

MODELING AND OPTIMIZATION OF THE PRODUCTION CAPACITY LOADING BY WAY OF FOUNDRY MANUFACTURE EXAMPLE

Doctor of technical sciences, professor, Antamoshkin, A., candidate of technical sciences
Antamoshkina, O.

Siberian state aerospace university named after academician M.F. Reshetnev, Krasnoyarsk, Russia

Abstract: The paper presents a multicriteria mathematical model of order execution optimization in a foundry production. The model is produced in terms of pseudo-Boolean optimization theory; different search optimization methods were used to solve the obtained problem.

KEYWORDS: PRODUCTION CAPACITY LOADING, FOUNDRY MANUFACTURE, PSEUDO-BOOLEAN OPTIMIZATION

1. Introduction

The topical problem of contemporary industrial association is the necessity to solve the problem of optimization of production capacity loading while orders from numerous partners are not regular, moreover they can be low-volume or even single along with mass-production or mass serial production orders. The majority of orders are characterized as non-regular and they cannot be planned in advance but, meanwhile, they are profitable enough for the enterprise. So, this leads to the necessity to solve the problems of planning production loading many times in random moments of time, but that demands the ability new small orders to put in intervals between high-volume orders loading. Moreover, it is necessary to take the time, which is needed for resetting of equipment for low-volume production, into consideration. It takes more time of production period in comparison to high-volume production. The existence of low-volume and single orders demands practically of continuous process of capacity loading planning. The making program of equipment loading for the industrial association and its branches is time-taking and logically difficult task. Let us consider this problem on the example of foundry manufacture.

2. The premises and the means of solving the problem

The foundry technology is the process of foundry work piece getting by casting molten metal in moulds, the hollow of which totally copies cast configuration. While cooling, the metal becomes hard and takes the configuration of form-hollow. The foundry branch produces small and large pigs, flat and cylindrical bars. The foundry branches' capacity includes semi continuous founding and conveyor machines. The casting of small pigs is done by different machines of conveyor type. Semi continuous casters are for getting aluminum bars of different shape and section area. Each foundry branch has concrete specialization on production types that might be done there. Certain number of orders must be done to the given moment of time. Each order has its volume, type and time of production. The type of production is characterized by shift productivity. The plant has three shifts. The change of production, done by the foundry machine, needs its reset and takes one shift except cases when the section of products is the same. The reset of a machine of the conveyor type takes some minutes, so it is not taken into account.

Thereby, it is necessary to load production capacity of foundry branches to make all orders, to get output in steady periods of time and to minimize the number of equipment resetting.

Introduce the following notations:

I – number of planning days;

J – number of foundry branches;

K_j – number of foundry machines in branch j , $j = \overline{1, J}$;

K'_j – number of conveyors in branch j , $j = \overline{1, J}$;

Q – number of orders for goods produced by foundry machines;

S – number of orders for small pig produced by conveyor;

V_q – productivity of foundry machines producing goods q

during a shift, $q = \overline{1, Q}$ (one considers that each order is equal to certain type of output);

V' – productivity of a conveyor producing small pigs during a shift;

T_q – time of order q fulfillment (for goods q production) by foundry machine, $q = \overline{1, Q}$;

T'_s – time of order s for small pig fulfillment, $s = \overline{1, S}$;

W_q – volume of order q (for goods q production) done by foundry machine, $q = \overline{1, Q}$;

W'_s – volume of order s for small pig, $s = \overline{1, S}$;

z_{jq} – characterizes the specialization of foundry branch on foundry machines:

$$z_{jq} = \begin{cases} 1, & \text{if foundry machines of foundry} \\ & \text{branch } j \text{ can produce goods } q, \\ 0, & \text{in opposite case,} \end{cases}$$

z'_{js} – characterizes the specialization of foundry branch on conveyors:

$$z'_{js} = \begin{cases} 1, & \text{if conveyors of foundry branch } j \text{ can produce} \\ & \text{goods } s, \\ 0, & \text{in opposite case;} \end{cases} M_e -$$

the set of goods the production of which does not need the reset of foundry machines (the goods have the same section), $e = \overline{1, N}$,

$$M_1 \cup \dots \cup M_N = [1, 2, \dots, Q],$$

$$M_{e_1} \cap M_{e_2} = \emptyset$$

$$\forall e_1, e_2 \in [1, 2, \dots, N]$$

α – coefficient of stiffness restrictions on demands of uniformity of production in days, $0 < \alpha < 1$, the bigger α , the stronger is the requirement of uniformity.

In order to make the model, it is necessary to introduce the following boolean variables:

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

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

$$Y' = \{y'_{ijks}\} \in B_2^{n'},$$

where $B_2 = \{0, 1\}$, $B_2^n = B_2 \times B_2 \times \dots \times B_2$ – is a set of boolean variables.

$$y_{ijkq} = \begin{cases} 1, & \text{if on day } i \text{ goods } q \text{ is produced on foundry} \\ & \text{machine } k \text{ of foundry branch } j, \\ 0, & \text{in opposite case;} \end{cases}$$

$$x_{ijkq} = \begin{cases} 1, & \text{if on day } i \text{ the production of goods } q \text{ on} \\ & \text{foundry machine } k \text{ of foundry branch } j \text{ starts,} \\ 0, & \text{in opposite case;} \end{cases}$$

$$y'_{ijks} = \begin{cases} 1, & \text{if on day } i \text{ small pig } s \text{ is produced on conveyor } k \\ & \text{of foundry branch,} \\ 0, & \text{in opposite case,} \end{cases}$$

The general dimension of boolean vector $Y(X, Y')$ is:

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

Remarks.

1. $x_{ijkq} \leq y_{ijkq} \quad \forall i, j, k, q$ - if goods q are produced on foundry machine k of foundry branch j on j -day, so their production will start that very or the previous day.

2. $x_{ijkq} = y_{ijkq} \cdot (1 - y_{i-1, jkq}) \quad \forall i, j, k, q$
 $(y_{0, jkq} = 0 \quad \forall j, k, q)$ - if goods q production on foundry machine k of foundry branch j starts on day i , so they should not be produced the previous day. It is obvious that $y_{i+1, jkq}$ and $y_{i-2, jkq}$ can be taken as zero and unit value as well.

3. The solution of the problem

The process of decision search has two successive stages. The first stage is loading of foundry machines in a way, to minimize the number of resetting of equipment and to fulfill the restrictions on demands of production output and uniformity of produced goods in time.

The second stage is loading conveyor machines in such a way, to make the volume of produced goods uniform in time (taking in account the goods being produced by foundry machines) and to fulfill the restrictions on demand output.

The first stage.

1. Objective function and main restrictions:

(1) $C(X) \rightarrow \min$

- the number of foundry machines resetting is being minimized,

(2) $A_q^1(Y) \geq W_q, q = \overline{1, Q}$

- the given volumes on each type of output should be produced,

$A_i^2(Y) \geq \alpha \cdot W^i, i = \overline{1, I}, \alpha \in (0, 1),$

(3) $W^i = \sum_{q=1}^Q W_q / I$

- the loading of foundry branches in planned period should be uniform in days,

$$C(X) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^{K_j} \sum_{q=1}^Q x_{ijkq} =$$

$$= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^{K_j} \sum_{q=1}^Q y_{ijkq} \cdot (1 - y_{i-1, jkq})$$

$$A_q^1(Y) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^{K_j} V_q \cdot y_{ijkq} \cdot \left(2 + \sum_{q_1 \in M(q)} y_{i-1, jkq_1} \right) - \quad \text{the}$$

resetting of foundry machines takes a shift, so in a day of resetting the goods are produced during two shifts, all the rest time – during three ones,

$$A_i^2(Y) = \sum_{j=1}^J \sum_{k=1}^{K_j} \sum_{q=1}^Q V_q \cdot y_{ijkq} \cdot \left(2 + \sum_{q_1 \in M(q)} y_{i-1, jkq_1} \right)$$

- see above,

$M(q) = \{M_e : q \in M_e, e = \overline{1, N}\}$ - sets of goods equal in section (the reset of foundry equipment is not required),

$$y_{0, jkq} = 0 \quad \forall j, k, q.$$

2. Additional restrictions:

$$(4) \sum_{q=1}^Q y_{ijkq} \leq 1 \quad \sum_{q=1}^Q x_{ijkq} \leq 1,$$

where $i = \overline{1, I}, j = \overline{1, J}, k = \overline{1, K_j},$

$$(5) y_{ijklq} \leq z_{jq} \quad x_{ijkq} \leq z_{jq},$$

where $i = \overline{1, I}, j = \overline{1, J}, k = \overline{1, K_j}, q = \overline{1, Q}.$

3. Auxiliary restrictions:

$$(6) y_{ijkq} = 0 \quad (x_{ijkq} = 0) \quad \forall i > T_q,$$

where $j = \overline{1, J}, k = \overline{1, K_j}, q = \overline{1, Q}.$

The second stage.

1. Objective function and main restrictions:

$$(7) \max_{1 \leq i \leq I} A_i^2(Y') \rightarrow \min$$

- uniform daily loading of conveyors in planned period of time,

$$(8) A_s^q(Y') \geq W'_s, s = \overline{1, S}$$

- required volumes of output should be produced,

$$\text{where } A_s^q(Y') = \sum_{i=1}^{T_q} \sum_{j=1}^J \sum_{k=1}^{K_j} 3 \cdot V' \cdot y'_{ijks},$$

$$A_i^2(Y') = A_i^2(Y) + \sum_{j=1}^J \sum_{k=1}^{K_j} \sum_{s=1}^S 3 \cdot V' \cdot y'_{ijks}$$

- the first summand defines the production of foundry machine.

2. Additional restrictions:

$$(9) \sum_{s=1}^S y'_{ijks} \leq 1,$$

where $i = \overline{1, I}, j = \overline{1, J}, k = \overline{1, K'_j},$

$$(10) y'_{ijks} \leq z'_{js},$$

where $i = \overline{1, I}, j = \overline{1, J}, k = \overline{1, K'_j}, s = \overline{1, S}.$

3. Auxiliary restrictions:

$$(11) y_{ijks} = 0, \forall s > T'_j$$

where $j = \overline{1, J}, k = \overline{1, K'_j}, s = \overline{1, S}.$

The features of the model.

The first stage.

1. There are two spaces of Boolean variables (denoted by B^X and B^Y), corresponding to vectors X and Y . Each point $Y \in B^Y$ is being corresponded by unique point $X \in B^X$, the components of which are determined by relation:

$$x_{ijkq} = y_{ijkq} \cdot (1 - y_{i-1, jkq}) \quad (y_{0, jkq} = 0 \quad \forall j, k, q).$$

Meanwhile the point $X \in B^X$ may be corresponded by several points $Y \in B^Y$ (with different value of restriction functions).

2. Objective function (1) is linear and, moreover, it is unimodal and monotonous in the space B^X with minimum point $X^0 = (0, \dots, 0)$. In the space B^Y objective function is quadratic, moreover, it is unimodal and non-monotonous with minimum point $Y^0 = (0, \dots, 0)$.

3. Restriction functions (2) and (3) in the space B^Y are quadratic and, moreover, they are unimodal and monotonous pseudo-boolean functions with minimum point $Y^0 = (0, \dots, 0)$. In the space B^X restriction functions are not defined identically.

4. The set of admissible decisions in the spaces B^X and B^Y are top-limited by $I \cdot \sum_{j=1}^J K_j - th$ level of minimum point (X^0

and Y^0) due to the restriction (4). In the space B^Y it corresponds to the fact that each day the goods are produced by foundry machine.

5. The set of admissible decisions in the general case are an unconnected set (in the set B^Y).

Hereby, the decision is absolutely determined by variables of Y in contrast to variables of X. But objective function C (X) has perfect constructive features, so the search of optimum on the set X is more effective in comparison to the set Y. Restriction functions (2) and (3) are determined on the set Y, so it is necessary to calculate the value of these functions going from a corresponding point Y. Such points can be several:

$$X \rightarrow Y_1, Y_2, \dots, Y_H,$$

where in some such points the decision can be admissible and in some not. In this case restriction functions are monotonous and for concrete point X one should find point Y, which should belong to the level with large function value:

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

where $Y_h = (y_{1111}, \dots, y_{LJK_Q})$, H- number of levels.

Nevertheless, the transform algorithms, given below, permit to calculate optimal value X that minimizes number of foundry machines' resets, as well as optimal value Y that gives maximum possibility (in accordance with determined X_{opt}) to load foundry equipment daily in an equal way.

Below possible algorithms of such transformation are given.

Algorithm 1 of transformation X into Y.

1. Suppose $N_{jk} = 0$, $j = \overline{1, J}$, $k = \overline{1, K_j}$; $i = 1$.
2. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $q = \overline{1, Q}$ make the following: if $x_{ijkq} = 1$, so $N_{jk} = q$.
3. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $q = \overline{1, Q}$ make the following: if $N_{jk} = q$, so $y_{ijkq} = 1$, otherwise $y_{ijkq} = 0$.
4. If $i < I$, so $i = i + 1$ and go to 2.

Here N_{jk} is intermediate indicator, connecting variables y_{ijkq} with variables x_{ijkq} .

Meanwhile, decision Y obtained from the found best vector X_{opt} can correspond to the situation when the volume of produced goods overcomes the demanded number greatly (that does not contradict the given model, but can influence on the uniformity of production capacity loading that is optimized during the next step). So, having finished the first stage of search, Y_{opt} one should define in accordance with rule:

$$X_{opt} \rightarrow Y_1, Y_2, \dots, Y_H,$$

$$Y_{opt} = \arg \min_{Y_h: A_q^1(Y_h) \geq W_q, q = \overline{1, Q}} \left(\sum_{i, j, k, q} y_{ijkq}^h \right).$$

In this case the transform algorithm differs a bit from the previous one.

Algorithm 2 of transformation X into Y.

1. Suppose $N_{jk} = 0$, $j = \overline{1, J}$, $k = \overline{1, K_j}$.
 - 1a. Suppose $y_{ijkq} = 0$, $i = \overline{1, I}$, $j = \overline{1, J}$, $k = \overline{1, K_j}$, $q = \overline{1, Q}$; $i = 1$.
2. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $q = \overline{1, Q}$ make the following: if $x_{ijkq} = 1$, so $N_{jk} = q$.
3. For $j = \overline{1, J}$, $k = \overline{1, K_j}$, $q = \overline{1, Q}$ make the following: if $N_{jk} = q$ and $A_q^1(Y) < W_q$, so $y_{ijkq} = 1$.
4. If $i < I$, so $i = i + 1$ and go to 2.

The condition $A_q^1(Y) < W_q$ given in the point 3 as well as point 1a gives the possibility to define final Y_{opt} - there is no strict necessity in such transformation during the search.

The second stage.

1. Objective function (7) is linear and unimodal monotonous pseudo-boolean function with minimum point $Y^{*0} = (0, \dots, 0)$.
2. Restriction functions (8) are linear and unimodal monotonous pseudo-boolean function with minimum point $Y^{*0} = (0, \dots, 0)$.
3. The set of admissible decisions in general are unconnected.

4. The results and discussion

The three perspective optimizing procedures were chosen in order to solve the problem due to the model made [1].

1. Dual greedy-algorithm for search optimization problems.
2. Dual algorithm of random search of boundary points.
3. Modified dual algorithm of random search of boundary points.

The dual algorithm here is an algorithm in which the search goes from the field of inadmissible decisions up to the boundary of admissible field opposed to straight algorithm.

The following data were taken for the tests:

$$I = 31, J = 3, K_j = 12,$$

$$K_2 = 9, K_3 = 7, Q = 36, \alpha = 0.5,$$

$$V_q \in [40, 50], W_i \in [20, 25000],$$

general dimension of boolean vector $n = 31248$.

On the bases of the experiments it was found out that modified dual algorithm of random search of boundary points is more effective than dual algorithm of random search of boundary points. The last one, in conditions of strict restrictions, such as great number of orders with shipping in different time periods, has not got acceptable solution. The solution was not also found, while using modified dual algorithm of random search of boundary points with such data as $R=10$ and $L=200$.

The modified dual algorithm of random search of boundary points is more flexible procedure in comparison to greedy-algorithm, as it allows choosing parameters L and R that influence the algorithm working time and accuracy of solution. Greedy-algorithm does not give such a possibility and, in case of large dimensions (about 30,000), searching period might be too long (0, 5-1 hours).

Minimal value $C_{opt} = 47$ was found, while using modified dual algorithm of random search of boundary points with the dimension of algorithm of about 1000 and $L=12$. In case $R=100$ and $L=60$, $C_{opt} = 52$. While using greedy-algorithm, in order to know the number of foundry capacity loading, the optimal solution was $C_{opt} = 49$.

The results of the experiments showed that the most effective (on the point of accuracy and working time) for the given problem are greedy-algorithm and modified dual algorithm of random search of boundary points. The rest algorithms that were studied do not find an admissible decision under strict restrictions on variables. It is connected with the specific character of the problem: big number of different restrictions that make admissible decisions field small enough.

5. Conclusion

The realization of suggested means in an automated control system of industrial association allows solving the problem of making capacity loading program in efficient and operative way.

6. References

1. Antamoshkin, A.N., Masich, I.S. Pseudo-Boolean Optimization in a Case of an Unconnected Feasible Set. Models and Algorithms for Global Optimization. Series: Springer Optimization and Its Applications, Vol. 4, 2006, p. 64-78.