ADVANCED GENETIC ALGORITHMS FOR ENGINEERING DESIGN PROBLEMS

Jan Roupec*

The study of analogy of the natural evolution and the technical object design dates back more than 50 years. The genetic algorithm (GA) is considered to be a stochastic heuristic (or meta-heuristic) optimisation method. The best use of GA can be found in solving multidimensional optimisation problems, for which analytical solutions are unknown (or extremely complex) and efficient numerical methods are also not known. GAs are inspired by adaptive and evolutionary mechanisms of live organisms, but they do not copy the natural process precisely. The paper describes the main terms, principles and original implementation details of GA. The main goal of this paper is to help readers to use proper GAs on the field of technical objects design.

Keywords : *genetic algorithm, stochastic heuristic optimisation methods, evolutionary computing, genetic operators*

Various challenging decision making problems are related to the area of engineering design. Recently studied problems usually involve uncertain parameters, composed structure, and the need to find optimal or suboptimal values of design variables. They can be found in traditional mechanical or civil engineering design problems, see, e.g., [15] and [19] or in the design of transportation and generally logistic networks [23] or in the problems of reliable system design [16]. The built mathematical models often belong in mathematical programming and its specific branches integer, multistage, and stochastic programming [22]. The classical deterministic algorithms, even utilizing advantages of decomposition (see [20]) applied to the instances of discussed problems may often need support of heuristics, see [21]. Therefore, we focus on the presentation of GA framework that was develop in [5] and later applied to aforementioned problems. Additional readings and alternative approaches can be found in literature, e.g., [2]–[7], [10]–[14]. Genetic algorithms (GA) belong in artificial evolutionary systems. For study, understanding, description and development of genetic algorithms is necessary to be familiar with mechanisms that nature uses for the selection of parents, population adaptability, keeping a various genetic fund while main traits of biological categories remain stable, etc. In addition the practical using of artificial evolutionary systems brings many other problems relating to their convergence, robustness and stability. The first chapter brings the simplified introduction to biological principles of genetics and evolution.

1. Introduction to evolution and genetics

Scientific discussion of evolution dates back more than 200 years, see [1] and [2]. In the 1760s, the French naturalist Buffon (Georges-Louis Leclerc, Comte de Buffon, 1707–1788)

* Ing. J. Roupec, Ph.D., Institute of Automation and Computer Science, Faculty of Mechanical Engineering, Brno University of Technology

had written his Natural History of Animals, which contain a clear statement of the possibility of evolution. Buffon's student Lamarck (Jean-Baptiste Pierre Antoine de Monet, chevalier de Lamarck, 1744–1829) wrote extensively about evolution. Lamarck was the first person to support the idea of evolution with logical arguments and was also the first person to put forth and hypothesis concerning the mechanisms of evolutionary change. He suggested that living organisms have the ability to change gradually over many generations by the inheritance of structure that have become larger and more highly developed as a result of continued use or, conversely, have diminished in size as a result of disuse. Now we do not believe that most evolutionary changes have been produced by the mechanisms proposed by Lamarck.

Like most great ideas, the Darwin-Wallace theory of natural selection, as presented in Darwin's book The Origin of Species, is remarkable for its simplicity. Darwin (Charles Robert, 1809–1882), who developed the theory in greater detail than Wallace, began with two familiar facts:

(1) The individuals of most kinds of organisms are not identical, and

(2) Offspring tends to resemble their parents.

Darwin suggested that slight variation among individuals significantly affects the change that a given individual will survive and reproduce. Darwin called this differential reproductive success of varying individuals *natural selection.*

Mendel (Johann Gregor, 1822–1884) accurately observed patterns of inheritance and proposed a mechanism to account for some of the patterns. Mendel showed that the inheritance follows particular laws, which were later named after him.

Mendelian genetics relates to organisms with sexual reproduction. Inheritable properties of living organisms are stored in structures called chromosomes. The location of a character being inherited is called gene, so genes are parts of chromosomes. Genes contain alleles. An allele is the alternate form of a genetic character found at a given locus on a chromosome. Living organisms using sexual reproduction have two alleles for each gene (two particular forms of the gene). Chromosomes holding information in the form of pairs of alleles are called diploid chromosomes (whereas haploid chromosomes don't contain pairs of alleles). Descendant of diploid parents obtains one allele from each parent. Individuals having both alleles coding specific character identical are called homozygotes; individuals with different set of corresponding alleles are called heterozygotes.

The physical appearance of a character is its phenotype, which is supposed to be the result of the genotype (e.g. the genetic code of the character). We can often observe similar or even identical look (phenotype) of individuals with different sets of genes (genotype). Three types of alleles exist: dominant, recessive and codominant. Dominancy means, that one allelic form is capable to determine the phenotype of heterozygous individual. The recessive allele cannot determine the phenotype if the second (dominant) allele is present. More than one dominant allele can exist for one specific trait. When the conjugate appearance of two dominant alleles leads to a different phenotype than the presence only one of them, we talk about codominancy (e.g. the AB blood type is the phenotypic symptom of the conjugate presence of the dominant allele for the A blood type and the dominant allele for the B blood type, these alleles are codominant).

It is not important, whether the dominant allele was inherited from mother of father. But the inheritance can be sex-linked. It means that the inherited trait is present only in the phenotype of individuals having specific sex. All cells in the body of a living organism contain the same number of chromosomes. This number is fixed for all individuals of the specific biological species. Usually the number of chromosomes in the cell is even. In this case chromosomes come in pairs and we talk about diploid cells. Sex cells are exceptional; they contain only half number of chromosomes than other cell of the same species. Sex cells are diploid cells.

The agents of evolution include nonrandom mating, mutation, genetic drift, migration, and natural selection. Sexual recombination generates an endless variety of genotypic combination that increases the evolutionary potential of the population [9], [10], [11]. Because it increases the variation among the offspring produced by an individual, it improves the change that some of them will be successful in varying and often-unpredictable environments they will encounter. However, sexual recombination does not influence the frequency of different alleles or the rate of evolutionary change.

Males of most species increase their reproductive success by mating with a large number of whereas females are unlikely to increase their reproductive success by increasing the number of males with which they mate. One male provides a female with enough sperm for a very large number of offspring – usually many more than the number of eggs the female can produce, or the number of young she can nourish. Therefore, males of many species attempt to increase the number of females they copulate with, and they fight with one another for access to females. Females, on the other hand, are usually more discriminating in their choice of mates, and they resist most attempts by males to copulate with them. Individuals may be larger or smaller than the average ones of the same age and sex. The reproductive roles of males and females are different in different groups. In cooperative system, however, advantages accrue to those individuals able to dominate others. The result was the evolution of dominant female, the queen (bees, wasps, and ants), who lays most of the eggs, and subordinate females who do most of the work. The dominant female, who has the most offspring in the nest, performs less parental care than the subordinate females do. Subordinate females, although they do worse then dominant females, nonetheless do better than they would as solitary breeders in the same habitat because groups compete more successfully for good breeding areas then do solitary pairs.

Most mammals evolved sociality via the familial route [12]. Some present-day mammals represent the first stage, in which solitary females or male-female pairs care for their young. Social organization in mammals is dependent on the group size, as follows:

Group size:	**social unit:**
1 or 2	pair,
2–12	male with small harem,
2–100	territory-defending male with harem,
up to thousands	herd in which males defend females,
up to thousands	herd with male dominance hierarchy.

Female primates, like other female mammals, usually remain with their natal troop, whereas young males are driven away by older males before they become reproductively active. In troops with more than one male, there are strong dominance hierarchies among the males, and just a few of the males do most of the copulating. Animals choose associates on the basis of their traits, the resources they hold, and their relatedness to the chooser. Close relatives share alleles with one another and can increase the frequency of these alleles in future generations by helping one another even if the aid is not returned. Among non-relatives, helping behavior can evolve only if individuals reciprocate. Complex social systems have evolved by two routes. The familial route begins when offspring remain with their parents and help to rear future broods. The parasocial route begins when adults or the same generation associate around clumped resources, and cooperation evolves among them. Among birds, males of promiscuous species that display on communal grounds are usually much larger an more brightly colored then females, whereas i a monogamous species, males are usually similar in size to females, whether or not they are more brightly colored.

Populations of most species are divided into a number of subpopulations that are at least partly isolated from one another. Different agents often affect subpopulation uniquely in different parts of the range. An individual organism, if it completes its entire life cycle, is born, grows to maturity, reproduces, and dies. Population dynamics determine whether the population as a whole is increasing, decreasing, or staying the same, and whether its structure is changing. Population dynamics are changes in population structure that result from the reproduction, growth, energy gathering, dispersal, and death of members of population. Populations in resource-rich environments increase exponentially, but population growth slows and ceases when the carrying capacity of the environment is reached. Birth rates are affected by the amount of resources available to the parents. Death rates of young individuals are usually high among species that produce large number of offspring and provide them with little energy or care.

The environment in which an organism normally lives is called its habitat. Everything that the organism needs must be provided by its habitat. One of the most spectacular responses to seasonal changes in habitat quality is migration, the regular seasonal movement of animals from one place to another. Parental care reduces juvenile death rates, allowing more individuals to survive to post reproductive age, when death rates again become very high.

2. Principles of genetic algorithms

Genetic algorithms belong in stochastic heuristic optimization methods. The main usage of GAs is the solving of problems of multi- dimensional optimization where no analytic solution is known or is not possible to be used.

We denote an optimization problem as

$$\bar{x}_{\mathrm{opt}} = \arg\mathrm{opt}\{f(\bar{x})|\bar{x} \in C\} \ , \tag{1}$$

where f is the cost function, C is the set of feasible solutions, $\bar{x}$ is a feasible solution and $\bar{x}_{\mathrm{opt}}$ is the optimum solution to be found. The strategy of searching through the set C is very important. The most primitive strategy is to generate all feasible solutions 'step by step', for every solution calculate the cost function and find the best solution. In practice this procedure cannot be used. Theoretically, the C is usually an infinite set. When using computer, the C is finite, because we cannot store infinite number of combination into the memory. But the number of different elements of the set C is always very high. So, we cannot explore all possible values of $\bar{x}$ in a suitable time. Heuristic methods use some kind of more clever strategy.

The strategy of searching through the set C used by GA is inspired with natural evolution, where the best individuals have biggest chance to survive and to become parents of new offspring. In addition, GA uses another mechanism existing in the nature – mutation. Mutation is a 'small' random change of genetic information. Using mutation individuals can adapt to changing living conditions easier. Mutation can also prevent degeneration (in optimization the deadlock in a local extreme can be supposed to be an analogy to degeneration).

GA has an iterative character. GA works not only with one solution in time but with the whole population of solutions. The population contains many (ordinary several hundreds) individuals ů bit strings representing solutions. The mechanism of GA involves only elementary operations like strings copying, partially bit swapping or bit value changing. GA starts with a population of strings and thereafter generates successive populations using the following three basic operations: reproduction, crossover, and mutation. Reproduction is the process by which individual strings are copied according to an objective function value (fitness). Copying of strings according to their fitness value means that strings with a higher value have a higher probability of contributing one or more offspring to next generation. This is an artificial version of natural selection. Mutation is an occasional (with a small probability) random alteration of the string position value. Mutation is needed since, in spite of reproduction and crossover effectively searching and recombining the existing representations, they occasionally become overzealous and lose some potentially useful genetic material. The mutation operator prevents such an irrecoverable loss. The recombination mechanism allows mixing of parental information while passing it to their descendants, and mutation introduces innovation into the population.

GAs are simplified versions of biological mechanisms. They can be solved only by using computers. Here we can see the reason, why the formal description of GA is neither unified nor commonly accepted. Nevertheless we need some formal description that may support future implementation. We can show one possibility of the GA formalization [3]. The GA is a stochastic heuristic algorithm containing operators and parameters as follows:

$$GA = (N, P, f, \Theta, \Omega, \Psi, \tau) \tag{2}$$

where P is the population containing N elements (individuals) $P = \{S_1, S_2, \dots, S_N\}$. Each element S_i, $i = 1, \dots, N$ is a string (or set) of integers of the fix length of n representing the solution of the problem, so $S_i \in Z^n$. f denotes the cost function (fitness function) which assigns a real number to every individual:

$$f\colon S_i \to R \ , \qquad i = 1, \dots, N \ . \tag{3}$$

Θ is the parent selection operator, which selects u elements from P:

$$\Theta\colon P \to \{S_1, S_2, \dots, S_u\} \ . \tag{4}$$

Ω is the set of genetic operators Ω_* containing the crossover operator Ω_c, the mutation operator Ω_m and eventually other problem dependent or implementation dependent operators. All these operators generate v descendants $\{O_1, O_2, \dots, O_v\}$ from u parents $\{S_1, S_2, \dots, S_u\}$:

$$\Omega_*\colon \{S_1, \dots, S_u\} \to \{O_1, \dots, O_v\} \ . \tag{5}$$

Ψ is a deletion operator, which deletes v selected elements from the population in the t-th iteration $P(t)$ modeled by a set of population elements. Then new v children generated by (5) are added to the population and the new population $P(t+1)$ is created:

$$P(t+1) = P(t) - \Psi(P(t)) \cup \{O_1, \dots, O_v\} \ . \tag{6}$$

τ is the stopping rule:

$$\tau(P(t)) \rightarrow (\text{true}, \text{false}) \ . \tag{7}$$

The parent selection operator Θ and the genetic operators Ω_* have the probabilistic character and the deletion operator Ψ is usually deterministic.

Depending on implementation an individual S_i contains one or more chromosomes. The genotype of chromosome is the inner representation of chromosome in the computer memory. Usually the inner representation is treated as a vector of genes. The phenotype $\bar{x}$ of chromosome is an abstract mathematical object representing the solution of solved optimization problem. We can treat it as a vector containing numbers, symbols, the graph structure description etc.

The use of computers is expected, so the chromosome will be stored in the memory in the form of field of bits. This field must have the finite length. It means that the set of all possible phenotypes of chromosomes is finite too:

$$X = \{\bar{x}, \bar{x}', \bar{x}'', \dots\} \ . \tag{8}$$

When the element of population S_i contains just one chromosome, we can write the cost function (3) in the following form:

$$f \colon X \rightarrow R \ . \tag{9}$$

Then the solution of the original problem (1) may be transformed to searching of a global minimum of the cost function f:

$$\bar{x}_{\text{opt}} = \arg \min_{\bar{x} \in X} f(\bar{x}) \ . \tag{10}$$

The ambition of evolution is to create a chromosome, which is very near (or even equal) to the optimal chromosome $\bar{x}_{\text{opt}}$.

As mentioned above, the mutation operator (as well as the crossover operator) has a probabilistic character; we can denote p_{m} the probability of mutation and p_{c} probability of crossover. The application of the mutation operator to the chromosome must satisfy the rule:

$$\lim_{p_{\text{m}} \rightarrow 0} \Omega_{\text{m}}(\{\bar{x}\}) = \{\bar{x}\} \ . \tag{11}$$

The probability of crossover p_{c} determines the probability of selection the specific individual (chromosome) for reproduction process. The value of p_{c} is different for every individual and depends on the fitness. The fitness F is a non-negative number bringing a relative measure of the quality of every individual in the current population. The mapping function

$$F \colon P \rightarrow R^+ \tag{12}$$

must satisfy the rule

$$f(\bar{x}_1) \leq f(\bar{x}_2) \Rightarrow F(\bar{x}_1) \geq F(\bar{x}_2) \ . \tag{13}$$

Naturally the rule (13) applies for searching a minimum. The cost function – fitness dependence is usually linear, but can be of different type. The relative (standardized) fitness can be denoted as

$$F'(\bar{x}) = \frac{F(\bar{x})}{\sum\limits_{\tilde{x} \in P} F(\tilde{x})} \ , \tag{14}$$

where

$$\forall \bar{x} \in P\colon 0 \leq F'(\bar{x}) \leq 1 \ , \tag{15a}$$

$$\sum_{\bar{x} \in P} F'(\bar{x}) = 1 \ . \tag{15b}$$

The relative fitness is frequently used directly as the value of probability of crossover p_c :

$$p_\mathrm{c}(\bar{x}) = F'(\bar{x}) \ . \tag{16}$$

The application of the p_c value is a part of implementation of the parent selection operator Θ.

The run of GA can be described using following sequence :

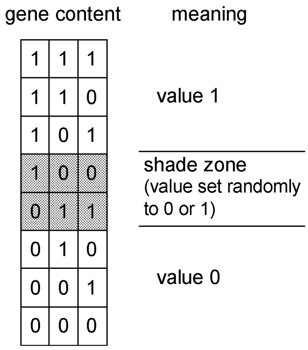
1. Generating of initial population (random generation is usually used).
2. Computing of fitness of all individuals.
3. Parent selection (4) and generating of offspring (5).
4. Creating of the new population using deletion operator and adding offspring generated in previous step (6).
5. Mutation.
6. If the stopping rule is not satisfied, go to step 3.
7. The result is the best individual in the population.

3. Implementation details and tricks

In spite of simple principles, the design of GA for successful practical using is surprisingly complicated. GA has many parameters that depend on the problem to be solved. In the first, it is the size of population. There are two opposite demands. The diversity (the amount of information in the population) should be as high as possible. As the rate of convergence should be high and it depends on the number of computations in every iteration. Larger populations usually decrease the number of iterations needed, but dramatically increase the computing time for each of iteration. The factors increasing demands on the size of population are the complexity of the problem being solved and the length of the individuals. A smaller population brings the problem of degeneration (premature convergence). The optimal size of population increases exponentially with the length of genes (when binary coding is used) [6]. Practical experiences show that the population size of 50–200 produces usually good results in many cases, for large problems up to 1000 individuals in population should be used.

Every individual contains one or more chromosomes holding value of potential solution. Chromosomes consist of genes. Genes are the elements of the parameters vector. The value represented by chromosome has to be decoded. To eliminate the influence of the Hamming's barrier, Grey code must be usually used.

We can consider the gene to be a structure representing one bit of solution value It is usually advantageous to use some redundancy in genes and then the physical length of genes can be greater than one bit. Such a type of redundancy was introduced by Ryan [4]. So called shades means, that chromosomes contain redundant information – every value bit is stored in one gene having length of more than one bit. E.g. for the genes of the length of three bits gene values 0, 1, 2 could represent value 0, gene values 5, 6, 7 could represent value 1, and values 3 and 4 could be an 'undetermined' (shade) area – their value is set



gene content			meaning
1	1	1	value 1
1	1	0	
1	0	1	
1	0	0	shade zone (value set randomly to 0 or 1)
0	1	1	
0	1	0	value 0
0	0	1	
0	0	0	

Fig.1: Shades – structure of the gene

randomly (once for the whole lifetime of each gene). The structure of these three bits genes is presented on Fig. 1.

The strategy of the generation of the initial population is also important. Usually it is generated randomly, but where we can also use solutions obtained by another methods (in this case we talk about hybrid GA). It is useful to reach the high diversity in the initial population, and so using of identical individuals in initial population is not recommended. There is also big problem with infeasible solutions. Initial population should not contain infeasible solutions. However the appearance of infeasible solutions is not only a problem of initial population generation. Infeasible solution can be also the product of the crossover and/or mutation operation. Such solutions could be excluded from population, but sometime it is better to use penalization of corresponding members of population (by assigning a very bad value of cost function to these individuals).

One of the most important details is the parent selection strategy. The diversity is dramatically decreased when only the best individuals are selected for crossover. Three basic strategies are used:

- proportionate selection – the probability of selection of each individual is proportionate to its fitness,
- ranking selection – population is sorted by the fitness, the probability of selection of each individual is proportionate to its rank in population,
- tournament selection – several individuals are selected randomly, from them the individual with the best fitness is taken.

New individuals are created by operation called crossover. In the simplest case crossover means swapping of two parts of two chromosomes split in randomly selected point (so called one point crossover). It is also possible to use multi-point crossover. In dependence of application the point of crossover can be located inside the genes or only on the borders of genes. Uniform crossover means that every gene (or even every bit) of the new individual is randomly chosen from one parent.

The main goal of mutation is to eliminate degeneration and to allow changes of genotype, which cannot be reached by crossover. There are different kinds of realisation of mutation operation (to negate one bit, the group of bits, to change one gene, arithmetic operation applied to a chosen element of vector of solution in chromosome, ...). The mutation has a probabilistic character (bits/genes/chromosomes are selected randomly); corresponding parameter is the probability of mutation.

The classical version of GA uses only three genetic operators – crossover, mutation and reproduction (e.g. the parent selection and the replacement scheme). One of the biggest disadvantages is a tendency of GA to reach some local extreme. In this case GA is often not able to abandon this local extreme in consequence of the low variability of members of population (analogy of the degeneration). To prevent the degeneration and the deadlock in local extreme the limited lifetime of individual can be used. Limited lifetime is realized by the 'death' operator [5], which represents something like continual restart of GA. This operator enables decreasing of population size as well as increasing the speed of convergence. Every individual has the additional information – age. A simple counter that is incremented in each of GA iteration represents the age. If the age of any member reaches the preset lifetime limit LT, this member 'dies' and is immediately replaced by a new randomly generated member. The age is not mutated nor crossed over. The age of new individuals (incl. individuals created by crossover) is set to zero. Lifetime limit is useful set from 5 to 20 in dependence of the population size and the typical number of iterations. It is necessary to store the best solution obtained separately – the corresponding individual need not to be always present in the population because of the limited lifetime.

Many GAs are implemented on a population consisting of haploid individuals (each individual contains one chromosome). However, in nature, many living organisms have more than one chromosome and there are mechanisms used to determine dominant genes. Sexual recombination generates an endless variety of genotype combination that increases the evolutionary potential of the population. Because it increases the variation among the offspring produced by an individual, it improves the change that some of them will be successful in varying and often-unpredictable environments they will encounter. The modeling of sexual reproduction is quite simple. The population is divided into two parts – males and females [13, 14]. One parent from every part is selected for crossover. The sex of individual is stored in the special gene; this gene is not mutated. The sex of descendant is determined by crossover the sexual genes of parents, descendant is placed to the corresponding part of population. There can be different selection strategies of parent of different sex. For example analogy of territory defending male can be used. In this case the best member of the 'male' part of the population is chosen for the crossover with some high probability, otherwise is chosen randomly. The second parent is chosen sequentially from the second ('female') part of the population.

The replacement scheme (implementation of the Ψ operator) brings another problem. The simplest possibility is the generational replacement where the whole population is replaced in every iteration. The behavior of GA will not be monotonous in this case, because genetic operators do not guarantee the enhancement of the best individual in every next iteration (the best individual in the next generation may be even worse then the best individual in the previous one). To protect monotonous behavior the incremental replacement (steady-state replacement) was introduced. We can use least-fit member replacement where one (or more) element with the worst fitness is replaced, or we can replace randomly chosen element (elements). The elitism brings a way to keep monotony while generational replacement is used. One or several best individuals represent the elite. The whole elite is directly taken to the next iteration. In general we may say, that it is not necessary to replace all members of population by newly created individuals. The crossover may generate the number of individuals between the quarter and the half of the population size. Created individuals should be sorted into the corresponding places in the population according to their fitness in such a way that the size of the population remains the same. Newly created individuals of low fitness do not have to be involved in the population.

Genetic algorithms commonly use heuristic and stochastic approaches. From the theoretical viewpoint, the convergence of heuristic algorithms is not guaranteed for the most of application cases. That is why the definition of the stopping rule of the GA brings a new problem. We must notice that a typical GA does not produce the solution enhancement in every iteration. The stagnation can be observed in following situations:

- Temporary stagnation, GA is able to continue.
- Local extreme was reached and population is degenerated, GA is probably not able to continue.
- The optimal solution was found (this solution is unknown, so this situation cannot be simply identified).

It can be shown [5, 7], that while using a proper version of GA the typical number of iterations can be determined.

4. Conclusions

The paper presents principle ideas behind the development of GAs, one of the possible formal schemes for the description of GAs related to implementation needs, and especially it discusses original ideas and implementation details. At the beginning, various application areas have been cited.

Acknowledgement

This contribution has been prepared with the financial support of the Ministry of Education, Youth and Sports, project MSM0021630519 'Progressive reliable and durable load bearing structures', by a grant from the Grant Agency of the Czech Republic reg. no. 103/08/1658 and by project CZ 0046 – FM EEA and Norway (subproject B/CZ0046/2/0021 and subproject B/CZ0046/3/0035)

References

[1] Purves W.K., Orians G.H., Heller H.C.: Life, the science of biology, Sinauer Associates, Inc., 1992

[2] Ošmera P.: Integrated Evolutionary Algorithms, In Proceedings of MENDEL'2000, Brno, Czech Republic, p. 113–117, 2000

[3] Kvasnička V., Pospíchal J., Tiňo P.: Evolučné algoritmy, STU Bratislava, 2000

[4] Ryan C.: Shades. Polygenic Inheritance Scheme, in Proceedings of the 3th International Conference on Soft Computing MENDEL '97, Brno, Czech Republic, 1997, pp. 140–147

[5] Roupec J.: Design of Genetic Algorithm for Optimization of Fuzzy Controllers Parameters, (In Czech) PhD Thesis, Brno University of Technology, Brno, Czech Republic, 2001

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

[7] Roupec J., Popela P., Ošmera P.: The Additional Stopping Rule for Heuristic Algorithms, in Proceedings of the 3th International Conference on Soft Computing Mendel '97, Brno, Czech Republic, 1997, pp. 135–139

[8] Dawkins R.: The Selfish Gene, Oxford University Press, Oxford 1976

[9] Ošmera P., Roupec J.: Limited Lifetime Genetic Algorithms in Comparison with Sexual Reproduction Based GAs, In Proceedings of MENDEL'2000, Brno, Czech Republic, p. 118–126, 2000

[10] Ošmera P., Roupec J., Matoušek R.: Genetic Algorithms with Diploid Chromosomes and Sexual Reproduction, In book: Sinčák P., Vaščák J.: Quo Vadis Computational Intelligence? Physica – Verlag, A Springer Verlag Company, pp. 317–323, 2000

[11] Ošmera P., Roupec J., Matoušek R.: Genetic Algorithms with Sexual Reproduction. In Proceedings of SCI 2000, Orlando, USA, pp. 306–311, 2000

[12] Ošmera P., Roupec J.: Integrated Evolutionary Algorithms, In book Abraham A., Köppen M.: Advances In Soft Computing – Hybrid Information Systems, Physica – Verlag, Springer – Verlag Company, 2001, pp. 353–360

[13] Ošmera P., Roupec J., Matoušek R.: Genetic Algorithms with Sexual Reproduction. In Proceedings of World Multiconference on Systemics, Cybernetics and Informatics SCI 2000, Orlando (Florida, USA), 2000, pp. 306–311

[14] Ošmera P., Roupec J., Matoušek R.: Genetic Algorithms with Diploid Chromosomes and Sexual Reproduction, In Studies in Fuzziness and Soft Computing, vol. 54, Physica – Verlag, Springer – Verlag Company, 2000, ISSN 1434-9922, pp. 317–323

[15] Laníková I., Štěpánek P., Šimůnek P.: Fully probabilistic design of concrete structures 16th International Conference on Soft Computing MENDEL'2010, p. 426–433

[16] Karpíšek Z., Lacinová V.: System Reliability Computed by Interval Arithmetic 16th International Conference on Soft Computing MENDEL, 2010, pp. 565–570

[17] Matoušek R.: Genetic Algorithm and Advanced Tournament Selection Concept, In Proceedings of NICSO 2008: Studies in Computational Intelligence, vol. 236, pp. 189–196, 2009

[18] Popela P., Vlk M.: Bayesian Approach and Other Updating Techniques, Chapter 16, In Marek P., Brozzetti J., Guštar M., Tikalsky P.: Probabilistic Assessment of Structures using Monte Carlo Simulation, 2nd edition, ITAM. Semily: TERECO, GLOS, 2003, s. 404–422

[19] Štěpánek P., Zlámal M.: Additional strengthening of masonry vaulted structures by nonprestressed reinforcement, ACEE – Journal of architecture, civil engineering and environment, 2009, 2 (3), p. 85–90

[20] Popela P.: Numerical Techniques and Available Software, Chapter 8 in Part II, In Dupačová J., Hurt J., Štěpán J.: Stochastic Modeling in Economics and Finance, Applied Optimization, Dordrecht/Boston/London: Kluwer Academic Publishers, 2002, s. 206–227

[21] Roupec J., Popela P.: Scenario generation and analysis by heuristic algorithms, In Proceedings of World Congress on Engineering and Computer Science (WCECS), pp. 931–935, San Francisco, CA 2007

[22] Žampachová E., Popela P., Mrázek M.: Optimum beam design via stochastic programming, Kybernetika, vol. 46, pp. 575–586, 2010

[23] Dauzère-Pérèz S., Nordli A. , Olstad A. , Haugen K.K., Myrstad P.O., Teistklub G., Reistad A.: Omya Hustadmarmor : Optimizing the supply chain of calcium carbonate slurry to the European paper making industry. INTERFACES – finalist in the Franz Edelemann competition, Vol. 37, No. 1: 1–13

Received in editor's office: April 1, 2010
Approved for publishing: January 20, 2011

Note : This paper is an extended version of the contribution presented at the international conference *STOPTIMA 2007* in Brno.