Hybrid Fuzzy Algorithm for Solving Operational Production Planning Problems

L.A. Gladkov(✉), N.V. Gladkova, and S.A. Gromov

Southern Federal University, Taganrog, Russia
{leo_gladkov,nadyusha.gladkova77}@mail.ru

Abstract. The article deals with the development of methods of solving operational production planning problems. Authors formulated the operational production planning problem statement, determined constraints and the objective function. The scheme of solutions encoding and modified genetic operators are developed to consider the problem character. Authors proposed the hybrid algorithm model based on integration of genetic search methods and fuzzy control approach. Experimental research of developed algorithms characteristics allows us to determine their time complexity. Obtained results show the effectiveness of suggested approach.

Keywords: Operational production planning problems · Genetic algorithm · Fuzzy logic · Scheduling theory · Optimization · Hybrid algorithm

1 Introduction

In accordance with Manufacturing Enterprise Solutions Association International (MESA) definition, operational (elaborate) planning is considered as the process of production schedule drawing up and estimation based on priorities, attributes, characteristics and methods related to a specific character of products and production technology.

Thus, operational planning is reduced to scheduling theory problems [1–4], which requires:

- to appoint an executor for each job;
- to put in order jobs of each executor, i.e. to find their optimal performance sequence to achieve the assigned goal.

In the scheduling theory ordering problems are considered with the requirement that all issues of what and how should be done are resolved. It is suggested that jobs nature do not depend on their performance sequence. Therewith, following assumptions are to use:

1. All assigned jobs are to be performed and to be defined fully. Decomposition of the jobs set into performed and non-performed classes is not included into the ordering problem.
2. Devices allocated to perform jobs are defined uniquely.

R. Silhavy et al. (eds.), *Artificial Intelligence Trends in Intelligent Systems*,
Advances in Intelligent Systems and Computing 573, DOI 10.1007/978-3-319-57261-1_44

3. A set of elementary operations related to each job performance and a set of constraints on the order of their performance are assigned. The manner in which these operations are carried out is defined. It is assumed, that there is at least one device able to perform each operation.

2 Problem Statement

In terms of mechanical engineering the operational production planning problem combines different classes of scheduling theory problems. There is a set of machines (production line) $\{M\}$, $|M| = m$, each line is characterized by definite list of parameters imposing additional constraints on jobs allocation. The timetable of production lines unavailability determines periods of service or repair works.

- sb_{jl} denotes the beginning of service i on line j;
- se_{jl} denotes the finishing of service i on line j;
- RQ denotes total amount of equipment;
- WT denotes the matrix of reconfiguration periods required for switching from job i to job j. Each element of the matrix $wt_{ij} \geq 0$.

The finite set of jobs is denoted by $\{N\}$, $|N| = n$, where each job i includes an operation. The job is an elementary problem required to be performed, which is characterized by following parameters:

- number of machine m_i allocated to perform job i, $1 < m_i < m$;
- duration of job performance;
- individual schedule date d_i of job i;
- the criterion of need to use the equipment $rq_i \in \{0, 1\}$;
- the incidence matrix of jobs and production lines R, the matrix element r_{ij} represents the selection priority of the line j to perform the job i,

$$r_{ij} = \begin{cases} r_{ij} \geq 0 \\ r_{ij} = \infty \end{cases},$$

where r_{ij} equals ∞, if the job i is not performed on the line j.

The problem requires finding such decomposition of the jobs set N into m disjoint subsets, which provides following conditions:

1. Distribution of jobs across lines corresponds to the incidence matrix $R, \forall m \in M, i \in N_m \Rightarrow r_{im} > 0$;
2. The lines to perform the jobs is selected with the use of the least priority value

$$\sum_{i \in N} r_{im} \rightarrow min$$

3. There is such schedule (ordering) σ_m: $N_m \to \{0, 1, \ldots, D\}$ for each subset N_m in terms of planning D, that:
   (a) the sequences of jobs performance on the one line are not to recur,

$$\forall n_{i+1} \in N_m \backslash n_i \Rightarrow \sigma_m(n_{i+1}) \neq \sigma_m(n_i)$$

   (b) the timetable of the line m availability is not to be broken,

$$\forall n_i \in N_m \Rightarrow \begin{cases} \sigma_m(n_i) \notin [sb_{ml}; se_{ml}] \\ \sigma_m(n_i) + t_i \notin [sb_{ml}; se_{ml}] \end{cases}$$

   (c) the number of simultaneously loaded lines is not to be exceeded,

$$\forall i \in \{0, 1, \ldots., D\}, |\{n_i \in N : \sum_{m=1}^{|M|} \sigma_m(n_i)/i\}| \leq m_{\max}, m_{\max} \leq m$$

   (d) conditions of lines reconfiguration are to be fulfilled,

$$\forall n_{i+1} \in N_m \backslash n_i, \sigma_m(n_i) < \sigma_m(n_{i+1}) \Rightarrow \sigma_m(n_{i+1}) - \sigma_m(n_i) - t_i \geq wt_{n_i n_{i+1}}$$

   (e) constraints on simultaneous usage of equipment are:

$$\begin{gathered} \forall i \in \{0, 1, \ldots., D\} : \{\forall n_j \in N : \\ [\sigma_m(n_j); \sigma_m(n_j) + t_j] \subset i \wedge rq_j = 1\} \leq RQ \end{gathered}$$

The common criterion for schedule organization is minimization of the objective function $F \to min$, where F is considered as the penalty function representing the total jobs deviation from individual schedule dates:

$$F = \sum_{i}^{N} |\sigma(n_i) + t_i) - d_i| \to min.$$

3 The Algorithm Description

In solving practical problems with the use of genetic algorithms following preliminary tasks are to be accomplished:

(1) to select the way of solutions representation;
(2) to develop genetic operators;
(3) to determine rules of solutions survival;
(4) to generate the initial population.

In terms of the stated problem let us to apply the encoding scheme, when each chromosome consists of required solution entirely. One agent (individual) includes encoded information about the whole plan for the planning period. The downside of the scheme is that chromosomes are very long. However, the objective function of each individual represents the common optimization criterion, and each generation includes a certain set of solutions, which provides faster convergence together with genetic operators with the use of diverse genetic material. Thus, the chromosome structure is represented as a set of production jobs for the planning period i.e. the whole operational plan. The chromosome contains a number of genes: $N_h = Nr_{max} A_r$, where Nr_{max} is the maximum number of production jobs for the planning period; A_r is a number of variable attributes of production jobs. The value of $Nr_{max} = M$ is determined by common number of production jobs formed on the basis of the main production schedule rows.

Thus, the obtained chromosome involves M groups of genes, each group determines corresponding production job completely, that is the first job corresponds to the first group, the second job corresponds to the second one, etc.

The gene value determines the value of corresponding attribute:

- The gene value of 'duration of job performance' attribute is a number of hours during which the selected line is loaded by the job.
- The gene value of 'production line number' attribute is a sequence number of nonzero element in corresponding row of products and production lines incidence matrix R.
- The gene value of 'production job order' is a number of an hour, when the production job begins.

It should be mentioned that the identifier of the product is not encoded individually, they are determined strictly, i.e. each job is related to a certain product clearly. In such encoding way relevant information is contained not only in gene values, but in their position in chromosome, too. This minimizes the chromosome length, providing reduction of the search space. As a result, the convergence time (number of generations to be processed for the purpose of convergence), and the time required for a generation processing decrease, too. Let us note some specific characteristics of such individuals' representation scheme:

- zero value of job performance duration or job beginning date is interpreted as the absence of production job – so in evolution process the genetic algorithm is able to convergence to optimal number of tasks under condition of $M > M_{opt}$;
- values M are artificial constraints imposed on the search space in such solutions representation.

Authors suggest to encode the chromosome in binary form (Fig. 1).

The chromosome length is calculated as follows:

- $L_h = (L_b + L_{rm} + L_{dm})\, N$,
- where L_b is a number of bits required to encode any moment of job beginning during planning period with selected accuracy (in terms of the stated problem with the accuracy of an hour);
- L_{rm} is a number of bits required to encode the alternative line for the product of the production job m. The value of this gene determines sequence number of nonzero element in corresponding row of incidence matrix R;

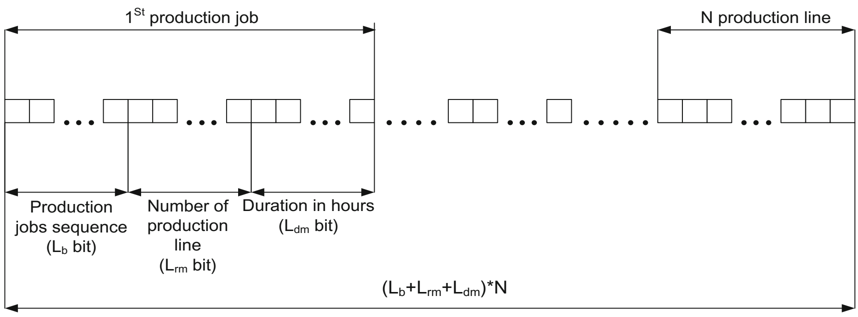

Fig. 1. The chromosome encoding scheme

- L_{dm} is a number of bits required to encode the duration of the job m (in terms of the stated problem the value is multiple of an hour).

Such encoding scheme is sufficiently flexible since it allows us to vary product jobs beginning dates, line numbers and performance durations.

In terms of the stated problem authors suggest to use following genetic operators:

- the selection operator of roulette wheel;
- the reproduction operator. In the process of reproduction which is implemented after selection, chromosomes are copied with the probability proportional to their objective function;
- the crossover operator.

Commonly, the specific characteristic of the stated problem includes penalty functions used in individual's objective function calculation. Besides, it is suggested to modify logic of basic genetic search operators. The modification idea contains the usage of rules applied by subject expert while drawing up the schedule. The essence of these rules is in directed adjustment of production tasks certain parameters to resolve conflicts arising due to violation of constraints conditioned by the problem specific character. Modification assumes changing basic mutation and crossover operators. In particular, there is the specific rule of the allele selection in chromosomes crossover implementation. To visualize modified logic authors show the simplified example considering five jobs and three alternative production lines. Two selected chromosomes represent different solutions that are alternatives of allocation and sequence of jobs performance by production lines. Each rectangle is marked with the index of corresponding job. Rectangles are located horizontally along lines denoting a certain production line. Thus, we obtain the variation of the Gantt diagram commonly used for schedule visualization (Fig. 2).

Jobs are selected for each production line, one from each individual. Genes inherited by a child are determined randomly for each pair (Fig. 3). Herewith, the solution should be tested for duplication. Figure 4 shows the example of jobs allocation during the crossover process.

The result of the crossover operator implementation is shown on Fig. 4. We obtained new solutions, at that excluded incorrect solutions or solutions breaking constraints of jobs sequence on lines.

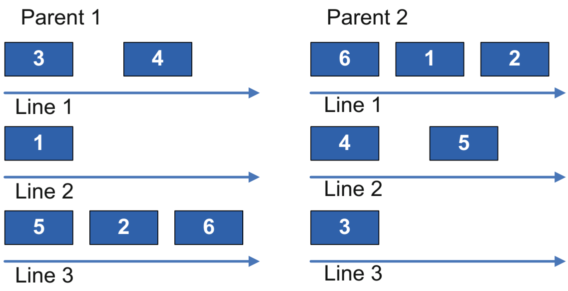

Fig. 2. Alternatives of production planning drawn up with the use of modified crossover operator

Line 1	Parent 1	Parent 2		Child 1	Child 2
Pair 1	3	6	->	6	3
Pair 2	4	1	->	4	1
Pair 3	NULL	2	->	NULL	2

Line 2	Parent 1	Parent 2		Child 1	Child 2
Pair 1	1	4	->	1	4
Pair 2	NULL	5	->	5	NULL

Line 3	Parent 1	Parent 2		Child 1	Child 2
Pair 1	5	3	->	3	5
Pair 2	2	NULL	->	2	NULL
Pair 3	6	NULL	->	NULL	6

Fig. 3. Jobs reallocation resulting from the use of modified crossover operator

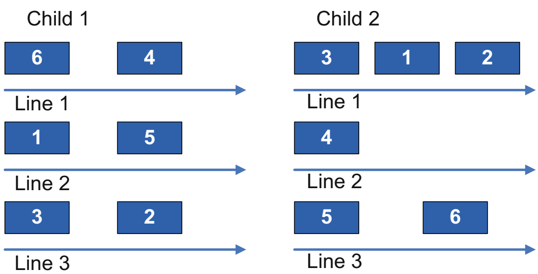

Fig. 4. Results of jobs reallocation with the use of the modified crossover

Thus, in basic crossover operator logic we added the predetermination factor allowing us to exclude potential solutions, i.e. children that do not meet constraints.

- the mutation operator. In terms of the stated problem let us use a multipoint mutation operator. In each chromosome we select N_{OM} pairs of different genes exchanging their values. The value of N_{OM} is calculated as $N_{OM} = \alpha_{OM}L$, where $\alpha_{OM} \in (0; 0{,}5)$ is a parameter determining the proportion of gene pairs involved in mutation of the total chromosome length.

The obtained value of N_{OM} is rounded to the nearest larger integer. The value of α_{OM} should be selected from the interval [0,01; 0,03].

- the migration operator is used to exclude premature convergence of the algorithm i.e. local optimum. Let us assume two parallel evolving populations:

$PR^1 = \{pr_1^1, \ldots, pr_i^1, \ldots, pr_{N_{pr}}^1\}$ and $PR^2 = \{pr_1^2, \ldots, pr_i^2, \ldots, pr_{N_{pr}}^2\}$. At a certain stage N_M individuals from the first population move to the second population. The same number of the second population moves to the first one. Thus, population size remains the same. The selection of individuals from both populations is realized on the basis of 'either best or worst' principle. In this case, we select N_M individuals with the best objective function value from the first population and from the second one – individuals with the worst objective function value. Number of individuals involved in the migration is sized by N_M, which value can be calculated as follows: $N_M = \alpha_M N_{pr}$, where $\alpha_M \in (0; 1)$ is a parameter determining the proportion of agents (individuals) involved in migration of total population size. The obtained value of N_M is rounded to the nearest larger integer. The value of α_M should be selected from [0,1; 0,3].

The objective function calculation contains following steps:

- to calculate the criterion F;
- to calculate corrections considering constraints;
- to calculate the objective function on the basis of values of the criterion F and corrections.

The criterion F is determined by the function of jobs number that breaks directive requirements during the whole planning period. One of the main factors of effective optimization search is the objective function sensibility. Even little adjustment of the criterion F is to result to the maximum change in relation to other individuals objective function values. The objective function sensibility to solutions varieties is directly connected to the effectiveness while estimating the importance of optimization constraints. Artificial constraints imposed on feasible region apart from encoding scheme strongly restrict the GA search possibilities, due to the creation of artificial local optimums on borders of feasible region formed by reimposed constraints.

The essence of GA corrections connected to penalties imposing assumes the reduction of individual objective function value if the solution represented by this individual overruns feasible region. It should be mentioned that constraints determining feasible region are related to a certain production job or to the whole schedule. They can be subdivided into two groups: jobs penalties or whole schedule penalties. Constraints related to schedule penalties are:

- constraints of technical equipment usage;
- constraints of lines number working a day;
- non-strict constraints of the least priority lines usage.

Penalties for breaking these constraints are calculated clearly in a form of criterion value correction. To calculate the correction let us use following coefficients:

- The coefficient of technical equipment overuse:

$$\alpha_{st} = \sum_{t=1}^{T} \frac{bs(t) - bs_{lim} + |bs(t) - bs_{lim}|}{2} \bigg/ bs_{lim},$$

where $bs(t)$ is a number of used equipment in a moment t; bs_{lim} is a total number of equipment RQ; T is a planning horizon.

- The coefficient of excess of working lines a day:

$$\alpha_{bq} = \sum_{t=1}^{T} \frac{bq(t) - bq_{\text{lim}} + |bq(t) - bq_{\text{lim}}|}{2} \bigg/ bq_{\text{lim}}$$

where $bq(t)$ is a number of used equipment in a moment t; bq_{lim} is a limiting number of lines working simultaneously m_{max}; T is a planning horizon.

- The coefficient considering the usage of the least priority lines usage:

$$\alpha_{M_r} = \sum_{t=0}^{T} \frac{M_r(t)}{M(t)},$$

where $M_r(t)$ is a number of production jobs in a day t assigned to production line with the priority value other than the least priority of alternative lines of corresponding job in accordance with the incidence matrix R. $M(t)$ is a total number of production jobs in a day t with nonzero duration.

Corrections are calculated as follows:

$$\begin{aligned}
\Delta_{st} &= \tilde{F}^* \alpha_{st} \\
\Delta_{bq} &= \tilde{F}^* \alpha_{bq} \\
\Delta_{M} &= \tilde{F}^* (1 - \alpha_{M_r})
\end{aligned}$$

Thus, the value $\Phi = \tilde{F} - \Delta_{st} - \Delta_{bq} - \Delta_{M_r}$ is represented as the objective function value used for the GA operators work.

The next GA step is the stop criterion checking. Stop criterion can be represented as:

- obtaining assigned number of generations;
- obtaining assigned algorithm execution time;
- remaining relative adjustment of population OF average value during a assigned number of generations.

Also, the common stop criterion is obtaining solution meeting the constraints in the optimization problem statement. The last criterion from the three criteria mentioned above represents the convergence factor in a most relevant way.

This criterion can be represented as follows:

$$\frac{|\Phi_{ij} - \Phi_{i(j-\delta)}|}{\Phi_{i(j-\delta)}} \cdot 100\% \leq \Delta_{\Phi},$$

where Φ_{ij} is the objective function value of individual i on the iteration j; $\Phi_{i(j-\delta)}$ is the objective function value of individual i on the iteration $(j-\delta)$; δ is a number of iterations to calculate relative objective function change; Δ_{Φ} is a threshold value in relation to the objective function change.

Values of δ and Δ_{Φ} are calculated empirically on the basis of research of obtained solutions quality with a certain values, and acceptable algorithm execution time. Initial values can be selected from intervals: $\delta \in [10; 50]$; $\Delta_{\Phi} \in [0.3\%; 3\%]$.

Under the stop condition the obtained solution is captured. Otherwise, genetic operators mentioned above are to be realized.

4 Integration of the Genetic Algorithm and the Adaptive Search

In recent years integrated and hybrid models involving genetic algorithms are of a great interest. We can distinguish approaches of 'external' hybridization, for instance, building hybrids of genetic algorithms and evolutionary strategy and neural network meta models [5–7] and 'internal' hybridization, when in the context of evolutionary design algorithms are integrated on the basis of evolutionary models, for instance, Darwin model and Lamarck model [8–10].

In this paper authors suggest an 'external' hybridization variant, when hybrid fuzzy genetic algorithm combines approaches of fuzzy logic and genetic search in terms of united optimization process. The algorithm scheme is shown on Fig. 5. The main idea of hybridization involves the mathematical tool of fuzzy logic theory used for encoding, calculation of genetic algorithm optimal parameters, genetic operators probability values, fitness function and stop criterion selection. The suggested algorithm can be applied in terms of parallel computing performed on corresponding resources. Modern processors have multicore architecture, which allows us to carry out parallel and distributed computing [11, 12]. In terms of hybrid algorithm logic solutions might be obtained and exchanged simultaneously during one hybrid algorithm iteration.

To improve the quality of genetic search results authors propose to solve the problem of including expert information in evolution process by building fuzzy logic controller which adjusts evolution process parameters values.

Following parameters are used as input [13]:

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)};\ e_2(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{worst}(t) - f_{best}(t)};$$
$$e_3(t) = \frac{f_{best}(t) - f_{best}(t-1)}{f_{best}(t)};\ e_4(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{ave}(t)},$$

Where t is a time step, $f_{best}(t)$ is the best objective function value on iteration t, $f_{best}(t-1)$ - is the best objective function value on iteration $(t-1)$, $f_{worst}(t)$ is the worst

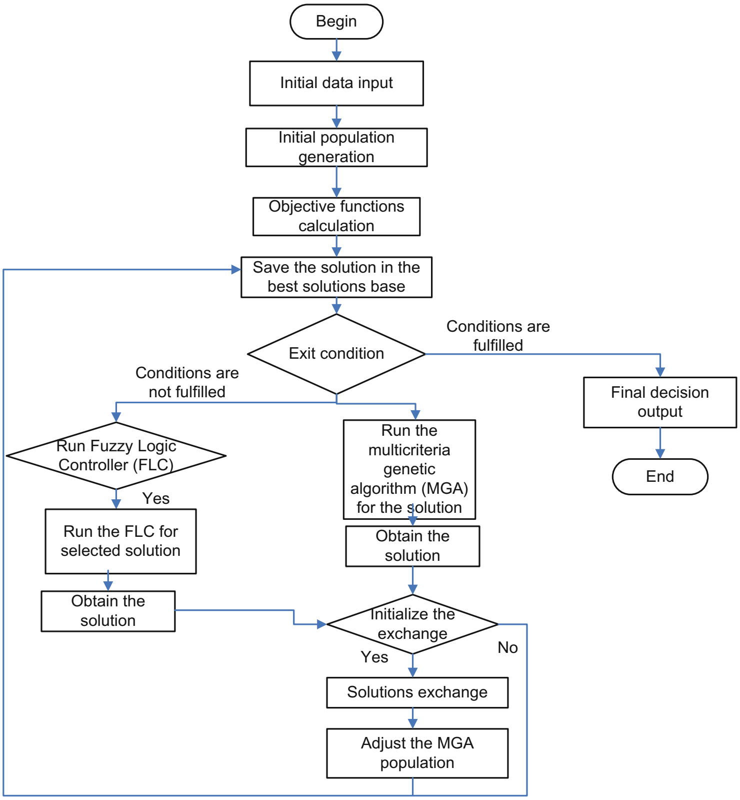

Fig. 5. Hybrid algorithm scheme

objective function value on iteration t, $f_{ave}(t)$ is the average objective function value on iteration t, $f_{ave}(t-1)$ is the average objective function value on iteration $(t-1)$ [14].

Obtained output parameters represent probabilities of crossover, mutation and migration operator.

5 Experimental Research

Let us consider the objective function value as an optimization criterion on the basis of a certain number of algorithm iterations analysis.

The purpose of experiments is to determine the character of problem dimension behavior (values of jobs number N and lines number M), and time spent on solutions search. The convergence is considered as obtaining such objective function D value, that during following δ iterations the change ΔD is less than $\Delta\%$ of previous value.

$$\frac{|D_{i-\delta} - D_i|}{D_{i-\delta}} * 100\% \leq \Delta$$

where $D_{i-\delta}$ is the objective function value on the iteration $(i-\delta)$; D_i is the objective function value on the iteration i; δ is a number of iterations to calculate relative objective function change; i is the current iterations; Δ is the threshold value of relative objective function change.

To carry out the experimental research authors took following values of convergence criterion parameters: $\delta = 10$; $\Delta = 1\%$.

The research was carried out on the basis of twelve points selected by experts. Each experiment point of production schedule is determined by initial data vector, which coordinates includes: total amount of jobs N, total amount of lines, number of paired lines $LD = \sum_{j_1=1}^{M} \sum_{j_2=(M-j_1-1)}^{M} ld_{j_1 j_2}$, planning horizon D in hours, number of equipment RQ.

The initial data is shown on the Table 1.

Table 1. Input data for algorithm convergence estimation

№	N	M/LD	D	RQ
1	5	3/2	72	1
2	10	3/2	96	1
3	20	6/2	96	2
4	50	12/4	120	3
5	70	16/6	120	3
6	100	24/8	120	4
7	150	26/8	144	5
8	200	30/10	168	6
9	250	34/10	192	6
10	300	40/12	240	7
11	400	50/16	240	8
12	500	54/18	240	8

For each experiment point authors took common input data:

- Jobs duration are selected randomly from the assigned interval $t_i \in [t_{min}, t_{max}]$;
- A service interval is assigned for the paired line.

As shown on graphics, the convergence time increases linearly at more problem dimension. The adaptive search converge faster that the modified genetic algorithm. Obtained results prove the common assumption that approximate algorithm is relevant for quasioptimal solution search. The behavior character represents almost linear calculation time dependence on the problem dimension, which allows us to assume polynomial time complexity of developed algorithms.

Acknowledgment. This research is supported by the grant from the Russian Foundation for Basic Research (project # 16-01-00715, 17-01-00627).

References

1. Conway, R.M., Maxwell, W.L., Miller, L.W.: Theory of Scheduling, 2nd edn. Dover Publications, Mineola (2004)
2. Pinedo, M.: Scheduling: Theory, Algorithms and Systems, 3rd edn. Springer, New York (2008)
3. Leung, J.Y.T.: Handbook of Scheduling. Chapman & Hall/CRC, Boca Raton (2004)
4. Luger, G.F.: Artificial Intelligence. Structures and Strategies for Complex Problem Solving, 6th edn. Addison Wesley, Boston (2009)
5. Michael, A., Takagi, H.: Dynamic control of genetic algorithms using fuzzy logic techniques. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 76–83. Morgan Kaufmann (1993)
6. Lee, M.A., Takagi, H.: Integrating design stages of fuzzy systems using genetic algorithms. In: Proceedings of the 2nd IEEE International Conference on Fuzzy System, pp. 612–617 (1993)
7. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. J. Soft Comput. **7**(8), 545–562 (2003). Springer
8. Gladkov, L.A., Kureichik, V.V., Kureichik, V.M.: Genetic Algorithms. Phizmatlit, Moscow (2010)
9. Gladkov, L.A., Gladkova, N.V., Leiba, S.N.: Hybrid intelligent approach to solving the problem of service data queues. In: Proceeding of 1st International Scientific Conference "Intelligent Information Technologies for Industry" (IITI 2016), vol. 1, pp. 421–433 (2016)
10. Gladkov, L.A., Gladkova, N.V., Legebokov, A.A.: Organization of knowledge management based on hybrid intelligent methods. In: Silhavy, R., Senkerik, R., Oplatkova, Z.K., Prokopova, Z., Silhavy, P. (eds.) Software Engineering in Intelligent Systems. AISC, vol. 349, pp. 107–112. Springer, Cham (2015). doi:10.1007/978-3-319-18473-9_11
11. King, R.T.F.A., Radha, B., Rughooputh, H.C.S.: A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In: Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, pp. 577–582 (2004)
12. Zhongyang, X., Zhang, Y., Zhang, L., Niu, S.: A parallel classification algorithm based on hybrid genetic algorithm. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 3237–3240 (2006)
13. Gladkov, L., Gladkova, N., Leiba, S.: Manufacturing scheduling problem based on fuzzy genetic algorithm. In: Proceeding of IEEE East-West Design and Test Symposium – (EWDTS 2014), Kiev, Ukraine, pp. 209–212 (2014)
14. Gladkov, L.A., Gladkova, N.V., Leiba, S.N.: Electronic computing equipment schemes elements placement based on hybrid intelligence approach. In: Silhavy, R., Senkerik, R., Oplatkova, Z.K., Prokopova, Z., Silhavy, P. (eds.) Intelligent Systems in Cybernetics and Automation Theory. AISC, vol. 348, pp. 35–44. Springer, Cham (2015). doi:10.1007/978-3-319-18503-3_4