

Improving short-term planning by incorporating scheduling consequences

Petra Heijnen*, Ivo Bouwmans, Zofia Verwater-Lukszo

Faculty of Technology, Policy and Management
Delft University of Technology

Delft, The Netherlands

Abstract

Decisions in short-term planning and scheduling in multi-product multi-purpose plants are often taken in isolation from each other and may be based on conflicting objectives. This may result in decisions that are sub-optimal for the plant as a whole. A more integral perspective on the decision-making could lead to a better overall performance. This paper describes a mathematical reformulation of the short-term planning and scheduling decisions in a bi-level top-down program. It is based on a recursive formula and a smart definition of the cutting rules, which greatly reduces the complexity of the optimization. A practical case shows that the reliability of the planning results increases significantly.

Keywords: Bi-level programming, short-term planning, scheduling, optimisation, decision support

1. Introduction

In a multi-product multi-purpose plant, decisions are usually taken at different locations and on different hierarchical levels, too often in isolation from each other (Shobry et al. 2002). Together, however, these decisions influence the overall performance of the plant to a high extent. Ryu, Dua and Pistikopoulos (2003) address plant-wide planning and network decisions. This paper aims at the planning and scheduling level. Planners as well as schedulers can only decide about their own decision factors and have no control over the others, but the decisions can mutually interfere, causing far from optimal results. This happens in particular when the respective decisions are based on conflicting objectives.

Therefore, a more integral perspective on short-term planning and scheduling decision-making in the plant is desirable to guarantee a better overall performance. Multi-level programming is a specific way of mathematically formulating such multi-level decision problems in a complex system, by incorporating the consequences of one decision on the other, aiming at a better performance of the system as a whole (e.g. Bialas et al. 1982).

* Author to whom correspondence should be addressed: p.w.heijnen@tbm.tudelft.nl

In this paper, the short-term planning and scheduling decisions in a multi-product plant, aiming at different, conflicting objectives, are mathematically reformulated in a bi-level top-down program. The method is based on a recursive formula by which the scheduling objective function can be calculated. The use of this recursive formula and a smart definition of the cutting rules significantly reduce the complexity of the search for an optimal planning.

2. Short-term planning and scheduling in a multi-product plant

In the short-term *planning*, customer orders are accepted or rejected based on a chosen criterion, which in our example is profitability. At the start of each week, the planners receive the set of customer orders from the sales department. Every customer order indicates the amount of product, the date the order should be delivered and the type of product. Since capacity is limited, only the most profitable set of orders from the order list is selected and passed on to the schedulers.

In the *scheduling*, the orders accepted in the planning stage are sequenced minimizing the total tardiness, i.e. the total time of delivery after due date. To reduce complexity, the assumption is made here that all orders are to be produced in one reactor.

The scheduling performance highly depends on the planning decision, since the schedulers have to schedule all the orders selected by the planners, whereas another selection of orders might have led to a better schedule with less tardiness.

Moreover, if in the final schedule two consecutive orders are of different types of product, extra changeover costs are made, reducing the profit. The planners, however, only estimate the profit by correcting every order with a fixed sum of changeover costs, disregarding the real consequences of the scheduling process. The schedulers, in turn, have no influence on the order selection procedure.

The problem situation described here might seem a little artificial, since planners and schedulers, working for the same company, should aim at a better *overall* performance and not only at their own objectives. Nevertheless, isolated decision making is not uncommon in practice for various reasons, such as habits, the complexity of integrated decision making, or the impossibility of unambiguously defining the influence of lower-level activities on strategic objectives.

2.1 Mathematical formulation of the planning and scheduling decisions

Each week, the planners solve the following decision problem:

|| *From the total set of customer orders, find the subset of orders that maximize the profit and for which the total available capacity is not exceeded.*

Let

$$C = \{[a_i, d_i, p_i] \mid i \in \{1, 2, \dots, m\}\} \quad (1)$$

be the total set of customer orders received from the sales department, where a_i is the ordered amount of product (in tons), d_i the due date (in working hours from the start of the week), and p_i the type of product ordered. The planners should then find the binary vector $\underline{x} = [x_1, x_2, \dots, x_m]$ for which the total profit

$$P(\underline{x}) = \sum_{i=1}^m (a_i f(p_i) - c) x_i \quad (2)$$

is maximized. Here, $f(p_i)$ is the profit that can be achieved with one ton of product of type p_i . The decision vector $\underline{x}$ is bounded by the constraint

$$\sum_{i=1}^m a_i t(p_i) x_i \leq T, \quad (3)$$

in which $t(p_i)$ is the time (in hours) needed to produce one ton of product of type p_i .

Also weekly, the schedulers solve the following optimization problem:

|| *For the selected orders received from the planners, find the production sequence that minimizes total tardiness.*

Let $S(\underline{x})$ be the set of n selected orders received from the planners,

$$S(\underline{x}) = [C_i \mid x_i = 1, i \in \{1, 2, \dots, m\}] \quad (4)$$

The schedulers should find the permutation $\underline{y} = \Pi S(\underline{x})$ for which the total tardiness

$$T(\underline{y}) = \sum_{j=1}^n \max(0, c_j(\underline{y}) - d_j), \quad (5)$$

in which d_j is the due date of the j -th order in $\underline{y}$ and

$$c_j(\underline{y}) = \sum_{l=1}^j t(p_l) \quad (6)$$

is the completion time (in working hours from the start of the week) of order j in schedule $\underline{y}$, with p_l the type of product of the l -th order in $\underline{y}$.

3. Solving the planning and scheduling problem

The planning and the scheduling decision can be solved separately by a branch-and-bound algorithm, first proposed by Land and Doig (1960). However, better results can be achieved when the consequences of the planning results on the scheduling level are incorporated in the planning process. Therefore, the planning and scheduling decisions are reformulated into a bi-level top-down decision-making problem here, assuming that the decision makers on the planning level have complete knowledge about the optimization problem solved on the scheduling level.

3.1 Bi-level programming

In a bi-level program, two levels are distinguished where closely related decisions are taken, and where the decision makers at one level can only decide about a limited set of decision variables and have no power on the decision variables of the other level.

A top-down program is a specific form of a bi-level program in which

- the decisions on the upper level determine the parameter values on the lower level, constraining the feasible region of the lower-level solution space;
- the decision on the lower level influences the degree to which the objective on the upper level is achieved;
- the decision on the upper level is taken before the decision on the lower level.

The objectives on both levels are generally conflicting in that the optimal values of the decision variables of either level influence the solution of the other level negatively. As a consequence, to determine the best performance of the overall system the different optimisation problems of both levels should be integrated and solved jointly. However, in general this is unfeasible.

The usual strategy to solve bi-level optimisation problems is to incorporate the resulting decisions of the lower level into the upper level decision and solve the optimisation problems of the lower level iteratively. This is done under the assumption that the decision makers on the upper level have complete knowledge about the objective functions and constraints of the lower level.

3.2 Planning and scheduling bi-level program

For the decisions on the planning level, a better performance of the system as a whole can be achieved when the consequences of the planning on the scheduling level are incorporated in advance.

On the scheduling level, the orders selected on the planning level are sequenced to minimize total tardiness (Pinedo, 1995). However, every changeover between different types of products in the chosen schedule will reduce the profit. Although the planners cannot decide about the final production schedule, they can expect that within the limited solution space caused by the planners' selection of orders, the schedulers will choose the optimal schedule with respect to their own objective of minimal tardiness. The planning decision problem is therefore reformulated to incorporate scheduling results:

Find the binary vector $\underline{x} = [x_1, x_2, \dots, x_m]$ for which the total profit

$$P(\underline{x}) = \sum_{i=1}^m a_i f(p_i) x_i - qC(\underline{y}^*) \quad (7)$$

is maximized, in which $C(\underline{y}^*)$ is the number of changeovers in the optimal schedule $\underline{y}^*$, belonging to the planned orders, and q is the fixed costs for one changeover.

As before, the decision vector $\underline{x}$ is bounded by the constraint (1.3). The formulation of the scheduling problem remains the same.

3.3 Top-down solving of the planning and scheduling problems

The new top-down formulation of the planning and scheduling decisions makes solving of the planning problem much more complex, since for the evaluation of the profit expected from a planning, the scheduling problem must be solved as well.

The planning problem is solved by a branch-and-bound algorithm. The complexity of a branch-and-bound search strongly depends on the efficiency of cutting branches from the solution tree. For the planning problem, the tree starts with an empty planning, $[\]$. Every node in the tree, corresponding with a binary vector $\underline{x}_1$ of length smaller than or equal to m , is split into two branches, indicating that the next order is added to the planning, resulting in node $[\underline{x}_1, 1]$, or that it is not, resulting in node $[\underline{x}_1, 0]$.

A branch of the planning tree can be excluded from further analysis if either

- the total completion time of the orders selected so far is larger than the available capacity T , or

- the potential profit, i.e. the profit obtained with the selected orders plus the total profit of all orders not considered so far, is smaller than a certain lower bound.

However, for the second cutting rule the maximum potential profit of an incomplete planning in a certain node can only be determined by solving the scheduling problem for every complete planning branching from that node. This procedure will be far from efficient.

Under the assumption that the profit achieved from a customer order will always be more than the costs for one extra changeover, q , the maximum profit achievable in a planning that branches from a certain node corresponding with $\underline{x}_1$ is at most

$$P(\lfloor \underline{x}_1, 1, 1, \dots, 1 \rfloor) - (N(\lfloor \underline{x}_1, 1, 1, \dots, 1 \rfloor) - 1) * q. \quad (8)$$

$N(\lfloor \underline{x}_1, 1, 1, \dots, 1 \rfloor) - 1$ is the number of different product types in the planned orders corresponding with $\underline{x}_1$ and in the remaining orders. Since $N(\lfloor \underline{x}_1, 1, 1, \dots, 1 \rfloor) - 1$ is a lower bound for the total number of changeovers in a schedule belonging to the planning $\lfloor \underline{x}_1, 1, 1, \dots, 1 \rfloor$, Equation (1.8) will yield a lower bound for the potential profit of an incomplete planning $\underline{x}_1$.

By using this lower bound for the profit, the scheduling problem only needs to be solved when a complete planning is achieved in one of the leaves of the planning tree. The actual profit can then be computed taking into account the real changeover costs that result from the optimal schedule for that planning. Still the complexity of the planning algorithm is high and the reuse of previous scheduling results is not that simple. Let schedule I and schedule II be the optimal schedules belonging to planning I and planning II, respectively. Even if planning I consists of the orders of planning II plus just one order extra, the orders in schedule I and schedule II may have a totally different sequence.

The complexity of calculating the tardiness of a schedule, however, can be reduced significantly by using the following recursive formula.

Let $S(\underline{x})$ be the set of selected orders in a certain planning, represented by the binary vector $\underline{x}$. Let S_1 be an ordered subset of $S(\underline{x})$ and let S_2 be the complement of S_1 , $S_2 = S(\underline{x}) \setminus S_1$. Let $T(S_2, S_1)$ be the tardiness of the orders in S_1 , when the orders in S_2 are completed before the orders in S_1 can start. Then

$$T(S_2 \setminus \{k\}, [k, S_1]) = T(S_2, S_1) + \max(0, C(S_2) - d_k) \quad (9)$$

in which $C(S_2)$ is the total completion time of all orders in set S_2 .

With this recursive formula, the planners can efficiently optimize the set of orders, which results in a better solution for the plant as a whole. The example in the next section illustrates this.

4. An illustrative example

Assume that for the coming week a company has received the fourteen customer orders given in table 1.

In this example the extra costs for one changeover amount to € 3000. The total available production time equals 90 h. The due date indicates the hour within this available time at which the order should be finished.

Table 1. Customer orders for the following week.

Order number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Amount (t)	85	55	58	50	44	36	31	29	28	26	21	19	16	15
Due date (h)	52	26	58	25	83	56	70	61	75	41	27	32	48	40
Product type	5	5	2	2	4	2	4	4	4	2	5	5	3	4

If the planning and scheduling activities are executed independently, the following results are obtained.

The optimal *planning* without incorporating the scheduling results is

$S = \{3,4,6,7,10,13,14\}$, with an estimated total profit of € 318,048.

The optimal *scheduling* of the planned orders, minimizing total tardiness, is $y^* = [4,14,10,13,6,7,3]$ (total tardiness 32 h). This schedule requires six changeovers and the actual profit is € 321,048.

Incorporating the scheduling results in advance results in a higher profit, with $S = \{3,4,6,7,9,10\}$. The optimal scheduling for this planning, with a tardiness of only 19.2 h, is $y^* = [4,10,6,3,7,9]$. Implementation of this planning and schedule algorithm, including the required changeover, results in a total profit of € 332,188.

5. Conclusions

The use of a bi-level top-down optimization algorithm significantly enhances the reliability of the planning results, in that the predicted profit is closer to the actual one that is obtained after the scheduling process. In addition, the set of orders handed over to the schedulers by the planners has the potential to yield a result that is better for the plant as a whole, in contrast with the sub-optimal result that can be expected when the planning and the scheduling are optimized separately. The mathematical reformulation that is presented in this paper effectively reduces the computational complexity. Using this process, that can be completed within reasonable time, the planners are able to anticipate the decision of the schedulers.

References

- Bialas Wayne F., Mark H. Karwan, 1982, On two-level optimization, IEEE transactions on automatic control, Vol. AC-27, No. 1.
- Land A. H., A. G. Doig, 1960, An Automatic Method for Solving Discrete Programming Problems, Econometrica, Vol.28, pp. 497-520.
- Pinedo M., 1995, Scheduling: Theory, Algorithms and Systems, Prentice Hall.
- Ryu, Jun-Hyung, Vivek Dua, Efstratios N. Pistikopoulos, 2003, A bilevel programming framework for enterprise-wide process networks under uncertainty, Computers & Chemical Engineering.
- Shobrys, Donald E., Douglas C. White, 2002, Planning, scheduling and control systems: why cannot they work together. Computers & Chemical Engineering 26, 149-160.

Acknowledgements

The research is part of the program Next Generation Infrastructures. For more information: www.nginfra.nl