Bulgarian Chemical Communications, Volume 48, Number 4 (pp. 713 – 719) 2016

Modified multi-population genetic algorithms for parameter identification of yeast fed-batch cultivation

T. K. Pencheva*, M. K. Angelova

Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences
105 Acad. Georgi Bonchev Str., 1113 Sofia, Bulgaria

Received February 8, 2016; Revised March 12, 2016

In this investigation two new modifications of the standard multi-population genetic algorithm have been developed. Modifications differ from each other in the sequence of implementation of main genetic operators selection, crossover and mutation. The main idea of newly developed modifications is the operator selection to be executed between the operators crossover and mutation, no matter their order. Both modifications, together with the standard one multi-population genetic algorithm, have been investigated for parameter identification of yeast fed-batch cultivation. The obtained results have been compared and the newly proposed modifications have been shown to be as accurate as the standard multi-population genetic algorithms and proven to be even faster.

Keywords: Multi-population genetic algorithms, Genetic operators, Fermentation process, Parameter identification.

INTRODUCTION

Fermentation processes (FP) as representatives of biotechnological processes attract sustained interest due to the fact that they are an indigenous part of many industries such as industrial biotechnology, microbiology and the pharmaceutical industry. FP combine the dynamics of both biological and non-biological processes but their specific peculiarities are predominantly determined by the characteristics of live microorganisms. Since FP are complex dynamic systems with interdependent and time-varying process variables, their modeling, optimization and future high quality control is a real challenge. Adequate modeling of the non-linear FP significantly depends on the choice of a certain optimization procedure for model parameter identification. Conventional optimization methods usually fail in leading to a satisfactory solution [1]. This fact provokes the idea to apply stochastic algorithms, i.e. genetic algorithms (GA). GA are known as a quite promising stochastic global optimization method and have been widely applied to solve different complicated engineering problems [2-5]. Among a number of searching techniques, GA are representatives of the methods inspired by biological evolution and the principle of Darwin's theory of "survival of the fittest". GA are a feature of hard problem solving, tolerant to noise, easy to interface and hybridize. All these properties make GA convenient and more workable for different optimization problems, among them parameter identification and optimization of fermentation processes [6-9].

The standard simple genetic algorithm (SGA) [10] imitates the processes that occur in nature and searches for a global optimum solution using three main genetic operators implementing them in a sequence selection, crossover and mutation. SGA works with "chromosomes" (coded parameters) and starts with a selection of such chromosomes that represent better possible solutions according to their objective function values. Then a new offspring is formed applying the crossover operator. Finally, mutation is applied with deterministic probability, aiming to prevent the failing of all the solutions into a local optimum of the solved problem.

If there are many populations (called subpopulations), that evolve independently from each other, the single-population GA is converted to a multi-population GA (MpGA) [10]. This feature presents MpGA as more similar to nature than SGA. After the isolation time (a certain number of generations), part of the individuals "migrate" – they are distributed between the subpopulations. Similar to SGA, the standard MpGA as originally presented in [10], implements the three main genetic operators in a sequence selection, crossover and mutation. In this investigation this algorithm will be denoted as MpGA_SCM, coming from selection, crossover and mutation. According to [10, 11] the working principle of MpGA_SCM can be shortly presented as shown in Fig. 1.

To imitate the mechanics of natural selection and genetics is enshrined in the "philosophy" of GA. Thus one can make an analogy with the processes occurring in nature and to speculate that

* To whom all correspondence should be sent:
E-mail: tania.pencheva@biomed.bas.bg

for the probability mutation to come first and then crossover it is comparable that both processes occur in reverse order; or perform selection after crossover and mutation, no matter their order. Following this idea altogether five modifications of MpGA_SCM, differing in the sequence of implementation of the main genetic operators, have been developed [12, 13]. They all aim to improve the model accuracy and the algorithm convergence time for the purposes of parameter identification of fed-batch cultivation of *S. cerevisiae*. Table 1 lists the order of the steps to create a new population for five of the developed up to the moment modifications of MpGA_SCM.

As seen from Table 1, there are two modifications of MpGA_SCM that have not yet been considered, namely MpGA_CSM (crossover, selection, mutation) and MpGA_MSC (mutation, selection, crossover).

The aim of the present investigation is two modifications of MpGA, namely MpGA_CSM and MpGA_MSC, to be developed and to be applied for parameter identification of *S. cerevisiae* fed-batch cultivation. Moreover, the influence of the most important GA parameters, namely the generation gap and rates of crossover, mutation, insertion and migration are going to be investigated towards model accuracy, presented by the optimization criterion, and algorithms convergence time.

MATHEMATICAL MODEL OF *S. CEREVISIAE* FED-BATCH CULTIVATION

The cultivation of the yeast *S. cerevisiae* is performed in the Institute of Technical Chemistry – University of Hannover, Germany. The cultivation conditions and full process description details can be found in [1]. The fed-batch cultivation of *S. cerevisiae* considered here corresponds to the so called *mixed oxidative state* according to the functional state modeling approach [1].

```
1.  [Start]
    Generate k random subpopulations each of them with n chromosomes
2.  [Object function]
    Evaluate the object function of each chromosome n in the
    subpopulations
3.  [Fitness function]
    Evaluate the fitness function of each chromosome n in the
    subpopulations
4.  [New population]
    Create a new population by repeating following steps:
     4.1. [Selection]
        Select parent chromosomes from the subpopulation according to
        their fitness function
     4.2. [Crossover]
        Cross over the parents to form new offspring with a crossover
        probability
     4.3. [Mutation]
        Mutate new offspring at each locus with a mutation probability
5.  [Accepting]
    Place new offspring in a new population
6.  [Replace]
    Use new generated population for a further run of the algorithm
7.  [Migration]
    Migration of individuals between the subpopulations after
    following isolation time
8.  [Test]
    If the end condition is satisfied, stop and return the best
    solution in current population, else move to Loop step
9.  [Loop]
    Go to Fitness step.
```

Fig. 1. Structure of the standard MpGA

Table 1. Sequence of algorithm steps implemented in MpGA

MpGA	Algorithm steps
MpGA_CMS (crossover, mutation, selection) [12]	1, 2, 3, *4.2*, *4.3*, *4.1*, 5, 6, 7, 8, 9
MpGA_SMC (selection, mutation, crossover) [12]	1, 2, 3, *4.1*, *4.3*, *4.2*, 5, 6, 7, 8, 9
MpGA_MCS (mutation, crossover, selection) [12]	1, 2, 3, *4.3*, *4.2*, *4.1*, 5, 6, 7, 8, 9
MpGA_SC (selection, crossover) [13]	1, 2, 3, *4.1*, *4.2*, 5, 6, 7, 8, 9
MpGA_CS (crossover, selection) [13]	1, 2, 3, *4.2*, *4.1*, 5, 6, 7, 8, 9

Particularly for this specific functional state, mathematical model of *S. cerevisiae* fed-batch cultivation is described as follows [1]:

$$\frac{dX}{dt} = \left(\mu_{2S} \frac{S}{S+k_S} + \mu_{2E} \frac{E}{E+k_E} \right) X - \frac{F}{V} X \,, \qquad (1)$$

$$\frac{dS}{dt} = -\frac{\mu_{2S}}{Y_{SX}} \frac{S}{S+k_S} X + \frac{F}{V} \left(S_{in} - S \right), \qquad (2)$$

$$\frac{dE}{dt} = -\frac{\mu_{2E}}{Y_{EX}} \frac{E}{E+k_E} X - \frac{F}{V} E \,, \qquad (3)$$

$$\frac{dO_2}{dt} = \left(\frac{\mu_{2E}}{Y_{EX}} \frac{E}{E+k_E} Y_{OE} - \frac{\mu_{2S}}{Y_{SX}} \frac{S}{S+k_S} Y_{OS} \right) X + k_L^{O_2} a \left(O_2^* - O_2 \right), \qquad (4)$$

$$\frac{dV}{dt} = F \,, \qquad (5)$$

where X, S, E, O_2 are respectively the concentrations of biomass, [g/l], substrate (glucose), [g/l], ethanol, [g/l], and dissolved oxygen, [%]; O_2^* – dissolved oxygen saturation concentration, [%]; F – feeding rate, [l/h]; V – volume of the bioreactor, [l]; $k_L^{O_2}a$ – volumetric oxygen transfer coefficient, [1/h]; S_{in} – initial glucose concentration in the feeding solution, [g/l]; μ_{2S}, μ_{2E} – maximum growth rates of the substrate and ethanol, [1/h]; k_S, k_E – saturation constants of the substrate and ethanol, [g/l]; Y_{ij} – yield coefficients, [g/g]. All the functions are continuous and differentiable and all the model parameters fulfill the non-zero division requirement.

The mean square deviation between the model output and the experimental data obtained during cultivation has been chosen as an optimization criterion:

$$J = \sum \left(Y - Y^* \right)^2 \rightarrow min \,, \qquad .. \qquad (6)$$

where Y is the experimental data, Y^* – the model predicted data, $Y = [X, S, E, O_2]$.

MODIFIED MPGA FOR PARAMETER IDENTIFICATION OF *S. CEREVISIAE* FED-BATCH CULTIVATION

This investigation aims to present the development of two modifications of MpGA in which the selection operator is performed between crossover and mutation, namely MpGA_CSM and MpGA_MSC. They are both going to be compared to the standard MpGA_SCM. Table 2 lists the order of the steps to create a new population only for the three kinds of MpGA considered here.

Many operators, functions, parameters and settings in GA can be improved or implemented specifically solving various problems [10]. In this study five of the main GA parameters, namely generation gap (GGAP), and rates of crossover (XOVR), mutation (MUTR), insertion (INSR) and migration (MIGR) have been investigated.

Table 2. Sequence of algorithm steps implemented in MpGA modifications considered here

MpGA	Algorithm steps
MpGA-SCM	1, 2, 3, *4.1*, *4.2*, *4.3*, 5, 6, 7, 8, 9
MpGA-CSM	1, 2, 3, *4.2*, *4.1*, *4.3*, 5, 6, 7, 8, 9
MpGA-MSC	1, 2, 3, *4.3*, *4.1*, *4.2*, 5, 6, 7, 8, 9

Higher values of GGAP do not improve the GA performance, especially when aiming to find a faster solution. Usually mutation is applied randomly, with a low probability – typically varying between 0.01 and 0.1. Higher XOVR leads to quicker introduction of new strings into the population, while a low XOVR may cause stagnation due to the lower exploration rate. INSR determines how many of the produced population individuals are inserted into the new generation. Each MIGR characterized the number of exchanged individuals. According to some statements [14], the range of the main GA parameters investigated here are as follows: GGAP = {0.5, 0.67, 0.8, 0.9}, XOVR = {0.65, 0.75, 0.85, 0.95}, MUTR = {0.02, 0.04, 0.06, 0.08, 0.1}, INSR = {0.5, 0.6, 0.8, 0.9, 1} and MIGR = {0.1, 0.2, 0.4, 0.6, 0.8}. When one of the parameters considered here GGAP, XOVR, MUTR, INSR or MIGR is investigated according to the values mentioned above, the basic values for the other four parameters are chosen as follows: GGAP = 0.8, XOVR = 0.95, MUTR = 0.05, INSR = 0.95 and MIGR = 0.2, hereafter termed as referent points.

The values of the rest GA parameters, type of genetic operators in considered here and MpGA modifications are tuned according to [12]. The values of the GA parameters except the ones investigated here have been accepted as follows: number of variables (NVAR) = 9; precision of binary representation (PRECI) = 20; number of individuals (NIND) = 20; maximum number of generations (MAXGEN) = 100; number of subpopulations (SUBPOP) = 5; number of generation, after which migration takes place between subpopulations (MIGGEN) = 20. The following types of genetic operators are chosen: encoding – binary; reinsertion – fitness-based; crossover – double point; mutation – bit inversion; selection – roulette wheel selection; and, fitness function – linear ranking.

Following model (1)-(5) of *S. cerevisiae* fed-batch cultivation, nine model parameters have to be estimated altogether. All three kinds of MpGA have been consequently applied for the purposes of parameter identification of *S. cerevisiae* fed-batch cultivation. All the computations are performed in a *Matlab 7* environment using the *Genetic Algorithm Toolbox* [15] on a PC Intel Pentium 4 (2.4 GHz) platform running *Windows XP*. All three kinds of GA are terminated when a certain number of generations (in this case 100) are fulfilled. The scalar relative error tolerance *RelTol* is set to $1e^{-4}$, while the vector of absolute error tolerances (all components) *AbsTol* is set to $1e^{-5}$.

The influence of the main GA parameters, namely GGAP, XOVR, MUTR, INSR and MIGR has been investigated for all three kinds of MpGA – two newly developed modifications MpGA_CSM and MpGA_MSC, as well as for the standard MpGA_SCM as a referent point. The investigation is performed in relation to model accuracy and convergence time. Tables 3 and 4 demonstrate the results obtained with respect to GGAP, XOVR, MUTR, INSR and MIGR. Because of the stochastic nature of GA, thirty runs have been performed for each GA parameter value and each algorithm in order for representative results to be achieved. Presented here are the average values obtained.

None of the three MpGA algorithms considered here are preferred towards time convergence. When investigating different GA operators, different MpGA modifications perform the best: i.e. MpGA_CSM is the fastest one at GGAP, XOVR and INSR, while MpGA_MSC is the "winner" at MUTR, and MpGA_SCM – at MIGR.

RESULTS AND DISCUSSION

As seen from Tables 3 and 4, the optimization criterion values obtained with three kinds of MpGA are very similar, varying between 0.0220 and 0.0222 which means less then 1% divergence. This result is very promising due to the fact that newly developed modifications do not cause a loss in accuracy. It is worth to note that with very few exceptions MpGA_CSM and MpGA_MSC lead to a decrease of the convergence time compared to the standard MpGA_SCM. As such, it can be speculated that processing the selection operator between crossover and mutation (no matter their order) needs much less computational time.

Table 3. Influence of GGAP, XOVR and MUTR on the model accuracy and convergence time

		MpGA_SCM		MpGA_CSM		MpGA_MSC	
		J	*t*, [s]	*J*	*t*, [s]	*J*	*t*, [s]
GGAP	0.5	0.0220	100.8910	**0.0221**	**97.6870**	0.0220	98.2970
	0.67	0.0221	112.1720	0.0221	128.8750	0.0221	121.8600
	0.8	0.0221	155.4680	0.0221	163.8590	0.0220	145.6710
	0.9	0.0220	170.2660	0.0221	165.6720	0.0220	166.0150
XOVR	0.65	0.0221	166.2500	**0.0221**	**143.4060**	0.0221	144.6570
	0.75	0.0221	151.1100	0.0222	149.6720	0.0220	153.6870
	0.85	0.0221	154.7660	0.0220	144.8750	0.0220	144.7340
	0.95	0.0221	166.7970	0.0221	149.7810	0.0221	149.7810
MUTR	0.02	0.0221	144.1880	0.0221	140.0000	**0.0221**	**122.0660**
	0.04	0.0221	140.4690	0.0220	145.8910	0.0220	150.4680
	0.06	0.0221	162.0940	0.0221	160.6870	0.0221	159.7660
	0.08	0.0221	156.0940	0.0221	155.9370	0.0221	142.4530
	0.1	0.0221	162.2970	0.0221	147.9060	0.0220	156.3280

Table 4. Influence of INSR and MIGR on the model accuracy and convergence time

		MpGA_SCM		MpGA_CSM		MpGA_MSC	
		J	*t*, [s]	*J*	*t*, [s]	*J*	*t*, [s]
INSR	0.5	0.0221	177.0310	0.0222	159.2970	0.0220	144.9530
	0.6	0.0221	174.2030	0.0221	146.2810	0.0220	149.7650
	0.8	0.0221	185.9530	0.0220	145.5160	0.0220	151.2810
	0.9	0.0221	157.1250	**0.0221**	**142.0310**	0.0220	150.4060
	1	0.0220	169.1410	0.0221	146.7820	0.0221	151.4060
MIGR	0.1	**0.0221**	**143.7190**	0.0221	146.4690	0.0220	147.2500
	0.2	0.0221	163.6880	0.0221	154.4530	0.0221	146.0320
	0.4	0.0221	174.0310	0.0222	149.6250	0.0221	146.6410
	0.6	0.0221	175.6250	0.0222	143.7960	0.0221	155.5310
	0.8	0.0221	181.2970	0.0221	157.4060	0.0220	147.7340

It should be noted that in this investigation the GGAP is the most sensitive of the five investigated parameters toward a convergence time. Considering three kinds of MpGA at GGAP = 0.8 (used as a referent value), there is no significant decrease of the convergence time. But, using GGAP = 0.5 instead of GGAP = 0.8 leads to the fastest performance of all the considered here three kinds of MpGA for all the values of the investigated parameters. The fastest algorithm is MpGA_CSM, while the other two are a bit more accurate. Comparing both MpGA modifications implemented at GGAP = 0.5 towards the standard one MpGA_SCM at GGAP = 0.8 (used as a referent value), MpGA_CSM appears as 1.59, while MpGA_MCS – as 1.58 times faster than MpGA_SCM. Thus, GGAP = 0.5 is chosen as the most appropriate one.

Considering XOVR, the biggest decrease in the convergence time is observed when using XOVR = 0.65 instead of XOVR = 0.95 (used as a referent value) in both MpGA modifications, respectively 14% when applying MpGA_CSM, and 13% for MpGA_MCS in a comparison to the standard MpGA_SCM at XOVR = 0.95. For these two out of three algorithms, XOVR = 0.65 leads to the fastest performances and as such this value is chosen as the most appropriate one.

Considering MUTR, using MUTR = 0.02 instead of MUTR = 0.04 or MUTR = 0.06 (closest to the used as a referent value MUTR = 0.05) leads to decrease of convergence time, respectively, of about 13% towards MUTR = 0.04 and about 25% towards MUTR = 0.06, both achieved when newly presented modification MpGA_MSC is applied and compared to the standard MpGA_SCM. In this case, two out of three algorithms lead to the fastest performances, and as such MUTR = 0.02 is chosen as the most appropriate value.

Respectively almost 16 and 11% of the convergence time can be saved using INSR = 0.9 instead of INSR = 1 (the closest to the used as a referent value INSR = 0.95) when applying MpGA_CSM and MpGA_MSC. In this case again two out of three algorithms, INSR = 0.9 leads to the fastest performances – the standard MpGA_SCM and MpGA_CSM, and as such this value is chosen as the most appropriate one.

Some very promising results are obtained when MIGR is investigated. Again about 10-11% decrease of convergence time is observed when using MIGR = 0.1 instead of MIGR = 0.2 (used as a referent value) in the case of presented MpGA modifications towards the standard MpGA_SCM at MIGR = 0.2. As it can be seen from Table 4, in this case the standard MpGA_SCM is the fastest one. For MIGR the value of 0.1 is chosen, although not all of the algorithms perform the best at this value, but the obtained results are very close to the best results achieved.

As a summary of the detailed analysis presented above, the following values of the GA parameters have been chosen as the most promising ones: GGAP = 0.5, XOVR = 0.65, MUTR = 0.02, INSR = 0.9 and MIGR = 0.1. Developed here are two MpGA modifications that lead to a decrease of the convergence time: MpGA_CSM is the fastest one for three of the GA parameters – GGAP, XOVR and INSR, while another modification of MpGA_MSC is the fastest one for MUTR. Only considering MIGR, the fastest algorithm is the standard one – MpGA_SCM, but two modifications are with very close results with about a 2% bigger convergence time. Finally, if one compares the fastest algorithm, which in this investigation is MpGA_CSM at GGAP = 0.5, to the slowest one, which in this investigation is MpGA_SCM at MIGR = 0.8, it is 1.90 times faster, yielding almost the highest model accuracy.

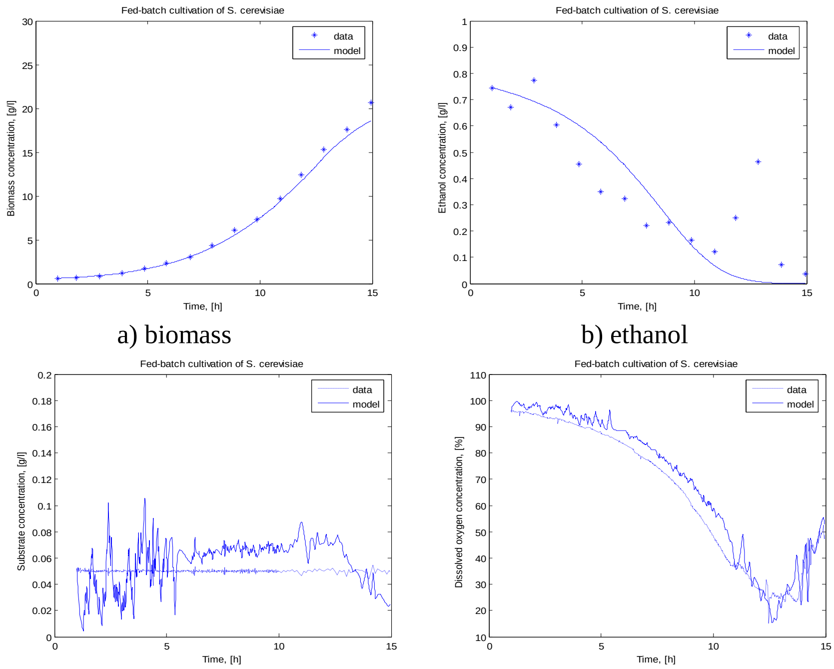
Distinguished as the fastest, the newly developed and presented algorithm, MpGA_CSM is applied for parameter identification of *S. cerevisiae* fed-batch cultivation. The identification procedure is performed with the values chosen due to five GA parameters investigated here and Table 5 lists the evaluated model parameters.

Fig. 2 shows the results from the experimental data and the model prediction, respectively, for biomass, ethanol, substrate and dissolved oxygen when MpGA_CSM is applied.

The results presented in Fig. 2 demonstrate the workability and efficacy of MpGA_CSM as one of the two newly elaborated modifications of MpGA presented here.

Table 5. Results from parameter identification when MpGA_CSM is applied.

Parameter	J	t	μ_{2S}	μ_{2E}	k_S	k_E	Y_{SX}	Y_{EX}	k_La	Y_{OS}	Y_{OE}
Value	0.0221	97.5940	0.90	0.12	0.15	0.80	0.41	1.64	65.20	509.82	360.17



a) biomass b) ethanol

c) substrate d) dissolved oxygen

Fig. 2. Model prediction compared to experimental data when MpGA_CSM is applied.

CONCLUSIONS

In this investigation two newly developed modifications of the standard MpGA are presented. In both modifications, MpGA_CSM and MpGA_MSC, the operator selection is executed between crossover and mutation, no matter their order. The workability and efficacy of the newly elaborated modifications have been demonstrated, together with the standard MpGA_SCM, for the purposes of parameter identification of fed-batch cultivation of *S. cerevisiae*. The investigation of the influence of the most important GA parameters with respect to the convergence time and generation gap have been recognized as the most sensitive among the five parameters examined. About 45% of the convergence time can be saved using GGAP = 0.5 instead of the referent value of GGAP = 0.8 in both MpGA_CSM and MpGA_MSC without a loss in accuracy.

As a whole, newly proposed modifications of MpGA have been shown to be as accurate and effective as the standard one even proved to be faster.

It is noteworthy that the proposed two modifications of MpGA, as representatives of the global search optimization technique, might be considered convenient for model parameter identification in different branches of GA implementations.

Acknowledgements*: This work is partially supported by National Scientific Fund of Bulgaria, Grant DMU 03-38.*

REFERENCES

1. T. Pencheva, O. Roeva, I. Hristozov, Functional State Approach to Fermentation Processes Modelling, Prof. Marin Drinov Academic Publishing House, Sofia, 2006.
2. O. Cordon, F. Herrera, *Fuzzy Sets Syst.*, **118**, 235 (2001).
3. R. Kuo, C. Chen, Y. Hwang, *Fuzzy Sets Syst.*, **118**, 21 (2001).
4. G. Carrillo-Ureta, P. Roberts, V. Becerra, *IEEE Int. Symp. Intell. Control (2001)*, Mexico City, 2001, p. 391.
5. P. Vasant, N. Barsoum, *Eng. Appl. Artif. Intell.*, **22**, 767 (2009).
6. M. Ranganath, S. Renganathan, C. Gokulnath, *Bioprocess Eng.*, **21**, 123 (1999).
7. J. Adeyemo, A. Mehablia, *Sci. Res. and Essays*, **6**(7), 1464 (2011).
8. M. Angelova, P. Melo-Pinto, T. Pencheva, *WSEAS Trans. Syst.*, **11**(7), 256 (2012).
9. M. Angelova, T. Pencheva, *C. R. de l'Acad. Bulg. Scie.*, **65**(3), 299 (2012).
10. D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Wiley, Massachusetts, (1989).
11. D. Gupta, S. Ghafir, *Int. J. Emerg. Techn. Adv. Eng.*, **2**(5), 56 (2012).
12. M. Angelova, S. Tzonkov, T. Pencheva, *Lect. Notes Comput. Scie.*, **6046**, 224 (2011).

13. M. Angelova, T. Pencheva, Improvement of Multi-population Genetic Algorithm Convergence Time, Chapter in: *Sabelfeld K. K., I. Dimov* (*Eds.*), *Monte Carlo Methods and Applications*, 2013, 1-9, De Gruyter, Berlin, Germany, eBook ISBN: 9783110293586,
14. K. Schuegerl, K.-H. Bellgardt (Eds.), Bioreaction Engineering, Modeling and Control, Springer-Verlag, Berlin Heidelberg New York, 2000.
15. A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca, Genetic Algorithm Toolbox for Use with MATLAB, User's guide, version 1.2., UK, 1994.

МОДИФИЦИРАНИ ГЕНЕТИЧНИ АЛГОРИТМИ ЗА ПАРАМЕТРИЧНА ИДЕНТИФИКАЦИЯ НА ПОЛУПЕРИОДИЧНА КУЛТИВАЦИЯ НА ДРОЖДИ

Т. К. Пенчева, М. К. Ангелова

Институт по биофизика и биомедицинско инженерство, Българска академия на науките
ул. Акад. Георги Бончев, бл. 105, София 1113, България

Получена на 8 февруари 2016, Рецензирана на 12 Март 2016

(Резюме)

В настоящото изследване са разработени две нови модификации на стандартните мултипопулационни генетични алгоритми. Модификациите се различават една от друга по реда на изпълнение на основните генетични оператори селекция, кръстосване и мутация. Основната идея на новоразработените модификации е операторът селекция да бъде изпълняван между операторите кръстосване и мутация, без значение от техния ред. Двете модификации, заедно със стандартния мултипопулационен генетичен алгоритъм, са изследвани при параметрична идентификация на полупериодична култивация на дрожди. Получените резултати са сравнени и новопредложените модификации са демонстрирани като също толкова точни, колкото и стандартния мултипопулационен генетичен алгоритъм, но с доказана по-добра сходимост.