Working Paper
n°: 2018-100-01

Semi-V-shape property for two-machine no-wait proportionate flow shop problem with TADC criterion

Sergey Kovalev [a], Mikhail Y. Kovalyov [b], Gur Mosheiov [c], Enrique Gerstl [c,d]

a INSEEC Business School, 25 rue de l'Université, 69007 Lyon, France
b United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus
c The Jerusalem School of Business Administration, The Hebrew University, Jerusalem, Israel
d School of Industrial Engineering, Jerusalem College of Technology, Jerusalem, Israel

February 2017

An ulterior version of this article appeared in International Journal of Production Research.
It can be purchased at:
https://www.tandfonline.com/doi/full/10.1080/00207543.2018.1468097

Semi-V-shape property for two-machine no-wait proportionate flow shop problem with TADC criterion

Sergey Kovalev[1*], Mikhail Y. Kovalyov[2], Gur Mosheiov[3], Enrique Gerstl[3,4]

[1] *INSEEC Business School, 25 rue de l'Université, 69007 Lyon, France, e-mail: skovalev@inseec.com*

[2] *United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova 6, 220012 Minsk, Belarus, e-mail: kovalyov_my@newman.bas-net.by*

[3] *The Jerusalem School of Business Administration, The Hebrew University, Jerusalem, Israel, e-mail: msomer@mscc.huji.ac.il*

[4] *School of Industrial Engineering, Jerusalem College of Technology, Jerusalem, Israel, e-mail: enrique.gerstl@mail.huji.ac.il*

Abstract

The problem of minimizing total absolute deviation of job completion times (TADC) in a two-machine no-wait proportionate flow shop has been recently studied. It was shown that the LPT (largest processing time first) job sequence is optimal if the number of jobs n does not exceed 7, and that the LPT sequence is not optimal for instances with $n \geq 8$. We prove that there exists an optimal *semi-V-shaped* job sequence, in which the first job has the largest processing time, a certain number, greater than $n/2$, of the following jobs appear in the LPT order, and jobs following job with the minimum processing time are sequenced in the SPT (shortest processing time first) order. We also present an $O(n^3)$ time dynamic programming algorithm to find the best V-shaped job sequence, in which the jobs on the left of the job with the minimum processing time are sequenced in the LPT order and those on the right of this job are sequenced in the SPT order.

Keywords: Scheduling; Two-machine no-wait flow shop; V-shape property; Dynamic programming.

*Corresponding author

1 Introduction

The studied problem can be formulated as follows. There are n jobs to be sequenced for processing in a two-machine flow shop. Each job j is processed first on machine 1 during p_j time units and then it instantly transfers to machine 2 where it is processed during the same time p_j. Because of the instant job transfer between the machines, this flow shop is called *no-wait*, and because of the same processing time on both machines, it is called *proportional*. No machine can process more than one job at a time. Given a job sequence, let C_j denote the completion time of job j. The objective is to minimize total absolute deviation of job completion times (TADC), which is equal to $\sum_{1\le i<j\le n} |C_i - C_j|$. A solution to this problem is graphically illustrated in Fig. 1. The TADC value is calculated based on formula (1) in the

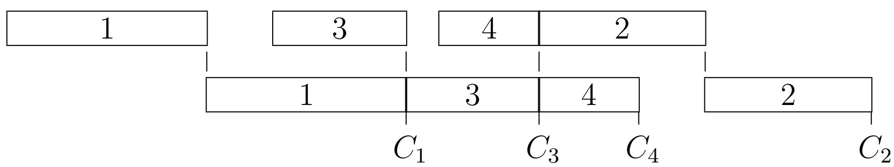

Figure 1: An example solution. TADC=$3(C_3 - C_1) + 4(C_4 - C_3) + 3(C_2 - C_4)$.

next section. Following the traditional scheduling notation, we denote the studied problem as $F2|no-wait, p_{ij} = p_j|TADC$.

The machine environment of a no-wait flow shop has been a popular topic in scheduling, see, e.g., Hall and Sriskandarajah [11] and Emmons and Vairaktarakis [7]. Among the numerous applications, steel manufacturing, plastic molding and silverware industries are important examples [7]. Equal job processing times on both machines often happen, for example, in galvanic and painting operations. The scheduling measure of minimum TADC has been introduced by Kanet [15]. The applicability of this measure is reflected in his statement: "This type of problem has applications in any service or manufacturing setting whenever it is deemed desirable to provide jobs (customers) the same treatment; i.e., each customer spends approximately the same time as every other customer." Kanet developed a polynomial time algorithm for the problem of minimizing TADC on a single machine, based on matching job processing times to job positions. TADC is an example of a *non-regular* objective function. A function $F(C_1, \ldots, C_n)$ is called non-regular if it is increasing or non-monotone in C_j, $j = 1, \ldots, n$. Job earliness/tardiness and completion time variance are other examples of non-regular objective functions. Optimal solutions for many problems with the earliness/tardiness objective and a common due date are *V-shaped* such that the earlier jobs appear in the non-increasing order of the processing times (LPT order), and the remaining jobs appear in the non-decreasing order of the processing times (SPT order). Existence of a V-shaped optimal schedule is an important property of the problem because it limits the number of candidate solutions to $O(2^n)$, and in many cases it leads to an efficient polynomial or pseudo-polynomial dynamic programming algorithm, simple and accurate heuristics and lower bounds. V-shaped schedules have attracted a number of researchers in the last decades, starting with Merten and Muller [19], Eilon and Chowdhury [6], Kanet [15, 16], Hall [9], Raghavachari [20], Bagchi et al. [1], and including Baker and Scudder [2], Biskup and Feldmann [4], Cai [5], Hall et al. [10], Hassin and Shani [13], Janiak et al. [14], Mani et al. [17, 18], Gerstl and Mosheiov [8], to name a few.

To the best of our knowledge, the only paper studying the problem of minimizing TADC in a no-wait proportionate flow shop is that of Ben-Yehoshua et al. [3]. They have shown that the LPT job sequence is optimal for $n \leq 7$. Clearly, LPT is a special case of V-shape, where all the jobs are assigned to the left side of the V. For $n = 8$ and $n = 9$, they proved that the optimal schedule is LPT or a single job (the 4-th largest job) is assigned to the right side of the V. For $n = 10$, they showed that the optimal schedule may contain two jobs on the right side of the V (the 4-th and the 6-th largest jobs).

Problem $F2|no-wait, p_{ij} = p_j|TADC$ can be modified by omitting the no-wait constraint and/or the assumption of equal job processing times on both machines. No result is available in the literature for these modifications. We also failed to establish any essential properties of optimal solutions for these modifications.

In the next section, we prove that there exists an optimal *semi-V-shaped* job sequence for the problem $F2|no-wait, p_{ij} = p_j|TADC$.

Definition 1 *A sequence of jobs* $(j_1, \ldots, j_n)$ *is called semi-V-shaped, if* j_1 *has the largest processing time, jobs* $j_1, \ldots, j_k$, $k > n/2$, *appear in the LPT order, and if* j_r, $r \geq k$, *is the job with the minimum processing time, then jobs* $j_r, j_{r+1}, \ldots, j_n$ *appear in the SPT order. Note that the order of jobs between* j_k *and* j_r *is not specified.*

Unfortunately, we are unable neither to prove that there exists an optimal job sequence that is V-shaped, nor to give an example for which all optimal sequences are not V-shaped. In Section 3, we present an $O(n^3)$ time dynamic programming algorithm to find the best V-shaped job sequence. The last section contains conclusions and suggestions for future research.

2 Semi-V-shape property

Assume that the jobs are re-numbered in the LPT order such that $p_1 \geq \cdots \geq p_n$. We employ the following result of Ben-Yehoshua et al. [3].

Lemma 1 (Lemma 1 in [3]) *There exists an optimal sequence which starts with job 1.*

Let us make few useful observations that facilitate understanding of the further proofs. Consider an optimal job sequence $S = (j_1, j_2, \ldots, j_n)$, in which $j_1 = 1$. Denote $\Delta_{j_{k-1},j_k} = |C_{j_k} - C_{j_{k-1}}|$ and

$$f_k := (n+1-k)(k-1) = -k^2 + k(n+2) - n - 1,$$

$k = 2, \ldots, n$. Note that $\Delta_{j_{k-1},j_k} \geq p_{j_k}$, $k = 1, \ldots, n$. It can be easily verified that the TADC value of S is equal to

$$F(S) = \sum_{k=2}^{n} \Delta_{j_{k-1},j_k} f_k. \tag{1}$$

We call function f_k *semi-concave* in k if $2f_k - f_{k-r_1} - f_{k+r_2} \geq 0$ for any indices k, r_1 and r_2 such that $2 \leq k-r_1 \leq k+r_2 \leq n$ and $(r_1-r_2)(n+2-2k) \geq 0$. The latter relation is equivalent to the statement that either $r_1 \geq r_2$ and $k \leq n/2+1$ or $r_2 \geq r_1$ and $k \geq n/2+1$. An informal verbal definition of a semi-concave function is that it is a discrete opening-down parabola-shaped function of one argument satisfying the property that for any three points such that the middle point is closer to the top point than to the bottom point, the middle point lies above the middle of the interval connecting bottom and top points. Since $2f_k - f_{k-r_1} - f_{k+r_2} = (r_1-r_2)(n+2-2k) + r_1^2 + r_2^2$, the function $f_k = (n+1-k)(k-1) = -k^2 + k(n+2) - n - 1$ is semi-concave. We also observe that maximum of the function f_k is attained at $n/2+1$ if n is even, it is attained at $(n+1)/2$ and $(n+3)/2$ if n is odd, and $f_{n-r} = f_{2+r}$ for $r = 0, 1, \ldots, \lceil n/2 \rceil$. Define $k^* = n/2+1$ if n is even and $k^* = (n+3)/2$ if n is odd. Define j^* as the right-most job with the minimum processing time among jobs in positions $1, \ldots, k^*$ in the optimal sequence S.

Lemma 2 *There exists an optimal sequence which starts with job 1 and jobs preceding and including job j^* in this sequence appear in the LPT order.*

Proof: Assume that j^* is in position r of the optimal job sequence S, i.e., $j^* = j_r$, $r \leq k^*$. Consider sub-sequence $(1, j_2, j_3, \ldots, j_r)$ of this sequence. Its contribution to the objective function is $F_1 = \sum_{i=2}^{r} f_i \Delta_{j_{i-1},j_i} \geq \sum_{i=2}^{r} f_i p_{j_i}$. Since $f_2 \leq f_3 \leq \cdots \leq f_r$, the LPT sequence minimizes $\sum_{i=2}^{r} f_i p_{j_i}$ by the result of Hardy et al. [12]. An observation that $\Delta_{j_{i-1},j_i} = p_{j_i}$, $i = 2, 3, \ldots, r$, if $(j_1, \ldots, j_r)$ is an LPT sequence, and the fact that job j^* is in the same position in the original and the LPT sub-sequences complete the proof. □

Lemma 3 *There exists an optimal sequence which starts with job 1, and jobs in the positions* $1, \ldots, k^*$ *are sequenced in the LPT order.*

Proof: If the job j^* is in the position k^* of the optimal job sequence S, then this lemma holds due to Lemma 2. Assume that j^* is in a position $r \leq k^* - 1$. Let $S = (S_1, S_2)$, where $S_1 = (1, j_2, j_3, \ldots, j_r)$ is the LPT sequence, whose last job is $j_r = j^*$ and $S_2 = (j_{r+1}, j_{r+2}, \ldots, j_n)$. Determine the right-most job j_i in S_1 such that

$$p_{j_{i-1}} \geq p_{j_{r+1}} > p_{j_i}, \quad 2 \leq i \leq r. \tag{2}$$

Since relation $p_1 \geq p_{j_{r+1}} > p_{j_r}$ holds by the definition of the jobs 1 and $j_r = j^*$, job j_i always exists. Move job j_{r+1} to position i immediately after the job j_{i-1}. Further reasoning is illustrated in Fig. 2.

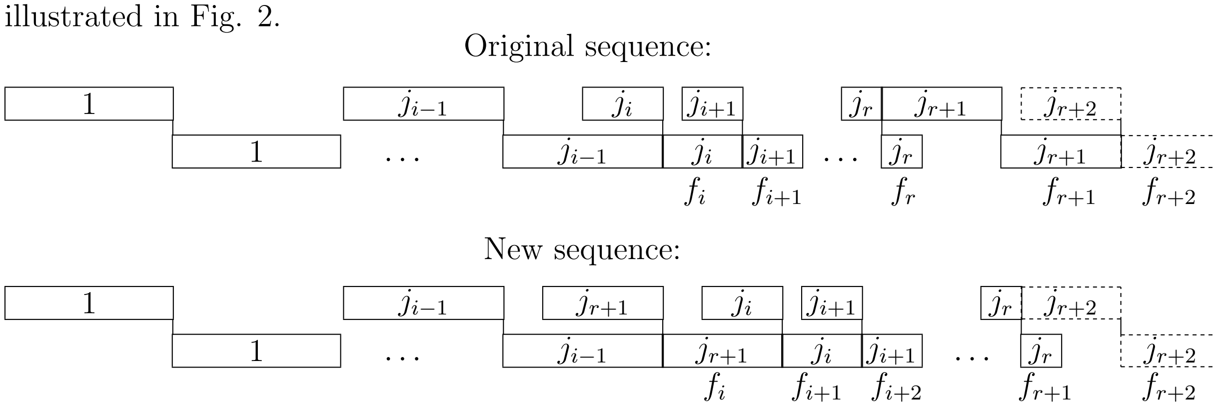

Figure 2: Original and new sequences

The part of the TADC value of the original sequence S, which is affected by this interchange, is $F_1 = \sum_{h=i}^{r} p_{j_h} f_h + (2p_{j_{r+1}} - p_{j_r}) f_{r+1} + \Delta_{j_{r+1},j_{r+2}} f_{r+2}$. The corresponding part of the TADC value of the new sequence is $F_2 = p_{j_{r+1}} f_i + \sum_{h=i}^{r} p_{j_h} f_{h+1} + \Delta_{j_r,j_{r+2}} f_{r+2}$. Let us evaluate the difference $F_1 - F_2$. Since $p_{j_r} < p_{j_{r+1}}$, there are three cases to consider: a) $p_{j_r} < p_{j_{r+1}} \leq p_{j_{r+2}}$, b) $p_{j_r} \leq p_{j_{r+2}} \leq p_{j_{r+1}}$, and c) $p_{j_{r+2}} \leq p_{j_r} < p_{j_{r+1}}$.

In the case a), $\Delta_{j_{r+1},j_{r+2}} = 2p_{j_{r+2}} - p_{j_{r+1}}$, $\Delta_{j_r,j_{r+2}} = 2p_{j_{r+2}} - p_{j_r}$ and

$$F_1 - F_2 = p_{j_{r+1}}(2f_{r+1} - f_i - f_{r+2}) + p_{j_r}(f_{r+2} - f_{r+1}) + \sum_{h=i}^{r} p_{j_h}(f_h - f_{h+1}) =$$

$$p_{j_{r+1}}(2f_{r+1} - f_i - f_{r+2}) - p_{j_r}(2f_{r+1} - f_r - f_{r+2}) + \sum_{h=i}^{r-1} p_{j_h}(f_h - f_{h+1}) =$$

$$(p_{j_{r+1}} - p_{j_r})(2f_{r+1} - f_r - f_{r+2}) + p_{j_{r+1}}(f_r - f_i) + \sum_{h=i}^{r-1} p_{j_h}(f_h - f_{h+1}).$$

Since $i \leq r \leq k^* - 1$, we have $f_h - f_{h+1} \leq 0$, $h = i, i+1, \dots, r$. Furthermore, $p_{j_h} \leq p_{j_i}$, $h = i, i+1, \dots, r$, $\sum_{h=i}^{r-1}(f_h - f_{h+1}) = f_i - f_r \leq 0$, and we continue evaluation

$$F_1 - F_2 \geq (p_{j_{r+1}} - p_{j_r})(2f_{r+1} - f_r - f_{r+2}) + (p_{j_{r+1}} - p_{j_i})(f_r - f_i) \geq 0.$$

The latter relation follows from the semi-concavity of f_k and the definition of j_i. We deduce that the new sequence is optimal. The first $r+1$ jobs in this sequence are arranged in the LPT order.

In the case b), $\Delta_{j_{r+1},j_{r+2}} = p_{j_{r+2}}$, $\Delta_{j_r,j_{r+2}} = 2p_{j_{r+2}} - p_{j_r}$ and

$$F_1 - F_2 = p_{j_{r+1}}(2f_{r+1} - f_i) + p_{j_r}(f_{r+2} - f_{r+1}) - p_{j_{r+2}}f_{r+2} + \sum_{h=i}^{r} p_{j_h}(f_h - f_{h+1}) \geq$$

$$p_{j_{r+1}}(2f_{r+1} - f_i) + p_{j_r}(f_{r+2} - f_{r+1}) - p_{j_{r+1}}f_{r+2} - p_{j_r}(f_{r+1} - f_r) + \sum_{h=i}^{r-1} p_{j_h}(f_h - f_{h+1}) =$$

$$(p_{j_{r+1}} - p_{j_r})(2f_{r+1} - f_r - f_{r+2}) + p_{j_{r+1}}(f_r - f_i) + \sum_{h=i}^{r-1} p_{j_h}(f_h - f_{h+1}).$$

After this, we can continue in the same fashion as in the case a). Again, the new sequence is optimal, and the first $r+1$ jobs in it are arranged in the LPT order.

In the case c), $\Delta_{j_{r+1},j_{r+2}} = p_{j_{r+2}}$, $\Delta_{j_r,j_{r+2}} = p_{j_{r+2}}$ and

$$F_1 - F_2 = p_{j_{r+1}}(2f_{r+1} - f_i) - p_{j_r}f_{r+1} + \sum_{h=i}^{r} p_{j_h}(f_h - f_{h+1}) =$$

$$p_{j_{r+1}}(2f_{r+1} - f_i) - p_{j_r}(2f_{r+1} - f_r) + \sum_{h=i}^{r-1} p_{j_h}(f_h - f_{h+1}) \geq$$

$$(p_{j_{r+1}} - p_{j_r})(2f_{r+1} - f_r) + (p_{j_{r+1}} - p_{j_i})(f_r - f_i) \geq 0.$$

Thus, in all cases, after the move of a job in position $r+1$ to position i, the new sequence remains optimal, and the first $r+1$ jobs in it are arranged in the LPT order. Repetition of this job move argumentation for positions $r+1, r+2, \dots, k^* - 1$ of the job j^* completes the proof. □

We call a sequence which satisfies Lemma 3 as a *half-LPT sequence*. Assume without loss of generality that job n with the minimum processing time is sequenced in such a position k^0 of the optimal half-LPT sequence that $k^0 \geq k^*$. The following lemma holds.

Lemma 4 *There exists an optimal half-LPT sequence such that the last $n - k^0$ jobs are sequenced in the SPT order.*

Proof: Consider sub-sequence $(j_{k^0+1}, j_{k^0+2}, \dots, j_n)$ of the last jobs in the optimal half-LPT sequence. Its contribution to the objective function is $F^{(p)} + F^{(\delta)}$, where $F^{(p)} = \sum_{i=k^0+1}^{n} f_i p_{j_i}$, $F^{(\delta)} = \sum_{i=k^0+1}^{n} f_i \delta_{j_i}$ and $\delta_{j_i} = \Delta_{j_{i-1},j_i} - p_{j_i} \geq 0$, $i = k^0+1, \dots, n$. Let job with the minimum processing time in this sub-sequence be j_r, $k^0 + 1 < r \leq n$. Move this job to position $k^0 + 1$. Observe that the δ value in position $k^0 + 1$ of the old sub-sequence is equal to the sum of the δ values in positions $k^0 + 1$ and $k^0 + 2$ of the new sub-sequence, the δ values in positions $k^0+2, k^0+3, \dots, r-1$ of the old sub-sequence are moved to positions $k^0+3, k^0+4, \dots, r$ of the new sub-sequence, the δ value in position $r + 1$ did not increase, and the δ values in positions $r+2, r+3, \dots, n$ did not change. Therefore, since $f_{k^0+1} \geq f_{k^0+2} \geq \cdots \geq f_n$, the value of $F^{(\delta)}$ did not increase. The value of $F^{(p)}$ did not increase by a similar reason. Therefore, the new sub-sequence is part of an optimal half-LPT sequence. The described job moving argument can be repeated until an optimal half-LPT sequence is obtained in which the last $n - k^0$ jobs are sequenced in the SPT order. □

Lemmas 3 and 4 can be re-formulated as the following theorem.

Theorem 1 *There exists an optimal semi-V-shaped job sequence.*

We do not know if the best semi-V-shaped job sequence can be found in polynomial time. The computational complexity of the problem $F2|no-wait, p_{ij} = p_j|TADC$ remains open.

3 Dynamic programming

In this section, we present a dynamic programming algorithm that finds the best V-shaped job sequence. It is an optimal algorithm if there exists an optimal V-shaped job sequence. However, the latter issue is open. Assume that the jobs are re-numbered in the *SPT order* such that $p_1 \leq \cdots \leq p_n$. Re-call that $k^* = n/2 + 1$ if n is even and $k^* = (n + 3)/2$ if n is odd. Our dynamic programming algorithm constructs partial V-shaped schedules by assigning job 1 to a certain position $k \in \{k^*, k^*+1, \dots, n\}$, and by assigning jobs $2, 3, \dots, n-1$ to the left and to the right of job 1. Finally, job n is scheduled in position 1.

Define $F(l, r, j)$ as the minimum TADC value for a partial schedule, which includes jobs $1, \dots, r - l + 1$ and no other job, the left-most job is in the position l, the right-most job is in the position r, and this right-most job is job j. The initialization is $F(k,k,1) = p_1 f_k$ and $F(k,k,j) = \infty$ for $j = 2,3,\ldots,n-1$, $k = k^*, k^*+1,\ldots,n$. The recursion for $r = k^*, k^*+1,\ldots,n$, $l = r-1, r-2,\ldots,2$, $j = 2,3,\ldots,r-l+1$ is

$$F(l,r,j) = \begin{cases} F(l+1,r,j) + p_{r-l+1} f_l, & \text{if } j \le r-l \\ \min_{1\le q\le r-l}\{F(l,r-1,q) + (2p_{r-l+1} - p_q) f_r\}, & \text{if } j = r-l+1. \end{cases}$$

By the definition of the state variable j, if $j \le r-l$ then the current job $r-l+1$ is assigned to the left-most position l, and if $j = r-l+1$ then this job is assigned to the right-most position r. The best TADC value is equal to $F^* = \min_{1\le j\le n-1}\{F(2,n,j)\}$ and the corresponding best V-shaped job sequence starting with job n can be found by backtracking.

Let us establish the running time of the described algorithm. It is determined by the recursive calculation of the values $F(l,r,j)$. This calculation has two mutually exclusive cases: $j \le r-l$ and $j = r-l+1$. In the case $j \le r-l$, the formula for $F(l,r,j)$ includes a constant number of elementary operations and it is applied for triples (l,r,j) whose number is $O(n^3)$. Thus, this case requires $O(n^3)$ time. If $j = r-l+1$ then the formula for $F(l,r,j)$ includes $O(n)$ elementary operations and it is applied for triples $(l,r,r-l+1)$ whose number is $O(n^2)$. Therefore, this case requires $O(n^3)$ time as well, and the overall running time of the algorithm is $O(n^3)$.

We have conducted computer experiments to see the practical performance of the dynamic programming algorithm. The algorithm was programmed in C and run on a personal computer Mac BookPro 2.4 GHz with a 16GBytes RAM. Job processing times were generated uniformly in the interval [1,100]. For a given number of jobs, 25 random instances were prepared and solved. The average and worst case running times are given in Table 1.

n	Average	Worst
50	11.39	12.35
100	160.88	201.05
150	881.45	1174.03
200	2924.99	4148.92
250	6991.41	7717.09
300	16602.14	21527.99

Table 1: Running time in milliseconds

Fig. 3 shows cubic function $y(n) = 0.0015n^3 - 0.3311n^2 + 20.054n$ built by the Microsoft Excel function Trend based on the data in the first column of Table 1. The correlation coefficient is $R^2 = 0.9966$.

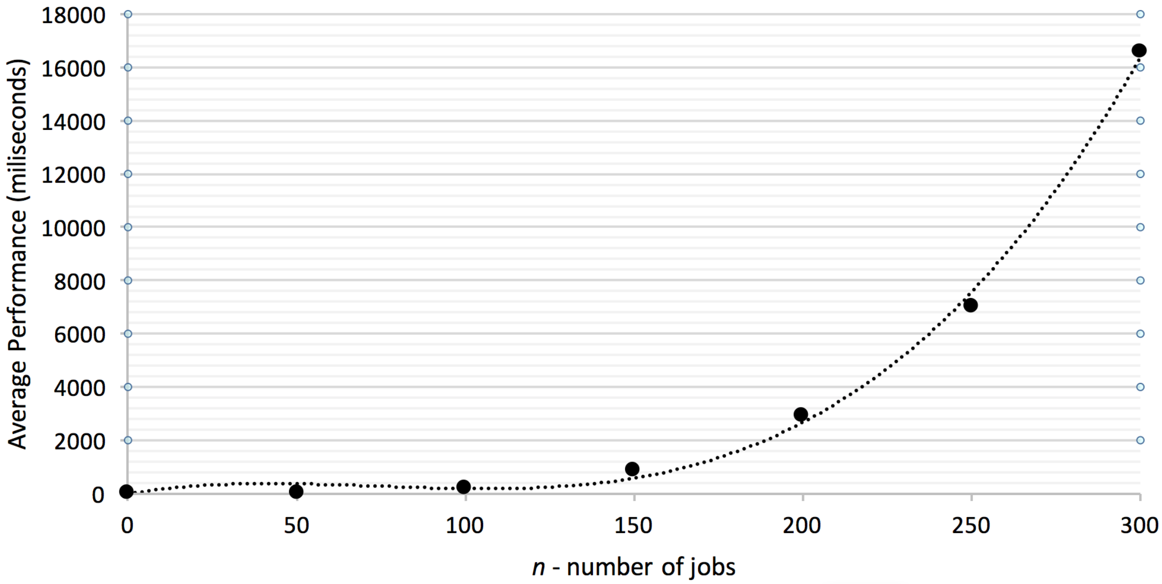

Figure 3: Approximation function $y(n) = 0.0015n^3 - 0.3311n^2 + 20.054n$ of running time

It can be observed that the algorithm finds the best V-shaped sequence within few minutes for instances with hundreds of jobs, which is acceptable for proactive planning in any hypothetical application. As for the reactive planning, a faster approach can be needed.

4 Conclusions and suggestions for future research

We study the problem of minimizing TADC in a two-machine no-wait proportionate flow shop. We prove that an optimal schedule exists which is semi-V-shaped with respect to the job processing times and suggest an $O(n^3)$ time dynamic programming algorithm to find the best V-shaped job sequence. The experimental performance of the algorithm conforms with its theoretical estimate.

The following topics are interesting for future research:

- prove or disprove the conjecture that there exists an optimal V-shaped job sequence;
- find minor modifications of the problem $F2|no-wait, p_{ij} = p_j|TADC$ which make its optimal solution V-shaped;
- develop an algorithm faster than $O(n^3)$ to find the best V-shaped solution;

- considering value of the best V-shaped solution as an approximation for the problem $F2|no-wait, p_{ij} = p_j|TADC$, evaluate its theoretical and experimental deviation from the optimum;
- develop efficient optimal algorithms and heuristics for the problem $F2|no-wait, p_{ij} = p_j|TADC$;
- establish computational complexity of the problem $F2|no-wait, p_{ij} = p_j|TADC$ and its modifications.

References

[1] U. Bagchi, Y. Chang, R. Sullivan (1987) Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties and a common due date, *Naval Research Logistics Quarterly*, 34, 739-751.

[2] K.R. Baker, G.D. Scudder (1990) Sequencing with earliness and tardiness penalties: A review, *Operations Research*, 38(10), 22-36.

[3] Y. Ben-Yehoshua, E. Hariri, G. Mosheiov (2015) A note on minimising total absolute deviation of job completion times on a two-machine no-wait proportionate flowshop, *International Journal of Production Research*, 53(19), 5717-5724.

[4] D. Biskup, M. Feldmann (2005) On scheduling around large restrictive common due windows, *European Journal of Operational Research*, 162, 740-761.

[5] X. Cai (1996) V-shape property for job sequences that minimize the expected completion time variance, *European Journal of Operational Research*, 91, 118-123.

[6] S. Eilon, I.G. Chowdhury (1977) Minimizing waiting time variance in the single machine problem, *Management Science*, 23, 567-575.

[7] H. Emmons, G. Vairaktarakis (2013) Flow shop scheduling. Theoretical results, algorithms, and applications. International Series in Operations Research & Management Science, volume 182.

[8] E. Gerstl, G. Mosheiov (2013) Due-window assignment problems with unit-time jobs, *Applied Mathematics and Computation*, 220, 487-495.

[9] N.G. Hall (1986) Single- and multiple-processor models for minimizing completion time variance, *Naval Research Logistics Quarterly*, 33, 49-54.

[10] N.G. Hall, W. Kubiak, S.P. Sethi (1991) Earliness-tardiness scheduling problems, II: Deviation of completion times about a restrictive common due date, *Operations Research*, 39(5), 847-856.

[11] N.G. Hall, C. Sriskandarajah (1996) A survey of machine scheduling problems with blocking and no-wait in process, *Operations Research*, 44, 510-525.

[12] Hardy, G.H., J.E. Littlewood, G. Polya (1934) Inequalities. Cambridge University Press, Cambridge, England.

[13] R. Hassin, M. Shani (2005) Machine scheduling with earliness, tardiness and non-execution penalties, *Computers & Operations Research*, 32, 683-705.

[14] A. Janiak, W. Janiak, M.Y. Kovalyov, F. Werner (2011) Soft due window assignment and scheduling of unit time jobs on parallel machines, *4OR*, 10(4), 347-360.

[15] J. Kanet (1981) Minimizing variation of flow time in single machine systems, *Management Science*, 27, 1453-1459.

[16] J. Kanet (1981) Minimizing the average deviation of job completion times about a common due date, *Naval Research Logistics Quarterly*, 28(4), 643-651.

[17] V. Mani, P.-C. Chang and S.H. Chen (2010) A parametric analysis for single-machine scheduling with past-sequence-dependent setup times, *International Journal of Innovative Computing*, 6(3A), 1113-1121

[18] V. Mani, P.C. Chang, S.H. Chen (2011) Single-machine scheduling with past-sequence-dependent setup times and learning effects: a parametric analysis, *International Journal of Systems Science*, 42 (12), 2097-2102.

[19] A.G. Merten, M.E. Muller (1972) Variance minimization in single machine sequencing problems, *Management Science*, 18, 518-528.

[20] M. Raghavachari (1986) A V-shape property of optimal schedule of jobs about a common due date, *European Journal of Operational Research*, 23(3), 401-402.