IOP Conf. Series: Materials Science and Engineering **122** (2016) 012001 doi:10.1088/1757-899X/122/1/012001

Combinatorial optimization in foundry practice

A N Antamoshkin[1,2] and I S Masich[1,3]
[1] Department of System Analysis and Operation Research, Siberian State Aerospace University named after academician M. F. Reshetnev, Krasnoyarsk, Russia
[2] Department of Economics and Information Technology of Management, Siberian Federal University, Krasnoyarsk, Russia
[3] Department of Informatics and Mathematical Modeling, Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia

E-mail: i-masich@yandex.ru

Abstract. The multicriteria 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.

1. Introduction

1.1. Production Capacity Optimization

One of the actual problems in modern industry is production capacity optimization under irregular orders from numerous partners. Along with mass serial production this orders may have small serial or single nature. The most orders are irregular, i.e. they cannot be planned beforehand but nevertheless they are enough profitable for 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 take significantly more part of execution time then 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.

1.2. Pseudo-Boolean optimization problems

Unconstrained pseudo-Boolean optimization is an issue that 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, these are local optimization methods [1, 2] and stochastic and regular algorithms based on local search for special function classes [3-6]. Moreover, there are a number of algorithms for optimization of functions 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, taboo search [12, 13].

If there are constraints on the binary variables, one of 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 can 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 respect to optimization space. That makes difficult searching 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 applying are given.

2. Search Algorithms for Pseudo-Boolean Optimization

2.1. 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 $B_2 = \{0,1\}$, $B_2^n = B_2 \times B_2 \times \cdots \times B_2$.
- 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 $W(X^0, X^l) = \{X^0, X^1, \ldots, X^l\} \subset B_2^n$ is called a *path* between points X^0 and X^l if for $\forall\ i = 1,\ldots,l$ 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 $\forall X^k \in O_k(X^*), k = \overline{1,n}$: $f(X^{k-1}) \le f(X^k), \quad \forall X^{k-1} \in O_{k-1}(X^*) \cap O_1(X^k)$.

Example: A polynomial of binary variables

$$\varphi(x_1,\ldots,x_n) = \sum_{j=1}^{m} a_j \prod_{i\in\beta_j} x_i \ ,$$

where $\beta_j \subset \{1,...,n\}$, is an unimodal and monotonic pseudo-Boolean function for $a_j \geq 0$, $j=\overline{1,m}$.

2.2. Problem Statement

Consider the problem of the following form

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

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:

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

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

2.3. Properties of a Feasible Set

- A point $Y \in \mathbf{A}$ is a *boundary point* of the set $\mathbf{A}$ 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 $\mathbf{A}$ with the basic point $X^0 \in \mathbf{A}$ if for $\forall X \in O_1(Y) \bigcap O_{i+1}(X^0)$ $X \notin \mathbf{A}$ holds.
- A constraint that determine a subspace of the binary variable space is called *active* if the optimal solution of the conditional problem do not coincide with the optimal solution of the appropriate problem without taking the constraint into account.

Consider some properties of a feasible set [15].

- 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:

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

- Consider a problem (1) with a constraint (2). If the constraint function (2) is an unimodal function then the set *S* of feasible solutions of the problem (1) is a connected set.

A number of limiting points of a connected feasible set for the problem (1) equals $s \leq s_{\max} = C_n^{[n/2]}$, where $[n/2]$ is the integer part of the value $n/2$. In case of $s = s_{\max}$ all limiting point belong to $O_{n/2}(X^0)$ if *n* is even and to $O_{(n-1)/2}(X^0)$ (or $O_{(n+1)/2}(X^0)$) if *n* is odd.

2.4. Transfer to Unconstrained Optimization

One of ways to take into account constraints in conditional problems is construction a generalized penalty function.

Consider the generalized function

$$F(X) = C(X) - r \cdot \sum_{j=1}^{m} \max\{0, A_j(X) - H_j\} \tag{3}$$

for a problem (1) with constraints (2).

In this case the following lemma occurs [15]:

The limiting points of the set *S* of the feasible points of the primary problem (1)-(2) with the basic point X^0 are point of local maxima for generalized penalty function (3) with parameter *r* satisfying the condition

$$r > \frac{C(Z) - C(X')}{\sum_{j=1}^{m} \max\{0, A_j(Z) - H_j\}}$$

for every limiting point $X' \in O_k(X^0)$ and every point $Z \in O_{k+1}(X^0) \cap O_1(X')$.

So, a problem (1)-(2) is identical to the problem

$$F(X) \to \max_{X \in B_2^n}, \tag{4}$$

in which a number of local maxima (in case of connectivity of the feasible set) $s \le s_{\max} = C_n^{[n/2]}$. In general case, when feasible set is not connected, a number of local maxima theoretically can amount to 2^{n-1}.

The problem (4) is solved by known search methods: local search, genetic algorithms, simulated annealing, etc.

The essential shortcoming of this approach is loss of the monotonicity property for an object function $C(X)$. By addition of even simple (for example, linear) constraints the generalized function becomes a polymodal nonmonotonic function with exponential number of local maxima.

2.5. Heuristic Algorithms for Boundary Point Search

For any heuristic of boundery point search we will consider a pair of algorithms – primary and dual [16]. 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 step 3; else $i = i + 1$ and repeat the step.
3. $X_{opt} = X_{i+1}$.

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 step 3; else $i = i + 1$ and repeat the step.
3. $X_{opt} = X_{i+1}$.

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 useful, 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 choice of a next point in step 2 of the total schemes.

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 accordance 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 problem specific, 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.

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

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 accordance with the condition

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

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

A greedy algorithm is regular algorithm, 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 found solutions. Run time of each algorithm start is constrained by

$$T \le \frac{n(n+1)}{2},$$

but average run time of a greedy algorithm and ARSB is significant larger than for other because they looks over all point of the next level in each step in distinction from RSB and MRSB which look over only one and R points correspondingly in each step in the dual scheme.

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

3. Foundry Branches Production Capacity Optimization

Production of different kind is produces in foundry branches (FB). There is specialization in every FB by 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 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 that orders were carried out all, production was made in regular more intervals in time, and the number of recustomizings of FM equipment was minimal.

Input data:

I is a number of day 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 on days, $0 < \alpha < 1$.

Variables:

For model construction introduce following binary variables:

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

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

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

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

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

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

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

Remarks:

1. For $\forall i, j, k, l$:

$$x_{ijkl} \le y_{ijkl} .$$

2. For $\forall i, j, k, l$:

$$x_{ijkl} = y_{ijkl} \cdot (1 - y_{i-1,jkl}) \quad (5)$$

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

3.1. Optimization Model

1. *The objective function and the main constraints*

$$C(X) \to \min , \quad (6)$$

$$A_l^1(Y) \ge W_l , \ l = \overline{1, L} , \quad (7)$$

$$A_i^2(Y) \ge \alpha \cdot W^I , \ i = \overline{1, I} , \ \alpha \in (0,1) , \ W^I = \sum_{l=1}^{L} W_l \Big/ I , \quad (8)$$

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

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

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

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

2. *The additional constraints*

$$\sum_{l=1}^{L} y_{ijkl} \le 1 \ \left(\sum_{l=1}^{L} x_{ijkl} \le 1 \right), \ i = \overline{1, I} , \ j = \overline{1, J} , \ k = \overline{1, K_j} , \quad (9)$$

$$y_{ijkl} \le z_{jl} \ (x_{ijkl} \le z_{jl}), \ i = \overline{1, I} ,$$

$$j = \overline{1, J} , \ k = \overline{1, K_j} , \ l = \overline{1, L} . \quad (10)$$

3.2. Model Properties

1. There are two spaces of binary variables (denote their by B^X and B^Y) corresponding 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 (5). 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 (6) 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 quadratic and unimodal nonmonotonic with the minimum point $Y^0 = (0,...,0)$.

3. The constraint function (7) and (8) in space B^Y are quadratic and unimodal monotonic pseudo-Boolean functions with the 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 (9). In space B^Y this level corresponds to the case when production is produces on each BM in 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 then on *Y*. As the constraint function (7), (8) from *X* are not defined, we should find values of these functions from the corresponding point *Y*. There are perhaps several such points

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

in some of them 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):

$$Y = \arg \max_{Y_h, h=\overline{1,H}} \left(\sum_{i,j,k,l} y_{ijkl}^h \right), \text{ where } Y_h = (y_{1111}, ..., y_{IJK_JL}).$$

One of 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 go to step 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 higher than the requisite value (this does not contradict to the constructed model but this can influent on uniformity of capacities loading which is optimized on the next stage). So when the first stage of search has ended, we should define Y_{opt} by the rule

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

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

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 go to step 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 are no any needs in this transformation during the search. It is necessary only for determining result Y_{opt}.

3.3. 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 than 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 accessible solution at all. It is a sequel of problem specific: a large amount of different constraints and, as a result, a comparative small accessible region.

The average results of solving 10 problems of month planning capacity loading are presented in the table 1. The average values of input data:

$I = 31$, $J = 3$, $K_1 = 12$, $K_2 = 9$, $K_3 = 7$, $L = 36$, $\alpha = 0.5$,

$V_l \in [40,50]$, $W_l \in [20,25000]$.

Herewith the total dimension of the binary vector is $n = 31248$.

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 $T \approx 8 \cdot 10^5$.

Table 1. Optimization results

Algorithm	Number of starts L	Found solution C_{opt}
RSB	2000	Not found
Greedy	1	49
MRSB, R=1000	12	47
MRSB, R=100	60	52
MRSB, R=10	200	Not found
GA*	-	Not found

* - GA parameters: turnir selection with the turnir value 5, population value 100, the largest number of generations 8000, mutation probability 0.0001.

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 efficiency, precision of the found solutions differs unessentially under nearly equivalent run time.

So, the algorithms of boundary point search show high efficiency for solving the pseudo-Boolean optimization problem with unconnected accessible region. The most efficient algorithms for the considered problem are the dual algorithms MRSB and greedy.

Acknowledgments

The results were obtained within the project No. 346 of the state task No. 2014/211 of the Ministry of Education and Science of the Russian Federation.

References

[1] Aarts E H L and Lenstra J K 1997 Local Search in Combinatorial Optimization *Wiley-Interscience Series in Discrete Mathematics and Optimization* (John Wiley & Sons Ltd)

[2] Papadimitriou C H and Steiglitz K 1998 *Combinatorial Optimization* (New York: Dover Publications)

[3] Crama Y and Hammer P L 2011 *Boolean Functions: Theory, Algorithms, and Applications* (New York: Cambridge University Press) p 687

[4] Masich I S 2013 *Search Algorithms for Conditional Pseudoboolean Optimization* (Krasnoyarsk: Siberian State Aerospace University) p 160

[5] Björklund H, Sandberg S and Vorobjov S 2002 Optimization on Completely Unimodal Hypercubes *Technical report* 2002-018 p 39

[6] Wegener I and Witt C 2002 On the Optimization of Monotone Polynomials by Simple Randomized Search Heuristics *Technical Report* 1433-3325

[7] Hammer P L and Rudeanu S 1968 *Boolean Methods in Operations Research and Related Areas* (Springer) p 310

[8] Boros E and Hammer P L 2002 Pseudo-Boolean Optimization *Discrete Applied Mathematics* **123(1-3)** pp 155-225

[9] Allemand K, Fukuda K, Liebling T M and Steiner E 2001 A polynomial case of unconstrained zero-one quadratic optimization *Math. Program.* A **91** pp 49-52

IOP Conf. Series: Materials Science and Engineering **122** (2016) 012001 doi:10.1088/1757-899X/122/1/012001

[10] Boros E and Hammer P L 2008 A Max-Flow Approach to Improved Lower Bounds for Quadratic Unconstrained Binary Optimization (QUBO) *Discrete Optimization* **5(2)** pp 501-529

[11] Foldes S and Hammer P L 2000 Monotone, Horn and Quadratic Pseudo-Boolean Functions *Journal of Universal Computer Science* **6(1)** pp 97-104

[12] Goldberg D E 1989 *Genetic algorithms in search, optimization, and machine learning* (Addison-Wesley)

[13] Schwefel H P 1995 *Evolution and Optimum Seeking* (N.Y.: Whiley Publ.) p 612

[14] Antamoshkin A N 1989 *Regular optimization of pseudo-Boolean functions* (Krasnoyarsk: Krasnoyarsk university publ.) p 284

[15] Antamoshkin A N and Masich I S 2007 Pseudo-Boolean optimization in case of unconnected feasible sets *Models and Algorithms for Global Optimization* vol 4 of the series *Optimization and Its Applications* ed A Torn and J Zilinskas (Springer) pp 111-122

[16] Antamoshkin A N and Masich I S 2006 Heuristic search algorithms for monotone pseudo-boolean function conditional optimization *Engineering&automation problems* vol 5, N 1 pp 55-61