

# Cost approach to the flow-shop construction scheduling

Hubert Anysz<sup>1,\*</sup> and Michał Krzemiński<sup>1</sup>

<sup>1</sup>Warsaw University of Technology, Warsaw, Poland

**Abstract.** A flow-shop construction scheduling – especially for a higher number of tasks projected – requires the use of metaheuristic algorithms to find the optimal schedule. The criteria of optimality may vary as well as constraints assumed for the schedule. The paper presents base rules of the flow-shop scheduling as well as the methods of making the work of engaged brigades continuous. When the cost of work execution is analyzed, it can occur that the optimal schedule understood as providing the shortest total time of project execution or the shortest total slack times of working brigades is not the cheapest solution. The following cost generators are taken into account: a daily cost of each brigade (working or waiting for the work), the technology applied to work execution, penalties for not keeping the milestones. The model of a time-cost evaluation of the flow-shop time schedule is formulated, analyzed and discussed. For the illustration of the presented model, the exemplary flow-shop time schedule with some constraints assumed is analyzed with and without the lowest cost as a criterion of optimality.

## 1 Introduction

The flow-shop scheduling was adopted from production programming to the construction industry. It is suitable for modelling, when the set of works executed in the same sequence is to be executed on more than one section. Examples can be found easily:

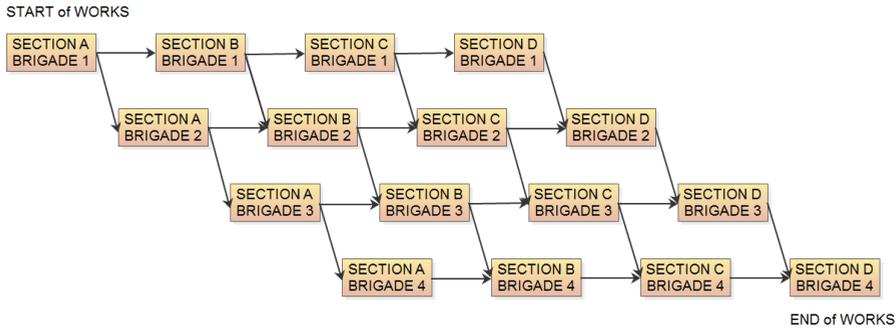
- finishing works as plastering, painting, floor finishing, door fitting in several rooms on one storey,
- the set of road construction works as excavation, dewatering system, crush-stone placing, compaction, asphalt layers placing for several sections of the road, and many other examples.

The section of a structure requires a different number of different works but the sequence of works should be kept on every section. There are three more assumptions:

- every type of work is executed by a specialized brigade,
- each brigade consists of a constant number of workers and machines,
- each brigade after the start of work stays on the building site up to completing their works on the last section (even if they have to wait some days between work execution on consecutive sections).

---

\* Corresponding author: [h.anysz@il.pw.edu.pl](mailto:h.anysz@il.pw.edu.pl)



**Fig. 1.** The exemplary network model for flow-shop schedule. Four work sections (A, B, C, D) and four brigades.

As a result of above-mentioned assumptions, the time a given brigade spend on a given section can be different. The classical problem of a flow-shop scheduling is to make a decision about the sequence of sections where the set of works should be executed in order to minimize the time when brigades wait for their work on consecutive section (to minimize so called slack-time). But there can be more goals to achieve [5]. The specific sequence of work can be found that minimizes the total time of works execution for all section [15]. Additional optimization is introduced in this paper – cost minimizing.

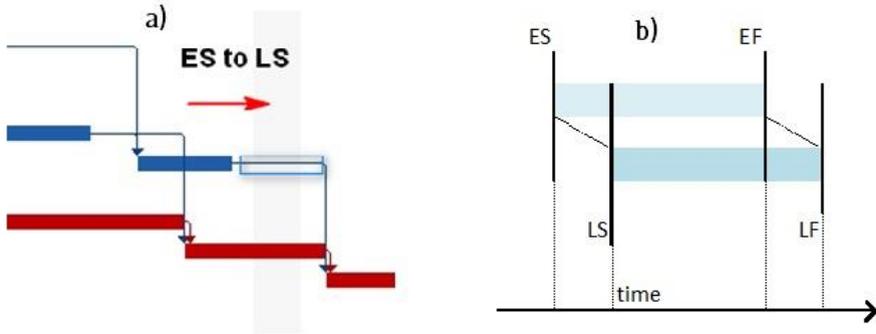
## 2 Methods of Finding the Optimal Solution

### 2.1 The Shortest Total Time of Works

There are several algorithms and methods for finding the shortest time of works. The exact solution can be found applying Johnson's algorithm or algorithms NEH (Navaz-Encscore-Ham), CDS (Campbell-Dudek-Smith). Another approach is time-coupling method introduced by Afanasjev [1] and being developed now by Mrozowicz (e.g. [9]) and Hejducki (e.g. [4]). For the limited number of tasks, i.e. 15, KASS software [7, 8] (by a complete review) produce the exact solution too. LPT (longest processing time first), SPT (shortest processing time first) and MS (minimum slack) methods give an approximate solution of the optimal sequence [10-11]. Metaheuristic methods of the shortest time finding are also developed [14].

### 2.2 Minimization of slack-time

The widely known method of minimizing slack-times is by planning the work execution based on the latest possible times (LS). The Critical Path Method (CPM) of scheduling assumes that the execution of every task starts at the earliest possible moment of time (ES) [12]. It creates the time floats for the task not lying on the critical path of the schedule.



**Fig. 2.** a) shifting start of the task not lying on critical path reduces the float b) the latest start of the task makes its finish the latest too.

Assuming the latest start of all tasks not lying on critical path cancels all floats in the time schedule [6]. Applying this for flow-shop scheduling it reduces the slack-time to zero [13]. Each brigade stays the shortest time on a building site.

### 2.3 The Minimal Cost

Independently from the variant of the work execution chosen (i.e. sequence of sections for work execution) the total cost can be calculated using the formula:

$$C^{(v)} = CL^{(v)} + CS^{(v)} + P^{(v)} \quad (1)$$

where:

- $C^{(v)}$  total cost of the variant  $v$
- $CL^{(v)}$  cost of brigades' working time
- $CS^{(v)}$  cost of brigades' slack-time days
- $P^{(v)}$  penalties for not keeping the milestones

$$CL^{(v)} = \sum_{i=1}^m ((ef_{in}^{(v)} - tf_{i1}^{(v)})B_i) \quad (2)$$

and

$$CS^{(v)} = \sum_{i=1}^m (tf_{i1}^{(v)}B_i) \quad (3)$$

where:

- $v$  number of variants
- $m$  number of brigades (or number of type of works)
- $n$  number of sections
- $ef$  the earliest possible time of a given work finish
- $B$  daily cost of a given brigade
- $tf$  the total float of time for a given task

Penalties can appear, when the client – besides the end of a project – force some midterms to be kept, so-called milestones [8]. The change of the base sequence of sections (assumed by the client) – where it is technologically possible – to the optimum sequence (minimum slack-time or minimum total time of executing the project) causes the scheduled milestones not kept. Then penalties can be calculated as follows:

$$P^{(v)} = \sum_{j=1}^n \left( (mlst_{mj}^{(v)}) * \left( \sum_{i=1}^m B_i * p_j \right) \right) \quad (4)$$

where:

$$mlst_{mj}^{(v)} = \begin{cases} ef_{mj}^{(v)} - ef_{mj}^{(b)} > 0 \rightarrow ef_{mj}^{(v)} - ef_{mj}^{(b)} \\ ef_{mj}^{(v)} - ef_{mj}^{(b)} \leq 0 \end{cases} \quad (5)$$

and

b base variant (base time schedule)

p the rate of penalty as a percent of planned total value of works carried on a given section.

There is no separate algorithm for searching minimal cost. It is calculated for the results i.e. for the variants achieved as optimal (minimum slack-time or minimum total time of executing the project). Then, the choice of a variant – considering its cost – can be made.

### 3 Results for cost minimization – example of calculation

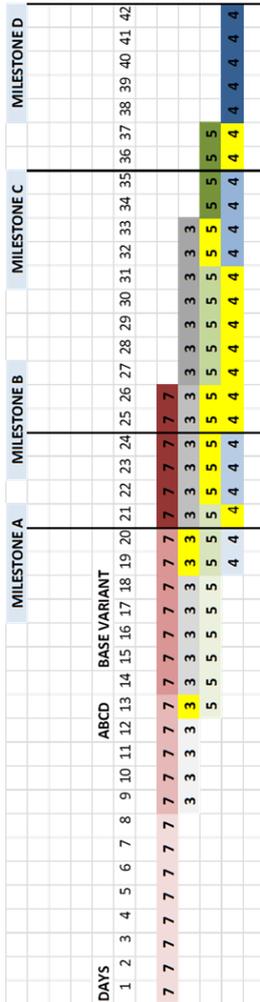
#### 3.1 Input data

It was assumed that four brigades (marked from 1 to 4) execute works on four sections (A, B, C, D). Input data were shown in tab.1.

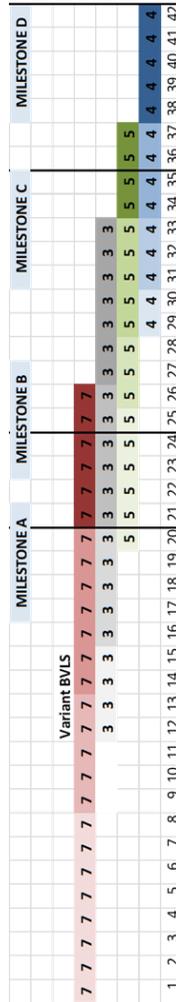
**Table 1.** Input data.

Brigade number	Days of engagement on the section				Cost per day
	A	B	C	D	
1	8	5	7	6	7
2	4	5	6	7	3
3	6	3	5	4	5
4	2	3	4	5	4

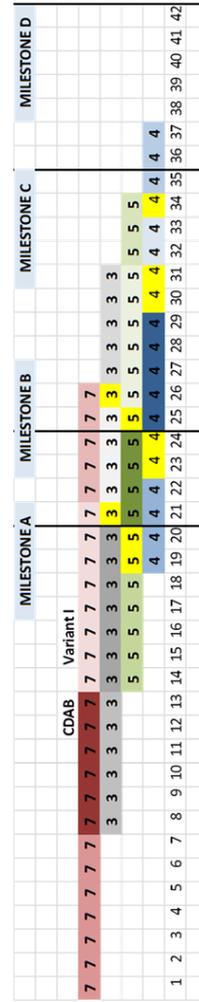
The next assumption is that the base sequence of sections (A-B-C-D) is set by the client. The time schedule (calculated for ES) is shown in fig. 3 along with four milestones stated by the finishing dates of each section. Searching for the minimal total time of the project KASS software was applied, as well as, LPT method. They produced the same result i.e. C-D-A-B sequence of the sections (shown in fig. 5). The total time is shortened from 42 to 37 days. The number of slack-times days is reduced from 20 to 10. The next step is to recalculate two aforementioned schedules from ES to LS. That creates two more possibilities (up to four in total).



**Fig. 3.** Base Variant A-B-C-D scheduled for ES

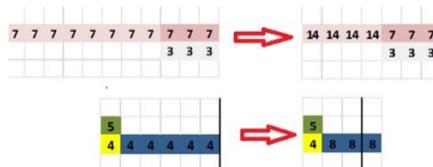


**Fig. 4.** Base variant A-B-C-D scheduled for LS



**Fig. 5.** The shortest variant C-D-A-B scheduled for ES

The base sequence A-B-C-D calculated for LS is presented in fig. 4. There is no slack-time days at all, but the total time is still 42 days. In order to shorten the total time of the project, one of assumption has been broken. The technology of work execution for the very first task and the last one have been changed. It is modeled by doubling the day cost of the brigade resulting in shortening the time execution for these two tasks. The change of technology the first task and the last one are shown in fig. 6.



**Fig. 6.** Modeling the change of technology of the very first task and the last one.

Applying that for all four analyzed previously variants, it doubles them up to following eight variants:

- BV base variant A-B-C-D, scheduled for ES,
- BVLS base variant A-B-C-D, scheduled for LS,
- VI the shortest variant C-D-A-B, scheduled for ES,
- VILS the shortest variant C-D-A-B, scheduled for LS,
- BVT base variant A-B-C-D, scheduled for ES with a change of technology,
- BVLST base variant A-B-C-D, scheduled for LS with a change of technology,
- VIT the shortest variant C-D-A-B, scheduled for ES with a change of technology,
- VILST the shortest variant C-D-A-B, scheduled for LS with a change of technology,

To calculate penalties for not keeping the milestones, it is assumed that the rate of penalty is equal to 2,5 % of the value of works on a given section.

### 3 Results

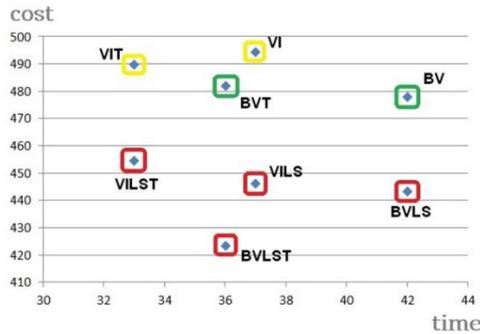
Results achieved are shown in tab. 2. It summarizes the total cost of each variant, total execution time and number of slack time days

**Table 2.** Total time, total cost and number of slack time days for each variant.

Schedule variant	Cost				Total time	Slack time days
	CL	CS	P	Total		
BV	398	84	0	482	42	20
VI	398	41	59	498	37	10
BVLS	398	0	49	447	42	0
VILS	398	3	49	450	37	1
BVT	398	84	0	482	36	20
VIT	398	41	44	483	33	10
BVLST	398	0	26	424	36	0
VILST	398	3	46	447	33	1

### 4 Discussion

There is no single, the best solution. The change of the technology shortens the total time of works. The influence of this is significant here, but when the number of sections is higher the shortening effect will be much lower. Moreover, it doesn't change the number of slack-time days – the cost of them should be covered. Changing the sequence of sections can generate shorter total time (compared to base variant) but – according to penalties appearing while changing the base sequence – total cost is raised. When applying LS scheduling, slack-time days disappear and the total cost lowers. On the other hand, slack-times in a flow-shop scheduling are the same phenomenon as floats in scheduling based on ES are. They provide time buffers which are helpful in unexpected delays in the work execution. Unfortunately, time buffers have to be paid. The cost of brigades waiting for the next section to start their work there, has to be covered. The engineer's decision about a sequence of sections, where the same set of works have to be executed, has strong financial aspect and it influence the effectiveness of the project [2]. To make the decision process easier, all variants are presented in the time-cost-risk chart (see fig. 7).



**Fig. 7.** The time-cost-risk chart. Red frames – 0 or 1 slack-time day. Yellow frame – 10 slack-time days. Green frame – 20 slack-time days.

## 5 Conclusion

When the milestones are not defined, it is possible to find the sequence of sections (in flow-shop scheduling) providing the minimal total time of works execution. Then the total cost will be lowered by the cost of excluded slack-time days. It is recommended to check all possibilities (e.g. by KASS v.2.2) than apply LPT rule. In a simple example analyzed, these algorithms work equally good but calculations made for more types of works with LPT method may not reach the minimum time (KASS v.2.2 finds the minimum for up to  $15! \cong 1,3E12$  variants). Taking into consideration penalties for not keeping the milestones – in most cases – the shortest variant will not be the cheapest one. The location of the variants found, on the time-cost-risk chart depends on number of slack-time days in the base variant, cost of brigades' working days, value of penalties, the relation of slack-time cost to the value of penalties, the base schedule. When there are more types of works to be executed on each section, covering costs of shortening the very first and the last task in the schedule will not bring so strong effect as in the analyzed example. To make a choice of a preferred variant easier, preparing the time-cost-risk chart is suggested. The issue of risk (of not keeping the completion date) should be considered. When choosing the variant where some milestones are not kept, it should be considered that it is not only a matter of penalties value. The trademark of the contractor suffers too. This effect, as being more intangible, is omitted in the presented example. It is assumed that there is no reward for total time shortening. If a reward is considered, it will affect the calculation and then affect the decision of the user of the calculation (time-cost-risk chart) as well.

## References

1. V.A. Afanasev, *Učiet zatrat vremieni na perieboizirovanije stroitielnykh organizacii pri formirovani i optimalizacii kompleksov potokov* (Aktualnyje problemy sovietskovo stroitielstva, Sankt-Petersburg, 1994)
2. H. Anysz, MATEC (2017) DOI: 10.1051/mateconf/201711700009
3. M.W. Hammad, A. Abbasi, M.J. Ryan, *Journal of Management in Engineering* **32**, 6 (2016) DOI: 10.1061/(ASCE)ME.1943-5479.0000447
4. Z. Hejducki, M. Rogalska, *Oficyna Wydawnicza Politechniki Wrocławskiej* (2011)
5. N. Ibadov, *Acta Physica Polonica A* **130(1)**, 112 – 114 (2016) DOI: 10.12693/APhysPolA.130.107

6. M. Juszczuk, Technical Transaction **2-B(6)**, 193-202 (2014) DOI: 10.4467/2353737XCT.14.127.2577
7. M. Krzemiński, Archives of Civil Engineering **LXII(2)**, 89-100 (2016) DOI: J0.1515/ace-2015-0067, ISSN 1230-2945
8. M. Krzemiński, Acta Physica Polonica A **130(6)**, 1439 – 1442 (2016) DOI: 10.12693/APhysPolA.130.1439
9. J. Mrozowicz, *Dolnośląskie Wydawnictwo Edukacyjne* (Wrocław, 1997)
10. M.L. Pinedo, Springer (2009) DOI: <http://dx.doi.org.00004dil0c92.eczyt.bg.pw.edu.pl/10.1007/978-1-4419-0910-7>
11. M.L. Pinedo, Springer (2012) DOI: <http://dx.doi.org.00004dil0c92.eczyt.bg.pw.edu.pl/10.1007/978-1-4614-2361-4>
12. E. Plebankiewicz, M. Juszczuk, J. Malara, Archives of Civil Engineering **LXI(3)**, 51-62 (2015) DOI: J0.1515/ace-2015-0024, ISSN 1230-2945
13. S.A. Mubarak, John Wiley & Sons **528** (2015) ISBN 111884579X, 9781118845790
14. H. Wang, W. Wang, H. Sun, Soft Computing **21**, 4297 (2017) DOI: 10.1007/s00500-016-2062-9
15. F. Xiong, K. Xing, F. Wang, European Journal of Operational Research **240(2)**, 338-354 (2015) DOI: 10.1016/j.ejor.2014.07.004