Computing and Informatics, Vol. 28, 2009, 209–223

APPLICATION OF A HIERARCHICAL CHROMOSOME BASED GENETIC ALGORITHM TO THE PROBLEM OF FINDING OPTIMAL INITIAL MESHES FOR THE SELF-ADAPTIVE HP-FEM

Anna Paszyńska

Faculty of Physics, Astronomy and Applied Computer Science
Jagiellonian University, Cracow, Poland
e-mail: anna.paszynska@uj.edu.pl

Maciej Paszyński

Department of Computer Science
AGH University of Science and Technology, Cracow, Poland
e-mail: paszynsk@agh.edu.pl

Revised manuscript received 9 December 2008

Abstract. The paper presents an algorithm for finding the optimal initial mesh for the self-adaptive *hp* Finite Element Method (*hp*-FEM) calculations. We propose the application of the hierarchical chromosome based genetic algorithm for optimal selection of the initial mesh. The selection of the optimal initial mesh will optimize the convergence rate of the numerical error of the solution over the sequence of meshes generated by the self-adaptive *hp*-FEM. This is especially true in the case when material data are selected as a result of some stochastic algorithm and it is not possible to design optimal initial mesh by hand. The algorithm has been tested on the non-stationary mass transport problem modeling phase transition phenomenon.

Keywords: Finite element method, hp adaptivity, genetic algorithms

1 INTRODUCTION

The paper focuses on the problem of finding optimal initial mesh for the self-adaptive *hp*-FEM (*hp* Finite Element Method) [1, 7] calculations. Here h stands for the element dimension, p stands for the polynomial order of approximation. The problem of the mass transport during the simulation of the austenite-ferrite phase transformation with moving boundary interface [10] is considered. The model parameters are provided by the Cellular Automata (CA) simulation [10]. The simulation involves solving the non-stationary mass transfer problem on a sequence of meshes generated by the self-adaptive *hp*-FEM. The solution at a current time step depends on the solution from the previous time step as well as on the material data obtained from CA simulation. The self-adaptive *hp*-FEM starts from an arbitrary initial mesh, selected by the user. It generates a sequence of meshes delivering convergence of the accuracy of the numerical solution with respect to the mesh size. The sequence of meshes is obtained by performing multiple h, p or hp refinements. The h refinement consists in breaking selected finite element into smaller son elements, and the p refinement consists in adjusting polynomial order of approximation on selected element edges and interiors. The convergence rate of the solution process over a sequence of meshes depends on the quality of the selected initial mesh. The regular initial mesh is often insufficient – it doesn't provide good convergence. The best convergence is obtained when the initial mesh fits material data. In other words, the coefficients of the partial differential equation being solved, e.g. the diffusion coefficient in our example, usually are non regular functions, since they result from measurements or from execution of some stochastic algorithms. The FEM approximation is constructed over finite element mesh, and the initial approximation must roughly capture singularities (local extremum) of material data functions, to provide guidelines for the adaptation process.

Designing such an optimal initial mesh by hand is very difficult, and sometimes even not possible. The problem of finding the optimal initial mesh is similar to the shape optimization problem. It can be solved by some deterministic algorithms, analyzing the material data properties, however genetic algorithms [2] are widely respected as a proper tool for solving shape optimization problems. Thus, we propose the application of the genetic algorithm for selection of the optimal initial mesh. We will use the hierarchical chromosome based genetic algorithm (HCBGA). The algorithm works on tree based data structures [1]. It allows to code artifact with different number of components (in our case - meshes with different number of elements). The hierarchical genetic operators utilized by the genetic algorithm create offspring that differs in size and shape from their parents. The described genetic algorithm can solve the design problem even if the optimal number of artifact components is not known. The paper is organized as follows. In the following section the non-stationary mass transport driven by CA simulation is described. The results obtained by the self-adaptive *hp*-FEM code for different initial meshes are presented in the next section. Finally, the HCBGA applied for the selection of the optimal initial mesh is presented. Some conclusions are summarized in the last section.

2 NON-STATIONARY MASS TRANSPORT PROBLEMS

The two-dimensional self-adaptive *hp*-FEM code [1] generates a sequence of finite element meshes, improving the accuracy of the numerical problem under consideration. The two-dimensional code was extended to support non-stationary mass transport problems [7]. The problem of mass transport during the simulation of the austenite-ferrite phase transformation with moving boundary interface [10] is considered here. There are many phase transition models considered in the literature, some of them focus on the heat transport problem only [13], the mass transport only [4], or combined heat and mass transport problems [14, 6]. The interface between the fluid and solid phases can be captured by using the front tracking technique by decoupling solid and fluid phases and introducing some interface conditions [13, 4, 14], or by utilizing the Cellular Automata (CA) technique [6]. In our model, the material data, such as the heat transfer coefficients, density and specific heat are computed on the basis of the CA simulation results [10]. This approach provides the possibility to replicate phase transformation during solidification and to take into account the progressing movement of the crystallization front. The simulation consists in solving the non-stationary mass transfer problems on a sequence of meshes generated by the self-adaptive *hp*-FEM. The considered strong form of the non-stationary mass transport problem is as follows: Find the concentration distribution scalar field $c \in C^2(\Omega)$ satisfying

$$\begin{cases} \frac{\partial c}{\partial t} - \nabla \cdot (D \nabla c) = 0 & \text{on } \Omega \times I \\ \mathbf{n} \cdot \nabla c = 0 & \text{on } \partial\Omega \times I \\ c(\mathbf{x}, 0) = c_0 & \text{on } \Omega. \end{cases} \tag{1}$$

The weak form of the non-stationary mass transfer problem is as follows. Find the concentration scalar field $c \in V$, where $V = \{v \in H_1(\Omega) : v = 0 \text{ on } \Gamma_D\}$, satisfying

$$(\dot{c}, v)_\Omega + \int_\Omega D \nabla c \circ \nabla v d\Omega = 0 \qquad \forall v \in V \tag{2}$$

$$(c(0), v)_\Omega = (c_0, v)_\Omega \qquad \forall v \in V. \tag{3}$$

The values of the diffusion coefficient D are defined by utilizing the general formula

$$D = D_1 f_s + (\mathbf{1} - f_s) D_2 \tag{4}$$

where f_s is the fraction of the solid phase given by the cellular automata (CA) [10] over the entire domain. Exemplary CA simulation results are presented in Figure 1 a). The CA computations are performed concurrently with respect to the FEM computations. For the mass transfer problems, FE – discretization in time gives the following matrix system:

$$M\dot{u} + Ku = f. \tag{5}$$

Applying the trapezoidal rule for the time discretization we obtain

$$(M + \alpha\delta K)u^{k+1} = [M - (1-\alpha)\delta K]u^k + \delta f^k \tag{6}$$

where M is the mass matrix, δ is the time step, $\alpha \in [0,1]$ gives different time integration schemes.

3 RESULTS FOR DIFFERENT INITIAL MESHES

The self-adaptive *hp*-FEM is utilized to solve the problem at every time step. The solution at a new time step depends on the solution from the previous time step as well as on the material data obtained from the cellular automata (CA) simulation. The *hp*-FEM generates a sequence of meshes delivering exponential convergence of the numerical error with respect to the mesh size.

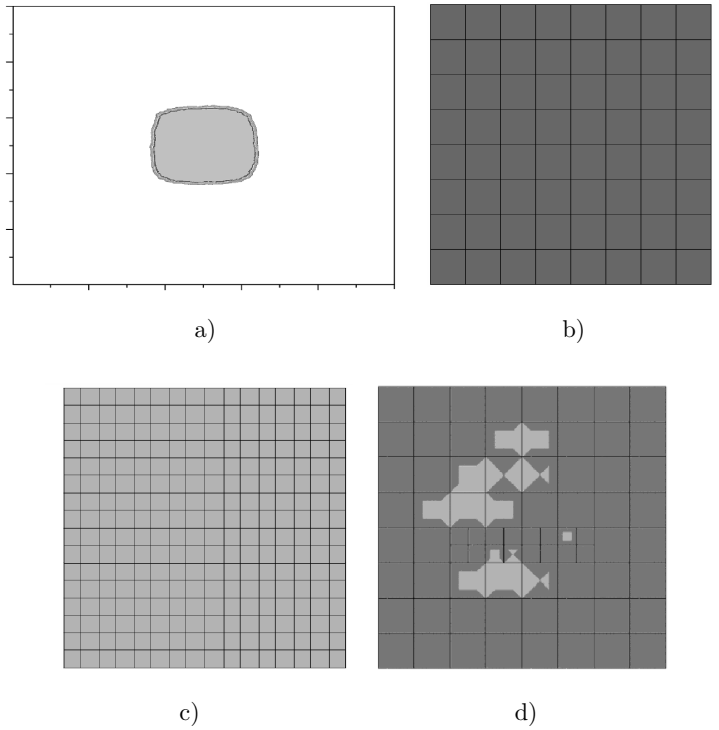


Fig. 1. a) Exemplary fraction of the solid phase obtained by CA simulation described in [7]; b) c) d) Coarse, fine and optimal meshes for the first iteration. Different colors denote different polynomial orders of approximation on element edges and interiors

The *hp*-FEM starts from an initial mesh. The exemplary initial mesh is presented in Figure 1 b). This mesh is called the coarse mesh. The *hp*-FEM executes

global *hp* refinement on the initial mesh to obtain the so-called fine mesh, presented in Figure 1 c). It is done by breaking each finite element into 4 new elements, and increasing polynomial order of approximation by one, on all elements edges and interiors. The variational problem under the consideration is solved on both the coarse and fine meshes, to obtain the coarse u_{hp} and the fine $u_{h/2,p+1}$ mesh solutions. The following strategy is utilized to obtain the so-called optimal mesh, presented in Figure 1 d). Various refinement strategies are considered for each finite element from the coarse mesh. An element can be either h refined (broken in horizontal, or vertical or both directions into new smaller finite elements), or p refined (polynomial orders of approximation are updated on selected element edges or in the element interior) or *hp* refined (both h and p refinements are performed at the same time). The corresponding error decrease rate is computed for each considered strategy:

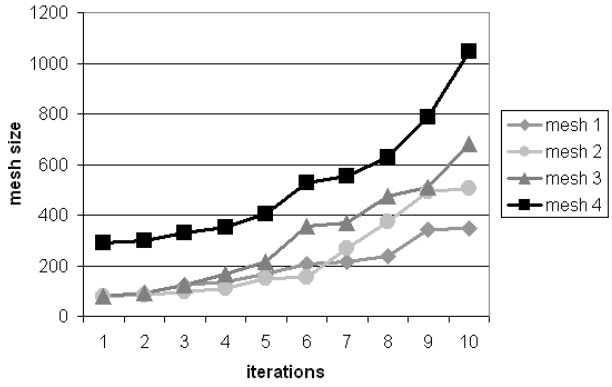
$$rate = \frac{\|u_{h/2p+1} - u_{hp}\|_{K,1} - \|u_{h/2p+1} - w\|_{K,1}}{\Delta nrdof} \tag{7}$$

where w denotes the local solution corresponding to a considered refinement strategy, obtained by utilizing the projection based interpolation technique [1], and $\Delta nrdof = nrdof_w - nrdof_{hp}$ denotes the number of degrees of freedom required to execute the considered refinement. The difference between the coarse and fine mesh solutions is measured in the energy norm (H^1 Sobolev space norm) restricted over the considered element K. For a given finite element from the coarse mesh, the refinement strategy providing maximum error decrease rate is selected. This is because the maximum error decrease expressed in the nominator in (7) is expected, while the minimum investment of the new degrees of freedom expressed in the denominator of (7) is wanted. The optimal mesh obtained by performing selected h or p or *hp* refinements becomes the coarse mesh for the next iteration, and the entire process is repeated. The stopping criterion is to obtain the required accuracy of the coarse mesh solution

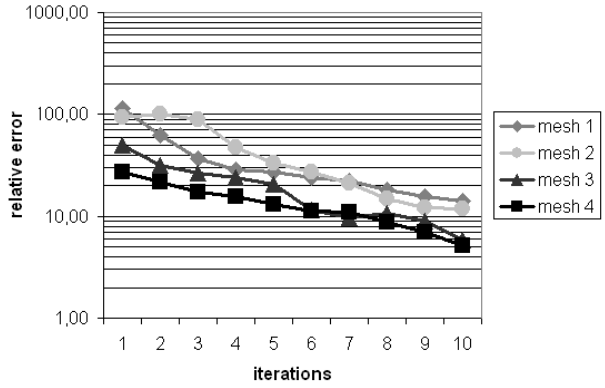
$$\frac{\|u_{h/2p+1} - u_{hp}\|_{\Omega,1}}{\|u_{h/2p+1}\|_{\Omega,1}}. \tag{8}$$

Let us discuss the computational cost related to the generation of the optimal mesh from an initial mesh. We focus on four initial meshes presented in Figure 3. The first regular initial mesh, presented on panel a), has been created manually. The three remaining meshes, presented on panels b), c) and d) in Figure 3, have been obtained by executing the HCBGA algorithm. The histories of convergence of the self-adaptive *hp*-FEM (mesh sizes and corresponding relative errors) over the initial meshes from Figure 3 are presented in Figure 2. The relative error presented on panel b) is defined according to (8) and multiplied by 100 %. The resulting optimal meshes, obtained after 10 iterations of the *hp*-FEM, are presented in Figure 4.

The computational cost for the whole sequence of ten meshes involves ten calls of direct solver routine for the coarse and fine mesh problems, and ten generations of consecutive optimal meshes. The number of degrees of freedom over a single *hp* finite element with polynomial orders of approximation (p_1, p_2) is $(p_1+1)(p_2+1) =$



a)



b)

Fig. 2. Mesh size and relative error for a sequence of 10 meshes generated from the initial meshes presented in Figure 3

$(p_1p_2+p_1+p_2+1)$, compare [1]. The fine mesh is obtained by performing global *hp* refinement - each finite element is broken into four new elements and the polynomial order of approximation is uniformly raised by one. The number of degrees of freedom over four new elements is of the order of $4(p_1+2)(p_2+2) = 4(p_1p_2+8p_1+8p_2+16)$. Thus, the number of degrees of freedom (d. o. f.) over the refined element is between 4 up to 10 times larger than original element size $N_{fine} = \alpha N_{coarse}$, where $\alpha \in [4, 10]$ depends on the polynomial orders of approximation. We can estimate the computational cost of a single iteration as

$$N^2_{coarse} \log N_{coarse} + N^2_{fine} \log N_{fine} = N^2_{coarse} \log N_{coarse} + (\alpha N_{coarse})^2 \log(\alpha N_{coarse}) \tag{9}$$

where $O(N^2 \log N)$ is the estimation of the solver computational cost. We can

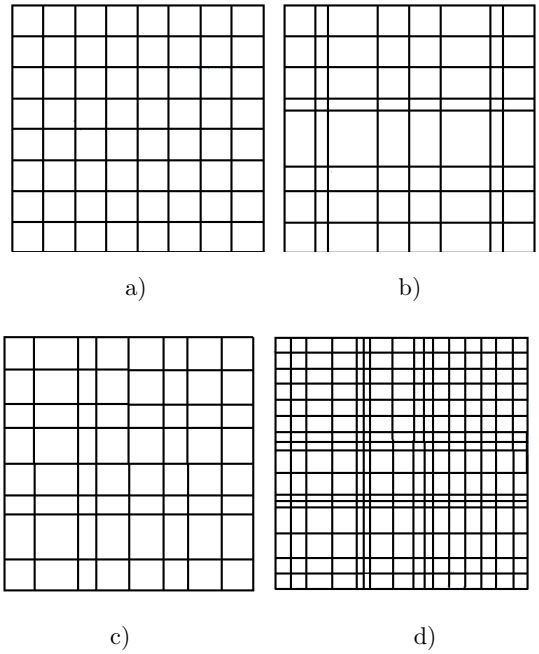


Fig. 3. Four exemplary initial meshes utilized in the experiment reported in Figure 2

estimate total computational cost for all iterations for a given time step by assuming the average $\alpha = 6$ and utilizing Equation (9). The total computational cost of generation the optimal mesh from the regular mesh is

$$\begin{aligned}
&81^2 \log 81 + 6^2 81_2 \log(81 \times 6) + 81_2 \log 91 + 6^2 93^2 \log(93 \times 6) \\
&+ 125^2 \log 125 + 6^2 125^2 \log(125 \times 6) + 136_2 \log 136 + 6^2 136^2 \log(136 \times 6) \\
&+ 167^2 \log 167 + 6^2 167^2 \log(167 \times 6) + 205_2 \log 205 + 6^2 205^2 \log(205 \times 6) \\
&+ 216^2 \log 216 + 6^2 216^2 \log(216 \times 6) + 236_2 \log 81 + 6^2 236^2 \log(236 \times 6) \\
&+ 343^2 \log 343 + 6^2 343^2 \log(343 \times 6) + 349_2 \log 349 + 6^2 349^2 \log(349 \times 6) \\
&= 55\,430\,825
\end{aligned} \tag{10}$$

while the total computational cost related to the initial mesh estimation is just $81^2 \log 81 + 6^2 81^2 \log(81 \times 6)$. It implies that a single estimation of the initial mesh is more than $55\,430\,825/663\,405 = 83$ times less expensive than performing 10 iterations of the algorithm. These estimations don't include computationally expensive algorithm making decisions about optimal refinements and generating the optimal mesh. These additional expensive components are not executed during a single initial mesh evaluation, but only during the process of the optimal mesh generation. Moreover, the presented estimation includes only 10 iterations. In order to obtain the solution with 1 % relative error it is necessary to perform more than 20 iterations, and 0.1 % accuracy requires more than 40 iterations. Thus, the estimation of

the initial mesh is hundreds times less expensive than the entire generation of the high accuracy optimal mesh. In three dimensions this difference is several orders of magnitude higher.

		Relative error over the mesh			
Mesh type	Size	Simulation 1	Simulation 2	Simulation 3	Final accuracy
2	81	92.48 %	99.98 %	115 %	12 %
3	81	49.99 %	49.98 %	49.98 %	6 %
4	289	25.03 %	27.23 %	29.53 %	5 %

Table 1. Summary of best individuals

Thus, we have performed the following experiments. We have executed the 20 steps of the HCBGA algorithm three times from initial population with 20 individuals. We utilized the relative error over the individual as the estimator of the individual's quality. In the first sequence of experiments, the individuals from the initial populations have fixed four partitions in horizontal and vertical directions, and some additional random partitions. The first row in Table 1 presents the best individuals obtained after these simulations. The two individuals presented in the first row, resulting from the first and the second simulations, are visually undistinguishable, and one of them is the initial mesh presented on panel b) in Figure 3. These individuals provide relative error estimation of the level of 90–100 %, and the final accuracy on the level of 12 %, which is not satisfactory.

With the lesson learnt after the first three executions, we have changed the locations of horizontal and vertical partitions on the initial populations, still with some additional random partitions. We have executed the three experiments with 20 steps of the HCBGA from the new initial population again with 20 individuals. The second row in Table 1 presents the best individuals obtained after these simulations. All these three best individuals presented in the second row are visually undistinguishable, and one of them is the initial mesh presented on panel c) in Figure 3. These individuals provide relative error estimation of the level of 50 %, with the final accuracy on the level of 6 %, which is a very satisfactory result.

To check the quality of the obtained individuals, we have increased the number of horizontal and vertical partitions in the initial population from four to eight, and executed three new simulations with 20 steps of the HCBGA from the new initial populations with 20 individuals. The third row in Table 1 presents the best individuals obtained after these simulations. All these three best individuals presented in the third row are again visually undistinguishable, and one of them is the initial mesh presented on panel d) in Figure 3. These individuals provide relative error estimation of the level of 25–30 %, with the final accuracy on the level of 5 %. This implies that the best individuals obtained in the previous sequence of experiments are high quality meshes, since they provide similar accuracy after 10 iterations of the *hp*-FEM, with lower computational cost, since the meshes utilized in the previous experiments were much smaller. It follows from Figure 2 that the

history of convergence depends on the quality of selected initial mesh. The regular mesh from panel a) in Figure 3 doesn't provide the best convergence. The best convergence is obtained when the initial mesh fits material data generated by the CA algorithm. We also conclude that the numerical error over the initial mesh can be utilized as the estimator for prediction of the quality of convergence on the sequence of meshes obtained from the considered initial mesh. Smaller numerical error over the initial mesh implies better accuracy obtained after 10 iterations. The numerical error over the initial mesh from panel c) is about three times smaller than the numerical error over the regular initial mesh, and the accuracy obtained after 10 iterations from the initial mesh from panel c) is also about three times smaller than the accuracy obtained from the regular initial mesh. The initial mesh from panel d) provides somewhat better accuracy after 10 iterations. However, the size of the initial mesh is much larger than sizes of all other presented initial meshes. The optimal mesh obtained after 10 iterations of the *hp*-FEM from the initial mesh from panel d) is also much larger than the optimal meshes obtained from other initial meshes.

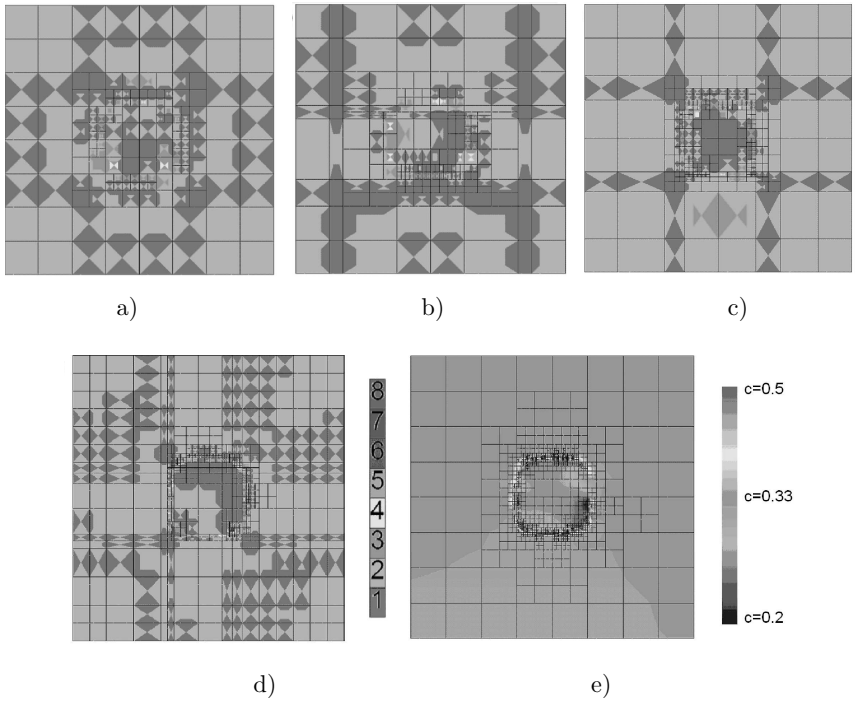


Fig. 4. Four optimal meshes obtained after 10 iterations from the first, second, third or fourth initial mesh, respectively. Different colors denote different polynomial orders of approximation on finite element edges and interiors. The last panel presents resulting concentration field distribution computed on one of the optimal meshes

We can make the following conclusions. It is necessary to start computations from optimal initial mesh, to obtain fastest convergence on the generated sequence of meshes. Such the optimal initial mesh must fit material data as well as predict location of singularities of the numerical solution. We propose the application of the genetic algorithm for optimal selection of the initial mesh. The selection of the optimal initial mesh will optimize the convergence rate of the numerical error of the solution over the sequence of meshes generated by the self-adaptive *hp*-FEM. This is especially true in the case when material data are selected as a result of some stochastic algorithm and it is not possible to design optimal initial mesh by hand.

4 APPLICATION OF HIERARCHICAL CHROMOSOME BASED GENETIC ALGORITHM FOR INITIAL MESH SELECTION

We propose the hierarchical chromosome based genetic algorithm as the tool for finding the optimal initial mesh. This algorithm can be applied to solve several design and engineering problems. The presented genetic algorithm works on tree based structures (called hierarchical chromosomes). The term "genetic programming" [5] is sometimes used for such kind of algorithms. However, the HCBGA can be rather called the genetic algorithm [3] to underline the most important differences between those two types of the algorithms. First of all, there are two disjoint spaces defined for the genetic algorithm: the search space (the phenotype space) and the solution space (the genotype space), while there is only one space (space of programs represented as parse trees) defined for the genetic programming. Secondly, in the genetic algorithm the evaluation process of an individual consists in decoding the genotype into the phenotype and evaluating it according to the fitness function, while in the genetic programming the same process consists in running the program. HCBGA can be described according to the scheme (μ, λ), where $\mu = \lambda$ [11]. The reproduction in the algorithm is done by fitness proportional selection and succession is done by elite selection, where $\sharp Elite = \mu$. Because of the hierarchical representation of the phenotypes, the hierarchical mutation and the hierarchical crossover have to be used, in spite of the traditional genetic operators. The hierarchical genetic operators are described later in this section. The hierarchical chromosome, which represents an object in our genetic algorithm, codes the structure and the meaning of an object from the point of view of the designer. Figure 5 presents an exemplary mesh and the hierarchical chromosome coding the mesh. Our exemplary mesh (denoted by $S1$) is described by two components: X and Y. Component X consists of two subcomponents: PX_1 and PX_2. Component Y consists of three subcomponents: PY_1, PY_2 and PY_3. Each subcomponent is described by two parameters: the left and the right limit of the interval, denoted by x_i. These values are coded as the binary string $(x_i = b_0 b_1 \dots b_m)$.

The whole process of finding the optimal initial mesh can be described as follows. At the beginning the initial population is created. We assume that each mesh from the initial population has the same number of components on the second level.

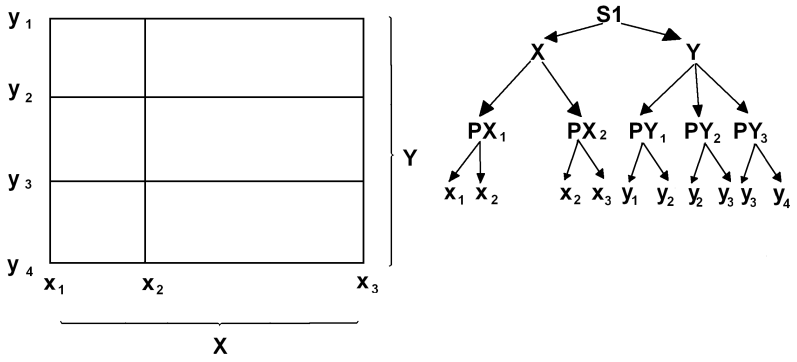


Fig. 5. An exemplary mesh and the hierarchical chromosome coding it

However, the overall size and shape of the hierarchical chromosomes coding initial meshes can be different. After generation, the new initial population should be evaluated. The computational problem is solved for each individual representing a single initial mesh, and the global *hp* refinememnt is performed. Then, the problem is solved again, and the relative error estimator for the individual is computed. After that the new offsprings are created (by using the hierarchical crossover and the hierarchical mutation) and added to the population. Then the whole process of evaluation and generation of offsprings is repeated. The algorithm stops, if the optimal initial mesh is found. The HCBGA allows for generating offsprings from parents of different size and shape (in our case – from initial meshes with different number of components), thank to the hierarchical representation of objects and hierarchical genetic operators. It also allows for creating offsprings with different number of components than their parents. As was mentioned above, the hierarchical mutation and hierarchical crossover are used to generate offsprings from artifacts coded as hierarchical chromosomes. The hierarchical mutation of group of alleles (coded components) allows for adding or removing the whole component from the genotype. This genetic operator is responsible for changing of the genotypes size. The algorithm for mutation of the group of alleles can be described as follows. First randomly choose the group of alleles (a component) and decide randomly, whether it should be remover or split. If removing was chosen, then remove the whole group of genes from the chromosome. If splitting was chosen then randomly choose the clipping coordinate, calculate the genes values for both components and add new group of alleles to the hierarchical chromosome. Figure 6 a) presents a phenotype before mutation. Figure 6 b) presents the phenotype after removing the component. Figure 6 c) presents the phenotype after splitting of the component. Figure 6 d) presents hierarchical chromosomes corresponding to the mesh from Figure 6 c). To sum up, the two kinds of the hierarchical mutation can be described as follows: removing of the component consists in removing of the suitable subtree from the

hierarchical chromosome, while splitting of the component consist in adding new subtrees to the corresponding node of the hierarchical chromosome.

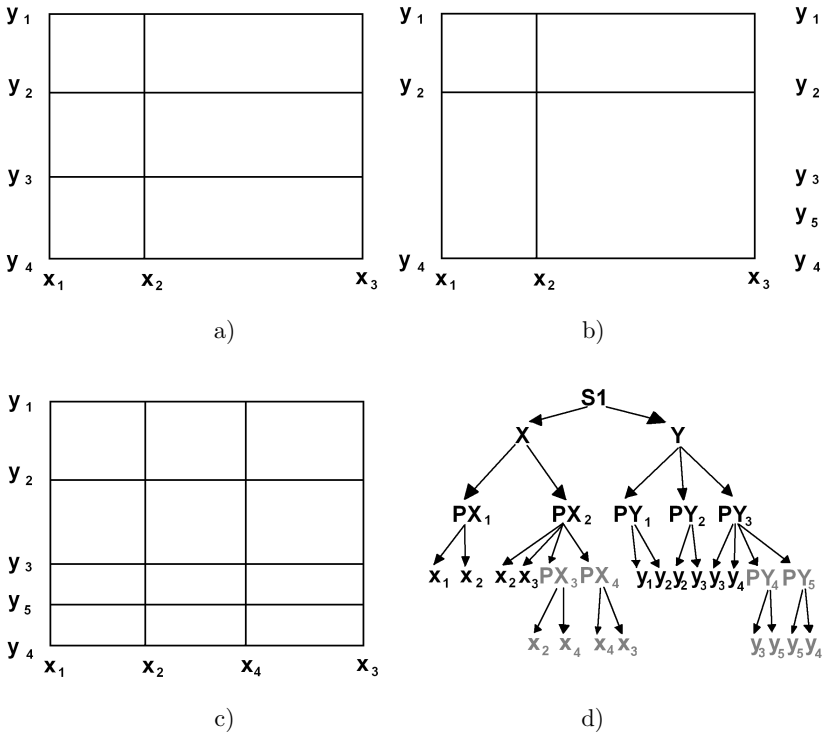


Fig. 6. a) phenotype before mutation, b) phenotype after mutation – removing of the component, c) phenotype after mutation – splitting of the component, d) hierarchical chromosomes corresponding to the mesh from panel c)

The next genetic operator used is the hierarchical crossover [8]. The hierarchical crossover is the two stage process, which consist in finding the suitable crossover point P in the parent individuals (in our case the number of the component from the second level of the hierarchy tree) and applying the crossover to the offspring generation. The second step consists in copying all the components located left from P and their subtrees to the children being generated – from the first parent to the first child, from the second parent to the second child. After that the components of number P with their subtrees and all primitives located right from P and their sub-trees are copied to the children – from the first parent to the second child, from the second parent to the first child. Thank to such a definition of the crossover, the generated children can have different size and shape than their parents. Figure 7 presents hierarchical chromosomes of the parents, and their offspring, created by crossing at crossover point P. The described hierarchical chromosome based genetic

algorithm was used to find optimal initial two-dimensional mesh, but it can be easily extended to three-dimensional meshes. Based on [12], according to the theorem proved in [9], the Markov chain modeling the algorithm is ergodic and the presented HCBGA has asymptotic guarantee of the success in the probabilistic sense.

5 CONCLUSIONS

The paper presented an application of the hierarchical chromosome-based genetic algorithm for finding the optimal initial mesh for the self-adaptive *hp*-FEM calculations. The application of the genetic algorithm for the initial mesh optimization was motivated by the fact that the ratio of convergence of the self-adaptive *hp*-FEM depends on the quality of the initial mesh, and it is reasonable to invest some computational time for the initial mesh optimization. The presented genetic algorithm can be simply extended to three-dimensional meshes.

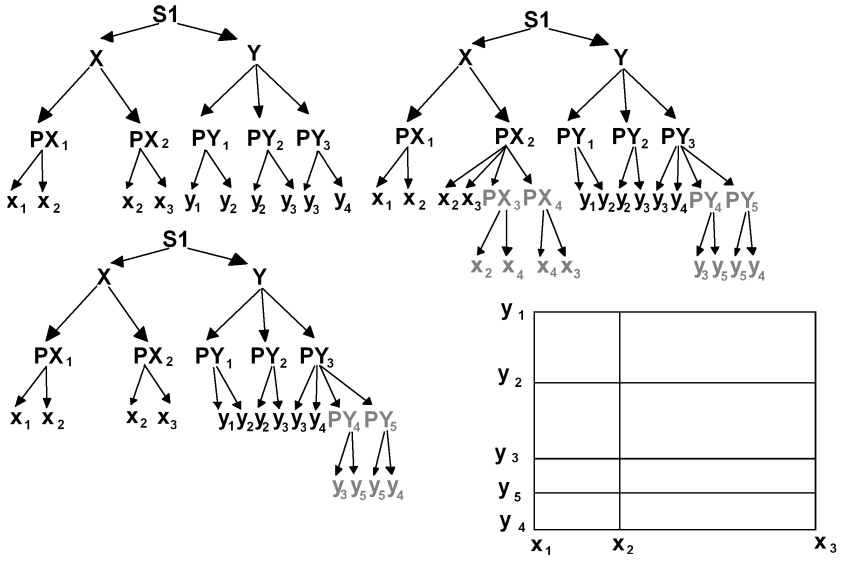


Fig. 7. Hierarchical chromosomes of the parents, an offspring created from the parents, and the hierarchical chromosome coding it

Acknowledgement

The presented work has been supported by the MNiSW grant No. 3 T08B 055 29.

REFERENCES

[1] Demkowicz, L.: Computing with hp-Adaptive Finite Elements: One and Two Dimensional Elliptic and Maxwell Problems. Chapmann & Hall/Crc Applied Mathematics & Nonlinear Science, Vol. 1, 2006.

[2] Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA, Addison-Wesley, 1989.

[3] Holland, J.: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, Massachusetts, 1992.

[4] Jacot, A.—Rappaz, M.: A Pseudo-Front Tracking Technique for the Modelling of Solidification Microstructures in Multi-Component Alloys. Acta Materialia, Vol. 50, 2002, No. 8.

[5] Koza, J. R.: Genetic Programming: On the Programming of Computers by Natural Selection. MIT Press, Cambridge, MA, USA, 1992.

[6] Liu, Y.—Xu, Q.—Liu, B.: A Modified Cellular Automaton Method for the Modeling of the Dendritic Morphology of Binary Alloys. Tsinghua Science & Technology, Vol. 11, 2006, No. 8.

[7] Matuszyk, P.—Paszyński, M.: Fully Automatic 2D hp-Adaptive Finite Element Method for Non-Stationary Heat Transfer Problems. Proceedings of COMPLAS, Barcelona, September 2007.

[8] Paszyńska, A.: An Extension of Voses Markov Chain Model for Genetic Algorithms. Proceedings of Genetic and Evolutionary Computation Conference, Washington, D. C., USA, 2005.

[9] Paszyńska, A.—Jabloński, M.—Grabska, E.: Markov Chain Model for Tree-Based Genetic Algorithm Used in Computer Aided Design. Proceedings of 5^{th} International Conference on Computer Recognition Systems, Wroclaw, 2007.

[10] Paszyński, M.—Matuszyk, P.—Gawad, J.—Madej, L.: Phase Transition Modeling with CA Merged with hp-Adaptive FEM for the Heat and Mass Transport Problems. XV. Conference Computer Methods in Material Science, Korbielow, 2008 (in Polish).

[11] Schaefer, R.: Podstawy genetycznej optymalizacji globalnej. Wydawnictwo Uniwersytetu Jagiellonskiego, Kraków, 2002 (in Polish).

[12] Vose, M. D.: The Simple Genetic Algorithm. MIT Press, 1999.

[13] Zhao, P.—Heinrich, J. C.: Front-Tracking Finite Element Method for Dendritic Solidification. Journal of Computational Physics, 2001, p. 173.

[14] Zhu, M. F.—Stefanescu, D. M.: Virtual Front Tracking Model for the Quantitative Modeling of Dendritic Growth in Solidification of Alloys. Acta Materialia, Vol. 55, 2007, No. 5.

Anna Paszyńska received her Ph. D. (2007) in computer science from the Institute of Fundamental Technological Research of Polish Academy of Sciences in Warsaw, Poland. She currently works as a Postdoctoral Fellow at the Jagiellonian University in Cracow, Poland. Her research interests include evolutionary algorithms, graph grammar and computer aided design.

Maciej Paszyński received his Ph. D. (2003) in mathematics with applications to computer science from the Jagiellonian University of Cracow in Poland. He holds a position as an Assistant Professor in the Department of Computer Science at AGH University of Science and Technology of Cracow. His research interests include adaptive hp-FEM, parallel algorithms and computational science.