STUDIA UNIVERSITATIS BABEŞ-BOLYAI, CHEMIA, L, 2, 2005

GENETIC ALGORITHMS OPTIMISATION AND ITS USE IN CHEMICAL ENGINEERING

ROMAN BOCHENEK, JACEK JEŻOWSKI*, ALINA JEŻOWSKA

* *Rzeszów University of Technology, Department of Chemical and Process Engineering,*
35-959 Rzeszów, Al. Powstanców Warszawy 6, Poland,
* email: *ichjj@prz.rzeszow.pl*

ABSTRACT. The paper deals with applications of genetic algorithms (GA) optimisation in chemical and process engineering. A novel optimisation procedure has been developed that is aimed at reducing an effect of premature degeneration. The description of optimisation algorithm is given. Solutions of two chemical engineering problems are also presented. The results show that the developed GA optimisation subroutine is efficient tool for solving complex nonlinear problems of chemical and process engineering.

Keywords: optimisation, genetic algorithms, reaction equilibrium, batch plant

1. INTRODUCTION

Genetic algorithm is relatively modern stochastic optimisation strategy. Applications in chemical in process engineering range from structural and parametric optimisation of processes, apparatus and systems (see e.g. [1]) via process modelling (e.g. [2]) to expert systems (for instance [3]).

GA strategy has an opinion of efficient optimisation technique for global optimisation of combinatorial problems, such as, for instance, scheduling of batch processes. The application to problems with only continuous variables is possible though rare. Deterministic approaches are expected to be more reliable tools for such problems. However, one should take into account such difficulties with deterministic methods as e.g. discontinuities of functions.

Similarly to other stochastic optimisation methods such as adaptive random search (ARS) or simulated annealing (SA) there are many versions of GA. In fact, only few common rules are applied in majority of GA optimisation techniques published in the literature, such as:

- fixed number of members in generated populations,
- given number of generated populations applied commonly as stopping criterion,
- crossover operator is the basic mean for genetic changes (though there exist many types of such operator) while mutation is treated as auxiliary operator.

Some procedures developed in the literature are quite complex while others simple. It is difficult to assess effectiveness of some versions suggested in the literature since insufficient tests had been carried out. More sophisticated versions usually ensure higher probability of locating global optimum. Such approaches need, however, more control parameters which have to be tuned by the user for specific problem. This requires a lot of trials and, also, user's experience.

In this work we present a GA-based optimisation method that is designed for general nonlinear mixed-integer optimisation problem (MINLP). We have attempted to reach a compromise between reliability of locating global optimum (that usually requires complex algorithm, many operators and numerous control parameters) and easiness of application (that requires a simple algorithm and a small number of control parameters).

In the following we present a description of the developed method. Then, examples of applications from chemical and process engineering are given.

2. GENETIC ALGORITHMS

The developed approach is aimed at solving general problem as follows:

$$\min FC(\boldsymbol{x},\boldsymbol{y}) \tag{1}$$

s.t.

$$h_k(\boldsymbol{x},\boldsymbol{y}) = 0\ , \qquad k=1,\ \ldots,\ K \tag{2}$$

$$g_l(\boldsymbol{x},\boldsymbol{y}) \le 0\ , \qquad l=1,\ \ldots,\ L \tag{3}$$

$$x_i^L \le x_i \le x_i^U\ , \qquad i=1,\ \ldots,\ I \tag{4}$$

$$y_j^L \le y_j \le y_j^U\ , \quad y_j \in \boldsymbol{D}_j, \qquad j=1,\ \ldots,\ J \tag{5}$$

where: $\boldsymbol{x}$ is vector of continuous variables and $\boldsymbol{y}$ is vector of discrete variables.

In contrast to early GA methods our algorithm does not use traditional chromosome code. We have applied easy and "natural" for technical problems representation where chromosome is defined by a vector of decision parameters. To cope with inequality constraints a well known penalty function approach was adapted, i.e. the original goal function was augmented by penalty terms.

Equality constraints are dealt for in two ways:

1) by replacing each equality by a pair of inequalities:

$$h_k(\boldsymbol{x},\boldsymbol{y})=0 \quad \rightarrow \quad h_k(\boldsymbol{x},\boldsymbol{y}) \ge -\varepsilon \ \text{ and } \ h_k(\boldsymbol{x},\boldsymbol{y}) \le \varepsilon \tag{6}$$

where: ε is a sufficiently small number

2) by determining values of some variables sequentially from equations or by solving simultaneous equations set.

The latter approach is more efficient since it reduces number of degrees of freedom. However, it is necessary to find such independent variables (i.e. the variables that are generated by optimisation algorithm) that linear equations are created in regards to the dependent variables. This is greatly facilitated by a possibility of applying sequential calculations in frame of GA.

In order to increase chances of survival of the best fitted individuals and to decrease an effect of "premature degeneration" we applied the so called "genetically modified sub-population". Members of this sub-population are created by using genetic operators to randomly chosen members of the parent population. Next, we form an intermediate population that is the superset of the parent population and genetically modified sub-population. The selection mechanism chooses the individuals that create the parent population of next generation. It is worthwhile noting that the sub-population has less number of members than the parent population and the number of its members is determined by a control parameter called modification ratio *u_mod*.

The procedure of genetic algorithm optimisation is given in the following. Figure 1 illustrates the main steps.

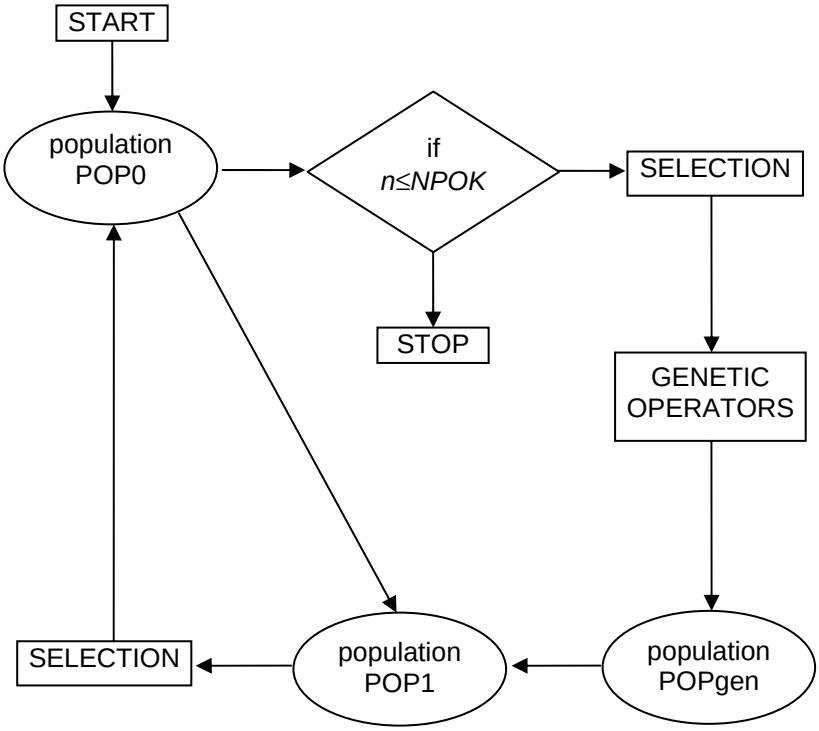

Figure 1. Flow sheet of optimisation algorithm

1) input values of control parameters
- population size *NPOP*,
- total number of populations generated in optimisation *NPOK*,
- number of genetic operators *NOPER*,
- modification ratio *u_mod*,
- probabilities of using the operators p_gen_i (i=1,..., *NOPER*)

[As in many others GA optimisation methods the population size is kept constant during optimisation. Also, given total number of generated populations is used as the stopping criterion.]

2) Calculation of parameters for fitness function.

Fitness function with penalty terms is calculated from:
- for minimisation: $FP=C_{max}-FC-CK$ (7)
- for maximisation: $FP=C_{min}+FC-CK$ (8)

Parameters C_{max}, C_{min} are calculated as follows. A set of many individuals (e.g. 10.000) is randomly generated and parameters C_{max}, C_{min} are determined from:

- for minimisation: C_{max}= abs(max{*FC*+*CK*}) (9)
- for maximisation: C_{min}=abs(min{*FC*-*CK*}) (10)

where abs () is the operator transforming negative into positive value

3) calculation of genetically modified population size
NPOPgen=int (*u_mod***NPOP*)
where int () is the operator that transforms real number into integer one

4) creation of initial population *POP0*

5) calculation of fitness function for members of population *POP0* from (7) for minimisation or from (8) for maximisation.

6) calculation of probabilities $p0_i$ (*i*=1, ...,*NPOP*) for selecting individuals from population *POP0*.

7) creation of genetically modified sub-population
a. selection of genetic operator *i* with probability p_gen_i
b. selection of individual (individuals) *i* for genetic modifications with probability $p0_i$
c. creation of modified individual (individuals) *i* by using selected operator
d. points a) and b) are performed until given size of sub-population *NPOPgen* is reached

8) calculation of fitness function for members of sub-population *POPgen*.

9) calculation of probabilities $p1_i$ (*i*=1,...,*NPOP*+*NPOPgen*) for selecting individuals from the superset consisting of sub-population and parent population

10) creation of offspring population having *NPOP* members by choosing members from the superset consisting of parent population and sub-population with probability $p1_i$ (the superset is intermediate population)

11) the population from previous step is copied into parent population of next generation

12) points (5) to (11) are performed until generation number is greater than *NPOK*

In the present version of the algorithm we applied the following genetic operators:

1. Simple crossover. Two parents exchange information by interchanging parts of their chromosomes. For selected parents and randomly chosen cutting position "*k*" in chromosomes the operation is given by:

$$\{x_1,x_2, ...,x_n\} + \{y_1,y_2, ...,y_n\} \rightarrow \{x_1,x_2,...,x_k,y_{k+1},...,y_n\} + \{y_1,y_2,...,y_k,x_{k+1}, ...,x_n\}$$

2. Arithmetic crossover. This operation though operates on two parents gives only one offspring that is linear combination of parents as follows:

$$\{x_1,x_2, ...,x_n\} + \{y_1,y_2, ...,y_n\} \rightarrow \{r*x_1+(1-r)*y_1,r*x_2+(1-r)*y_n,..., r*x_n+(1-r)*y_n\}$$

3. Heuristic crossover. Similarly to the arithmetic crossover also one offspring is produced from two parents. However, offspring's chromosome depends on fitness functions of parents according to:

$\{x_1,x_2, ...,x_n\} + \{y_1,y_2, ...,y_n\} \rightarrow P$

$P=\{x_1+r(x_1- y_1), x_2+r(x_2- y_2),...,x_n+r(x_n- y_n)\}$, if $FP(X)>FP(Y)$

or

$P=\{y_1+r(y_1- x_1), y_2+r(y_2- x_2),...,y_n+r(y_n- x_n)\}$, if $FP(Y)>FP(X)$

4. Uniform mutation. One parent is selected and, then, randomly chosen position "*k*" of chromosome is mutated in such the way that its value is changed by the value chosen from the entire range - see eqs (4, 5). Selection of the position in chromosome and choice of the value is performed using uniform distribution.

5. Non-uniform mutation. This operator differs from the previous one since the value of parameter $x[k]$ is chosen from the range given by:

$<x[k]-\Delta, x[k]+\Delta>$. Parameter Δ is defined by the formula:

$$\Delta= (x^U - x^L)*(1-r^{(1-n/NPOK)*B}) \quad (11)$$

where: *n*- current number of generation, *r* - random number from the range <0,1>, *B* - constant parameter.

It is important to note that "Δ" is the function of number of populations generated to the moment and "Δ" increases with an increase of this number. In consequence, range of values for parameter "*k*" diminishes in course of optimisation.

6. Local mutation. Randomly selected position $x[k]$ from also randomly chosen individual is changed according to: $p[k]=x[k]\pm\Delta$, where: $\Delta=(x^U-x^L)/C$, *C* – operator's parameter.

7. Range limit mutation. Randomly chosen position $x[k]$ of individual's chromosome is given upper (x^U) or lower (x^L) limiting value with uniform distribution. The operator may have significant effect in case where optimum values of some variables are at their range limit.

3. EXAMPLES OF APPLICATIONS

At present the optimisation method is in course of testing procedure. The aim is to test its robustness and computational efficiency for a wide variety of optimisation problems that can be met in chemical and process engineering. Also, we intend to find possibly narrow ranges of values for control parameters. The results will be published in forthcoming work. Some general guidelines are given in summary section. In this paper we present results for two problems from chemical engineering.

3.1. Chemical reaction equilibrium

The problem was first given in [4] and considered also in [5-7]. This is minimisation of Gibbs free energy for chemical reaction system, i.e. determination of reaction equilibrium composition. For *N* species, temperature *T* and pressure *P* the free energy function is approximated by equation (12) according to [4]:

$$\frac{G_M}{RT} = \sum_{i=1}^{N} x_i \left(\frac{G_i^o}{RT} + \ln P + \ln \frac{x_i}{\sum_{i=1}^{N} x_i} \right) \quad (12)$$

where: G_i^o – standard free energy of specie *i* at temperature *T*, *i* – denotes specie, x_i – moles number of specie *i*, *P* – pressure, G_M –Gibbs free energy of a system.

Constraints of material balance have to be added, such that:

$$\sum_i a_{ij} x_i = b_j \quad j=1, \ldots, \text{number of atom types in the system} \tag{13}$$

where: a_{ij} – number of type „*j*" atoms in specie „*i*" after reaction, x_i –moles number of specie „*i*", b_j – number of type „*j*" atoms

We consider here literature example from [4] - calculation of the composition of gases after combustion of stoichiometric mixture of hydrazine and oxygen at 3500 K and 750 kPa. Ten species can potentially exist in a system after reaction: (i=1,...,10): H, H_2, H_2O, N, N_2, NH, NO, O, O_2, OH (H, O are radicals). There are three (3) types of atoms: (j = 1, ..., 3, j = 1 denotes H, j = 2 denotes N and j = 3 denotes O).

The optimisation model for hydrazine combustion is given in the following (symbol c_i denotes parameters (G_i^o/RT+ln$P$) in eq. (12)). Values of c_i were taken from [4] .

$$min \sum_{i=1}^{10} x_i \left(c_i + ln \frac{x_i}{x_1 + \ldots + x_{10}} \right) \tag{14}$$

subject to:

$$x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2 \tag{15}$$

$$x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1 \tag{16}$$

$$x_4 + 2x_5 + x_6 + x_7 = 1 \tag{17}$$

vector of parameters c_i:

$\boldsymbol{c}$=[-6.089 -17.164, -34.054 -5.914 -24.721 -14.986 -24.100 -10.708 -26.662 -22.179]

Optimal solution according to [7] is:

$\boldsymbol{x}^*$ = [0.04034785 0.15386976 0.77497089 0.00167479 0.48468539
0.00068965 0.02826479 0.01849179 0.03849563 0.10128126],

$f(\boldsymbol{x}^*)$ = - 47.760765

The problem has seven degrees of freedom since three equality constraints can be used to eliminate three variables. We solved the problem in this way and obtained the global optimum $f(\boldsymbol{x}^*)$ =-47.7602 using 6 members in population, 5000 populations and with modification ratio *u_mod*=0.95. The genetic operators were given the following probabilities: simple crossover – 0.1, arithmetic crossover – 0.1, heuristic crossover – 0.2, uniform mutation – 0.05, non-uniform mutation – 0.25, local mutation – 0.25, range limit mutation – 0.05.

3.2. Design of a Multi-product Batch Plant

The plant consists of a sequence of M batch processing stages which are used to manufacture N different products. In each stage j, R_j units operate independently in parallel and all the units within a given stage j have the same size V_j. The time required to process one batch of product i in stage j is given by t_{ij}, where $t_{ij} \geq 0$.

The required unit size, V_{ij}, for processing product i in stage j is given by

$$V_{ij} \geq S_{ij} B_i , \quad i=1,\ldots,N, \; j=1,\ldots,M \tag{18}$$

where B_i (kg) is the batch size for product i, S_{ij} is a constant. The unit sizes V_j must then be chosen to satisfy

$$V_j \geq V_{ij} , \quad i=1,\ldots, N, \; j=1,\ldots, M \tag{19}$$

and they are available in a continuous range of sizes

$$V_j^L \le V_j \le V_j^U \,,\ j=1,\ldots, M \tag{20}$$

where V_j^L, V_j^U are given limits.

The plant operates in a steady cyclic condition, and no auxiliary storage is available, so each product batch, once started, must be processed through all stages without any waiting time. Over a given period of operation H (h), the number of batches, n_i, of each product, and their sizes B_i, must be chosen to achieve the required production Q_i (kg).

$$Q_i = n_i B_i \tag{21}$$

The number of units in each stage must be chosen so that all the batches can be processed. The resulting sequencing and capacity constraints are complex, and require the introduction of a large number of auxiliary zero-one variables. However, a much simpler constraint, which gives good sub-optimal solutions was obtained in [9] by the following reasoning.

One average, the time taken to process the n_i batches of product i in stage j will be $(n_i t_{ij}/R_j)$, and clearly we must have

$$H \ge \sum_{i=1}^{N} n_i t_{ij}/N_j \,,\quad j=1,\ldots, M \tag{22}$$

The sequencing constraints prevent the continuous use of all units in all stages, so that equality cannot in general be obtained for all j in (22). On the other hand, it is clear that the scheduling can always be done if the period H is long enough to allow the maximum cycling time, T_{Li}, given by

$$T_{Li} = \max_{j=1,\ldots,M} (t_{ij}/N_j) \,,\quad j=1,\ldots, N \tag{23}$$

in every stage. This leads to the constraint

$$H \ge \sum_{i=1}^{N} n_i T_{Li} \,, \tag{24}$$

The true optimum lies between the limits (22) and (24) and if this difference is large it might be worthwhile to consider the general sequencing constraints, but here we are using (23) and (24) following other authors.

The design problem is thus to choose the B_i, V_j, and R_j to minimize the capital cost C of the plant, given by

$$C = \sum_{j=1}^{M} \alpha_j R_j V_j^{\beta_j} \tag{25}$$

It is clear that the number of batches (n_i) treated in a given arbitrary period H need not be integer-value, but the number of units (R_j) in each stage must of course be positive integers so the problem is a mixed integer nonlinear program (MINLP). This class of problem is very difficult to solve.

By elimination of the intermediate variables, this relaxed problem may be formulated as follows. Chose the B_i, T_{Li}, V_j, R_j to minimize (25) subject to:

$$\sum_{i=1}^{N} \frac{Q_i \cdot T_{Li}}{B_i} \le H \tag{26}$$

$$V_j \geq S_{ij} B_i \,; \quad R_j \cdot T_{Li} \geq t_{ij} \,, \quad i=1,...,N,\ j=1,...,M \tag{27}$$

Problem has been solved for two sets of data that differs largely as for scale of combinatorial complexity:

- N=2 products in M=3 stages what is equivalent to 27 combinations of discrete variables with 13 inequalities
- N=5 products in M=6 stages what is equivalent to 4096 combinations of discrete variables with 61 inequalities.

Two formulations of the optimisation problem has been applied, original one as described above that requires solution of MINLP problem and modified one developed by the authors which requires solution of NLP problem only.

The global optima according to [8] are:

for data set (i):

C=38499.8; R_j[1 1 1]; V_j[480 720 960]; B_i[240 120]; T_{Li}[20 16].

for data set (ii):

C=285510; R_j[2 2 3 2 1 1]; V_j[3000 1891.64 1974.683 2619.195 2328.100 2109.797]; B_i[379.7467 770.3054 727.5089 638.2978 525.4531]; T_{Li}[3.2 3.4 6.2 3.4 3.7].

This problem is difficult for GA methods. At first it has many continuous variables and relatively few discrete ones, hence, it is not of highly combinatorial nature. As it was mentioned such problems are rather seldom solved in chemical engineering using GA methods. At second, the problem is ill-defined in regards to constraints at the global optimum. The analysis showed that changes of decision variables of order 0.01% around optimum values (in both directions) cause that the solution becomes infeasible. It can be considered that many constraints are practically active. Such optimisation problems are extremely difficult for GA since the method isn't able to ensure high precision for continuous variables.

Optimisation of this multi-product batch plant has been considered in several works and researchers attempted to solve it using both deterministic as well as stochastic optimisation methods. Authors [9] successfully solved this problem but only for data set (i) applying branch and bound approach. Advanced General Benders Decomposition approach, the same as used in well known commercial solver DICOPT from GAMS [10] was employed by authors [8]. It was found that solutions obtained depend on starting point and only linearisation by logarithmic transformation ensured the global optimum independent on initialisation.

Stochastic strategy from adaptive random search (ARS) class was applied in work [11]. It gave sufficient reliability only for data set (i). At last, authors [12] applied simulating annealing (SA) strategy. Also in this case sufficient reliability was reached for data set (i).

The authors developed on the basis of problem analysis a modification that allowed to eliminate discrete variables R_j and continuous ones V_j. Additionally inequality constraints were eliminated, too. In result, the MINLP problem was simplified to NLP one with reduced number of continuous variables and number of inequality constraints as well.

The reasoning which leads to the modification is as follows.

It can be concluded from the analysis of the goal function that variables N_j and V_j (j=1,...,M) have to reach admissible minimal values that meet constraints (27), which, in turn limit the variables from the lower bounds. For each j-th stage at least one of the constraints on products i=1,...,N must be active in the optimal solution. Since lower bounds on N_j and V_j determined by (27) depend only on B_i and T_{Li} , hence, the latter should be chosen as decision (independent) variables in optimization. Other variables become dependent ones and can be determined from active constraints (27), which, after transformation, become equations (28,29).

$$N_j = \max_i \left[int\left(\frac{t_{ij}}{T_{Li}} \right) \right], \quad \text{for } i=1,...,N,\ j=1,...,M \tag{28}$$

$$V_j = \max_i \left(S_{ij} B_i \right), \quad \text{for } i=1,...,N,\ j=1,...,M \tag{29}$$

where operator int(x) transform real x to integer one such, that: int(x)>x, function max[B] returns the element from set B of maximal value.

Thus, the original problem with both continuous and discrete variables defined as MINLP has been reformulated as NLP one with significantly reduced number of decision variables and, also, inequality constraints. It is of importance that such reformulation is possible only in frames of stochastic approaches, such as GA, where one can make direct use of computer programming language applied to code the optimisation method.

Using this NLP formulation it was possible to reach significantly better results even for data set (ii). Table 1 summarises results of our computations for data set (i) and table 2 for data set (ii).

Table 1.

Results of calculations for data set (i)

Optimisation model type	Goal function values	Total no. of population	Population size	Number of function evaluations
MINLP	38557.5	8000	28	224000
NLP	38499.5	5000	16	80000

Table 2.

Results of calculations for data set (ii)

Optimisation model type	Goal function values	Total no. of population	Population size	Number of function evaluations
MINLP	312386.3	20000	24	480000
NLP	285787.3	10000	56	560000

As one can expect better reliability was achieved for NLP formulation. For both data sets the global optimum was achieved with moderate number of goal function evaluations. In case of MINLP formulation the method located the global optimum for data set (i). For the second data set (ii) the results differ from global optimum but

this is negligible difference. Additionally, it is worth noting that our results are close to those obtained with application of other stochastic approaches as ARS from [11] or SA from [12]. The optimal rate of evolution was reached with modification ratio *u_mod* from range 0.15-0.30. The probabilities of using operators that we consider proper for the problem are of order: simple crossover – 0.2, arithmetic crossover – 0.12, heuristic crossover – 0.5, uniform mutation – 0.02, non-uniform mutation – 0.1, local mutation – 0.02, range limit mutation –0.04.

4. SUMMARY

The GA based optimisation method has been developed. The approach has a novel feature that was developed to decrease an effect of premature degeneration which prevents locating global optimum. In our method an intermediate population is created that is a superset of parent population and genetically modified sub-population. The latter consists of modified individuals created by genetic operators applied to randomly selected members of parent population. The size of sub-population is smaller than that of parent one and determined by control parameter *u_mod*. Members of intermediate population compete with each others during selection to the next generation. It was proved in our tests that addition of such the sub-population prevents premature degeneration. The algorithm of optimisation approach is given in the paper as well as description of seven genetic operators used in the method.

Application of the developed method for two chemical engineering problems is reported. The first problem consists in minimisation of Gibbs free energy of chemically reacting system – see also papers [4-7]. This is NLP task with seven degrees of freedom. The second problem is of MINLP type. This is structural and parametric optimisation of multi-product batch plant considered also in [8,9,11,12]. For second data set it is very difficult for solution to global optimum for deterministic and stochastic optimisation approaches.

For both problems global optimum solution has been obtained with our method. It is worthwhile noting that less number of goal function evaluations were needed in comparison with other stochastic approaches as e.g. simulating annealing from [12] or adaptive random search of authors [11].

The GA method has been also successfully applied to solve complex heat exchanger network retrofit design problem [13] as well as to smaller MINLP tasks from process engineering. The optimisation method was also tested using some typical problems for stochastic optimisation. The approach showed sufficient robustness and can be recommended for solving typical NLP problems.

It can be concluded that the developed optimisation approach is robust technique for solving MINLP problems of chemical and process engineering. We are in course of investigations aimed at finding good values of control parameters. Here, we limit ourselves to some general guidelines:

- number of members (*NPOP*) should 4 to 10 times higher than number of variables
- parameter *u_mod* should be of order 0.5-0.55
- probabilities of the genetic operators should be higher than 0.0 (i.e. we should use all the operators available). However, the influence of their values is complex and problem dependent.

SYMBOLS

CK	sum of penalty terms in augmented goal function
$\boldsymbol{D}_j$	set of values for "j-th" discrete variable
FC	original goal function in optimisation
FP	fitness function
g	inequality constraint
h	equality constraint
MINLP	Mixed Integer Non-Linear Programming
NLP	Non-Linear Programming
x/***x***	continuous variable / vector of continuous variables
y/***y***	discrete variable / vector of discrete variables

Subscripts

L	refers to lower limit
U	refers to upper limit

REFERENCES

1. C. Azzaro-Pantel, L. Bernal-Haro, P. Baudet, S. Domenech, L. Pibouleau, *Comput. Chem. Engng,* **1998**, *22*, 1461-1481.
2. D.J. Greeff, C. Aldrich, *Comput. Chem. Engng,* **1998**, *22*, 995-1005.
3. D.A. Manolas, G.A. Efthimeros, D.T. Tsahalis, *Int. Conference CHISA'98*, Prague, Czech Republic **1998**.
4. W.B. White, S.M. Johnson, G.B. Dantzig, *J. Chem. Phys.*, **1958**, *5*, 751-755.
5. M.W. Heuckroth, J.L. Gaddy, L.D. Gaines, *AIChE Journal*, **1976**, *22*, 744-750.
6. G.P. Rangaiah, *Comput. Chem. Engng*, **1985**, *9*, 395-400.
7. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, 1996.
8. G.R. Kocis, I.E. Grossmann, *Ind. Eng. Chem. Res.*, **1988**, *27*, 1407-1421.
9. Grossmann, I.E., Sargent, R.W.H., *Ind. Eng. Chem. Process Des. Dev.*, **1979**, *18*, 343-348.
10. A. Brooke, P. Kendrick, A. Meeraus, "GAMS. A users guide. Release 2.25", The Scientific Press Series 1992.
11. R.L. Salcedo, *Ind. Eng. Chem. Res.*, **1992**, *31*, 262-273.
12. M.F. Cardoso, R.L. Salcedo, S.F. Azevedo, D. Barbosa, *Comput. Chem. Engng*, **1997**, *21*, 1349-1364.
13. R. Bochenek, Optimal retrofit design of heat exchanger networks by genetic algorithms, Ph D Thesis, Rzeszow University of Technology, Poland (supervisor J. Jezowski, in Polish language) 2003