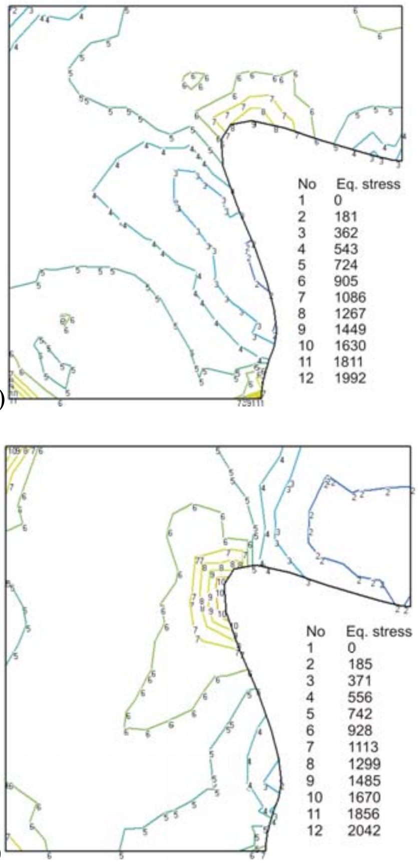

Figure 6. The best result, a) equivalent stresses map for best found chromosome for the first load case, b) for the second load case

The LCG2 middleware and Crossgrid testbed were used during test. The three sites were used during preparation of the test Laboratorio de Instrumentacao e Fisica Experimental de Particulas in Portugal, CESGA-Centro de Supercomputacion de Galicia in Spain and CYFRONET, Cracow in Poland.

The amount of work needed to perform optimization under grid is not big. The linux operating system GNU compilers and MPI communication library is available in most grids. The program need to be written using these resources. The time need to execute problem using grids is little bigger (under one minute) than time need to execute problem under queue manager on clusters. The execution time on a grid system is the same as on a cluster when the same computers are used. The biggest advantage of grids are access to many clusters and supercomputers distributed in different places without needs of obtaining logins on each resource. The other advantage is that the resources which are closest to our needs and with lowest number of jobs in queues are used when the resource broker is invoked.

5. Conclusions

The coupling of distributed evolutionary algorithm, finite element method and computational grid creates modern, powerful and efficient structures optimization tool. The evolutionary computation using grid environments opens new possibilities. The access to powerful computational distributed resources allows to perform computationally intensive jobs. The one time login and use of resource broker allow simple access to many clusters in virtual organization. The location and performance of the resources are not need to be known before submitting jobs. The numerical example of the evolutionary structural optimization is presented in the paper.

Acknowledgement

The work presented in the paper is sponsored by The Foundation for Polish Science, Professorial Grant.

References

[Burczynski02] Burczyński T., Kuś W., Distributed evolutionary algorithms in shape optimization of nonlinear structures. Lectures Notes on Computational Science 2328, Springer, 2002.

[Burczynski03] Burczyński T., Kuś W., Optimiza tion of structures using distributed and parallel evolutionary algorithms Parallel Processing and Applied Mathematics, PPAM2003, Revised papers, Lecture Notes on Computational Sciences 3019, Springer, pp. 572-579, 2004.

[Burczynski04a] Burczyński T., Kuś W., Długosz A., Poteralski A., Szczepanik M., Sequential and Distributed Evolutionary Computations in Structural Optimization. Lecture Notes on Artificial Intelligence 3070, Springer, 2004.

[Burczynski04b] Burczyński T., Kuś W., Długosz A., Orantek P., Optimization and defect identification using distributed evolutionary algorithms, Engineering Applications of Artificial Intelligence, No. 4, Vol. 17, pp. 337-344, 2004.

[Condor] Condor, High Throughput Computing, http://www.cs.wisc.edu/condor/

[Crossgrid] Crossgrid project home page, http://www.crossgrid.org

[EGEE] EGEE Enablig Grids for E-Science in Europe home page, http://www.eu-egee.org

[Kleiber98] Kleiber M. (red.), Handbook of Computational Solid Mechanics, Springer-Verlag, 1998.

[Kus00] Kuś W., Burczyński T., Evolutionary optimization of elasto-plastic solids. In: Methods of Artificial Intelligence in Mechanics and Mechanical Engineering (eds. T.Burczyński and W.Cholewa), Gliwice, 2000.

[Kus04a] Kuś W., Burczyński T., Computer implementation of the coevolutionary algorithm with Condor scheduler, KAEIOG 2004, Kazimierz, 2004.

[Kus04b] Kuś W., Burczyński T., Distributed evolutionary algorithm plugin for UNICORE system, Proc. 4th Cracow Grid Workshop, Cracow, 2004.

[LCG] LCG project home page, http://lcg.web.cern.ch/LCG/default.htm

[Michalewicz96] Michalewicz Z., Genetic algorithms + data structures=evolutionary algorithms. Springer-Verlag, Berlin, 1996.

[MPICH] MPICH project home page, http://www-unix.mcs.anl.gov/mpi/mpich/

[Tanese89] Tanese R., Distributed Genetic Algorithms. Proc. 3rd ICGA, pp.434-439, Ed. J.D. Schaffer. San Mateo, USA, 1989.

[Unicore] UNICORE Plus Final Report - Uniform Interface to Computing Resources, Joint Project Report for the BMBF Project UNICORE Plus, Grant Number: 01 IR 001 A-D, 2003.

[Zienkiewicz00] Zienkiewicz O. C., Taylor R. L., The Finite Element Method. The Basis, Vol. 1-2, Butterworth, Oxford, 2000.