# HIERARCHICAL METHOD OF RESCHEDULING FOR ASSEMBLY LINES WITH INTERMEDIATE BUFFERS

## *Hierarchiczna metoda reharmonogramowania montażu w liniach montażowych z buforami międzyoperacyjnymi*

**Marek MAGIERA** ORCID 0000-0002-9026-4020

**DOI: 10.15199/160.2020.2.1**

**Abstract:** A method of scheduling assembly in flexible assembly lines without parallel machines is presented. The method applies to lines fitted with intermediate buffers with limited capacities. The developed method is distinguished by the possibility of rescheduling. This is very significant in the case of a need to provide for new, urgent orders, as well as machine failures. The first level of the method concerns balancing the load of the assembly machines. Starting times of individual operations are determined at the second level of the method. Integer programming was used to solve the tasks allocated to both levels of the method. The results of computational experiments regarding the method are described.

**Keywords:** scheduling, optimization, assembly routes, integer programming, heuristic

**Streszczenie:** Zaprezentowano metodę budowy harmonogramów montażu w elastycznych liniach montażowych bez maszyn równoległych. Metoda dotyczy linii wyposażonych w bufory międzyoperacyjne o ograniczonych pojemnościach. Opracowaną metodę wyróżnia możliwość reharmonogramowania. Ma to duże znaczenie w przypadku konieczności uwzględnienia nowych, pilnych zleceń, a także awarii maszyn. Pierwszy poziom metody dotyczy równoważenia obciążeń maszyn montażowych. Wyznaczenie czasów rozpoczęcia wykonywania poszczególnych operacji dokonywane jest na poziomie drugim metody. Do rozwiązania zadań przyporządkowanych obu poziomom metody zastosowano programowanie całko-witoliczbowe. Zamieszczono wyniki eksperymentów obliczeniowych dotyczących metody.

**Słowa kluczowe:** harmonogramowanie, optymalizacja, marszruty montażu, programowanie całkowitoliczbowe, heurystyka

### Introduction – the reasons for construction of the method

Variable conditions of the assembly process often disturb performance of the designated assembly schedule. Malfunctions of the assembly machines and the resulting limited availability of the machines often cause inability to assemble the products in accordance with the original schedule. In such a case, it is recommended to rebuild the assembly schedule, taking into account the limited availability of the assembly machines. This action is referred to as rescheduling.

Another reason for rescheduling is to take into account new, urgent orders for product assembly. This gives a competitive advantage, as new orders can be completed in a relatively short time. Rescheduling can also be the result of modified requirements of the customers [7]. The reasons for rescheduling and the associated issues are described in detail in papers [3] and [14].

The aforesaid technical (limited machine availability) and economic (ability to complete the new orders in a short time) aspects are the cause of creating the assembly rescheduling method, described further on.

It should be emphasized that each rescheduling is a specific type of scheduling. Assembly scheduling consists in assignment of assembly operations to the machines and determining the starting times for these actions [8]. In the case of the developed method, it is possible to retain a part of the original schedule, and rescheduling is only performed for certain products.

The rescheduling method concerns simultaneous assembly of numerous products of different types. This involves the need to take into account numerous parameters and variables, which affects the computational complexity and duration of the computations related to the creation of a new schedule. The issues related to processing of large amounts of data are broadly discussed in study [12]. In order to reduce the size of the problems to be solved, the hierarchical concept was applied. The developed method is two-level. At the first level, operations are assigned to the machines; the second level is about separating the operations in time. An alternative concept is the monolithic approach, where both tasks are solved simultaneously. The advantages and disadvantages of both concepts are described in the study [12].

The literature covering the issues of task scheduling for production flow systems is very rich. The papers [10] and [15] are dedicated to the description of the review of the used methods. The authors of the paper [10] have classified these methods. They have presented monolithic and hierarchical methods used for determination of optimum solutions, heuristics, and hybrid approaches. The classification of the methods used for task scheduling for multi-stage assembly lines is described in the paper [9]. Creation of presented in the paper mathematical models concerning the developed method was inspired, e.g., by the studies: [8], [11], and [13]. These papers show very good perspectives for using mathematical programming in assembly planning.

## General description of the hierarchical method

The method applies to unidirectional assembly lines with intermediate buffers of limited capacities. In these buffers, the products await performance of subsequent operations only when it is not possible to transport the product to the next machine, or the next machine performs operations on another product. An example of an assembly line setup is shown in Fig. 1. It is an assembly system without parallel machines.

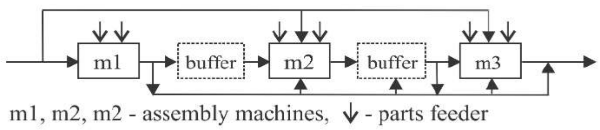


Fig. 1. Unidirectional assembly line with intermediate buffers

A block diagram of the developed method is shown in Fig. 2. At the first level, the assembly operations are assigned to the machines. Some of the operations can be performed in accordance with the original schedule, if the decision-maker so chooses. The machine loads are equivalent. At the second level, operation starting times are determined for the rescheduled products. The shortest possible schedules are determined.

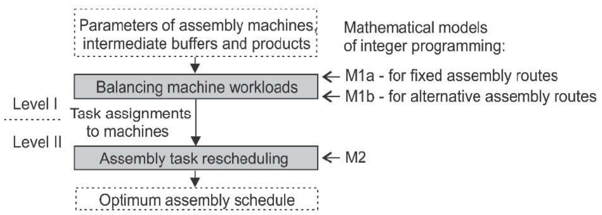


Fig. 2. Block diagram of the two-level method

The rescheduling method includes two types of assembly routes. In the case of fixed routes, every type of operation is assigned to exactly one machine. Alternative assembly routes are characterized by the fact that each type of operation is assigned to at least one machine.

In order to solve the problems regarding individual levels, integer programming was applied. Designations of the constructed linear mathematical models assigned to individual levels are listed in Fig. 2.

Use of the integer programming made it possible to computation solutions to optimum partial tasks concerning the individual levels of the method. Yet, as the hierarchical concept was applied, the designated schedule may be encumbered with a deviation from the optimum. The developed method is classified as heuristic. The issues concerning heuristics are described, e.g., in studies [4] and [5]. One of the advantages of the heuristics applied is the possibility of solving large tasks in a relatively short time.

The following chapters present the mathematical description of the developed method, followed by the results of the computational experiments concerning the method.

## Mathematical description of the hierarchical method

A list of designations used in the created mathematical models regarding individual levels of the method can be found in Tab. 1. Among the parameters and sets listed in this table, one can distinguish data concerning the original schedule (e.g. set Z), as well as data describing the products which require rescheduling.

Table 1. Notation of sets, parameters and variables used in the mathematical models

| | | |
|---|---|---|
| **Basic sets:** | | |
| $I$ | – | the set of assembly machines; $I = \{1, \ldots, M\}$; |
| $J$ | – | the set of types of assembly operations; $J = \{1, \ldots, N\}$; |
| $K$ | – | the set of types of assembly products; $K = \{1, \ldots, W\}$; |
| $L$ | – | the set of periods; $L = \{1, \ldots, H\}$; |
| **Others sets:** | | |
| $I_j$ | – | the set of the assembly machines capable of performing operation $j$; |
| $J_k$ | – | the set of assembly operations required for product $k$, $J_k \subset J$; |
| $J^C$ | – | the set of operations which require using the part feeder, $J^C \subset J$; |
| $K^1$ | – | the set of products which are to be assembly in accordance with the original schedule, $K^1 \subset K$, |
| $K^2$ | – | the set of products which are to be assembly in accordance with the new schedule, $K^2 \subset K$; |
| $R$ | – | the set of three elements $(k, r, j)$, in which operation $j \in J_k$ is performed immediately before operation $j \in J_k$, and $k \in K^2$; |
| $Z$ | – | the set of three elements $(i, k, l)$, in which product $k \in K^2$ is assembly in accordance with the original schedule using the machine $i \in I$ during the period $l \in L$; |
| **Parameters:** | | |
| $a_{ij}$ | – | working space required for assembly operation $j$ at machine $i$; |
| $b_i$ | – | total working space of the assembly machine $i$; |
| $d_i$ | – | capacity of buffer located before assembly machine $i$; |
| $g_{\tau i}$ | – | transport time between assembly machines $\tau$ and $i$; |
| $n_{il}$ | = | 1, if machine $i$ is available during period $l$, otherwise $n_{il} = 0$; |
| $p^1_{ik}$ | – | assembly time for assembly of product $k \in K^1$ loading machine $i$; |
| $p^2_{jk}$ | – | assembly time for operation $j$ of product $k \in K^2$; |
| $r_{il}$ | – | reserved space of the buffer located before the machine $i$, where, during period, $l$, the product assembled as per the original schedule will be stored; |
| **Variables declared for the M1a and M1b models:** | | |
| $x_{ij}$ | = | 1, if type of assembly operation $j$ is assigned to machine $i$, otherwise $x_{ij} = 0$; |
| $z_{ijk}$ | = | 1, if product $k$ is assigned to machine $i$ to perform assembly operation $j$, otherwise $z_{ijk} = 0$; |
| $P_{\max}$ | | maximum machine workload; |
| **Variables declared for the M2 model:** | | |
| $q_{ikl}$ | = | 1, if product $k$ is assigned to machine $i$ in period $l$, otherwise $q_{ikl} = 0$; |
| $y_{ikl}$ | – | capacity of the buffer located before machine $i$ occupied by product $k$ in period $l$; |

Level I of the developed method concerns assignment of operations to the assembly machines. This is done by solving the task of balancing the machine load, formulated in the form of integer programming task models. These models are presented below.

The mathematical models M1 (for fixed routes) and M2 (for alternative routes)

Minimize: $P_{\max}$; (1)

Subject to: $\sum_{k\in K^2}\sum_{j\in J_k} p_{jk}^2 z_{ijk} + \sum_{l\in L}(1-n_{il}) + \sum_{k\in K^1} p_{ik}^1 \le P_{max}\,;\ \ i\in I$ (2)

$\sum_{j\in J} x_{ij} = 1\,;\ \ i\in I$ - only for the M1 model; (3)

$\sum_{j\in J} x_{ij} \ge 1\,;\ \ i\in I$ - only for the M2 model; (4)

$\sum_{j\in J^c} a_{ij}x_{ij} \le b_i;\ \ i\in I;$ (5)

$x_{ij} = 0;\ \ i\notin I_j;\ \ j\in J;$ (6)

$z_{ijk} \le x_{ij};\ \ i\in I;\ \ j\in J_k;\ \ k\in K^2;$ (7)

$\sum_{i\in I} z_{ijk} = 1;\ \ j\in J_k;\ \ k\in K^2;$ (8)

$\sum_{i\in I} z_{irk} \le \sum_{i\in I} z_{ijk}\,;\ \ (k,r,j)\in R;$ (9)

$x_{ij}, z_{ijk} \in \{0,1\};\ \ i\in I;\ \ j\in J;\ \ k\in K^2$ (10)

The function of the objective (1) concerning models M1a and M1b represents the workload of the machine which forms a bottleneck in the assembly system. It is designated in constraint (2), which takes into account both the assignment of product operations to the machines and limited availability of the machines. This constraint takes into account the products which require rescheduling, as well as those assembles in accordance with the original schedule. Constraint (3) applies only to model M1a and guarantees fixed assembly routes. In turn, constraint (4) applies to model M1b and enables alternative assembly routes in the created schedule. The following constraints guarantee: (5) – verification of working spaces for individual machines, in order to arrange the part feeders for individual products; (6) – elimination of assigning the operations to the wrong machines; (7) – assignment of operations regarding individual products to those machines which have the ability to perform the specific types of operations; (8) – assignment of all the operations involving rescheduled products to the machines; (9) – taking into account the limitations concerning the sequence of the operation and the one-way flow of the products along the assembly line; (10) – binarity of the decision variables.

The determined variable values are the input data for the task solved at the 2nd level of the method. These include the durations of the machines being loaded by the rescheduled products. They are computed using equation (11). Machine workloads by products assembled in accordance with the original schedule are given, provided for in set Z (table 1).

$t_{ik} = \sum_{j\in J_k} p_{jk}^2 z_{ijk};\ \ i\in I;\ \ k\in K^2$ (11)

The 2nd level of the method includes scheduling of the operations concerning products which require a new schedule. Below is the mathematical model, concerning the 2nd level of the method.

The mathematical model M2

Minimize: $\sum_{i\in I}\sum_{k\in K}\sum_{l\in L} q_{ikl}\,;$ (12)

Subject to: $\sum_{l\in L} q_{ikl} = t_{ik};\ \ i\in I;\ \ k\in K^2;$ (13)

$q_{ikl} = 1;\ \ (i,k,l)\in Z;$ (14)

$\sum_{k\in K} q_{ikl} = n_{il}\,;\ \ i\in I;\ \ l\in L;$ (15)

$l q_{ikl} - f q_{ikf} \le t_{ik} - 1 + (H+1)\big(1-q_{ikf}\big);\ \ i\in I;\ \ l,f\in L;\ \ l>f;\ \ k\in K^2;$ (16)

$$\frac{\sum_{l\in L} lq_{ikl}}{t_{ik}} - \frac{\sum_{l\in L} lq_{\tau kl}}{t_{\tau k}} - \frac{t_{ik}+t_{\tau k}}{2} \geq g_{\tau i};\ \ k \in K^2;\ \ \tau, i \in I;\ \ \tau < i;\ \ t_{ik}, t_{\tau k} > 0; \tag{17}$$

$$\frac{\sum_{l\in L} lq_{ikl}}{t_{ik}} - \frac{\sum_{l\in L} lq_{\tau kl}}{t_{\tau k}} - \frac{t_{ik}+t_{\tau k}}{2} - g_{\tau i} = \sum_{l\in L} y_{ikl};$$

$$k \in K^2;\ \ \tau, i \in I;\ \ \tau < i;\ \ t_{ik}, t_{\tau k} > 0;\ \ \sum_{s=\tau}^{i} t_{sk} = t_{\tau k} + t_{ik}; \tag{18}$$

$$l y_{ikl} \geq \frac{\sum_{f\in L} f q_{\tau kf}}{t_{\tau k}} + \frac{t_{\tau k}+1}{2} + g_{\tau i} - (H+1)(1-y_{ikl});$$

$$k \in K^2;\ \ \tau, i \in I;\ \ i > 1;\ \ t_{\tau k} > 0;\ \ l \in L;\ \ \sum_{s=\tau}^{i} t_{sk} = t_{\tau k} + t_{ik}; \tag{19}$$

$$\frac{\sum_{f\in L} f q_{ikf}}{t_{ik}} - \frac{t_{ik}-1}{2} - l y_{ikl} \geq 1;\ \ k \in K^2;\ \ l \in L;\ \ \ i \in I;\ \ i > 1;\ \ t_{ik} > 0; \tag{20}$$

$$\sum_{k\in K} y_{ikl} + r_{il} \leq d_i;\ \ i \in I;\ \ i > 1;\ \ l \in L; \tag{21}$$

$$q_{ikl}, y_{ikl} \in \{0, 1\};\ \ i \in I;\ \ k \in K;\ \ l \in L; \tag{22}$$

The minimized sum (12) guarantees construction of the shortest possible assembly schedules.The constraints regarding linear mathematical model M2 ensure: (13) – allocation of all the operations which cannot be performed as per the original schedule between the assembly machines; (14) – assembly as per original schedule for the assembly products to which the rescheduling does not apply; (15) – loading an assembly machine during its availability in a given period with a maximum of one assembly operation; (16) – indivisibility of operation performance in time and space – the operation is assigned to only one machine; (17) – order of operation performance in an unidirectional assembly line in accordance with a given assembly sequence and provision of time required for transport between machines; (18) – determination of time that given assembly products spend in the buffers; (19) and (20) – location of assembly products in appropriate buffers at a given period, before performing subsequent assembly operations; (21) – maintenance of the limited buffer capacities; (22) – binarity of all decision variables.

## Computational experiments with the proposed hierarchical method

The presented method was verified using computational experiments. The mathematical models were coded in the AMPL language (*A Mathematical Programming Language*) [2], and *.mps files were generated. Computations were performed using the GUROBI optimizer [16]. The computational experiments enabled comparison of schedules including fixed and flexible assembly routes. Equation (23) defines the index $f$, intended to compare the length of the schedules. The schedule lengths were determined based on equation (24).

$$f = \frac{C_{\max}^{M1a,M2} - C_{\max}^{M1b,M2}}{C_{\max}^{M1b,M2}} \cdot 100\%; \tag{23}$$

$C_{\max}^{M}$ – the length of schedule determined by means of the the M model.

$$C_{max}^{M} = \max_{i\in I, k\in K, l\in L} lq_{ikl} \tag{24}$$

The computational experiments concerned four groups of test tasks. For each of these groups, 25 test examples were solved. The parameters of these task groups and the average values of the defined $f$ index are shown in Tab. 2.

Table 2. Parameters of 4 groups of test tasks and average values of indexes f [%]

| Group | Parameters of test tasks | | | | | Index $f$ |
|---|---|---|---|---|---|---|
| | *M* | *W* | *W¹* | *N* | *H* | |
| 1 | 3 | 4 | 1 | 10 | 16 | 9.2 |
| 2 | 4 | 5 | 2 | 12 | 18 | 8.9 |
| 3 | 4 | 6 | 2 | 14 | 20 | 10.3 |
| 4 | 5 | 7 | 3 | 16 | 22 | 12.7 |

Numbers of: $M$ – assembly machines, $W$ – types of products, $W^1$ – types of products assembled according to the original schedule, $k \in K^1$; $N$ – types of assembly operations, $H$ – periods.

The makespan has been divided into periods $l$ (unitary time intervals) where $l \in L = \{1,\dots, H\}$. Taking into consideration too high a number of the periods may result in a major increase in the size of the solved problem, which may result in a relatively long computation time or lack of the possibility of finding any solution to the problem due to limited possibilities of the discrete optimization packets. A low value of the periods may result in the inability to solve the problem when assembly machines should be loaded for a longer time. The parameters $H$ were determined based on the procedure described in the paper [6].

Thanks to the computational experiments, it was possible to measure the lengths of the schedules and compare them. The results presented in the table show that the fixed route schedules are about 9.2–12.7% longer than those which enable alternative routes. This results mostly from the fact that in the case of alternative routes, the load durations of individual machines are slightly different, unlike the schedules concerning alternative routes.

The computations times for fixed routes were about 18% shorter than those needed to determine schedules with alternative routes.

## Conclusions

The most important advantage of this method is the possibility of rescheduling. This method enables creation of schedules where new operations can be added to pre-made schedules. This way, new, urgent orders can be completed in a relatively short time. This is a response to the market requirements which grants a competitive edge to any company that uses rescheduling. Another advantage of rescheduling is the possibility of quickly building new schedules in the case of machine malfunctions. The new schedule is adapted to the updated setup of the assembly line from which the damaged machine was eliminated.

The short times of building the new schedules result from application of the hierarchical concept. As the problem to be solved is divided into two tasks, the problems are smaller and the tasks can be solved in a shorter time. Another benefit of the presented multi-level method is an ability to solve problems of relatively larger sizes, compared to the monolithic method.

The presented two-level method, like any other hierarchical method [1], is characterized by a certain deviation from the optimum. However, for each partial task, an optimum solution is determined. This was achieved by applying integer programming in the constructed mathematical models. Of course, these models can be modified and adapted to the variable conditions of assembly and the requirements of the production market.

***Wydanie publikacji finansowane przez Akademię Górniczo--Hutniczą im. Stanisława Staszica w Krakowie (dotacja podmiotowa na utrzymanie potencjału badawczego).***

## References

[1] Ebadian M., Rabbani M., Torabi S.A. & Jolai F. 2009. „Hierarchical production planning and scheduling in make-to-order environments: reaching short and reliable delivery dates”. International Journal of Production Research 47(20): 5761–5789.

[2] Fourer R., Gay D., and Kernighan B. 2003. AMPL: A Modelling Language for Mathematical Programming. Pacific Grove, CA: Duxbury Press.

[3] Katragjini K., Vallada E., Ruiz R. 2013. „Flow shop rescheduling under different types of disruption”. International Journal of Production Research 51(3): 780–797.

[4] Magiera M. 2016. Wybrane metody planowania przepływów produktów przez linie produkcyjne i łańcuchy dostaw (Selected methods of planning product flows through production lines and supply chains). Kraków: Wydawnictwa AGH.

[5] Magiera M. 2019. „Heuristic method of assembly scheduling of multi-option products for assembly line with intermediate buffers”. Technologia i Automatyzacja Montażu (3): 4–10.

[6] Magiera M. 2013. „A relaxation heuristic for scheduling flowshops with intermediate buffers”. Bulletin of the Polish Academy of Sciences. Technical Sciences 61 (4): 929–942.

[7] Modrak V., Soltysova Z., Stadnicka D. 2018. „Success factors and future trends in mass customization”. Technologia i Automatyzacja Montażu (2): 42–48.

[8] Pindeo M.L. 2009. Planning and Scheduling in Manufacturing and Services. New York: Springer-Verlag.

[9] Quadt D., Kuhn H. 2007. „A taxonomy of flexible flow line scheduling procedures”. European Journal of Operational Research 178 (3): 686–698.

[10] Ribas I., Leinstein R., Framinan J.M. 2010. „Review and classification of hybrid flowshop scheduling problems from a production system and a solution procedure prespective”. Computers & Operations Research 37 (8): 1439–1454.

[11] Ronconi D.P, Birgin E.G. 2012. „Mixed-Integer Programming Models for Flowshop Scheduling Problems Minimizing the Total Earliness and Tardiness”. In Ríos-Mercado R., Ríos-Solís Y. (eds) Just-in-Time Systems. Springer Optimization and Its Applications, 60. New York, NY: Springer.

[12] Soualhia M., Khomh F., Tahar S. 2017. „Task Scheduling in Big Data Platforms: A Systematic Literature Review”. The Journal of Systems and Software 134: 170–189.

[13] Tóth A., Knuutila T., Nevalainen O.S. 2018. „Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly”. International Journal of Computer Integrated Manufacturing (9): 815–830.

[14] Vieira G.E., Herrmann J.W. & Lin E. 2003. „Rescheduling manufacturing systems: a framework of strategies, policies, and methods”. Journal of Scheduling 6: 39–62.

[15] Yadov A., Jayswal S.C. 2017. „Modelling of flexible manufacturing system: a review”. International Journal of Production Research 56(7): 2464–2487.

[16] www.gurobi.com (Gurobi Optimizer, 2019).

dr inż. Marek Magiera
AGH Akademia Górniczo-Hutnicza w Krakowie, Katedra Badań Operacyjnych
al. A. Mickiewicza 30, 30-059 Kraków, Polska
e-mail: mmagiera@zarz.agh.edu.pl