Computer Assisted Mechanics and Engineering Sciences, **17**: 13–26, 2010.

A distributed implementation of parallel genetic algorithm for slope stability evaluation

Piotr E. Srokosz and Chi Tran
Faculty of Technical Sciences
University of Warmia and Mazury in Olsztyn, Poland
e-mails: psrok@uwm.edu.pl, tran.chi@uwm.edu.pl

(Received in the final form October 13, 2010)

Parallel processing, the method of considering many small tasks to solve one large problem, has emerged as a key enabling technology in modern computing. Parallel computers can be simply classified into shared memory systems and distributed memory systems. The shared memory computers have a global memory attached to a number of processors enabling several processors to work concurrently on different parts of the same computation. A different approach towards building large parallel computers is to connect several processors via a network. Each processor has its own local memory. The cost of building these computers increases with the number of processors. The distributed memory multiprocessor systems are scalable over a wider range than the shared memory computers. There are many intermediate computer architectures, each with its distinct programming model. Common between them is the notion of message passing. In all parallel processing, data must be exchanged between cooperating tasks. Several research groups have developed software packages such as **P**arallel **V**irtual **M**achine (**PVM**), the **M**essage **P**assing **I**nterface (**MPI**), and others. In this paper, hardware implementation of parallel information processing is introduced by application of a multicellular computer idea, in which working cells were composed of general purpose one-chip microcomputers. The influence of the cellular computer's structure size on quality and efficiency of calculations was analyzed. The optimal structure consisted of 4x4 cells which guaranteed achieving satisfactory recurrence of results for an assumed set of working parameters. This paper presents an idea and the results of trial computations regarding the problem of slope stability evaluation by variational calculus assisted by genetic algorithm.

Keywords: Hardware implementation, slope stability, variational calculus, parallel genetic algorithm

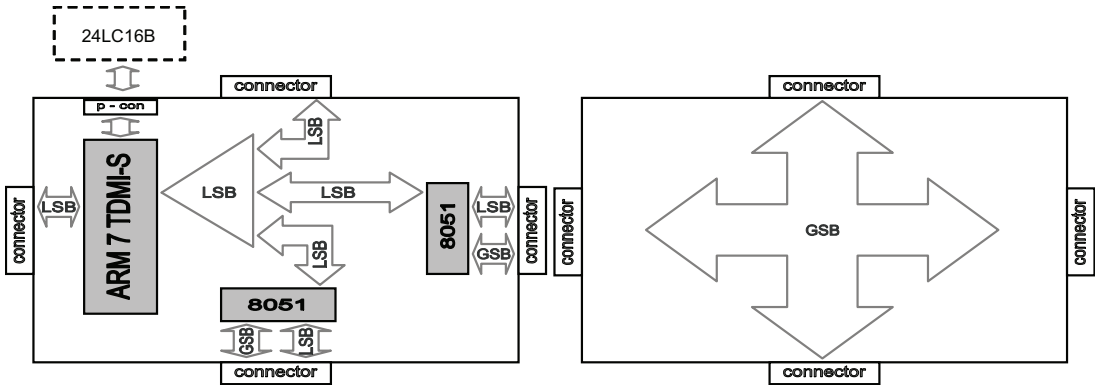
1. Cellular computer idea

The majority of all computation programs coded in C-language can be easily recompiled and executed on one-chip microcomputers (including floating-point operations). The advantages of such an idea are small dimensions (7×7 mm **A**dvanced **R**ISC **M**achine **ARM** chip, 20×52 mm dipARM unit, 65×100 mm computational cell) very low cost (about 20 EUR for the dipARM unit and 30 EUR for the single cell) and extremely low power consumption (less than 1W per one cell) which should come into prominence for the sake of environment protection. Taking into account the number of cells realizing parallel calculations, the idea of a simple external computational unit has originated. The unit called a cellular computer (cellputer) consists of a main calculation processor and two buffering microcontrollers. The main calculation unit is based on NXP LPC2106 microcomputer with ARM7TDMI-S processor. The ARM7TDMI-S is a high-performance 32-bit CPU, based on RISC principles with pipeline technique application, which assures continuous operation of processing and memory systems. The LPC2106 microcomputer incorporates a 128kB reprogrammable flash memory, 64kB internal static RAM supporting 32-bit accesses, vectored interrupt controller, inter-integrated circuit interface (I2C), serial peripheral interface, two universal asynchronous receiver and transmitter interfaces, pulse width modulation unit with six outputs, two timers with eight capture/compare channels, real-time clock, watchdog timer (WT), 32 general purpose input/output

pins (I/O), on-chip crystal oscillator and on-chip phase-locked loop enabling timing CPU up to 60MHz. The device is supplied in twofold manner: the CPU needs 1.8VDC and the I/O lines need 3.3VDC (but are tolerant to the TTL signals with up to 5VDC levels). In-system and in-application programming (ISP, IAP) capabilities enable code uploading and updating without breaking any connections with external peripherals, i.e. flash memory can be filled with new instructions filed directly in the cellputer's board. To avoid jamming of the executed program, the watchdog timer is enabled (with reload time about 2 minutes) to guard proper flowing of the code. The external memory is needful for storing working data necessary for continuation of calculations after restart action caused by WT.

1.1. Data buffering

The two buffers applied in the cellputer's board are Atmel AT89C2051 microcontrollers with 8051 architecture cores. This chip is a low-cost, 8-bit, high-performance CMOS microcomputer with a 2kB reprogrammable flash memory, 128B of internal RAM, a five vector two-level interrupt architecture, 15 I/O lines, two 16-bit timers/counters, a full duplex serial port and precision analog comparator. The maximum CPU rate can reach 2MHz when the built-in oscillator is set to working at frequency of 24MHz. On devices installed on the cellputer's board the first internal timer is enabled and guards proper flowing of the code (avoids jamming in the case of loss of synchronization in transmission among the buffers, ARM-CPU and processor in the supervision unit). AT89C2051 is supplied by 5VDC and is compatible with the industry-standard MCS-51 instruction set, which facilitates the creation of an executable code using many of generally accessible compiler-tools. Clear programming algorithm enables one to design and build home-made software and hardware programmers. The cellputer's board is shown in Fig. 1. The scheme of a complete cellputer system is shown in Fig. 2.



GSB – global serial bus (connection among cellputers and supervision unit)
LSB – local serial bus (connection between ARM-CPU and buffers)

Fig. 1. Cellputer's board scheme

1.2. The functional structure of the cellputer

All cellputer's boards are supplied from the global serial bus, which also assures free exchange of data among all components of the system. A view of cellputer boards is shown in Fig. 3.

Besides calculation boards, a few additional modules are engaged into the system's work, such as power supply and supervision units. They monitor the whole system by particularly written WatchDog procedure which in special cases (e.g. end of calculations, break due to appearance of intolerant errors) and critical exceptions (e.g. accidental interruptions and external disturbances caused by electric and electromagnetic noise) sends information via GSM network using an installed

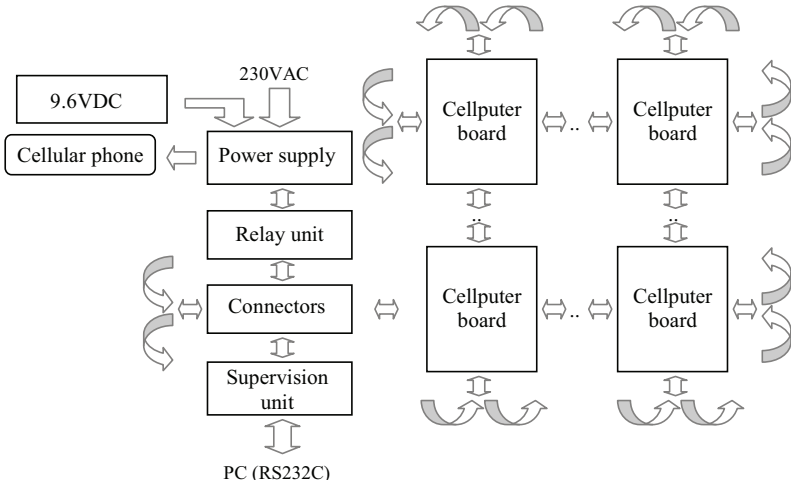


Fig. 2. Scheme of complete system

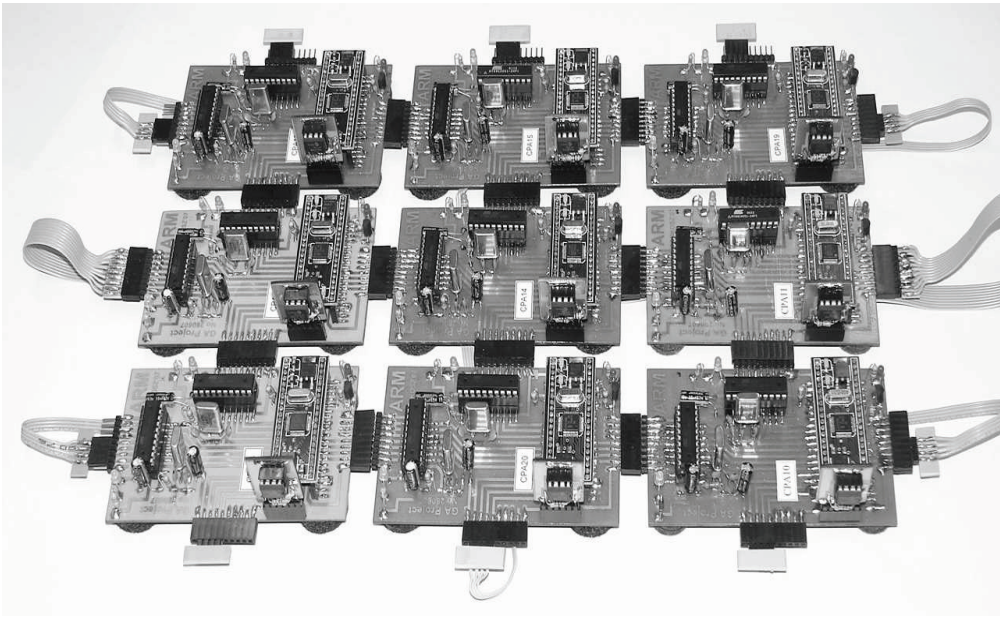


Fig. 3. View of the cellputer boards

cellular phone. Further, the power supply module inspects programmed restarts of the system and secures emergency boost should external power be absent. The supervision module provides communication with a personal computer via RS232C standard protocol. Each unit of this additional equipment is managed by its own AT89C2051 microprocessor.

2. Backgrounds of slope stability computation

The problem of reliable slope stability evaluation remains one of the most difficult issues in modern soil mechanics. In spite of sophisticated and highly specialized computer applications commonly exploited in engineering practice (based on classical slices methods as well as on advanced FEM techniques with complex constitutive models), it is a very difficult matter to avoid the situations in which doubts are appearing and verification of performed analyses using alternative approaches is highly recommended. The idea of the variational calculus technique of limit equilibrium analysis applied to evaluation of slope's safety factor was developed in 1977 by Baker and Garber [1].

The theoretical background presented on the 9th International Conference on Soil Mechanics and Foundation Engineering in Tokyo became the starting point for many subsequent modifications of variational limit equilibrium solutions (e.g., Chen and Liu [2], Baker [3] – application of nonlinear Coulomb's criterion, Leshchinsky and Reinschmidt [4], Lemonnier *et al.* [5, 6] – analysis of reinforcement, Srokosz [7, 8, 9] – calculations optimised by genetic algorithms GA). It is not the task of this paper to give a comprehensive mathematical background of the variational limit equilibrium method – the reader can find the details in adducted publications. However, for clarity of the reasoning presented here, a few principles of the calculations should be mentioned (because of the simplification and specificity of the mathematical notation, some of the equations are defined in polar coordinate system and some – in Cartesian coordinates, which is regarded as appropriate). The critical height of the slope is studied, so the value of safety factor is constant and assumed $F = 1.0$. The linear failure criterion of Coulomb is used: $\tau = \sigma \tan \Phi + c$. Hence, only three three constants define the material, namely: F – the internal friction angle, c – the cohesion, g – the weight by volume. For simplicity of the problem, the slope is homogeneous and self-weight loaded only. The shape of potentially critical slip surface is log-spiral and has a simple form in polar coordinates

$$r(\Theta) = r_0 e^{\tan \Phi(\Theta - \Theta_0)} \tag{1}$$

and its derivative in Cartesian system can be evaluated as

$$\frac{dy}{dx} = \frac{\tan \Phi(y - y_c) + x_c - x}{y - y_c - \tan \Phi(x_c - x)} \tag{2}$$

where x_c and y_c are the coordinates of pole center, sometimes called "rotation center" (Fig. 4).

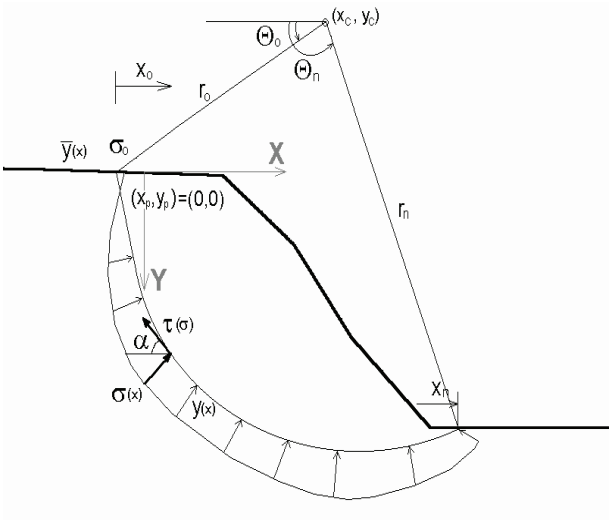


Fig. 4. Slip surface with normal stress distribution with Cartesian and polar coordinate system

The normal stress distribution along slip surface is not needed in the analysis with a linear failure criterion, although it will be necessary to evaluate the accuracy of static equilibrium equations providing two functionals of variational analysis, namely:

$$1^\circ: \quad T = \int Q dx, \qquad 2^\circ: \quad R = \sqrt{R_H^2 + R_V^2}, \tag{3}$$

where the integrated function Q has the following form

$$Q = \left(\sigma \frac{dy}{dx} - \tau\right)(y - y_c) + \left(\tau \frac{dy}{dx} + \sigma - \gamma\,(y - \bar{y})\right)(x - x_c)\,. \tag{4}$$

Here, $\overline{y}$ is the vertical coordinate of slope geometry (see Fig. 4), τ and σ are the tangent and normal components of stress distribution along the slide line (considered in Cartesian coordinates). The components of resultant force R are defined as

$$R_H = \int \left(-\tau + \sigma \frac{dy}{dx} \right) dx, \qquad R_V = \int \left(-\tau \frac{dy}{dx} - \sigma + \gamma (y - \bar{y}) \right) dx. \tag{5}$$

The explicit form of normal stress distribution can be found in polar coordinate system only, and assumes the following shape:

$$\sigma(\Theta) = \gamma r_0 e^{\tan\Phi(\Theta - \Theta_0)} \frac{3\tan\Phi\cos\Theta + \sin\Theta}{9\tan^2\Phi + 1} - \frac{c}{\tan\Phi} + C_1 e^{-2\Theta\tan\Phi}, \tag{6}$$

where

$$C_1 = e^{2\Theta_0\tan\Phi} \left(\sigma_0 - \gamma r_0 \frac{3\tan\Phi\cos\Theta_0 + \sin\Theta_0}{9\tan^2\Phi + 1} + \frac{c}{\tan\Phi} \right). \tag{7}$$

The whole idea of slope stability evaluation consists in finding its minimal height H_{crit} defined by the exit point of potentially critical slide line. The slide line is described uniquely by three geometrical parameters x0, xc, yc, although an additional unknown $\sigma 0$ is also needed to guarantee the agreement of the general requirements:

$$T = 0, \qquad R = 0, \tag{8}$$

or better:

$$\min_{x_0, x_c, y_c, \sigma_0} (|T|) \approx 0 \quad \wedge \quad \min_{x_0, x_c, y_c, \sigma_0} (R) \approx 0 \tag{9}$$

An optimisation process requires a definition of an objective function which evaluates the proximity of a currently analyzed set of variables to the optimal solution. The set of two equations (8) can be transformed into the objective function f in the following way:

$$f(w, Y) = w_T Y_T + w_R Y_R + w_\kappa Y_\kappa \tag{10}$$

where w_T, w_R, w_κ are weight coefficients, whereas

$$Y_T = \begin{cases} Y_{\mathrm{cut}} - |T| & \text{for } |T| < Y_{\mathrm{cut}}, \\ 0 & \text{otherwise}, \end{cases} \tag{11}$$

$$Y_R = \begin{cases} Y_{\mathrm{cut}} - R & \text{for } R < Y_{\mathrm{cut}}, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

$$Y_\kappa = \begin{cases} \left(1 - \frac{\sigma_0}{\kappa_0 \gamma H} \right) Y_{\mathrm{cut}} & \text{for } \sigma_0 < \kappa_0 \gamma H, \\ 0 & \text{otherwise}. \end{cases} \tag{13}$$

Here, Y_{cut} is the limit of the maximum value of $|T|$ and R, κ_0 is the prior information on σ_0, and H is the height of the slope defined by the exit point of the currently analyzed slide line. Selection of weight coefficients constitutes evolutionary management of the whole finding process. It should be emphasized that usefulness of GA strongly depends on the "shape" of the objective function in N-dimensional space of N optimised parameters. Nevertheless, complex relation between GA and optimised problem can be adjusted by proper values of the GA-controlling parameters. On the other hand, preservation of a particular inspected "shape" defined by tested and accepted forms of objective functions enables application of constant values of GA-parameters and high efficiency of evolutionary optimisation of different types of problems, even for different domains and branches.

The optimal values of x_0, x_c, y_c and σ_0 maximize the objective function f, which can take its highest value peaking assumed Y_{cut} (considering $w_R + w_T + w_\kappa = 1.0$). The reader can notice that the specificity of T and R facilitates the definition of a minimised objective function, (with its optimum value equal zero). However, in view of fitness maximisation in the canonical form of genetically controlled calculations, the presented model of f is ready for direct implementation in GA-computations. Another question is prior information usage in numerical investigations. An example of introduction of such a datum into the objective function was presented in Eq. (13), where κ_0 with an *a priori* known value is defined as (see [2]):

$$\kappa_0 = \frac{\sigma_0^{\text{crit}}}{\gamma H_{\text{crit}}} \tag{14}$$

where H_{crit} and $\sigma_{0\text{crit}}$ are the minimal height of analyzed slope and the optimal σ_0, respectively. It should be emphasized that this kind of knowledge is the most desirable support in computation management and is mainly formed according to the actual state of the problem just as the researcher's experience (in the analyzed case, the inexplicit form of prior information could be expressed by the acceptance of proper ranges of the solution components taken to the optimization process). Sometimes, the accessibility of prior information is a question of time needed to detect the "nature" of the "problem behaviour" – numerical and/or experimental simulations are very helpful, but in the majority of geotechnical problems prior knowledge is usually reserved for specialized and highly experienced engineers. The details of standard algorithm application with trial-and-error technique and prior information usage can be found in [9].

3. Genetic algorithm application

Investigation of four unknowns: x_0, x_c, y_c and σ_0 is usually realized by means of partially or totally trial-and-error methods (see [2, 9]). Finding a global solution is rather difficult because of infinite possible solutions related to the finite accuracy of computations: substantial numerical technique applied in the computer program leads to some truncation and rounding errors, which accumulated during recursive executions of the program. Another question is a difficulty of accepting close-to-zero numbers as indicators of sufficiently accurate results (see requirements (8) and (9)). Usually, most of the sharp equalities are softened to inequalities with assumed tolerance (e.g. $(|T|, R) < e_{\text{tol}}$ – prior information on error structure is highly valuable and desired here). The efficiency of finding a satisfactory solution can be significantly increased using genetically governed processing of the input data. However, such computations need an additional set of parameters which are necessary for controlling the activity of the assumed model of an evolutional algorithm. The usefulness of GA in optimisation of variational calculus has already been proven (see [7, 8]), although it is a generally known truth that GAs are ideal tools for finding good solutions, but obtaining the best ones needs more sophisticated techniques and approaches. Therefore, except for the canonical form of GA, a cycled renewing together with sub-hypercube technique were used in computations. A renewing population procedure consists of creation of a randomly generated population after an assumed number of iterations. This simple technique helps to remove individuals concentrated around a local optimum. The renewing can be interpreted as a specific form of the dispersion impact technique (DI, see [8]) acting on all bits in a chromosome chain. Assuming different lengths of DI-period in the computations, the GA's usefulness can be easily tested and for very frequent population revivals the exploration process comes close to the trial-and-error method. On the other hand, long periods between DIs enable the evolutionary techniques to manage the whole finding process with full intensity. A sub-hypercube searching procedure is a type of local searching technique based on consideration that an optimum "volume" of searched hyperspace referred to GA-controlling parameters exists (specifying: the number of individuals in a population and grain size of binary discretization constrain the dimensions of the ransacked space of unknowns). N_{Var}-dimensional subspace cut from the hypercube defined by the admissible limits of searched parameters is continuously investigated

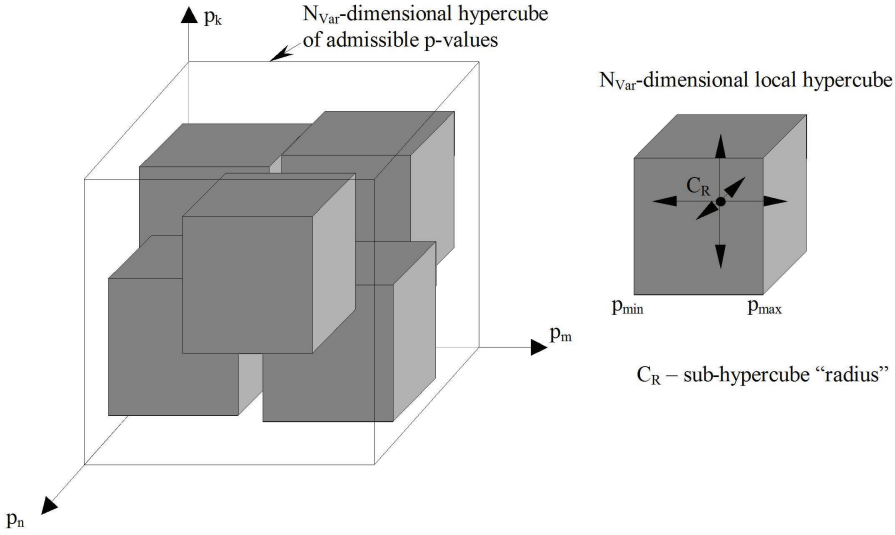


Fig. 5. Sub-hypercube idea

for period of one population's existence; the sub-hypercube usually represents a smaller space, more suitable for GA-findings with a constant number of individuals. Its limits are defined using the "radius" C_R (assumed value between 0.0 and 0.5) after each renewing action by random selection of its centre located in admissible space appointed on condition that the edges of a newly created sub-hypercube do not violate the global limits defined for all unknowns (i.e. global hypercube, Fig. 5).

The information processing techniques provided by GAs are based on different kinds of random number generators. Here, the Mersenne Twister (MT) pseudo-random number generator with $2^{19937} - 1$ period is used (see [10]). The discrepancy for a sequence of numbers created by MT is better than the one for a sequence generated by usually simplified routines which are built-in standard libraries provided with the most of popular code compilers (problem of satisfactory exploration performed by GAs, see [11]). The application of genetic algorithms in slope stability evaluation is well noted in publications worldwide (e.g. [12, 13, 14, 15]). Nonetheless, little information is related to the parallel GA (PGA) implementations in such calculations. The authors have already examined the coarse-grain model of PGA using a metacomputer: a cluster of personal computers joined together by Parallel Virtual Machine (PVM, see [16, 17]). The results were very promising; hence the research has been continued although a different kind of computational hardware has been singled out for the investigations. The idea of parallel evolutionary computations examined in this paper is based on the hybrid model of PGA (HPGA). The HPGA model consists of an elementary component of a coarse grain system, represented by an island of multi-individual population, and organized in fine grain structure (known also as a massively parallel network). Each population can communicate via migration with immediate neighbours only. The implementation of regular network topology (rectangular grid) for this model (known also as islands isolated by distance [18], extorts communication of each single computational cell with its four neighbours only (NEWS: North-East-West-South), taking into account the Linear-5 topology (more complex topologies of communication are possible too and depend on features of software implemented in working cells, see [9, 16]).

The cells at the endings of network are connected via wrapping around with the opposite ending cells, forming a toroidal grid. In spite of a typical fine grain structure of interconnections between islands, this model has been classified as hybrid because of a significant amount of individuals handled by each GA-worker (for a pure, massively parallel model each computational point works on 2-4 individuals typically). On the other hand, inconsiderable number of working points (max. 20) cannot create a pure fine grain model, which is usually characterised by a large number of GA-workers.

The "parallelization" of GA-calculations extorts introduction of additional parameters controlling frequency of data exchange and ruling type of incorporation process. The assumed structure of Linear-5 PGA model is ideal for hardware implementation using the cellular computer idea.

4. Example of calculations

In the test calculations, an example of slope taken into validation procedure presented in [2, 9] was used with parameters (see Fig. 6a): $\beta = 70°$, $\phi = 20°$, $c = 10\,\mathrm{kPa}$, $\gamma = 16\,\mathrm{kN/m^3}$.

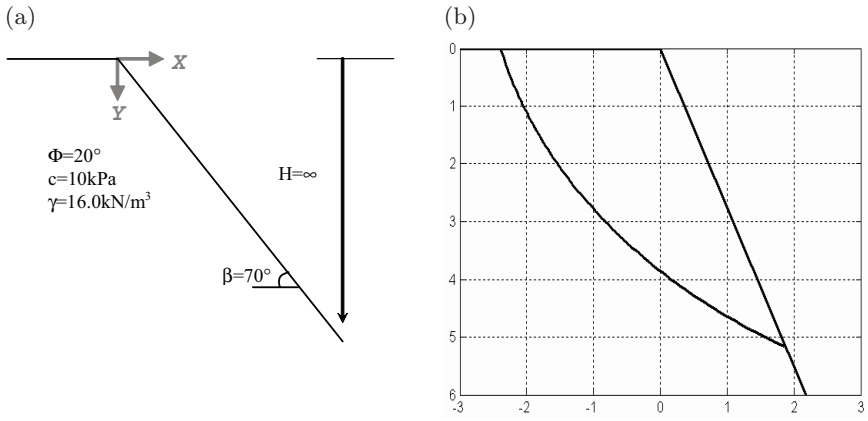


Fig. 6. Calculation example (a) and its global solution (b)

This particular case possesses an established global solution: $Hcrit = 5.18$ m, $x_c = 3.16$ m, $y_c = -3.85$ m, $x_0 = -2.39$ m, $\sigma_0 = -5.08$ and $\kappa_0 = -0.06$. The position and shape of critical slide line are shown in Fig. 6b. To evaluate the accuracy of the genetic management of computations, a normalized distance between the solution obtained from i-th run of the GA and the average solution of all performed runs was used:

$$||d_i|| = \left(\sum_{j=1}^{N_{\mathrm{Var}}} \left(\frac{p_j - \bar{p}_j}{\frac{1}{2} D_j} \right)^2 \right)^{\frac{1}{2}} \tag{15}$$

where p_j are the parameters that are being optimised, $\bar{p}_j$ are the average values of optimised parameters, $D_j = p_{j\max} - p_{j\min}$ is the range of admissible values for p_j parameter assumed in calculations, and i is the number of solution that is being analyzed. The normalized distance between an average result and the global solution $||d_p||$ can be evaluated using Eq. (15) assuming for p_j the optimal values of parameters and taking full range D_j in place of $1/2D_j$. Concentration of solutions found by the GA around their average value can be defined as follows:

$$C_p = 1 - \frac{\left\| \bar{d} \right\|}{N_{\mathrm{Var}}^{\frac{1}{2}}}, \tag{16}$$

where $\left\| \bar{d} \right\|$ is the average value of normalized distances for all obtained solutions. The main advantage of the suggested concentration analysis is the absence in formulas of any information about location and even existence of the global optimum; instead, simple convergence of final results is taken into considerations. However, the bounds on unknowns were assumed in a way which guarantees existence of global solution inside the created hypercube. High discrepancy described by low concentration values (lower than about 0.8 and 0.6 with and without prior information usage, respectively) indicates low usefulness of chosen GA-techniques in the optimisation process and undermines the fundamental merit of genetically controlled computations, their ability to indicate the

area of global optimum's existence. On the other hand, the concentrations reaching over 0.95 prove high efficiency of GA-calculations and correct selection of cooperating GA-techniques. Tables 1 and 2 present the sets of GA-constants which were already positively examined in previously performed calculations with assistance of coarse grain model powered by PVM (see also [16]). The analyses were carried out for 20 independent runs of computations.

Table 1. Working parameters of HPGA

Population size	Number of generations	Type of crossing	Crossing probability	Mutation probability	Migration probability	Incorporation method
60	60	uniform	0.8	0.005	0.0÷1.0	tournaments

Selection method	Hypercube radius	Scaling coefficient	Strategy	Copied individuals	Renewing period	Number of cells
tournaments	0.1÷0.5	2.0	elitist	10%	15	20

Table 2. Coding parameters

Number of variable	Description	Minimum value	Maximum value	Number of bits	Type of coding
1	x_c [m]	2.0	5.0	9	
2	y_c [m]	-5.0	-2.0	9	Gray
3	x_0 [m]	-3.0	-1.0	8	binary reflected
4	σ_0 [kPa]	-7.0	-3.0	8	

In each of 60 iterations (generations), a set of 60 individuals is created and each individual contains 4 coded parameters (Gray binary coding). The probability of uniform crossing of two individuals ("parents") selected by tournaments is assumed as 0.8, taking into consideration mutation probability of 0.005. Some individuals chosen by tournaments can migrate among the nearest neighbors with variable incorporation probability in range of 0.0 to 1.0. The elitist strategy guarantees that in each generation process, 10% of best individuals are copied into next population without any changes. To avoid a premature convergence, the objective function is scaled using coefficient 2.0. This operation forms a fitness function (see [20]). After each 15 generations a renewing procedure is applied and the next population is created using a random number generator (as at the beginning of the calculations). However, migrants remain saved in buffers. The sub-hypercube technique is applied with the radius variable from 0.1 to 0.5, which means that 20% to 100% of the range assumed for each unknown (defined limits) is taken into account during the actually performed cycle of 15 iterations. The whole investigation process includes 60 iterations (generations), i.e., 4 renewing processes occur. Gray binary coding technique was chosen to eliminate Hamming Cliffs phenomenon (see [11, 18]). The range of unknowns and number of bits in their binary representations were chosen subjectively, although these values of control data were already exploited in advanced performed investigations and, in order to retain the possibility of direct comparisons between results, they were left without any changes. Figure 7a presents the concentration of obtained results and their distance to the global optimum neglecting prior information on σ_0 value.

In Fig. 7b, the results considering prior information are shown ($w_T = 0.2$, $w_R = 0.7$, $w_\kappa = 0.1$). It is seen that the usage of prior information has a substantial influence on quality of obtained results. Migration probability and hypercube radius interact jointly both on concentration and distance to global optimum. It should be mentioned, that the incorporation of selected migrants is a resultant process of three independent actions: election of immigrants using migration probability, verification of affiliation of immigrants to the actually analyzed sub-hypercube space, selection of native individuals and execution of tournaments with immigrants. Positively examined immigrants

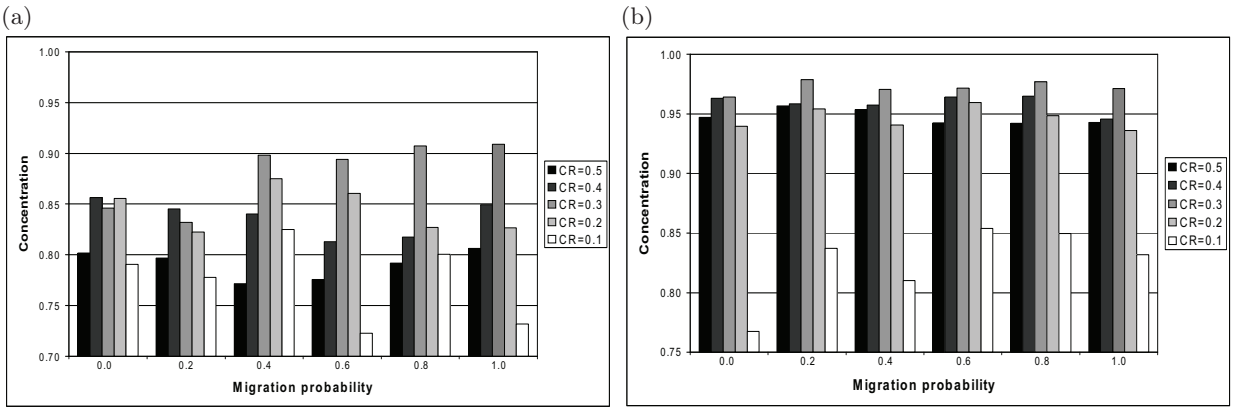


Fig. 7. Results of calculations: (a) no prior information used, (b) prior information used

replace native individuals and inherit all attributes of the native population, jointly with new fitness recalculated based on contribute objective value and actual statistics of such a modified population. In the analyzed slope stability optimization problem, the implementation of $C_R = 0.3$ and migration probability greater than 0.2 gave the best efficiency of computations in case of not using prior information. On the other hand, an application of the prior information made the whole optimization process insensible to migration probability value.

5. Influence of HPGA structure size on results

The main purpose of this analysis is to determine a minimum number of calculation units which would guarantee satisfactory computational capability of the system. The HPGA structure size denotes here three dimension components of the following quantities: population size analyzed in each working cell, maximum generations performed on population, number of cells building the cellputer system. To examine the influence of HPGA size on accuracy of repeated calculations, the variable dimensions of its components were taken into considerations. All other parameters remained unchanged. The whole investigation process included 20 to 100 iterations (generations), during which 1 to 6 renovations of population occurred. Though tests made previously revealed the optimum $C_R = 0.3$ giving the best results, subsequent tests were performed for $C_R = 0.4$ in order to decrease the thrust of local searching and to increase the influence of parallel processing itself. In the performed calculations, up to 20 computational cells were used, assembled in 8 different structures (systems), as shown in Fig. 8.

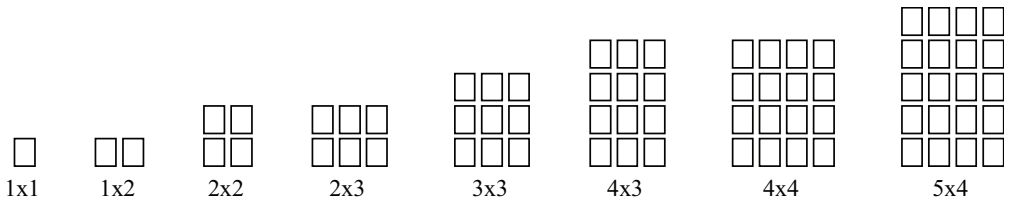


Fig. 8. Tested systems of cells

The behaviour of the cellputer as the system of cooperating cells can be visualized in the form of small rectangles filled with different colours indicating different values of the objective function – see Fig. 9; better fitness is represented by brighter colour, in contrast dark colours indicate poor fitness (detailed scale is not needed here, since a general idea is visualized although the real results of calculations were the basis of this analysis).

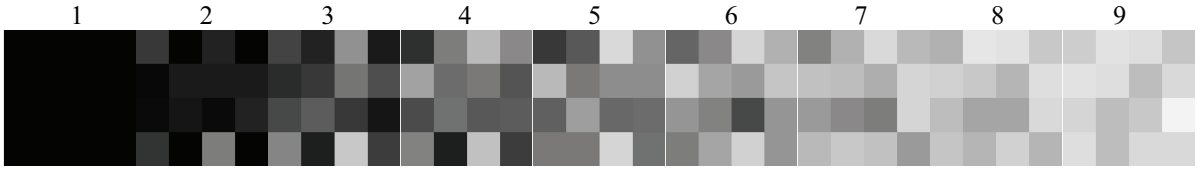


Fig. 9. Visualization of the typical cellputer's behavior for 4×4 structure

As it can be noticed, in the first cycle of calculations the system "comes into the stable phase" after 9 generations (nearly constant objective values in all cells during iterations), filling buffers with the best individuals – migrants. In the next cycles the "stable phase" is reached much faster – 5-6 generations after renewing complete the finding process, mostly thanks to migrants saved in the buffers. Though migrants feature fast convergence of calculations, most cycles perform investigation in a slightly different way due to three different accompanying operations: the hypercube redefinitions changing the hierarchy of individuals in the current population due to scaling operations, asynchronous reading of contents saved in the buffers – the variable time shift induces replacement of migrants and changes the pathway of searching (see [21]), different course of a pseudorandom number generator causes a genetic drift, which slightly affects the whole finding process.

It should be emphasized that for the smallest systems, like 1×1 and 1×2, each cell also incorporates its own individuals saved in the buffers; the bigger structures guarantee migration of the best solutions among adjacent neighbours, without "self-feeding" phenomenon. The two typical results obtained in all performed computations were presented in the form of maps, shown in Fig. 10.

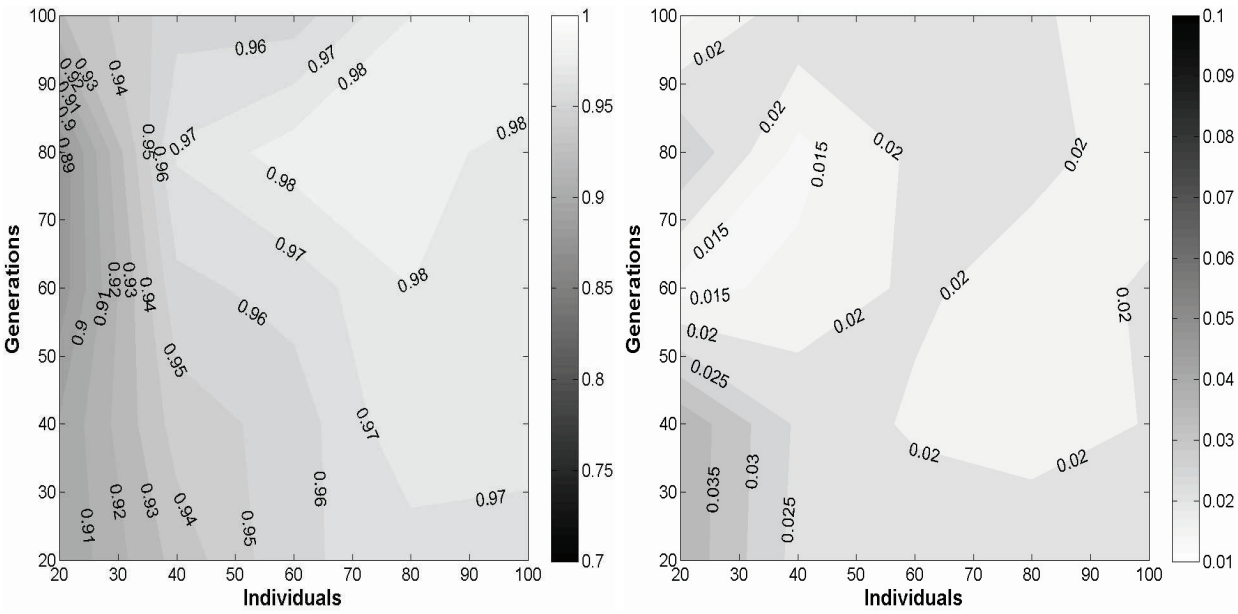


Fig. 10. Typical results of calculations

Analyzing the dependence between the size of a population, number of generations and obtained concentrations C_p for different numbers of cells working in the structure, it is remarkable that bigger systems processing a large amount of information give better concentration of the results. However, this conclusion is obvious and it was not the purpose of the investigation. The main importance lies in the high number of individuals which affects the concentration much stronger than the number of generations and size of the cellputer's structure. On the other hand, it is possible to detect the size of the structure and optimal number of individuals which guarantee the achievement of assumed limit of acceptable C_p. Differently, the analysis of the relative distances between average

and global solutions give non-unequivocal results. The lack of coincidence between $||d_p||$ and C_p can be observed. This fact is the consequence of the fact that dispersed results "can more easily" create an average point placed close to the optimum than tightly concentrated results occupying a smaller space. On the other hand, it should be noted that the global optimum taken into analyses was nominated by numerical investigations using differential rather than exact forms of Eqs. (1) and (6). Thus, a slight shift between the best results of different optimization processes can be justified. Let us define a normalized volume as follows:

$$||V|| = \frac{V_i}{V_{\max}}, \tag{17}$$

where V_i is the volume limited by the assumed horizontal facet (reference plane: for $C_p = 0.7$ and for $||d_p|| = 0.10$) and the surface created by typical results shown in Fig. 10, confined by vertical side facets, $V_{\max}$ is the maximum volume limited by reference and horizontal facet representing maximum $C_p = 1.0$ and minimum $||d_p|| = 0.01$: $(100 - 20)^2 \cdot (1.0 - 0.7) = 1920$ and $(100 - 20)^2 \cdot (0.10 - 0.01) = 576$ for C_p and $||d_p||$, respectively. Creating relation between size of the cellputer's structure and normalized volumes (Fig. 11), the favorable trend for $||d_p|| = 0.01$ can be observed, however, deceleration of changes for large structures implies achieving the limit of system configuration's means though permanent augmentation of C_p. The exemplary slide lines for two different values of C_p were collated in Fig. 12.

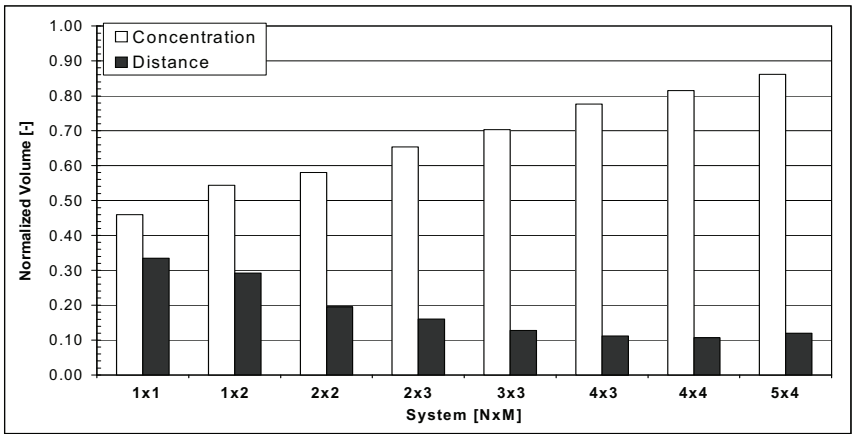


Fig. 11. Results of calculations. Dependence between the size of the cellputer's structure and normalized volumes for the concentrations C_p and distances $||d_p||$.

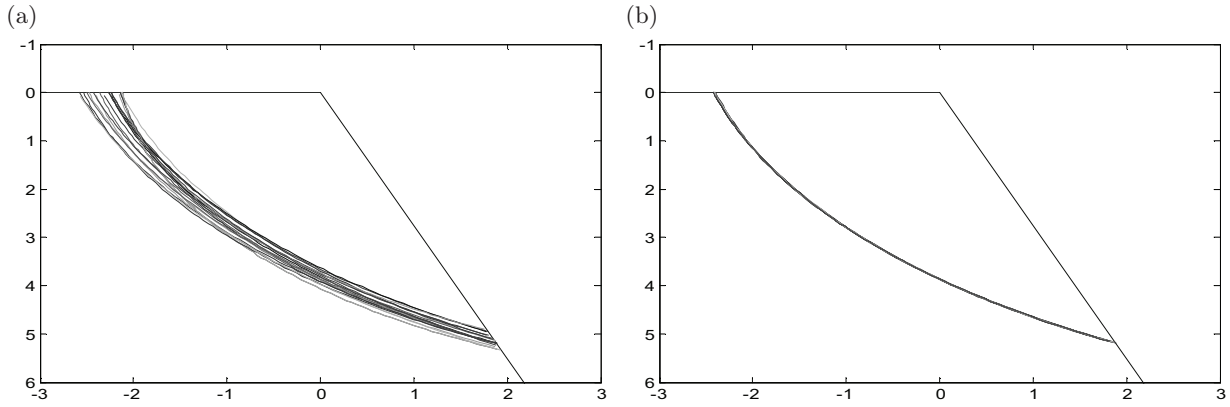


Fig. 12. Results of calculations. Slide lines for the concentrations C_p: (a) 0.75, (b) 0.99.

6. Concluding remarks

The general aspect of hardware implementation of PGA applied to multivariate and multicriterial optimization presented in this paper leads to the following conclusions:

- the application of single-chip microcomputers to parallel calculations using GA is possible and not difficult to realize;
- tested configurations of parameterized software – structured hardware gave information about a minimum number of needful processors which can guarantee achieving satisfactory recurrence of results for the assumed set of working parameters (in the analyzed slope stability problem it was 16 units);
- the simplicity of a single cell construction enables creation of different structures including a combination of various spatial configurations of interconnections among computational cells, i.e. multi-layered toroids or rings;
- the specificity of particular software working with adjusted controlling parameters can give perspicuous results, although it should be emphasized that the problem of slope stability evaluation by variational calculus can be solved in different numerical ways giving slightly different solutions.

Although the investigation process is based on the experimental basis only, the results can serve as the reference support to other similar research works. The time-measurements were not performed. The main aim of the investigation process was to verify helpfulness of DPGA structure in the identification of the global solution, neglecting at this moment the time-optimization aspect of computations. The presented project attained an adaptation of higher level HPGA structures, in which cells are divided into closed groups and the rivalry among them with spontaneous migration form the global solution. The work will continue. The sources of executable codes are available online at: `www.uwm.edu.pl/edu/piotrsrokosz/codes.htm`

Acknowledgements

The presented results of calculations were obtained exploiting computer applications created, using a package of Integrated Development Environment Delphi 7 Personal supplied by BSC Poland, a representative of Embarcadero Technologies Inc. The authors would like to thank these institutions for their support.

References

[1] R. Baker, M. Garber. Variational approach to slope stability. *Proceedings of 9th International Conference on Soil Mechanics and Foundation Engineering, Tokyo, Japan 1977*, pp. 9-12.

[2] W.F. Chen, X.L. Liu. Limit analysis in soil mechanics. *Developments in Geotechnical Engineering*, **52**, Elsevier, Amsterdam-Oxford-New York-Tokyo 1990.

[3] R. Baker. Variational Slope Stability Analysis of Materials with Nonlinear Failure Criterion. *Electronic Journal of Geotechnical Engineering*, pp. 2005-0514, 2005.

[4] D. Leshchinsky, A.J. Reinschmidt. Stability of membrane reinforced slopes. *Journal of Geotechnical Engineering ASCE*, **111**(11): 1285-1300, 1985.

[5] P. Lemonnier, A.H. Soubra, R. Kastner. Variational displacement method for geosynthetically reinforced slope stability analysis: Local stability. *Geotextiles and Geomembranes*, **16**: 1-25, 1998.

[6] P. Lemonnier, A.H. Soubra, R. Kastner. Variational displacement method for geosynthetically reinforced slope stability analysis. II: Global stability. *Geotextiles and Geomembranes*, **16**: 27-44, 1998.

[7] P. Srokosz. Slope stability analysis by variational method with genetic algorithm application. Part 1: Slopes with homogeneous cohesive soils. Simple genetic algorithm application. *Archives of Civil Engineering*, **53**(1): 85-108, 2007.

[8] P. Srokosz. Slope stability analysis by variational method with genetic algorithm application. Part 2: Genetic algorithm with advanced techniques. Simple modeling of uncertainties. *Archives of Civil Engineering*, **53**(2): 269-292, 2007.
[9] P. Srokosz. Slope stability analysis by variational method with genetic algorithm application. Part 3: Normal stress distribution. Parallel and distributed computing. *Archives of Civil Engineering*, **54**(2): 423-441, 2008.
[10] M. Matsumoto, T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, **8**(1): 3-30, 1998.
[11] J. Arabas. *Lectures on evolutionary algorithms* [in Polish]. Wydawnictwa Naukowo-Techniczne, Warsaw 2004.
[12] S.K. Das. Slope stability analysis using genetic algorithm. *Electronic Journal of Geotechnical Engineering*, pp. 2005-0504, 2005.
[13] A.T.C. Goh. Genetic algorithm search for critical slip surface in multiple-wedge stability analysis. *Canadian Geotechnical Journal*, **36**: 382-391, 1999.
[14] P. McCombie, P. Wilkinson. The use of the simple genetic algorithm in finding the critical factor of safety in slope stability analysis. *Computers and Geotechnics*, **29**: 699-714, 2002.
[15] N. Sabhahit, J. Sreeja, M.R. Madhav. Genetic algorithm for searching critical slip surface. *Indian Geotechnical Journal*, **32**(2): 86-101, 2002.
[16] P. Srokosz. Slope stability analysis by variational method with genetic algorithm application. Part 4: Parallel genetic algorithms. *Archives of Civil Engineering*, **55**(2): 229-256, 2009.
[17] *Parallel Virtual Machine System (PVM) version 3.4*. University of Tennessee, Knoxville TN; Oak Ridge National Laboratory, Oak Ridge TN; Emory University, Atlanta GA.
[18] D. Whitley. *A genetic algorithm tutorial*. Computer Science Department, Colorado State University, available online: http://samizdat.mines.edu/ga_tutorial/ (accessed 21.07.2009).
[19] E. Alba, J.M. Troya. Improving flexibility and efficiency by adding parallelism to genetic algorithm. *Statistics and Computing*, **12**: 91-114, 2002.
[20] D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Company, 1989.
[21] E. Alba, J.M. Troya. Analyzing synchronous and asynchronous parallel distributed genetic algorithms. Future Generation Computer Systems 2001; 17: 451-465.