ствам. Таким образом, последовательность P_{14} с таблицей умножения T_{14} образует группу, изоморфную G. Непосредственной проверкой находим, что $\omega(G) = \{1, 2, 4\}$. Группа G, порядок которой равен 2^{10}, является максимальной группой, удовлетворяющей условиям теоремы. Теорема доказана.

Следствие 1. Пусть $(10)^2 = e$, $(20)^2 = e$, $(21)^2 = e$. Тогда получим, что $|G| = 8$.

Следствие 2. Пусть $(10)^2 = e$, $(210)^2 = e$. Тогда получим, что $|G| = 16$.

Следствие 3. Пусть $(210)^2 = e$. Тогда получим, что $|G| = 32$.

Следствие 4. Пусть $(10)^2 = e$. Тогда получим, что $|G| = 64$.

Таким образом, проиллюстрированный в статье пример показывает эффективность моделирования периодических групп при помощи указанного выше алгоритма в задачах распознавания групп по их спектру.

Библиографический список

1. Кузнецов, А. А. Об одном алгоритме получения соотношений в свободной бернсайдовой группе $B(m, n)$ / А. А. Кузнецов, С. А. Тарасов, А. К. Шлепкин // Дискретные модели в теории управляющих систем : тр. VI Междунар. конф. М. : МГУ им. М. В. Ломоносова, 2004. С. 175–178.

2. Кузнецов, А. А. Некоторые комбинаторные вопросы в периодических группах : дис ... канд. физ.-мат. наук / А. А. Кузнецов. Красноярск : КрасГАУ, 2006.

3. Нерешенные вопросы теории групп. Коуровская тетрадь / под. ред. В. Д. Мазурова. Вып. 16. Новосибирск : ИМ СО РАН, 2006.

4. Кузнецов, А. А. Распознавание группы $L_2(7)$ по спектру в классе всех групп / А. А. Кузнецов, Д. В. Лыткина // Сиб. электрон. матем. изв. 2007. № 4. С. 136–140.

A. A. Kuznetsov

SET-THEORETIC ANALYSIS OF ALGEBRAIC PROBLEBMS OF GROUPS RECOGNIZABILITY BY SPECTRUM

It is shown the example of periodic groups modeling in problems of group recognizability by their spectrum.

Keywords: periodic groups, groups recognizability by spectrum, computational modeling of groups.

УДК 658.512.6

I. S. Masich

COMBINATORIAL OPTIMIZATION IN FOUNDRY PRODUCTION PLANNING

The mathematical model of foundry production capacity planning is suggested in the paper. The model is produced in terms of pseudo-Boolean optimization theory. Different search optimization methods were used to solve the obtained problem.

Keywords: combinatorial optimization, pseudo-Boolean function, heuristic algorithm, foundry.

Production capacity optimization. One of the actual problems in modern industry is production capacity optimization under irregular orders from numerous partners. Along with the mass serial production these orders may have small serial or single nature. The majority of orders are irregular, i. e. they cannot be planned beforehand but nevertheless they are profitable enough or the enterprise. It requires solving production scheduling problems many times in casual points of time.

So it is necessary to have means for including new small orders in capacity intervals of existing mass serial ones. Moreover, it is necessary to consider time of equipment revamping for small serial order since it takes significantly more part of execution time than in mass serial production.

Existing of small serial and single orders requires practically permanent process of production capacity planning. Construction of production capacity program for industrial enterprise and its subdivisions is laborious and logically intricate problem. Consider it by the example of foundry practice. Below we design an optimization model for production capacity and apply combinatorial methods to solve it.

Pseudo-Boolean optimization problems. Unconstrained pseudo-Boolean optimization is an issue studied enough by now. Algorithms that have been designed and investigated in the area of unconstrained pseudo-Boolean optimization are applied successfully for solving various problems. Particularly, they are local optimization methods [1–3] and stochastic and regular algorithms based on local search for special function classes [4–6]. Moreover, there are a number of algorithms for functions optimization given in explicit form: Hammer's basic algorithm that was introduced in [7] and simplified in [8]; algorithms for optimization of quadratic functions [9–11], etc. Universal optimization methods are also used successfully: genetic algorithms, simulated annealing, tabu search [12; 13].

If there are constraints on the binary variables, one of the ways to take into account it as is well known is construction and optimization of a generalized penalty function. Shortcoming of this approach is existence of a large number of local optima of the generalized function what will be shown below. If an accessible region is a connected one then this issue could be partly eliminated, for example, by using local search with a stronger system of neighborhoods. Extension of search neighborhood reduces the number of local optima which locate mainly not far one from another in this case. If an accessible region is unconnected then using penalty functions and unconstrained optimization methods get complicated because the accessible region is usually too small with the respect to optimization space. That makes difficult searching a feasible solution.

In this work heuristic procedures of boundary point search are considered for a constrained pseudo-Boolean optimization problem. Experimental investigation of the algorithms are described, recommendations for their apply are given.

Search algorithms for pseudo-Boolean optimization. Basic definitions. Consider some definitions that are necessary for describing optimization algorithm work [14].

A *pseudo-Boolean function* is called a real function of binary variables: $f : B_2^n \to R^1$, where $B_2 = \{0, 1\}$, $B_2^n = B_2 \times B_2 \times \Lambda \times B_2$.

Points $X^1, X^2 \in B_2^n$ are called *k-neighboring* points if they differ in *k* coordinate value, $k = \overline{1, n}$. 1-neighboring points are called simply neighboring.

The set $O_k(X)$, $k = \overline{0, n}$, of all point of B_2^n, that are *k*-neighboring to a point *X*, is called a *k-th level* of the point *X*.

A point set $W(X^0, X^l) = \{X^0, X^1, K, X^l\} \subset B_2^n$ is called a *path* between points X^0 and X^l if for $\forall\ i = 1, ..., l$ the point X^i is a neighboring to X^{i-1}.

A set $A \subset B_2^n$ is called a *connected set* if for $\forall X^0,\ X^l \in A$ the path $W(X^0, X^l) \subset A$ exists.

A point $X^* \in B_2^n$, for which $f(X^*) < f(X),\ \forall X \in O_1(X^*)$, is called a *local minimum* of pseudo-Boolean function *f*.

A pseudo-Boolean function that has an unique local minimum is called an *unimodal* on B_2^n function.

An unimodal function *f* is called *monotonic* on B_2^n if $\forall X^k \in O_k(X^*), k = \overline{1,n}\!:\ f(X^{k-1}) \le f(X^k),\ \forall X^{k-1} \in O_{k-1}(X^*) \cap O_1(X^k)$.

Problem statement. Consider the problem of the following form

$$C(X) \to \max_{X \in S \subset B_2^n}, \qquad (1)$$

where $C(X)$ is a monotonically increasing from X^0 pseudo-Boolean function, $S \subset B_2^n$ is a certain subspace of the binary variable space; it is determined by a given constraint system, for example:

$$A_j(X) \le H_j,\ j = \overline{1, m}. \qquad (2)$$

In the general case a set of feasible solutions *S* is an unconnected set.

One of ways to take into account constraints in conditional problems is construction a generalized penalty function that is solved by known search methods: local search, genetic algorithms, simulated annealing, etc. Shortcoming of this approach is loss of the monotonicity property for an object function. By addition of even simple (for example, linear) constraints the generalized function becomes a polymodal nonmonotonic function with exponential number of local maxima.

Properties of a feasible set. A point $Y \in \mathbf{A}$ is a *boundary point* of the set **A** if there exist $X \in O_1(Y)$ for which $X \notin \mathbf{A}$.

A point $Y \in O_i(X^0) \cap \mathbf{A}$ is called a *limiting point* of the set **A** with the basic point $X^0 \in \mathbf{A}$ if for $\forall X \in O_1(Y) \mathrm{I}\ \ O_{i+1}(X^0)\ \ X \notin \mathbf{A}$ holds.

A constraint that determines a subspace of the binary variable space is called *active* if the optimal solution of the conditional problem does not coincide with the optimal solution of the appropriate problem without taking the constraint into account.

If the object function is a monotonic unimodal function and the constraint is active then the optimal solution of the problem (1) is a point that belongs to the subset of limiting points of the feasible set *S* with the basic point X^0 in which the object function takes the lowest value:

$$C(X^0) = \min_{X \in B_2^n} C(X).$$

Properties of feasible sets are considered detailed in [15; 16].

Heuristic algorithms for boundery point search. For any heuristic of boundery point search we will consider a pair of algorithms – primary and dual. A primary algorithm starts search from the feasible area and moves in a path of increasing of the objective function until it finds a limiting point of feasible area. Otherwise, a dual algorithm keeps search in the unfeasible area in a path of decreasing of the objective function until it finds some available solution.

Total scheme of primary search algorithm:

1. Put $X_1 = X^0$, $i = 1$.

2. In accordance with a rule we choose $X_{i+1} \in O_i(X^0) \cap O_1(X_i) \cap S$. If there are no such point then go to 3; else $i = i + 1$ and repeat the step.

3. $X_{\text{opt}} = X_{i+1}$.

Total scheme of dual search algorithm:

1. Put $X_1 \in O_n(X^0)$, $i = 1$.

2. In accordance with a rule we choose $X_{i+1} \in O_{n-i}(X^0) \cap O_1(X_i)$. If $X_{i+1} \in S$ then go to 3; else $i = i + 1$ and repeat the step.

3. $X_{\text{opt}} = X_{i+1}$.

From these schemes we can see that a primary algorithm finds a limiting point of the feasible area, but a dual algorithm finds a boundary point which may be not a limiting one. So a primary algorithm finds a better solution then dual in most cases for problems with a connected set of feasible solutions. If we will use a primary algorithm for a problem with a unconnected feasible area then solution received in result may be far from the optimal because feasible and unfeasible solutions will rotate in a path of increasing of the objective function. For these cases a dual algorithm is more usefull, because this rotation does not play any role for it. For improving the solution that given by the dual algorithm, it is recommends to apply the corresponding primary algorithm. Such improving is very significant in practice.

Boundary point search algorithms considered below differs by only a rule of choise of a next point in step 2 of the total schemes.

Search rules. *Rule 1.* Random search of boundary points (RSB).

A point X_{i+1} is chosen by random way. Each point in the next step can be chosen with equal probabilities. For real-world problems these probabilities can be not equal but they are calculated in the basis of problem specific before search starts.

Rule 2. Greedy algorithm.

A point X_{i+1} is chosen acordance with the condition

$$\lambda(X_{i+1}) = \max_j \lambda(X^j),$$

where $X^j \in O_i(X^0) \cap O_1(X_i) \cap S$ for a primary algorithm and $X^j \in O_{n-i}(X^0) \cap O_1(X_i)$ for a dual one.

The function $\lambda(X)$ is chosen from the specific problem, for example:

– the objective function $\lambda(X) = C(X)$;

– specific value $\lambda(X) = C(X)/A(X)$ (for only constraint) and so on.

Rule 3. Adaptive random search of boundary points (ARSB).

A point X_{i+1} is chosen by random way in accordance with a probability vector

$$P^i = (p_1^i, p_2^i, ..., p_J^i),$$

where *J* is the number of points from which choice is made.

$$p_j^i = \frac{\lambda(X^j)}{\sum_{l=1}^{J} \lambda(X^l)}, \ j = \overline{1, J},$$

where $X^j \in O_i(X^0) \cap O_1(X_i) \cap S$ for a primary algorithm and $X^j \in O_{n-i}(X^0) \cap O_1(X_i)$ for a dual one.

ARSB is an addition to the greedy algorithm.

Rule 4. Modificated random search of boundary points (MRSB).

A point X_{i+1} is chosen in accordance with the condition

$$\lambda(X_{i+1}) = \max_r \lambda(X^r),$$

where X^r are points chosen in accordance with the rule 1, $r = \overline{1, R}$; *R* is an algorithm parameter.

A greedy algorithm is a regular algorithm [17], so it finds equivalent solutions under restart from a certain point. Other algorithms can be started several times and the best solution can be selected from the found solutions. An average run time of a greedy algorithm and ARSB is significant larger than for others because they look over all points of the next level in each step in distinction from RSB and MRSB which looks over only one and *R* points correspondingly in each step in a dual scheme.

Further we consider applying the described algorithms for one of real-world problems with large dimension and unconnected set of feasible solutions.

Foundry branches production capacity optimization. Production of different kinds is produces in foundry branches (FB). There is specialization in every FB by a kind of production which can be produced by its foundry machines (FM). There is a quantity of orders for production. To each order there corresponds the volume, a kind of production and term of performance. The kind of production is characterized by productivity for change (only 3 changes in day). Replacement made on FM production demands its recustomizing borrowing one change. It is necessary to load thus FB capacities the orders were carried out all, production was made in more regular intervals in time, and the number of recustomizings of FM equipment was minimal.

Input data:

I is a number of days for planning;

J is a number of FB;

K_j is a number of FM in *j*-th FB, $j = \overline{1, J}$;

L is a number of orders for production that produces on FM (and corresponding number of production kinds);

V_l is productivity of *l*-th kind of production for change on FM, $l = \overline{1, L}$;

T_l is term of performance of *l*-th order (for production of *l*-th kind) on FM, $l = \overline{1, L}$;

W_l is volume of *l*-th order (for production of *l*-th kind) on FM, $l = \overline{1, L}$;

z_{jl} characterizes specialization of FB:

$$z_{jl} = \begin{cases} 1, & \text{FM of } j\text{-th FB can make production of } l\text{-th kind,} \\ 0, & \text{otherwise;} \end{cases}$$

α is the factor of rigidity of restriction on demand of uniformity of production a days, $0 < \alpha < 1$.

Variables:

For the model construction introduce the following binary variables:

$$Y = \{y_{ijkl}\} \in B_2^n,$$

$$X = \{x_{ijkl}\} \in B_2^n,$$

where $B_2 = \{0, 1\}$, $B_2^n = B_2 \times B_2 \times \Lambda \times B_2$ is a set of binary variables.

$$y_{ijkl} = \begin{cases} 1, & \text{production of } l\text{-th kind is made} \\ & \text{in } i\text{-th day on } k\text{-th FM of } j\text{-th FB,} \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{ijkl} = \begin{cases} 1, & \text{production of } l\text{-th kind is started to make} \\ & \text{in } i\text{-th day on } k\text{-th FM of } j\text{-th FB,} \\ 0, & \text{otherwise;} \end{cases}$$

Total dimension of a binary vector *Y* (and *X*) is

$$n = I \cdot L \cdot \sum_{j=1}^{J} K_j.$$

Remarks:

1. $x_{ijkl} \le y_{ijkl} \ \ \forall i, j, k, l$.

2. $x_{ijkl} = y_{ijkl} \cdot (1 - y_{i-1,jkl}) \ \ \forall i, j, k, l \ (y_{0jkl} = 0 \ \ \forall j, k, l)$. (3)

Optimization model:

1. The objective function and the main constraints:

$$C(X) \to \min, \tag{4}$$

$$A_l^1(Y) \ge W_l, \ l = \overline{1, L}, \tag{5}$$

$$A_i^2(Y) \ge \alpha \cdot W^I, \ i = \overline{1, I}, \ \alpha \in (0,1), \ W^I = \sum_{l=1}^{L} W_l \Big/ I, \tag{6}$$

where

$$C(X) = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{l=1}^{L} x_{ijkl} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{l=1}^{L} y_{ijkl} \cdot \left(1 - y_{i-1,jkl}\right),$$

$$A_l^1(Y) = \sum_{i=1}^{T_l} \sum_{j=1}^{J} \sum_{k=1}^{K_j} V_l \cdot y_{ijkl} \cdot \left(2 + y_{i-1,jkl}\right),$$

$$A_i^2(Y) = \sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{l=1}^{L} V_l \cdot y_{ijkl} \cdot \left(2 + y_{i-1,jkl}\right),$$

$$y_{0jkl} = 0 \ \ \forall j, k, l.$$

2. The additional constraints:

$$\sum_{l=1}^{L} y_{ijkl} \le 1 \ (\sum_{l=1}^{L} x_{ijkl} \le 1), \ i = \overline{1, I}, \ j = \overline{1, J}, \ k = \overline{1, K_j}, \tag{7}$$

$y_{ijkl} \leq z_{jl}$ ($x_{ijkl} \leq z_{jl}$), $i = \overline{1, I}$, $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$.

Model properties:

1. There are two spaces of binary variables (denote their by B^X and B^Y) corresponding to vectors *X* and *Y*. For each point $Y \in B^Y$ a unique point $X \in B^X$ corresponds, components of which are determined by relation (3). Several points $Y \in B^Y$ (with different value of constraint function) can correspond to the point $X \in B^X$.

2. The objective function (4) is linear and unimodal monotonic in space B^X with the minimum point $X^0 = (0, ..., 0)$. In space B^Y the objective function is a quadratic and unimodal nonmonotonic one with a minimum point $Y^0 = (0, ..., 0)$.

3. The constraint function (5) and (6) in space B^Y are quadratic and unimodal monotonic pseudo-Boolean functions with a minimum point $Y^0 = (0, ..., 0)$. In space B^X the constraint functions unequivocally are not certain.

4. The feasible set in spaces B^X and B^Y is limited from above by $I \cdot \sum_{j=1}^{J} K_j$-th level of the minimum point (X^0 and Y^0) according to the constraint (7). In space B^Y this level corresponds to the case when production is produced on each BM every day.

5. The feasible set is an unconnected set in general case (in space B^Y).

Thus the problem solution is defined completely by the variables *Y*, but it does not hold for the variables *X*. But the objective function from *X* has good constructive properties so that optimum search on *X* is more efficient than on *Y*. As the constraint function (5), (6) from *X* are not defined, we should find values of these functions from the corresponding point *Y*. There exist perhaps several such points

$$X \to Y_1, Y_2, ..., Y_H.$$

Not at all points the solution may be feasible but in other not. As the constraint functions are monotonic here then for a certain *X* we should choose such *Y* that belongs to the most possible level (with the most values of the functions):

$$Y = \arg \max_{Y_h, h=\overline{1,H}} \left(\sum_{i,j,k,l} y_{ijkl}^h \right),$$

where $Y_h = (y_{1111}, ..., y_{IJK_jL})$.

One of the algorithms of this transformation is presented below.

Algorithm 1 of transformation *X* to *Y*:

1. Put $N_{jk} = 0$, $j = \overline{1, J}$, $k = \overline{1, K_j}$; $i = 1$.

2. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$ do: if $x_{ijkl} = 1$ then $N_{jk} = l$.

3. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$ do: if $N_{jk} = l$ then $y_{ijkl} = 1$ else $y_{ijkl} = 0$.

4. If $i < I$ then $i = i + 1$ and to 2.

At the same time the solution *Y* received from the found best vector X_{opt} by this way may corresponds to situation when a quantity of let out production is more higher than the requisite value (it does not contradict to the constructed model but this can influence uniformity of capacities loading which is optimized at the next stage). So when the first stage of search has ended, we should define Y_{opt} by the rule

$$X_{\text{opt}} \to Y_1, Y_2, ..., Y_H,$$

$$Y_{\text{opt}} = \arg \min_{Y_h : A_l^1(Y_h) \geq W_l, l=\overline{1,L}} \left(\sum_{i,j,k,l} y_{ijkl}^h \right).$$

In this case the transformation algorithm has some differences from the previous.

Algorithm 2 of transformation *X* to *Y*:

1. Put $N_{jk} = 0$, $j = \overline{1, J}$, $k = \overline{1, K_j}$.

1a. Put $y_{ijkl} = 0$, $i = \overline{1, I}$, $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$; $i = 1$.

2. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$ do: if $x_{ijkl} = 1$ then $N_{jk} = l$.

3. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $l = \overline{1, L}$ do: if $N_{jk} = l$ and $A_l^1(Y) < W_l$ then $y_{ijkl} = 1$.

4. If $i < I$ then $i = i + 1$ and to 2.

Here the condition $A_l^1(Y) < W_l$ is added in the step 3, and the step 1a is added also for possibility of this calculation. There is no necessity in this transformation during the search. It is necessary only to determine the result Y_{opt}.

Optimization algorithms. The dual algorithms RSB, greedy and MRSB have been used for solving the problem. The algorithm ARSB has not been considered for this problem because of its excessive large run time by frequent start. One start of ARSB can very rarely give a solution which is better then the solution given by the greedy algorithm. The start point of search is the point of the unconstraint minimum of the objective function $X^0 = (0, ..., 0)$. A found solution has been improved by the corresponding primary algorithm.

Moreover, the problem has been solved by the genetic algorithm (GA). To realize GA we have chosen a scheme that effective worked for multiple solving other combinatorial optimization problems.

Results of the experiments shows that the most effective algorithms (by precision and run time) from the considered ones for this problem are the greedy algorithm and MRSB. The other algorithms under hard constraints on the variables do not find any accesible solution at all. It is a sequel of problem specific: a large amount of different constraints and, as a result, a comparative small accesible regin.

The average results of solving 10 problems of month planning capacity loading are presented in the table. The average values of input data:

$$I = 31, J = 3, K_1 = 12, K_2 = 9, K_3 = 7, L = 36, \alpha = 0{,}5,$$

$$V_l \in [40, 50], W_l \in [20, 25\,000].$$

Herewith the total dimension of the binary vector is $n = 31\,248$.

The number of algorithm starts *L* has been chosen so that the run time nearly equals to the run time of one start of the greedy algorithm. In this case the run time equals to $T \approx 8 \cdot 10^5$.

MRSB is a more flexible procedure in compare with the greedy algorithm as the first one allows selecting the parameters *L* and *R* that influence on algorithm run time and solution precision. The greedy algorithm does not allow that possibility and run time may be overmuch large under high dimensions. What about their efficience, precision of the found solutions differs unessentually under nearly equivalent run time.

The optimization model for foundry production capacity planning is designed. It corresponds to the conditional pseudo-Boolean optimization problem with an unconnected feasible set and it is solved by the series of heuristic search algorithms. The algorithms of boundary point search show high efficiency for solving the pseudo-Boolean optimization problem with an unconnected accessible region. The most efficient algorithms for the considered problem are the dual algorithms MRSB and greedy one.

Bibliography

1. Aarts, E. H. L. Local Search in Combinatorial Optimization / E. H. L. Aarts, J. K. Lenstra // Wiley-Interscience Series in Discrete Mathematics and Optimization. England : John Wiley & Sons Ltd, 1997.

2. Antamoshkin, A. N. Comparative efficiency of two schemes of local search for optimization of pseudo-Boolean functions [Electronic resource] / A. N. Antamoshkin, I. S. Masich // Investigated in Russia : electronic journal. 2004. 51. P. 544–546. Access mode: http://zhurnal.ape.relarn.ru/articles/2004/051.pdf. Title from a display.

3. Papadimitriou, C. H. Combinatorial Optimization / C. H. Papadimitriou, K. Steiglitz. Prentice-Hall, Englewood Cliffs, NJ, 1982.

4. Antamoshkin, A. N. Optimization of unimodal monotone pseudoboolean functions / A. N. Antamoshkin, V. N. Saraev, E. S. Semenkin // Kybernetika. 1990. Vol. 26, № 5. P. 432–441.

5. Bjцrklund, H. Optimization on Completely Unimodal Hypercubes / H. Bjцrklund, S. Sandberg, S. Vorobjov // Technical report 2002-018/ May 2002/ ISSN 1404-3203.

6. Wegener, I. On the Optimization of Monotone Polynomials by Simple Randomized Search Heuristics / I. Wegener, C. Witt. Technical Report. ISSN 1433-3325.

7. Hammer, P. L. Boolean Methods in Operations Research and Related Areas / P. L. Hammer, S. Rudeanu. Berlin : Springer-Verlag ; New York : Heidelberg, 1968.

8. Boros, E. Pseudo-Boolean Optimization / E. Boros, P. L. Hammer // Discrete Applied Mathematics. 2002. Vol. 123. P. 155–225.

9. Allemand, K. A. Polynomial case of unconstrained zero-one quadratic optimization / K. Allemand, K. Fukuda, T. M. Liebling, E. Steiner // Math. Program. Ser. A. 2001. Vol. 91. P. 49–52.

10. Foldes, S. Monotone, Horn and Quadratic Pseudo-Boolean Functions / S. Foldes, P. L. Hammer // Journal of Universal Computer Science. 2000. Vol. 6, № 1. P. 97–104.

11. Hammer, P. L. Quadratic functions of binary variables / P. L. Hammer, B. Simeone // Combinatorial Optimization, Lecture Notes in Mathematics ; ed. by B. Simeone. Vol. 1403 Springer, Heidelberg, 1989.

12. Goldberg, D. E. Genetic algorithms in search, optimization, and machine learning / D. E. Goldberg. Reading, MA : Addison-Wesley, 1989.

13. Schwefel, H.-P. Evolution and Optimum Seeking / H.-P. Schwefel. New York : Whiley Publ., 1995.

14. Antamoshkin, A. N. Regular optimization of pseudo-Boolean functions / A. N. Antamoshkin. Krasnoyarsk: Krasnoyarsk university publ., 1989.

15. Antamoshkin, A. N. Heuristic search algorithms for monotone pseudo-boolean function conditional optimization / A. N. Antamoshkin, I. S. Masich // Engineering&automation problems. 2006. Vol. 5, № 1. P. 55–61.

16. Antamoshkin, A. N. Unimprovable algorithm of conditional optimization of monotonic pseudo-Boolean functions [Electronic resource] / A. N. Antamoshkin, I. S. Masich // Investigated in Russia : electronic journal. 2004. 64. P. 703–708. Access mode: http://zhurnal.ape.relarn.ru/articles/2004/064.pdf. Title from a display.

17. Antamoshkin, A. N. Greedy algorithm and local search for conditional pseudo-Boolean optimization [Electronic resource] / A. N. Antamoshkin, I. S. Masich // Investigated in Russia : electronic journal. 2003. 177. P. 2143–2149. Access mode: http://zhurnal.ape.relarn.ru/articles/2003/177.pdf. Title from a display.

Comparative results

Algorithm	Number of starts L	Found solution C_{opt}
RSB	2 000	Not found
Greedy	1	49
MRSB, R = 1 000	12	47
MRSB, R = 100	60	52
MRSB, R = 10	200	Not found
GA*	–	Not found

Note.* GA parameters: tournament selection with the tournament value 5, population value 100, the largest number of generations 8 000, mutation probability 0,000 1.

И. С. Масич

КОМБИНАТОРНАЯ ОПТИМИЗАЦИЯ ДЛЯ ПЛАНИРОВАНИЯ ЗАГРУЗКИ ЛИТЕЙНОГО ПРОИЗВОДСТВА

Предлагается математическая модель планирования загрузки литейного производства. Модель представлена в терминах теории псевдобулевой оптимизации, в качестве методов решения полученной задачи использовались различные поисковые методы оптимизации.

Ключевые слова: комбинаторная оптимизация, псевдобулевые функции, приближенные алгоритмы, литейное производство.