Computer Assisted Mechanics and Engineering Sciences, **12**: 175–182, 2005.

Computational grids in evolutionary optimization of structures

Wacław Kuś
Department for Strength of Materials and Computational Mechanics,
Silesian University of Technology,
Konarskiego 18a, 44-100 Gliwice, Poland

(Received March 31, 2005)

The paper is devoted to computational grids applications in evolutionary optimization of structures. The two grid middleware are used, UNICORE and LCG2. The distributed evolutionary algorithm is used for optimizataion. The fitness function is computed using finite element method. Numerical examples are presented.

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 [13]. 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) [8, 17]. 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 [15] can shorten the time of calculations [9, 1–4].

The computational grids allows to use distributed computational resources. The authorization is one of the most important elements of grids. The Public Key Infrastructure are 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 computational grids is effective when computational intensive tasks are performed. The additional time is needed to execute jobs in grids (when comparing with clusters). The time is not big and is under one minute in most cases.

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/programs. The first evolutionary optimization tests [10] were performed using Condor package [5]. The plugins and programs for evolutionary optimization of structures using UNICORE environment [16] were presented in [11]. The use of LCG middleware [12] and Crossgrid [6] 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 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 theoretical reduction of time could be bigger then the number of processing units. 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. 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

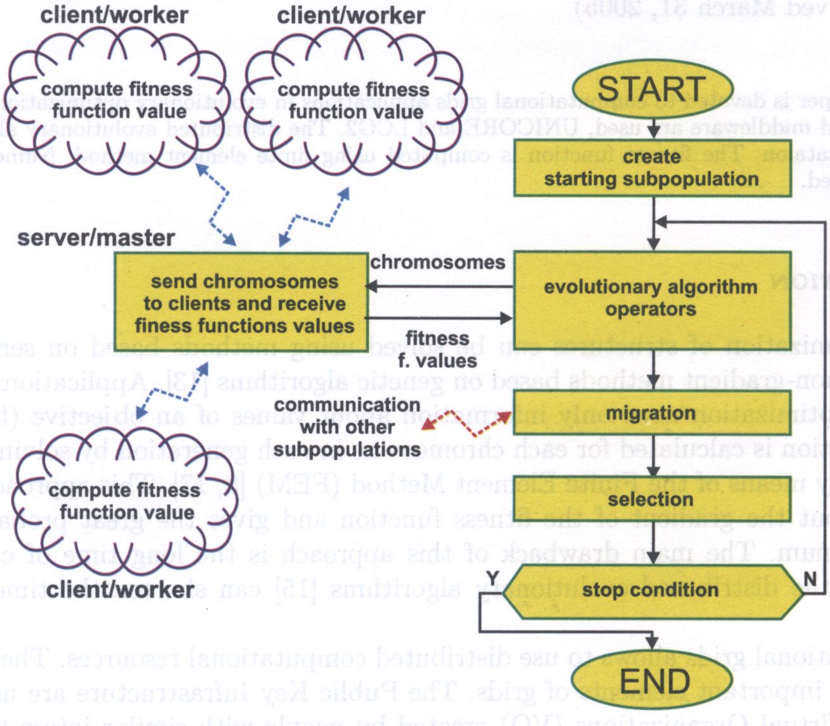
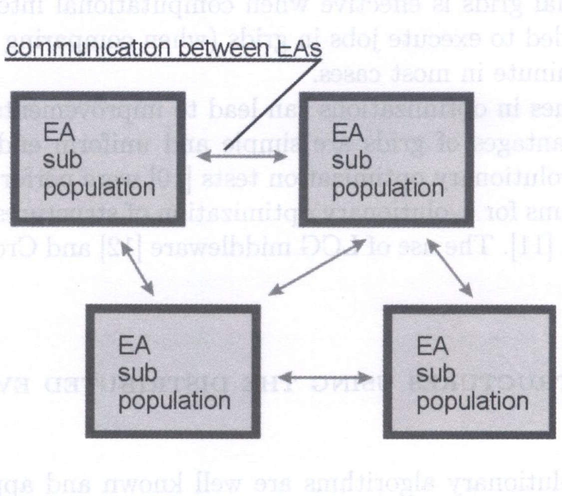


Fig. 1. The flowchart of the distributed evolutionary algorithm for one subpopulation



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.

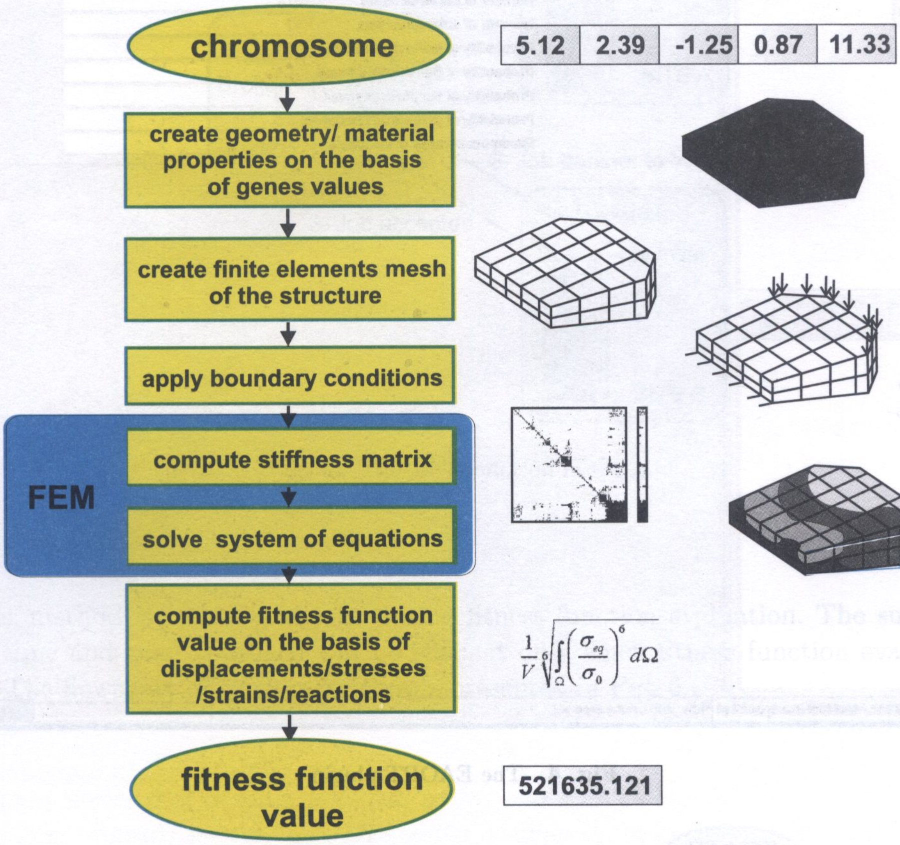


Fig. 3. Fitness function evaluation using FEM

3. Optimization using UNICORE EAOPT plugin

The UNICORE environment allows to perform computational tasks with use of computers without deep knowledge about target computers operating systems, directory structure etc. The UNICORE client module is written in Java and can be used on most computer systems currently available. The client is very flexible and prepared to use third party plugins. The UNICORE define special classes to be used in plugins like file browsers. The communication between client module and the target system can be performed with use of predefined classes.

The plugin can be used to prepare evolutionary optimization job. The parameters of the distributed evolutionary algorithm like number of chromosomes, genes, subpopulations, probabilities of operators, constraints on design variables can be load using plugin. The EAOPT plugin is also responsible for transferring files to target computer, job execution. The transfer of output files are also performed after computations.

The optimization is performed on the target computer/cluster using EAOPT – distributed evolutionary algorithm implementation. The EAOPT program and EAOPT plugin for UNICORE are both implemented by the author.

The EAOPT plugin is shown in Fig. 4.

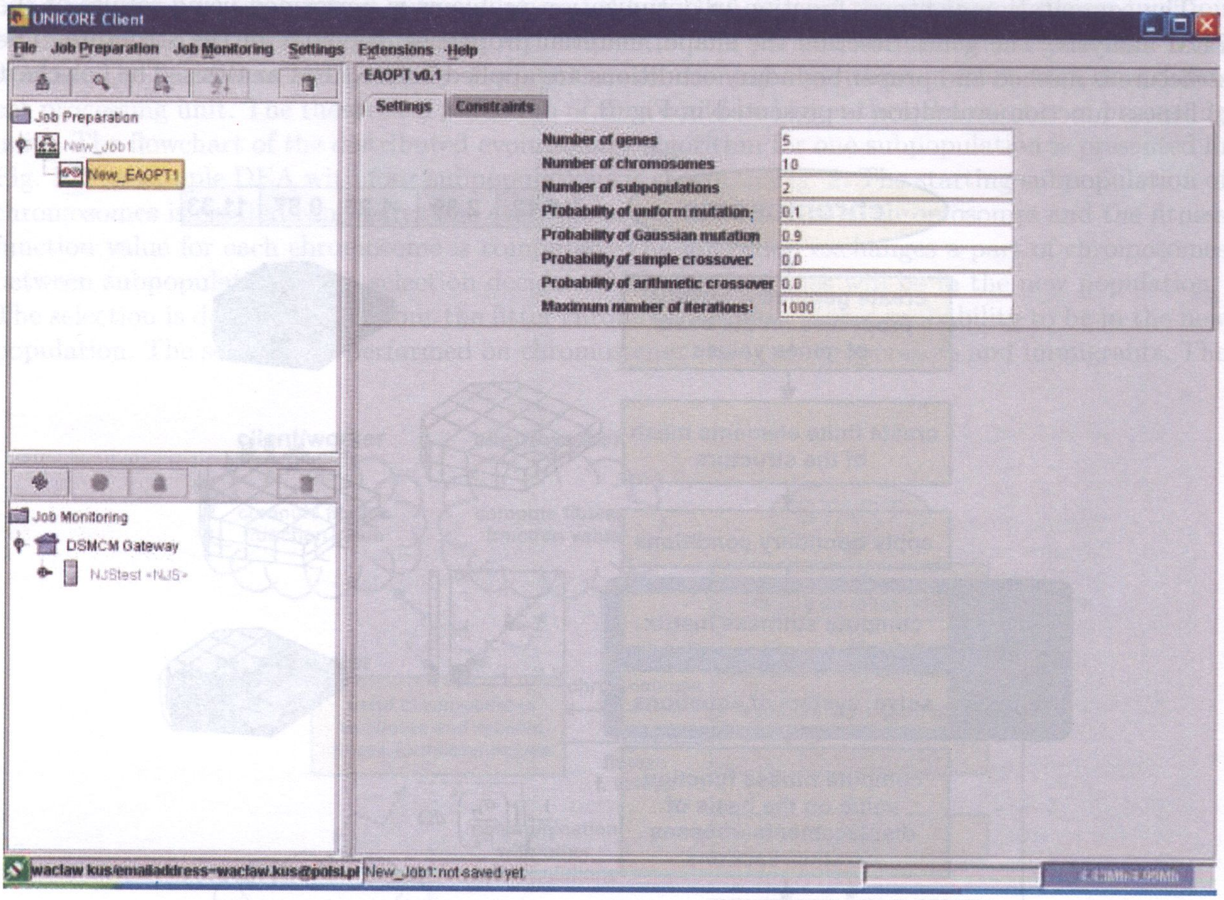


Fig. 4. The EAOPT plugin

4. Evolutionary optimization using grid based on LCG middleware

The goal of the LCG project [12] 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 [6], EGEE [7]. 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. 5. 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.

The simplest way to use such grids is to submit evolutionary optimization job. The Crossgrid project testbed allows to use MPICH [14] jobs. The distributed evolutionary algorithm can be implemented using MPICH library. There is submition of one job for one optimization problem.

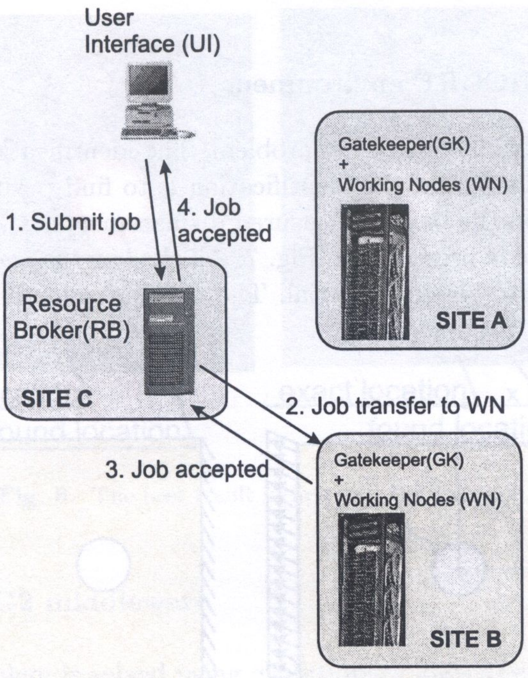


Fig. 5. The job submition in the grid

The other method is to use grid jobs during fitness function evaluation. The submition of job takes some time and such approach will be efficient only when fitness function evaluation is time consuming. The flowchart od such algorithm is presented in Fig. 6.

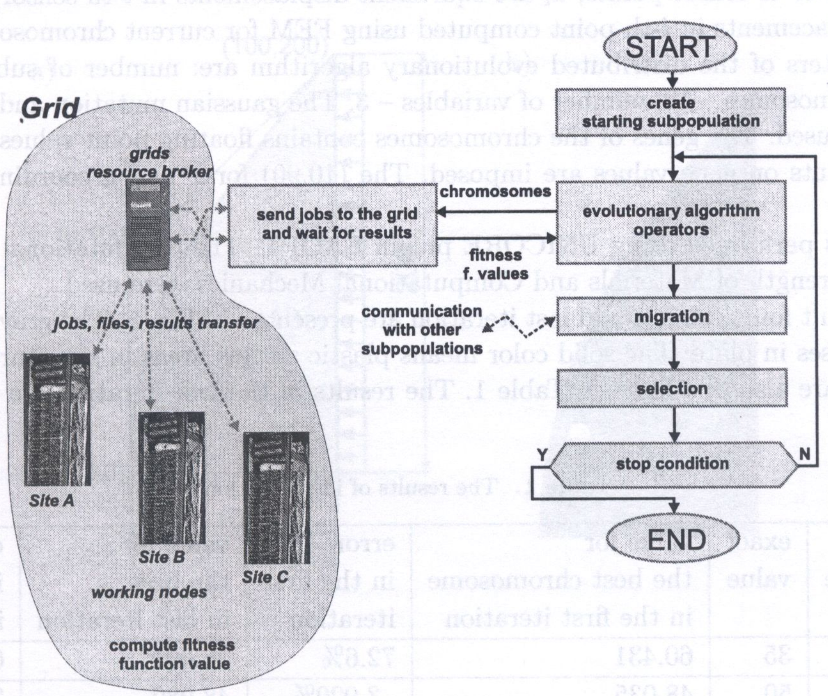


Fig. 6. The distributed evolutionary algorithm using many grid jobs

5. Numerical tests

5.1. Test with use of UNICORE environment

The goal of the test is to solve identification problem. The identification problem can be expressed as optimization problem. The aim of the identification is to find position of the center and radius of a void in the steel plate on the base of measured displacements.

The geometry of the plate is presented in Fig. 7. All the lengths and positions are in milimeters. The plate is made from elasto-plastic material. The displacements are measured in sensor points (Fig. 7).

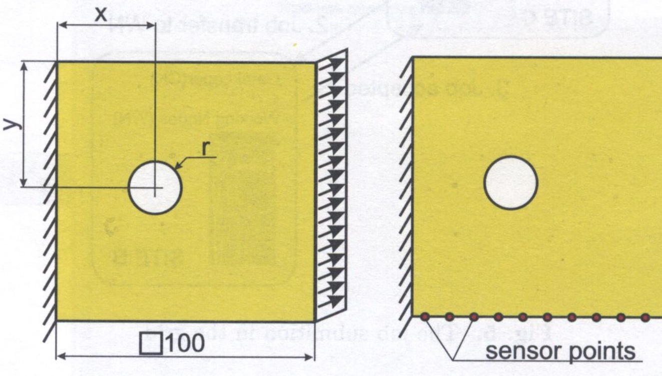


Fig. 7. The geometry of the plate

The fitness function is expressed as:

$$F = \sum_{i=1}^{n} |u_i - \hat{u}_i| \tag{1}$$

where n is number of sensor points, u_i are equivalent displacements in i-th sensor point and $\hat{u}_i$ are equivalent displacements in i-th point computed using FEM for current chromosome.

The parameters of the distributed evolutionary algorithm are: number of subpopulations – 2, number of chromosomes – 10, number of variables – 3. The gaussian mutation and simple crossover operators were used. The genes of the chromosomes contains floating point values.

The constraints on gene values are imposed. The (10.90) for x and y coordinates, (2.45) on r value.

The test was performed using UNICORE plugin EAOPT. The computational resources of Department for Strength of Materials and Computational Mechanics were used.

The best result found in first and last iteration are presented in Fig. 8. The gray levels represents equivalent stresses in plate. The solid color means plastic strains areas in structure.

The results are also presented in Table 1. The results in the last iteration are very close to the searched one.

Table 1. The results of identification

design variable	exact value	value for the best chromosome in the first iteration	error in the first iteration	value for the best in last iteration	error in last iteration
x	35	60.431	72.6%	35.001	0.005%
y	50	48.035	3.929%	48.980	2.040%
r	10	10.064	0.646%	10.064	0.646%

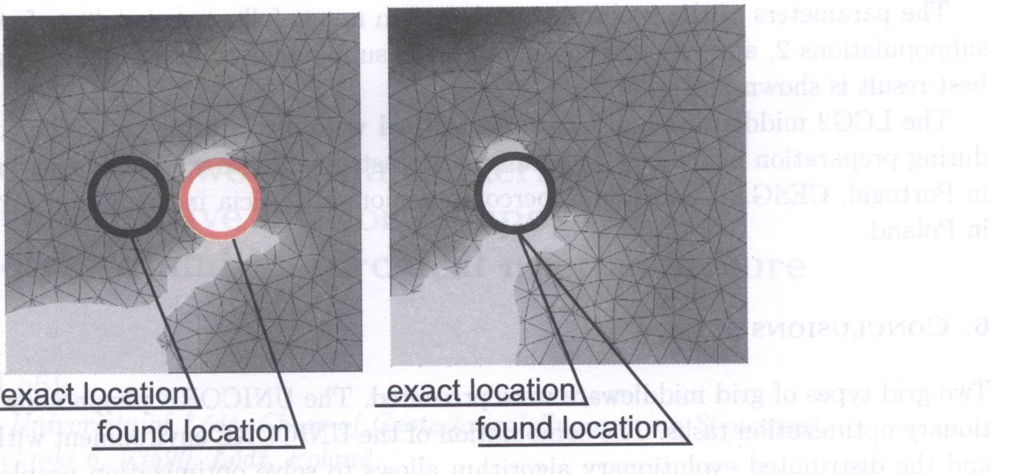


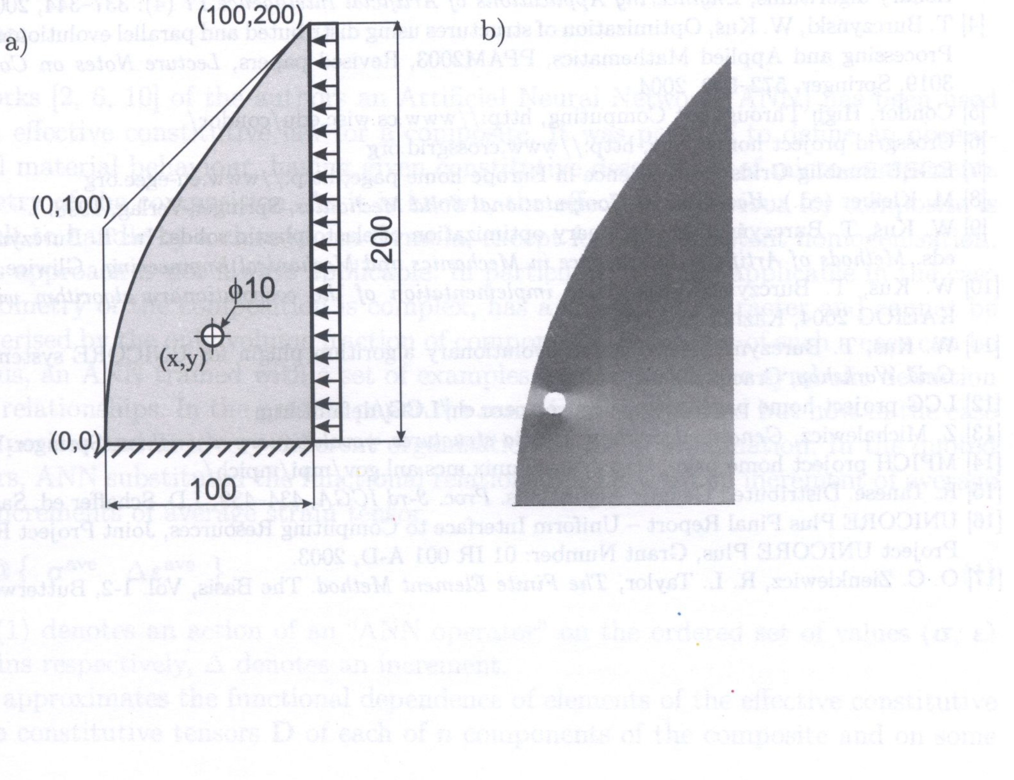
Fig. 8. The best result in first and last iteration

5.2. Test with use of LCG2 middleware

The shape optimization problem is solved using evolutionary algorithm. A steel plate with circular hole is considered (Fig. 9a). The optimal position of the center of the hole (x, y) is searched. The structure is made from an elastic material. The aim of the optimization is the minimization of equivalent stresses values in the structure. The area of the structure is constant. The fitness function is expressed as follows:

$$F = \int_{\Omega} \sigma_{\mathrm{eq}} \mathrm{d}\Omega \tag{2}$$

where: σ_{eq} are computed equivalent stresses.



The parameters of the evolutionary algorithm are as follows: a number of genes 2, a number of subpopulations 2, a number of chromosomes in subpopulation 4, a number of generations 10. The best result is shown in Fig. 9b.

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.

6. Conclusions

Two grid types of grid middleware were presented. The UNICORE plugin allows to perform evolutionary optimization tasks. The combination of the UNICORE environment with specialized plugin and the distributed evolutionary algorithm allows to solve optimization problems using easy user interface. The existing grids testbeds allows to perform numerical test using advanced middlewares and many computer resources. The coupling of distributed evolutionary algorithm, finite element method and computational grid creates modern, powerful and efficient structures optimization tool.

Ackowledgment

The research is financed from the Polish science budget resources in the years 2005–2008 as the research project.

References

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

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

[3] T. Burczyński, W. Kuś, A. Długosz, P. Orantek. Optimization and defect identification using distributed evolutionary algorithms, *Engineering Applications of Artificial Intelligence* **17** (4): 337–344, 2004.

[4] T. Burczyński, W. Kuś, Optimization of structures using distributed and parallel evolutionary algorithms Parallel Processing and Applied Mathematics, PPAM2003, Revised papers, *Lecture Notes on Computational Sciences* 3019, Springer, 572–579, 2004.

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

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

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

[8] M. Kleiber (ed.), *Handbook of Computational Solid Mechanics*, Springer-Verlag, 1998.

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

[10] W. Kuś, T. Burczyński. *Computer implementation of the coevolutionary algorithm with Condor schedule*, KAEIOG 2004, Kazimierz, 2004.

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

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

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

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

[15] R. Tanese. Distributed Genetic Algorithms. *Proc. 3-rd ICGA*, 434–439, J.D. Schaffer ed. San Mateo, USA, 1989.

[16] 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.

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