ESTIMATING OPTIMUM OF FLOW-SHOP SCHEDULES

Ph.D. Zoltán A.Vattai
Budapest University of Technology and Economics, Faculty of Architecture
zvattai@ekt.bme.hu

Abstract

At Flow-Shop Problems the job is to schedule work of *n* machines on *m* workpieces for to achieve a pre-set optimum – say, the shortest overall execution time. Analogy of it can be recognized at managing Industrialized Construction. Main challenge of Flow-Shop Schedules is that the mathematical problem is Non-Polynomial, and – except of some special cases – the correct solution can be gained by some kind of enumeration. For testing and demonstrating effect of sequences on overall execution time of a Master Schedule embracing execution of numerous sub-projects a special software had been developed at BUTE DCT&M. After testing numerous principal ways of achieving optimal solutions our researches lead us to the experiences:

1. Neither the simple nor the more sophisticated search algorithms for finding optimal solutions proved to be more effective than the most primitive way of random sampling;
2. Solutions later prove to be optimal or very close to optimum (pseudo-optimal solutions) may get in view in a relatively short time, but proving their optimality takes significantly longer.

The paper focuses on later lesson and introduces some ways of estimating optimum of $F||C^{max}$ problem via elaborating lower bounds by help of which we can guess how far the found or constructed solutions are from the optimum and/or how much efforts to find even better ones are likely necessiated.

Keywords: Scheduling, Flow-Shop, Industrialized Construction

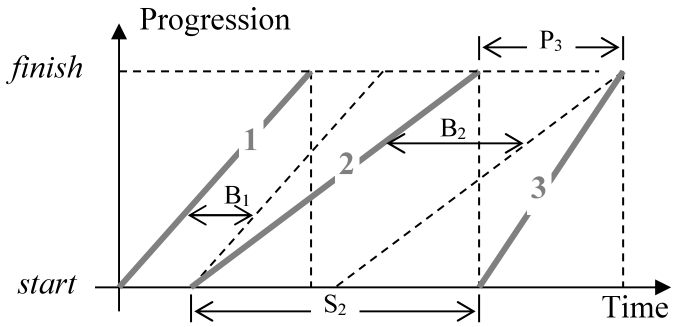
Accesses of the author:
Dr. Zoltán András Vattai, associate professor
Department of Construction Technology and Management (DCT&M)
Budapest University of Technology and Economics (BUTE)
1111-Hungary, Budapest XI. Műegyetem rkp. 3. K224
Tel.: 36-1 463-4106 Fax.: 36-1 463-3554 Mobile: 36-30 911-2253
E-mail: zvattai@ekt.bme.hu Home Page: www.ekt.bme.hu

PREFACE

For graphical representation of ideas to be discussed on following pages instead of using one-dimensional schedules (known as Bar Charts or Gantt Charts) usually applied at discussing scheduling (*sequencing*) problems, we illustrate our thoughts by two-dimensional *time-route diagrams* known as *progression curves*, *cyclograms* or as *linear schedules*, as seen below. Horizontal axis of these diagrams is *Time* while along vertical axis *Progression* can be read in any proper dimension such as m, m^2, m^3, ton, %, €, etc., that is a common characteristic of all processes (jobs) being scheduled.

Figure 1

Two-dimensional representation of timing processes (1,2,3) of a project (here: *workpiece*)



Processes are represented by individual lines (See: line „1", „2" and „3" above) slopes of which can be read as *intensity* of progression, while durations of processes (P_1, P_2, P_3) and timing of them (e.g. succession times between them either at start: S_1, S_2 or at finish: F_1, F_2) can be read as horizontal (time) views of their linear representations.

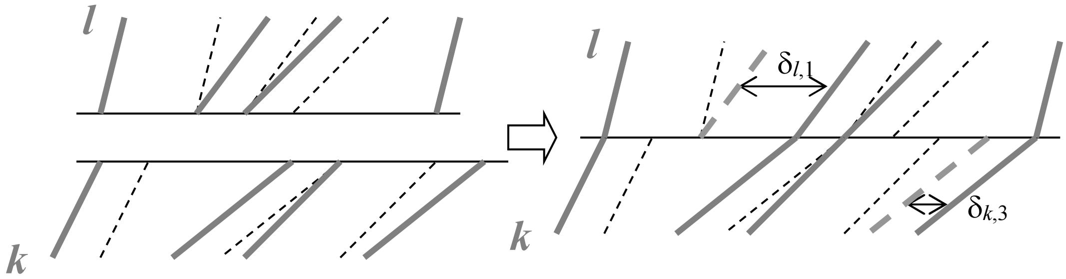
For succeeding processes with or without overlapping in time, minimum succession times (*technological breaks* or *critical approaches* succeeding the given processes represented by *B* values) are used to set minimum of their non-overlapping period.

Ranges/values of critical approaches can be defined/selected tipically by considering technical-technological constraints, such as time of hardening, cooling, drying, consolidating, etc. as constant values or can be set as functions of progression due to needed manipulation area or safety zones or on-site storage capacities and so on. Guiding relative timing of processes this way any relative time position, *lead* or *lag* time can be set.

The schedule of a workpiece in which all the succession times between succeeding processes are at their minimum (set by *B* values) we do refer as *own-schedule* or *most compact schedule* of it (as if it was processed in its own). These most compact own-schedules of workpieces will be released (succession times between neighbouring processes can and will be increased) as requires when combining or linking them up in a *Master Schedule*.

Thus the problem of forming a Master Schedule and finding on optimal sequence of workpieces for to achieve the shortest overall completion time can be derived back to the problem of matching pairs of succesion vectors (succession times „F_k" at finish on preceeding workpiece and succession times „S_l" at start on succeeding workpiece). (See: Figure 2.)

Figure 2 Releasing *own-schedules* of projects/workpieces (k and l) when linking them up in a Master Schedule with expectation of breakless performance of processes/machines. (Detail of the *Master Schedule* is to the right.)



INTRODUCTION

Though thinking it over in short it can be seen clearly, it is less widely known that sequence of buildings to be erected may have significant effect on completion time of the buildings themselves and on overall completion time of all when having the same resources assigned to them (see Figure 3).

The phenomenon is well-known in manufacturing industry and much effort have been made to help scheduling workpieces to achieve a pre-set optimal characteristic of the whole *production program*, thus calling into existence a special branch of applied mathematics (*Operations Research*) referred as *Scheduling*.

In 1973 researchers had suggested a reference system to identify scheduling problems by classification, indicating availability of resources and further restrictions completed by identification of actual target function [4]. Using their coding system *Scheduling Problem* discussed can be identified as $F|overlap,\ no\text{-}wait|C^{max}$ to be read as:

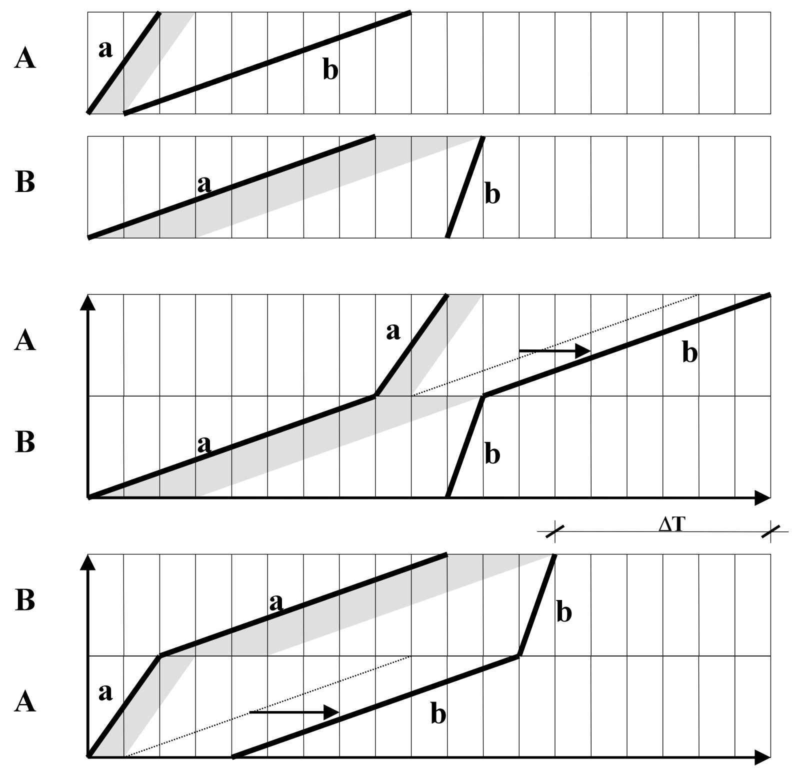
> The job is a so called *Flow-Shop* scheduling with no limit on number of machines ('F'), where overlapping in time is allowed when processing a workpiece by succeeding machines, machines should not wait when shifting from a workpiece to an other ('*overlap, no-wait*'), and the aim is to minimize the overall execution time identified by *completion time* of the last workpiece ('C^{max}').

Flow-Shop as a basic class of scheduling problems embraces a set of conditions on production environment:

- There are 'm' workpieces to be scheduled on 'n' machines (each workpiece must be processed by each machine);
- Order of machines at processing workpieces is given and is the same for each workpiece (technological order of machines is fixed and known);
- Order of workpieces must be the same on each machines;
- Machines are performing their only job (special machines for each process);
- Each process is performed by one machine (one machine is available for each kind of processes).

To demonstrate variety of scheduling problems and their coding we can mention *Job-Shop* scheduling (*J*), where technological orders of machines are given in advance but may differ by workpiece; *P* refers to situation when *parallel machines* are used and each machine can perform any job; *pre-emption* indicates when processing of a workpiece can be broken at machines; *idle* refers to allowed workless periods of machines between succeeding workpieces, while *no-wait* warns us when it is unacceptable; *res1,res2* identifies limited availability of resources; target function can be minimum of sum of *completion times* (ΣC_i), minimum of sum of *delays*-, *lateness*- or *tardiness* (ΣD_i, ΣL_i, ΣT_i) too; and so on – to mention the most frequent ones only.

Figure 3 Demonstrating effect of sequence of workpieces (‘A’ and ‘B’) to be processed by two machines (‘a’ and ‘b’). At top we see the most compact individual own-schedules of workpieces with technological breaks indicated (grey stripes, indicating minimum succession times between succeeding jobs). Below we see Master Schedules constructed off them sequencing the two wokpieces in order of B→A and another variant of it sequencing the workpieces in order of A→B. Both variants of the Master Schedule are the most compressed ones! Minimum overall completion time is (can be) strongly effected by the sequence of workpieces. (See ΔT)



Difficulty is that for to solve most of the Flow-Shop Problems there is no any existing close formula. For to find optimum some type of *enumerative algorithm* [7, 8] is needed, which may/would take enormous run-time. Computations of these kinds are referred as *NP-hard* in technical literature, which means that time (steps) needed to find solution is *non-polinomial*, it can not be defined as $n \bullet x$, or n^2, n^3, etc., where n represents the number of workpieces to be scheduled.

To demonstrate difficulties of these calculations let assume a computer calculating one million Master Schedules (permutations) of the same 20 projects in a second. Well, to examine all possible permutations that computer should work for more than

77 thousand years ($20! = 2.43 \cdot 10^{18}$; $2.43 \cdot 10^{12}$ sec > 77,000 years). So long machine time – for to solve a single problem – usually we do not have. ...

$F2||C^{max}$ (Scheduling *m* workpieces on *2* machines) is one of a few delighting problems for which there exists a polynomial solution – first published by S. M. Johnson in 1954 [1]. Having more than two machines the problem seems to be stubbornly *NP-hard*.

It is not by chance that much effort were made to derive $F||C^{max}$ scheduling problem back to $F2||C^{max}$. Szwarc published an *elimination method* in 1971 [2] later improved to an *optimal elimination method* published by him in 1973 [3]. Also Szwarc published *dominance conditions* for $F3||C^{max}$ problem in 1978 [5].

Z.A.Vattai proved *optimality* of Johnson's (1954) heuristic algorithm and generalized the solution for *overlapped* situations ($F2|overlap|C^{max}$) while introducing term of *o-shaped* (master) schedules (where succession times between the processes both at start and at finish are at their minimum and all necessary releases have been made mainly in inner sections of the Master Schedule) in 1993 [9].

Time by time comprehensive overviews are published reporting latest results of searching faster and faster methods to find solutions for Flow-Shop Scheduling problems. *Theoretical approaches* were comprehended in Proceedings of the NATO Advanced Study and Research Institute in 1981 [6]. Results are adopted in up to date *management techniques* [7] and resource pool of *finite mathematics* [8] is enriched permanently by more and more sophisticated methods of supporting *decision making*.

Recognizing difficulties of providing a sure optimum and also recognizing sensibility of optimal schedules (meaning: a slight modification of data can significantly effect final result) the empasis had turned to *pseudo-optimal* and practically acceptable solutions and to *probabilistic* and/or *combined* methods.

Amy D. Wilson et al tried to estimate the optimum on *statistical* basis and published their results in 2004 [10]. R. Ruiz and C. Maroto gave a comprehensive review on *heuristic methods* applied for Flow-Shop Scheduling problems in 2005 [11]. The latest endeavours are focusing on *generic algorithms* and on development of some kind of *hybrid heuristics* for the problem, some results of which were published by M. G. Ravetti at al. in 2006 [12].

For testing and demonstrating effect of „*Sequence*" (of projects) on minimum overall execution time of a Master Schedule comprehending execution of numerous building projects („*multi-project management*") a small software had been developed by the author at Department of Construction Technology and Management of Budapest University of Technology and Economics. Lessons of test runs had been published in 2008 [13].

Primary aim was to bring attention of our students (future managers) on extended considerations of construction management in case of executing large-scaled complex development works. After modelling and testing potential effects of sequences on total execution time we tested five principal ways of building/finding optimal sequences:

1. For to gain certain optimum, and to check any other trials, enumerative algorithms had been developed, later improved and accelerated by some methods of filtering (*Total-, Partial- and Implicite Enumeration*);
2. Building sequence as a kind of set of optimal matches of individual schedules – with the hope of deriving the problem back to a kind of *Assignment Problem* that can be solved by *Linear Programming* (*Arranged Branch & Bound*);
3. Finding partially optimal solutions for simpler cases and extending/combining them for more complex situations (*Johnson's Algorithm*);
4. Producing an initial sequence and improving it gradually via series of consecutive modifications (*Pair-wise Exchanges*).
5. Finally, for testing/measuring return of all our efforts against, a pure and primitive way of finding optimal sequence – by chance (*Random Sampling*).

Due to our principal aim of testing sequencing/scheduling as a tool of resolving some contradictions in Construction Industry and to get *real* optimum, manyfold heuristics had been deliberately excluded from our investigations.

After long times of examinations, after numerous trials and hypotheses falling apart as leaves from the trees none of the principal ways above proved to be either the only or the best way of constructing/finding *the optimal* sequence. None of the *advanced techniques* and/or approaches proved to be either deliberately or more outstandingly better or effective for our purposes than the most primitive way of *Random Trials*. But the same time, we found that elaborating a proper estimate on likely optimum is a more promising challenge. Having it, we could judge optimality of any sequence found or produced, and we could estimate likely return of our efforts to find an even better solution if the one produced did not seem to be optimal. ...

ESTABLISHING A LOWER BOUND

Before discussing data and algorithm some general recognition are worth to be highlighted. (For better understanding see Figure 4.)

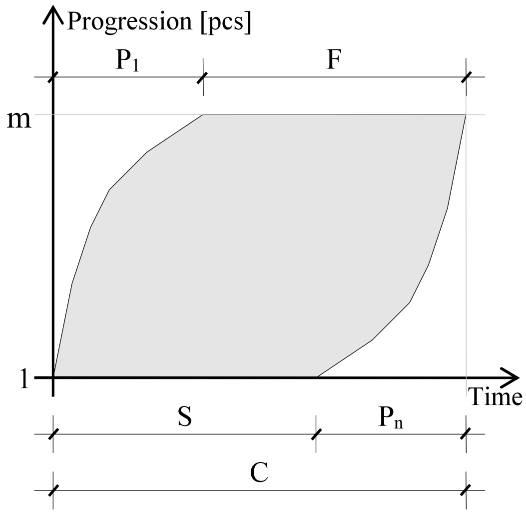
- Overall execution time of any Master Schedule can be divided in two segments:
  1. Overall processing time on the first machine (P_1), which is a constant value, not varying by the sequence of workpieces.
  2. Time span between finishing the first and finishing the last process on the last workpiece (F) actually effected by the sequence.

- Building up a Master Schedule from the very last or from the very first workpiece is a symmetrical problem, so the division of above can also be made as overall duration of processing on the last machine (P_n) and time span between starting the first and starting the last process on the first workpiece (S).

- In an optimal Master Schedule F and S values are at their minimum so these are the quantities the examinations should focuse on. Thus calculations should concentrate on *succession times*, while processing times have less or indirect importance.

Figure 4

Constant (P_1; P_n) and variable (F; S) parts of overall completion time (C) in a Master Schedule



$C = P_1 + F$

$C = S + P_n$

$P_1 = \text{const}$

$P_n = \text{const}$

$C_{min} \Rightarrow F_{min}$

$C_{min} \Rightarrow S_{min}$

Studying Figure 3 and Figure 4 it is easier to understand *o-shaped* characteristic [9] of an optimal Master Schedule: 'succession times between the processes both at start and at finish are at their minimum while all necessary releases have been made mainly in inner sections of the Master Schedule'. …

Denotions

For faster discussion we do introduce some mnemonic denotions:
(For better understanding recall Figure 1.)

<u>Denotions for own-schedules of workpieces</u>

p_{ij} = **p**rocessing time of machine j on workpiece i

$b_{i,j}$ = minimum succession time (technological **b**reak) between processes j and j+1 on workpiece i

$s_{i,j}$ = minimum succession time between **s**tarting processes j and j+1 on workpiece i

$f_{i,j}$ = minimum succession time between **f**inishing processes j and j+1 on workpiece i

s_i = minimum succession time between **s**tarting the first and the last process on workpiece i

f_i = minimum succession time between **f**inishing the first and the last process on workpiece i

c_i = minimum **c**ompletion time of workpiece i

<u>Denotions for a fictive „optimal" Master Schedule</u>

P_j = (overall) **p**rocessing time on machine j in the fictive Master Schedule

B_j = minimum succession time (technological **b**reak) between processes j and j+1 in fictive the Master Sschedule

S_j = minimum succession time between **s**tarting processes j and j+1 in the fictive Master Schedule

F_j = minimum succession time between **f**inishing processes j and j+1 in the fictive Master Schedule

S = minimum succession time between **s**tarting the first and the last process in the fictive Master Schedule

F = minimum succession time between **f**inishing the first and the last process in the fictive Master Schedule

C = minimum **c**ompletion time of the fictive Master Schedule

Denotions for estimating completion time of the optimal Master Schedule

E_1, E_2, E_3, E = partial and overall **e**stimates

At interpreting and reading the scheduling problem and the abbreviations above analogy between manufacturing and construction can be highlighted by changing some key words, such as:

- *workpiece* in manufacturing = *building* or *project* in construction;
- *machine* in manufacturing = *machine* or *team* or *subcontractor* in construction

Calculating own-schedules of workpieces

Overlapping processes in time when erecting a building is typical in Construction for to shorten overall completion time. In Manufacturing, when workpieces are moving from machine to machine this solution can not be applied. Before going on with processing a given workpiece on the succeeding machine, the preceeding machine must finish its job. Expectation of non-overlapping in time can be considered as a special case of overlapping, with succession time equal to minimum of durations of processes linked to each-other. (See processes „**2**" and „**3**" in Figure 1 – and B_2 value between them. $B_2=\min\{P_2, P_3\}$)

$b_{i,j} = \min\{p_{i,j}; p_{i,j+1}\}$ $\qquad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n-1$ $\qquad$ R 1.1

Assuming constant intensity of processing at each process, progressions can be represented as straight lines in linear schedules. Thus, minimum succession times (critical approaches, b_{ij} values) will appear between starts or between finishes (or both) of succeeding processes. Thus, succession times between starts and between finishes of processes in the most compressed individual schedules (*own-schedules*) of workpieces can be calculated:

$s_{i,j} = \max\{b_{i,j}; b_{i,j}+p_{i,j}-p_{i,j+1}\}$ $\qquad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n-1$ $\qquad$ R 1.2

$f_{i,j} = \max\{b_{i,j}; b_{i,j}-p_{i,j}+p_{i,j+1}\}$ $\qquad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n-1$ $\qquad$ R 1.3

Minimum of variable part of completion times of individual workpieces can be calculated (See analogy on Figure 4):

$s_i = \sum_{j=1}^{n-1} s_{i,j}$ $\qquad i = 1,2\ldots,m$ $\qquad$ R 1.4

$$f_i = \sum_{j=1}^{n-1} f_{i,j} \qquad i = 1,2\ldots,m \qquad \text{R 1.5}$$

As explained at Figure 3, having the constant part and having the variable part at minimum, minimum completion times and most compact (*own*-) schedules for each individual workipeces can be calculated.

$$c_i = p_{i,1}+f_i = s_i+p_{i,n} \qquad i = 1,2\ldots,m \qquad \text{R 1.6}$$

These schedules can/will/must be released (by increasing succession times between succeeding processes) when combining them into one single Master Schedule.

Calculating a fictive „optimal" Master Schedule

As most compact schedules were calculated for individual workpieces, the same way a *fictive* most compact Master Schedule can be caculated for the whole series for to estimate the theoretically achievable minimum (as lower bound). With the assumption that minimal succession times will appear between starts and/or between finishes of succeeding processes, similar logic and similar relations can be used for calculation.

The only difference is that when considering individual processing times we have to consider overall processing times on the individual machines, and when considering minimum succession times (technological breaks) we have to consider the minimum of techonolgical breaks between processing on succeeding machines.

$$P_j = \sum_{i=1}^{m} p_{i,j} \qquad j = 1,2\ldots,n \qquad \text{R 2.0}$$

$$B_j = \min_i\{b_{i,j}\} \qquad j = 1,2\ldots,n\text{-}1 \qquad \text{R 2.1}$$

$$S_j = \max\{B_j;\ B_j+P_j–P_{j+1}\} \qquad j = 1,2\ldots,n\text{-}1 \qquad \text{R 2.2}$$

$$F_j = \max\{B_j;\ B_j–P_j+P_{j+1}\} \qquad j = 1,2\ldots,n\text{-}1 \qquad \text{R 2.3}$$

$$S = \sum_{j=1}^{n-1} S_j \qquad \text{R 2.4}$$

$$F = \sum_{j=1}^{n-1} F_j \qquad \text{R 2.5}$$

$$C = P_1+F = S+P_n \qquad \text{R 2.6}$$

Estimating the optimum

Having the individual own-schedules of workpieces and having the fictive Master Schedule we can establish at least three estimates as lower bounds on the shortest overall completion time:

1.) Assuming that unreleased own-schedule of the workpiece with the shortes succession time between finishing the first and the last process on it can be the last in the optimal Master Schedule, overall completion time can not be shorter than E_1 value calculated below:

$E_1 = P_1+\min\{f_i\}$ R 3.1

2.) As it was mentioned earlier, building up a Mater Schedule is a symmetrical problem, so we can follow similar logic when constructing it from direction of start or of finish. Also, assuming that unreleased own-schedule of the workpiece with the shortes succession time between starting the first and the last process on it can be the first in the optimal Master Schedule, overall completion time can not be shorter than E_2 value calculated below:

$E_2 = \min\{s_i\}+P_n$ R 3.2

The two estimates (E_1 and E_2) can not substitute each other. We need both.

3.) We also can conclude that overall completion time of the optimal Master Schedule can not be shorter than completion time of the fictive Master Schedule where – by assumption – no inner conflict within the schedule necessitate releasing minimum succession times between succeeding processes.

$E_3 = C$ R 3.3

By our experiences none of the estimates above can be harder lower bound on the theoretically achievable minimum than the others. Accuracy of them is defined by the actual set of data.

$E = \max\{E_1; E_2; E_3\}$ R 3.4

Though the three estimates are hard lower bounds they may more or less fail the real optimum. There can be situations when the most compact own-schedule is an outstanding one and it can not be at the beginning and at the end of the Master Schedule in the same time – that is: overall processing times on individual machines override own-schedules of individual workpieces. Also, there can be situations when technological breaks are over-dominating extents of processing times, thus individual own-schedules are overriding overall processing times on the individual machines. And we can imagine any mixture of them.

For practical use a kind of combination of the three estimates had been integrated in the software mentioned above. That is, after calculating individual own-schedules, all workpieces are tested to be the first, while others to be the last one in the Master Schedule, and the rest of workpieces are substituted by a single fictive (inner, partial master) schedule with processing times equal to overall processing times on machines less processing times on the selected two workpieces and with zero minimum succession times between them. Thus $m \bullet (m-1)$ tests are performed to establish a lower bound to estimate theoretical minimum of overall completion time.

CONCLUSION

Due to standard technologies and to specialization of resources typical in Construction Industry problem of harmonizing preferences of Clients and those of Contractors can be demonstrated by the challenge of *Flow-Shop Problems*. Expectation of completing workpieces (buildings) as individual deliveries in the shortest times (Clients' interest) is not necessarily coinciding endeavour of firms contributing in completing the series of workpieces – also in the shortest overall execution time (Contractors' interest).

The question we faced was if computerized *tools of sequencing* could help us at resolving some contradictions of interest of clients and that of contractors in a way of finding proper arrangements/sequences of contracts, according to which each building could be delivered in an acceptably short period and performance of contractors could be managed in an effective *industrialized* way. …

For testing and demonstrating effect of sequence on minimum overall execution time of a Master Schedule a small computer appilcation had been developed at Department of Construction Technology and Management of Budapest University of Technology and Economics. Together with modelling and testing potential effects of the sequences more principal ways of building/finding optimal solutions had been tested.

Experiences show that none of the advanced techniques and/or approaches proved to be either deliberately or more outstandingly better or effective than the most primitive way of *Random Trials*. The same time it was found that elaborating a proper estimate on likely optimum is a more promising challenge. Having it, we can judge optimality of any sequence found or produced, and we can estimate likely return of our efforts to find an even better solution if the one produced did not seem to be optimal.

In this paper we introduced a way of estimating optimum of $F|overlap,\ no\text{-}wait|C^{max}$ problem.

REFERENCES

[1] Johnson, S. M. (1954): Optimal Two- and Three-Stage Production Schedules with Setup Times included, *Naval Research Research Logistics Quaterly*, Vol 1. PP. 61-68.

[2] Wlodzimierz Szwarc, *Elimination methods in the m·n sequencing problem*, Naval Research Logistics Quarterly 18, PP. 295-305 (1971)

[3] Wlodzimierz Szwarc, *Optimal elimination methods in the m·n flow-shop scheduling problem*, Operations Research 21, PP. 1250-1259 (1973)

[4] *Symposium on the theory of scheduling and its application*, ed. by S. E. Elmaghraby, Berlin-Heidelberg-New York, Springer, 1973

[5] Wlodzimierz Szwarc, *Dominance conditions for three-machine flow-shop problem*, Operations Research 26, PP. 203-206 (1978)

[6] *Deterministic and stochastic scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical approaches to scheduling*

problems held in Durham, England, 1981, ed. by M. A. H. Dempster, J. K. Lenstra, A. H. G. Rinnooy Kan

[7] Frank S. Budnick, Dennis McLeavy, Richard Mojena, *Principles of Operations Research for Management*, IRWIN Homewood, Illinois, 1988

[8] John W. Brown, Donald R. Sherbert, *Methods of finite mathematics*, John Wiley & Sons, Inc. 1989

[9] Zoltán A. Vattai, *A Descriptive Proof for Johnson's Algorithm for Solving Two-machine Flow-Shop Problem*, PERIODICA POLYTECHNICA SER. CIVIL ENG. VOL. 37, NO 3, PP. 249-260, Budapest, 1993

[10] Amy D. Wilson, Russell E. King, James R. Wilson, *Case study on statistically estimating minimum makespan for flow line scheduling problems*, European Journal of Operational Research 155 (2004) 439-454

[11] R. Ruiz and C. Maroto, A comprehensive review and evaluation of permutation flowshop heuristics, *European Journal of Operational Research*, Vol 165., Issue 2., pp 479-494, 2005

[12] M. G. Ravetti, F. G. Nakamura, C. N. Meneses, M. G. C. Resende, G. R. Mateus and P. M. Pardalos, Hybrid heuristics for the permutation flow shop problem, *Optimization Online*, 2006, no. 11, URL: http://www.optimization-online.org/DB_HTML/2006/11/1519.html

[13] Zoltán A. Vattai, *Laying Siege To* $F//C^{max}$ *Flow-Shop Problem – Is 'Random' The Likely Winner?*, 8th International Conference On Organization, Technology And Management In Construction, Umag, Croatia, 2008