Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 8, 2008, No. 2

IMMUNE AND EVOLUTIONARY SHAPE OPTIMIZATION IN FORGING

WACŁAW KUŚ[1)], TADEUSZ BURCZYŃSKI[1,2)]

[1)] Department for Strength of Materials and Computational Mechanics
Silesian University of Technology, Konarskiego 18a, 44-100 Gliwice, Poland
[2)] Institute of Computer Modelling, Artificial Intelligence Section
Cracow University of Technology, Warszawska 24, 31-155 Kraków, Poland
Corresponding Author: Tadeusz.Burczynski@polsl.pl (T. Burczyński)

Abstract

The paper deals with applications of methods of artificial intelligence: artificial immune systems and evolutionary algorithms in optimization of a forging process. The shape optimization of the anvils in a two-stage forging process is considered as a numerical example. The paper contains description of the evolutionary algorithm, the artificial immune system and parallel versions of bioinspired algorithms in grid environment.

Key words: evolutionary computing, immune computing, grid environment, shape optimization, forging process

1. INTRODUCTION

The optimization methods inspired by biological mechanisms have become very popular in last few decades. Most of them give good results in optimization problems where multimodal functional appears. The paper describes the computational intelligence algorithms: evolutionary algorithms and artificial immune system. The evolutionary algorithms (EAs) are based on mechanisms taken from biological evolution of species. The mechanism similar to biological one like a mutation, a crossover and a selection are used in EAs. EAs operate on population of chromosomes (individuals with one chromosome). The uniform and Gaussian mutations and the simple crossover are used in the paper as evolutionary operators. EAs have found several application in optimization of mechanical structures (Burczyński et al., 2004, Kuś and Burczyński, 2006). The artificial immune systems (AISs) are developed on the basis of mechanism discovered in biological immune systems. An immune system is a complex system which contains distributed groups of specialized cells and organs. The main purpose of the immune system is to recognize and destroy pathogens. The AISs take only a few elements from the biological immune systems. The most frequently used are the mutation of the B cells, proliferation, memory cells, and recognition by using the B and T cells. The unknown global optimum is the searched pathogen. The memory cells contain design variables and proliferate during the optimization process performed with use of AIS. The B cells created from memory cells undergo mutation. The B cells are evaluated and better ones exchange memory cells. The applications of AISs in optimization are not so common as applications of EAs. The optimization using AISs can be found in paper Burczyński et al., 2007.

The paper deals with optimization in a forging process. The optimization problems in forging were

ISSN 1641-8581

performed using classical gradient-based method in Badrinarayanan (1997), Zabaras et al. (2000), Zhao et al. (2002) and using genetic algorithms in Antonio et al. (2002), Chung and Hwang (1998). The optimization using AIS and EA is shown in this paper which is an extension version of methods shown in papers Kuś and Burczyński (2007a) and (2007b). The shape optimization of the anvils in two-stage forging process is considered as a numerical example.

The goal of the present paper is to demonstrate an application of intelligent optimization techniques in the forging process. The paper presents the details about computational grid used during optimization process. The results of optimization are verified by using two optimization methods.

The paper consists of six sections. The second section describes the evolutionary algorithm, the artificial immune system is presented in section 3. The parallel versions of bioinspired algorithms in grid environment are described in section 4. The optimization problem formulation and numerical example are shown in section 5. Section 6 contains general conclusions.

2. THE EVOLUTIONARY ALGORITHM

Evolutionary algorithms are well known and applied in many areas of optimization problems Michalewicz (1996). The main disadvantage of these algorithms is the long time needed for computation. The parallel evolutionary algorithms Kuś (2006) perform an evolutionary process in the same manner as the sequential evolutionary algorithm. The difference is in a fitness function evaluation. The parallel evolutionary algorithm evaluates fitness function values in the parallel way. Theoretically, maximum reduction of time needed to solve the optimization problem using parallel evolutionary algorithms is equal to the number of used processing units. The flowchart of the parallel evolutionary algorithm is shown in figure 1. The starting population of chromosomes is created randomly. The evolutionary operators change chromosomes and the fitness function value for each chromosome is computed. The server/master transfers chromosomes to clients/workers. The workers compute the fitness function and send it to server. The workers operate on different processing units. The selection is performed after computing the fitness function value for each chromosome. 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 next iteration is performed if the stop condition is not fulfilled. The stop condition can be expressed as a maximum number of iterations. The evolutionary operators used in the presented algorithms are a crossover and a Gaussian mutation. The crossover chooses randomly two parent chromosomes and creates a new one containing a part of genes from the first and a part from the second parent. The Gaussian mutation creates chromosome based on a randomly chosen one. The part of the genes in a new chromosome have values changed by adding random numbers with the Gaussian distribution. The selection is performed by the use of the ranking method. The probability of being in the new population does not depend on the fitness function value, but on the number of chromosomes ordered accordingly to the fitness function values.

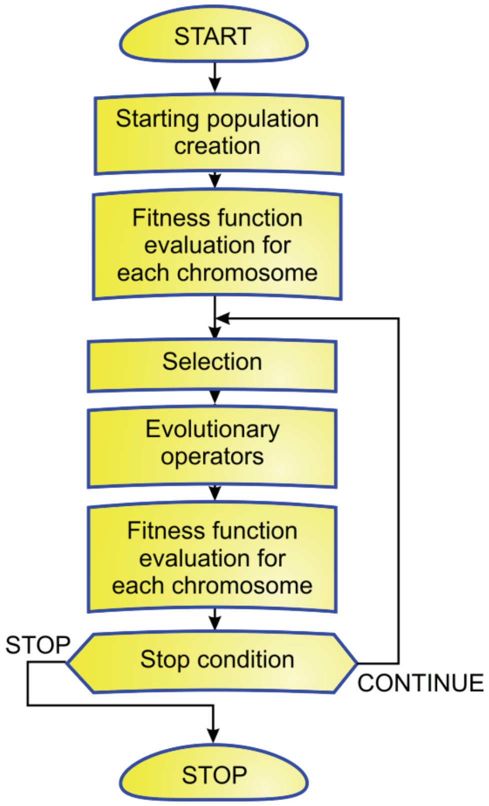

START
Starting population creation
Fitness function evaluation for each chromosome
Selection
Evolutionary operators
Fitness function evaluation for each chromosome
Stop condition
STOP
CONTINUE
STOP

Fig. 1. The flowchart of evolutionary algorithm

3. THE ARTIFICIAL IMMUNE SYSTEM

The artificial immune systems are developed on the basis of mechanism discovered in biological immune systems. An immune system is a complex system which contains distributed groups of specialized cells and organs. The main purpose of the immune system is to recognize and destroy pathogens - funguses, viruses, bacteria and improper functioning cells. The lymphocytes cells play a very important role in the immune system. The lymphocytes are divided into several groups of cells. There are two main groups B and T cells, both contains some subgroups (like B-T dependent or B-T independent). The B cells contains antibodies, which could neutralize pathogens and are also used to recognize pathogens. There is a big diversity between antibodies of the B cells, allowing recognition and neutralization of many different pathogens. The B cells are produced in bone marrow in long bones. A B cell undergoes a mutation process to achieve big diversity of antibodies. The T cells mature in thymus, only T cells recognizing non self cells are released to the lymphatic and the blood systems. There are also other cells like macrophages with presenting properties, the pathogens are processed by a cell and presented by using MHC (Major Histocompatibility Complex) proteins. The recognition of a pathogen is performed in a few steps. First, the B cells or macrophages present pathogen to a T cell using MHC, the T cell decides if the presented antigen is a pathogen. The T cell gives a chemical signal to B cells to release antibodies. A part of stimulated B cells goes to a lymph node and proliferate (clone). A part of the B cells changes into memory cells, the rest of them secrete antibodies into blood. The secondary response of the immunology system in the presence of known pathogens is faster because of memory cells. The memory cells created during primary response, proliferate and the antibodies are secreted to blood. The antibodies bind to pathogens and neutralizes them. Other cells like macrophages destroy pathogens. The number of lymphocytes in the organism changes, while the presence of pathogens increases, but after attacks a part of the lymphocytes is removed from the organism.

The artificial immune systems (AIS) de Castro and Timmis (2004), de Castro and Von Zuben (2002), Wierzchoń (2001) take only few elements from the biological immune systems. The most frequently used are the mutation of the B cells, proliferation, memory cells, and recognition by using the B and T cells. The artificial immune systems have been used to optimization problems, classification and also computer viruses recognition. The cloning algorithm Clonalg presented by von Zuben and de Castro de Castro and Von Zuben (2002) uses some mechanisms similar to biological immune systems to global optimization problems. The unknown global optimum is the searched pathogen. The memory cells contain project variables and proliferate during the optimization process. The B cells created from memory cells undergo mutation. The B cells are evaluated and better ones exchange memory cells. The crowding mechanism is used - the diverse between memory cells is forced in Wierzchoń (2001) version of Clonalg. A new memory cell is randomly created and substitutes the old one, if two memory cells have similar project variables. The crowding mechanism allows finding not only the global optimum but also other local ones. The presented approach is based on the algorithm presented in Wierzchoń (2001). The mutation operator is changed. The Gaussian mutation is used instead of the nonuniform mutation in the presented approach.

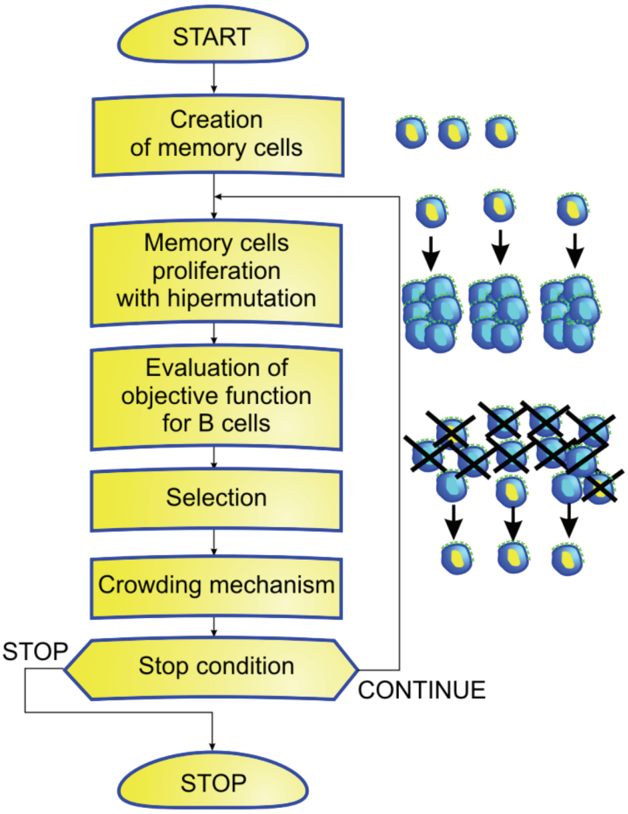

START
Creation of memory cells
Memory cells proliferation with hipermutation
Evaluation of objective function for B cells
Selection
Crowding mechanism
Stop condition
STOP
CONTINUE
STOP

***Fig. 2.** The flowchart of artificial immune system*

The flowchart of the artificial immune system is shown in figure 2. The memory cells are created randomly. They proliferate and mutate creating B cells. The number of clones created by each memory cell is determined by the memory cells objective function value. The objective functions for B cells are evaluated. The selection process exchanges some memory cells for better B cells. The selection is performed on the basis of the geometrical distance between each memory cell and B cells (measured by using design variables). The crowding mechanism removes similar memory cells. The similarity is also determined as the geometrical distance between memory cells. The process is iteratively repeated until the stop condition is fulfilled. The stop condition can be expressed as the maximum number of iterations.

4. THE EA AND AIS IN GRID ENVIRONMENT

The use of bioinspired optimization algorithms leads to evaluating hundreds or thousands objective functions. The computations of the objective function for physical problems takes a lot of time. The parallelization of objective function evaluations is very important in this case. The most time consuming part of the optimization algorithm should be parallelized. In the paper the objective function evaluation is parallelized.

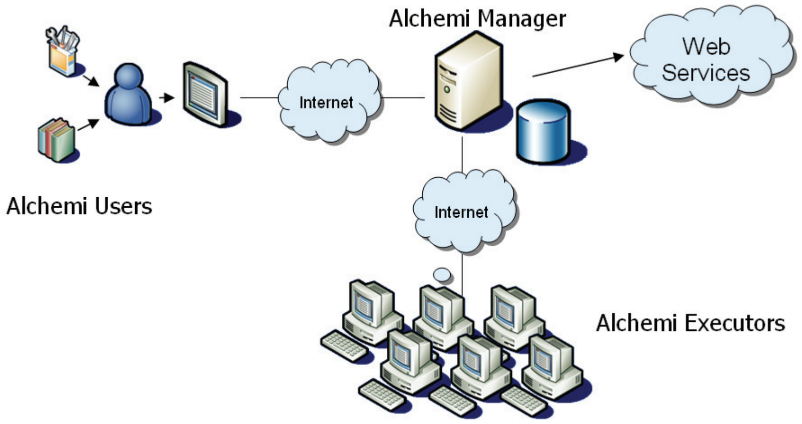


Fig. 3. Architecture of grid based on Alchemi framework (Akshay et al., 2005)

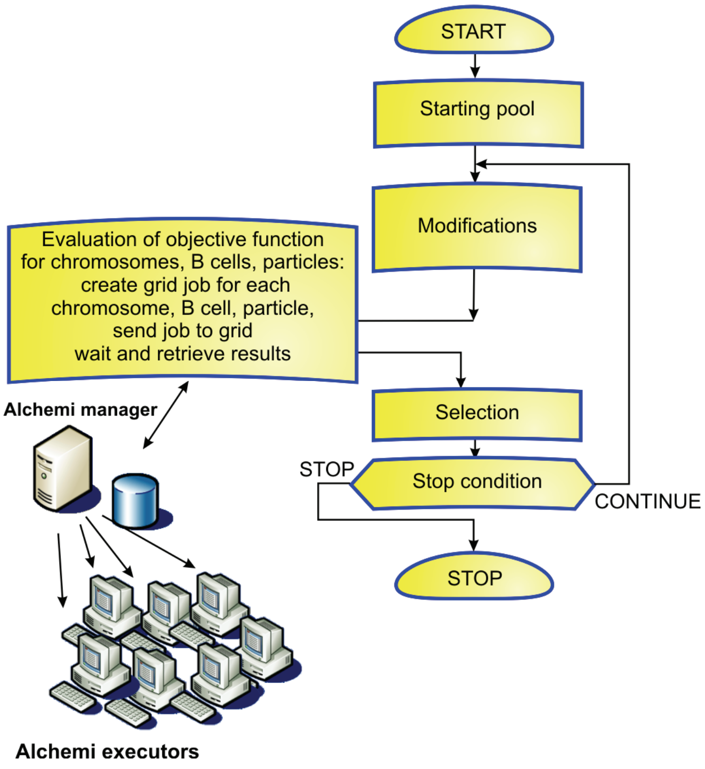

START
Starting pool
Modifications
Evaluation of objective function for chromosomes, B cells, particles: create grid job for each chromosome, B cell, particle, send job to grid wait and retrieve results
Alchemi manager
Selection
STOP
Stop condition
CONTINUE
STOP
Alchemi executors

Fig. 4. Flowchart of parallel grid based bioinspired optimization algorithm

The Alchemi framework Akshay et al. (2005), Alchemi web page (2008) was used to construct a grid. The Alchemi framework is based on Windows .NET. This makes Alchemi useful only on hardware using Windows operating system. The Alchemi consists of a few elements a) Alchemi Manager – the central host with scheduling capabilities, one manager is needed for grid or part of grid, b) Alchemi Executors – the hosts performing computations, c) Alchemi Cross Platform Manager – web services based manager with ability to communicate with non-Alchemi parts of grid. The security policy is based on usernames and passwords. The users can be grouped. The end-users, executors and administrator groups are available by default. The information about users, tasks, jobs, executor hosts are stored in database connected with Alchemi Manager (the SQL Server, MySQL and "in-memory" databases are supported). The architecture of grid based on Alchemi is shown in figure 3.

The most important advantage of Alchemi framework is API provided for grid applications developers. The execution of grid applications is performed using remote threads running on executors

The optimization problem can be submitted to the grid and executed on some grid resources on one hand. On the other hand when the computations of each objective function value is time consuming, each job can be submitted to the grid as separated grid job. The bioinspired algorithm with objective function evaluations as a separate jobs is shown in figure 4. The typical steps for considered bioinspired algorithms are presented. The poll of solutions is created in the first step. The modifications of pool are performed. The grid is used to evaluate objective function values in the parallel way. The next, the selection is performed and based on a stop condition, algorithm stops or iterates.

The grid threads are used during parallel computations. The objective function for each chromosome or B cell are computed separately by creating grid threads. The maximum number of threads working in parallel is equal to number of chromosomes or B cells in one iteration.

5. SHAPE OPTIMIZATION OF ANVILS

The shape optimization of anvils in the first stage of the two-stage forming is considered as a numerical example. The forging is performed using a flat die in the second stage. The goal of the optimization is to find such a shape of anvils which gives the cylindrical product of forging after the second stage. The two-stage forging using flat anvils is presented in figure 5a. The product after forging has the barreled shape. Figure 5b presents the two-stage forging with optimized anvils. The product should have cylindrical shape after two-stage forging.

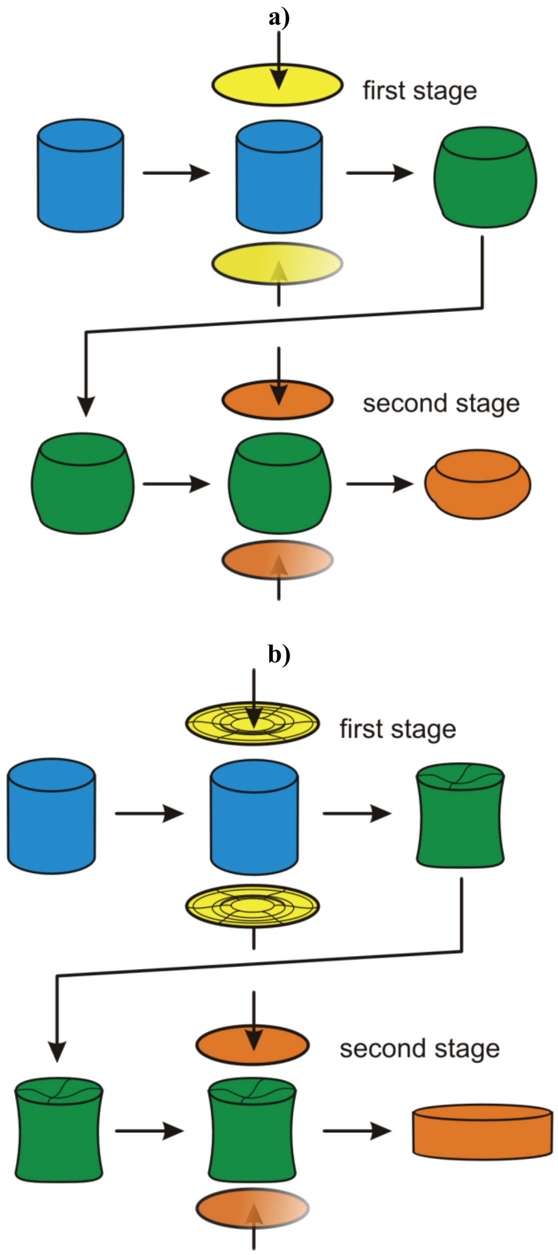


Fig. 5. Two-stage forging with: a) flat anvils, b) optimized anvils

The forging is simulated using MSC.Marc (User guide, 2002) program. The finite element method is used during the direct problem solution. The forging is analyzed as the axisymetrical problem, the model is shown in figure 6. The preform is made from a highly nonlinear material. The material nonlinearities and different shapes of the preform have influence on the objective function computing time. The objective function computation time for chromosomes and B cells can vary by as much as a factor of two.

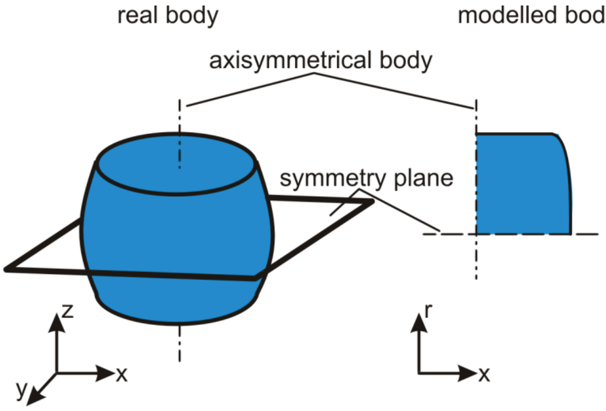


Fig. 6. Axisymetrical model of the perform

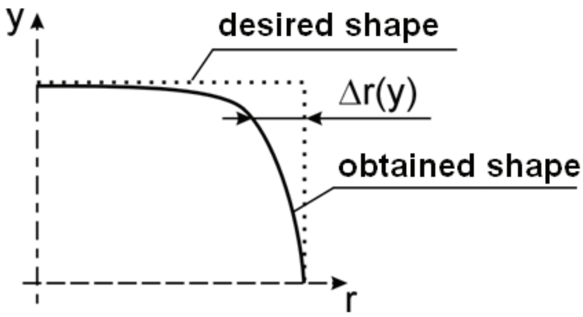


Fig. 7. Difference between obtained and desired shape

The objective function for a design vector **ch** is expressed as:

$$F(\mathbf{ch}) = \int_y \Delta r(y)\, \mathrm{d}\, y \qquad (1)$$

where $\Delta r(y)$ is a difference between desired and obtained shape after forging for y coordinate (figure 7).

The chromosome **ch** contains parameters defining anvils shape. The anvils were modeled using the NURBS curves. The genes contain information about the NURBS control polygon vertices. The sample NURBS anvil shape with polygon of control points is presented in figure 8.

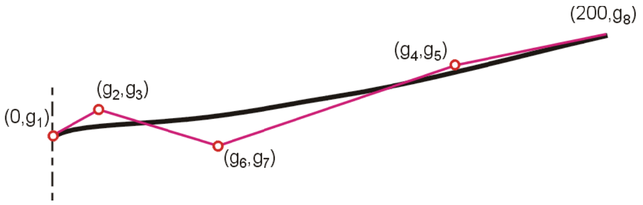
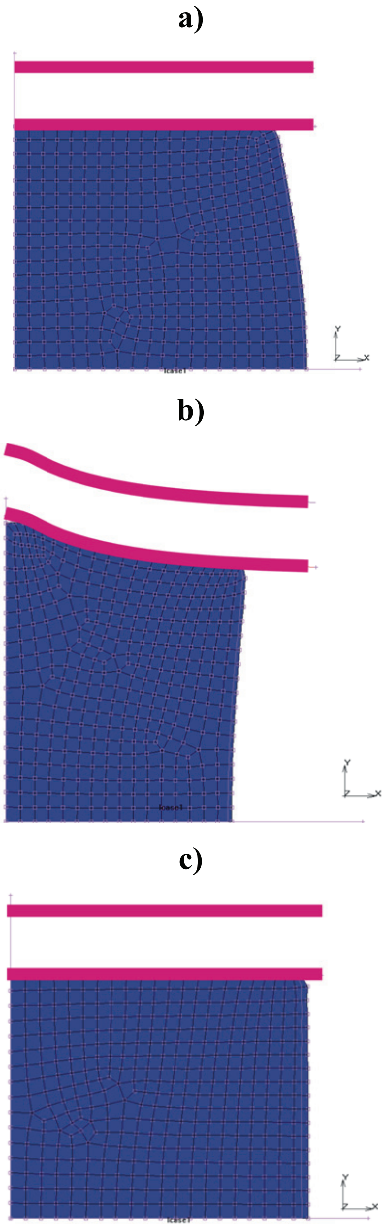


Fig. 8. *Shape of the anvil defined using NURBS control polygon vertices*

The numerical example was computed using EA and AIS. The 18 chromosomes and B cells each containing 8 parameters were used. The simple crossover and the Gaussian mutation were used for every chromosome in the population in EA. The two subpopulations were used in the evolutionary algorithm. The AIS uses 2 memory cells with 4 clones.


a)

b)

c)

Fig. 9. *a) Product after forging using flat anvils in both stages,product after forging using optimized anvil in first stage and flat anvils in second stage, b) result after first stage, c) final product*

The use of AIS and EA gave similar results. The results of the two-stage forging when only flat anvils were used are presented in figure 9a. The barreling of the product can be easily observed. The results after optimization using both algorithms are presented in figure 9b,c. The shape of the product is very close to the cylindrical one.

The EA and AIS worked in the grid environment. The testbed configuration is shown in figure 10. The speedup of computations were measured for EA and AIS. The speedup is the ratio of time need for computation using many computers and time need for computation using one computer. The speedup for 18 computers using EA was equal to 14. The AIS achieved speedup 7 by using 8 computers.

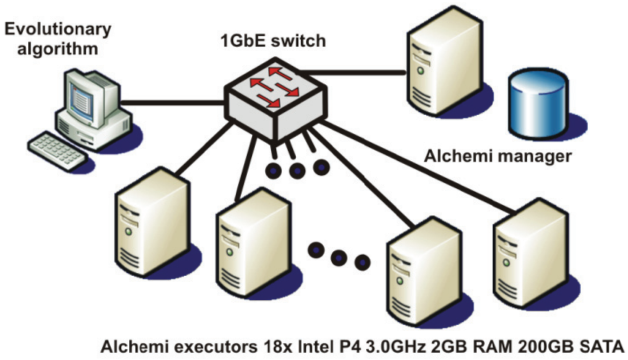


Fig. 10. *Grid testbed*

6. CONCLUSIONS

The evolutionary algorithm and artificial immune systems were presented in the paper. The parallelization of these algorithms and the use of the grid environment during optimization were described. The application of considered algorithms in forging is shown. The shape optimization of anvils is presented. The EA and AIS gave similar results. The measurements of speedup of computations in grid test environment were performed. The optimization results obtained by using both methods are very close to each other.

ACKNOWLEDGEMENT

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

REFERENCES

Akshay, L., Rajkumar, B., Rajiv, R., Srikumar, V., 2005, *Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework, High Performance Computing: Paradigm and Infrastructure*, Laurence Yang and Minyi Guo (editors), Wiley Press, New Jersey, USA, 2005.

Alchemi web page, 2002, http://sourceforge.net/projects/alchemi.

António, C.A.C., Douardo, N.M., 2002, Metal-forming optimization by inverse evolutionary search, *J. Mat. Proc. Techn.*, 121, 403-413.

Badrinarayanan, S., 1997, *Preform and die design problems in metalforming*, PhD. thesis, Cornell University.

Burczyński, T., Kuś, W., Beluch, W., John, A., Poteralski, A., Szczepanik, M., 2004, Zastosowanie algorytmów ewolucyjnych z MSC/Nastran i MSC/Marc w zagadnieniach optymalnego projektowania, *Biuletyn WAT*, nr 6 (622), (in Polish).

Burczyński, T., Kuś, W., Poteralski, A., Szczepanik, M., 2007, Global optimization using artificial immune systems and comparison with evolutionary algorithms, *Proc. 17-th Int. Conf. on Computer Methods in Mechanics CMM-2007*, Lódź-Spała, CD ROM.

Chung, J.S., Hwang, S.M., 1998, Application of a genetic algorithm to process optimal design in non-isothermal metal forming, *J. Mat. Proc. Techn.*, 80-81.

de Castro, L. N., Timmis, J., 2003, Artificial Immune Systems as a Novel Soft Computing Paradigm, *Soft Computing*, 7(8), 526-544.

de Castro, L.N., Von Zuben, F.J., 2002, Learning and Optimization Using the Clonal Selection Principle, *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems, 6:3, 239-251.

Kuś, W., Burczyński, T., 2006, Grid-based evolutionary optimization of structures, Parallel Processing and Applied Mathematics PPAM 2005, *Lecture Notes on Computational Science 3911*, Springer.

Kuś, W., 2006, Evolutionary optimization of forging anvils using grid based on Alchemi framework, *Proc. 2nd IEEE International Conference on e-Science and Grid Computing eScience 2006*, Amsterdam.

Kuś, W., Burczyński, T., 2007a, Biological inspired methods for shape optimization in grid environments, Proc. *7th World Congress on Structural and Multidisciplinary Optimization, WCSMO-7*, Seul, CD-ROM.

Kuś, W., Burczyński, T., 2007b, Applications of parallel artificial immune systems in shape optimization, *Proc. 17-th Int. Conf. on Computer Methods in Mechanics CMM-2007*, Lódź-Spała, CD ROM.

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

Users Guide, 2002, MSC.Marc.

Wierzchoń, S. T., 2001, *Artificial Immune Systems, theory and applications*, EXIT, Warsaw.

Zabaras, N., Bao, Y., Srikanth, A., Frazier W. G., 2000, A continuum Lagrangian sensitivity analysis for metal forming processes with applications to die design problems, *Int. J. Numer. Meth. Engng.*, 48.

Zhao, X., Zhao, G., Wang, G., Wang, T., 2002, Preform die shape design for uniformity of de-formation in forging based on preform sensitivity analysis, *J. Mat. Proc. Techn.*, 128, 25-32.

IMMUNOLOGICZNA I EWOLUCYJNA OPTYMALIZACJA KSZTAŁTU W PROCESIE KUCIA

Streszczenie

W artykule przedstawiono zastosowanie dwóch biologicznie inspirowanych metod obliczeniowych – algorytmów ewolucyjnych i sztucznych systemów immunologicznych w optymalizacji procesu kucia. Dwu-etapowy proces kucia swobodnego modelowany jest za pomocą metody elementów skończonych i rozwiązany za pomocą programu MSC Marc. W celu przyspieszenia obliczeń zagadnienie rozważane jest w środowisku gridowym. Przedstawiono przykład numeryczny ilustrujący skuteczność zastosowanych inteligentnych technik optymalizacji.

Submitted: April 14, 2008
Submitted in a revised form: July 16, 2008
Accepted: October 5, 2008