Worst-Case Analysis of Cash Inventory in Single Machine Scheduling

Mehdy Morady Gohareh[a], Ehsan Mansouri[a*], Reza Ghasemy Yaghin [b]

[a]Department of Industrial Engineering, Faculty of Engineering, Arak University, Arak, Iran

[b]Clothing Engineering and Management Group, Department of Textile Engineering,

Amirkabir University of Technology, Tehran, Iran

[*]Correspondence to: Department of Industrial Engineering, Faculty of Engineering, Arak University, Arak, Iran
Email address: e-mansouri@araku.ac.ir

Worst-Case Analysis of Cash Inventory in Single Machine Scheduling

Abstract

This paper studies the inventory of cash as a renewable resource in single machine scheduling problem. The status of cash is incorporated in to the objective function. Average and minimum available cash and maximum and average cash deficiency are contemplated subject to a worst-case scenario that for each job the cost [price] is paid [received] entirely at the start [end] of processing the job. For each objective, either proof of NP-hardness or optimal scheduling rule are provided. Some scenario-based numerical experiments are also carried out which reveal and emphasize the effect of financial assumptions of this research in single machine scheduling.

Keywords: single machine scheduling; worst-case cash management; renewable resource; cash deficiency; complexity analysis;

1. *Introduction and literature review*

Single machine scheduling problem is of paramount importance in practice and theory. Due to that, single machine models have been thoroughly analyzed under all kinds of conditions and with many different objective functions. Pinedo and Blazewicz et al. have provided a complete review of these models [1,2]. The result is a collection of easily applicable priority dispatching rules that often lead to optimal. Earliest Due Date (EDD) first [3] and Shortest Processing Time (SPT) first [4] are examples of these rules amongst others.

However, the issue that has rarely been addressed in the literature of scheduling, including single machine, is the availability of money (i.e. liquidity). Naturally, processing a job on a machine involves paying the cost and receiving the price. The conventional rules of single machine scheduling do not try to manage the inflow and outflow of cash. Consequently, they do not guarantee the availability of cash throughout scheduling.

In financial engineering, the concept of matching payments and incomes through time is categorized under wider topics such as asset-liability managements or cash management. Assets are cash incomes and liabilities are cash outgoes. The same terminology is adapted in this paper. In the literature, there exist some rather distinct lines of research pertinent to the topic of this paper. Scheduling in the presence of renewable and non-renewable resources, financial constraints and jobs with deteriorating values are somehow relevant to this paper.

In this research, money or cash is a renewable resource. However, some authors have studied non-renewable resources in scheduling [5–12]. Examples of a non-renewable resource include raw materials, energy, or even money. This line of research has been getting an increasing attention in recent years. Nonetheless, and to the best of our knowledge, the issue of resource management has rarely been addressed in the objective function and most authors have focused on classic time-based objectives. The only exception is Yedidsion et al. in which a bi-criteria single machine scheduling problem with controllable resource-dependent processing times was considered [12]. One of the objectives was from the traditional class but the other was the total resource consumption.

Some authors have explicitly considered renewable resource management in scheduling. Briskorn et al. studied the complexity of a single machine scheduling where, jobs either add an amount to the inventory of a resource or remove an amount from that inventory [13]. The authors assumed that the aforementioned inventory could not be negative. Hence, if the needed amount of resource is not present, the job may not be processed. The authors, however, did not incorporate the resource inventory in the objective function and went on to use the traditional time-based objectives. Kellerer et al. presented the stock size problem which is basically the scheduling of jobs (trucks) on a single machine (the stock) so that the jobs are processed within a fixed period of time [14]. Each truck can deliver to [take from] the stock a quantity of a good, hence forming the inflow [outflow] of the good. Unlike Briskorn et al. and similar to this paper, the authors involved the resource inventory into the objective function. The objective was to minimize the maximum required stock size. Figielska considered a preemptive two-stage flow shop scheduling to minimize the makespan in the presence of a renewable resource that could be shared between stages [15]. Singh and Ernst addressed a scheduling problem in which a renewable resource i.e. electricity must be shared among some mines [16]. The problem was modelled as a number of separate single machine scheduling problems. The authors developed a heuristic based on Lagrangian relaxation to minimize the total weighted tardiness.

Explicit financial considerations have also been applied to single machine scheduling. As opposed to our pessimistic and worst-case scenario of cash receiving and incurring, Morady Gohareh et al. considered more general patterns for the flow of money and presented the complexity analysis [17,18]. Xie developed polynomial algorithms for single machine scheduling problem with multiple financial constraints [6]. Some authors have also considered jobs with deteriorating values over time (see for example [19–26]). The objective function in these problems is mostly about total job values. Doe to its practical value, resource-constrained project scheduling in renewable and non-renewable form has also spawned a vast body of literatures [27,28]. Artigues et al. carried out a rather thorough review of the models, algorithms, extensions and applications of this type scheduling [29,30].

As the above literature review reveals, the issue of resource management is rarely introduced in the objective function of scheduling problems. In this paper, a single machine scheduling is considered in which processing a job involves paying the cost and receiving the price. The somehow worst-case scenario is assumed in which the cost should be paid before the processing starts and the price cannot be collected until the processing is finished. This research extends the current literature by introducing the aformentioned worst-case scenario in to the objective function of single machine scheduling. In Section 2 notations and assumptions are put forward. Sections 3 to 7 introduce the new financial single machine models. Four objective functions are defined in these models; (i) minimizing the maximum cash deficiency which reflects the amount of external financing required to accomplish the schedule; (ii) maximizing the minimum available cash which is an index of the level of resistance in financially instable environments; (iii) maximizing the average available cash which is an index of the productivity of the scheduling system, and (iiii) minimizing the average cash deficiency. For each model, either an optimal rule is developed and/or complexity analysis is presented. Section 8 concludes the paper by carrying out a numerical experiment to compare scheduling rules of this research with SPT.

2. Notations and assumptions

Consider a single machine scheduling problem with *n* jobs. The following notations are used in this paper ($j = 1,2,\ldots,n$):

- p_j : processing time of job *j*,
- r_j : release date of job *j*,
- d_j : due date of job *j*,
- w_j : weight or importance factor of job *j*,
- C_j : completion time of job *j*,
- L_j : lateness of job *j* i.e. $L_j = C_j - d_j$,
- E_j : earliness of job *j* i.e. $E_j = \max(-L_j, 0)$,
- ct_j: cost of processing job *j* paid at the beginning of its processing,
- pr_j: price of job *j* received upon finishing its processing ($pr_j > ct_j$),
- t_j: the time that processing job *j* starts,
- $al(t)$: net value of available cash at time *t*,
- al_j *:* net value of available cash during processing job *j*,
- y_j *:* amount of debt (i.e. liability) incurred while processing job *j*,
- aL_j *:* net value of available cash at the end of processing job *j* (i.e. just before processing the immediate succeeding job starts),
- A_j *:* set of jobs that precede job *j* in the schedule,

- al_0 : initial net value of available cash before any processing starts.

$al(t)$ is called the asset-liability function. This function can be positive or negative at each time. $al(t)$ only changes at discrete time horizons i.e. when the processing of a job is finished and the price is collected or when the processing of a job starts and the cost is paid. Hence, during processing a job $al(t)$ does not change. Therefore, although time is assumed to be continuous, the system is discrete-event in nature. A suitable method for analyzing and describing $al(t)$ is to depict available cash versus time. The result may be called the cash-time diagram.

Considering the above notations, the following expressions can be stated for any schedule. It is assumed that job *i* is immediately followed by job *j* and j_1 is the first job in the schedule:

$$al_j = al_i + pr_i - ct_j . \tag{1}$$

$$al_j = aL_i - ct_j . \tag{2}$$

$$al_i = aL_i - pr_i . \tag{3}$$

$$al_{j_1} = al_0 - ct_{j_1} . \tag{4}$$

$$al_i = al_0 + \sum_{o \in A_i} (pr_o - ct_o) - ct_i . \tag{5}$$

$$aL_j = aL_i + pr_j - ct_j . \tag{6}$$

$$aL_{j_1} = al_0 + pr_{j_1} - ct_{j_1} . \tag{7}$$

$$al_i = al(t), \quad t \in (t_i, t_j) . \tag{8}$$

$$y_i = \max(0, -al_i) . \tag{9}$$

The concept of the above notations and expressions is demonstrated in Figure 1. The figure depicts the cash-time diagram for a single machine scheduling with three jobs, namely j_1, j_2 and j_3 and the sequence j_1-j_2-j_3.

Figure 1: Cash-time diagram for a three-job single machine scheduling problem and the sequence j_1-j_2-j_3.

The financial objectives considered in this research are described below:

- Minimizing the maximum liability ($y_{\max}$): maximum liability is defined as:

$$y_{\max} = \max(y_1, y_2, ..., y_n). \tag{10}$$

It measures the maximum amount of financing (i.e. borrowing) required for a schedule to be accomplishable. According to Figure 2, a 200 loan is required.

- Minimizing the average liability ($\bar{y}$): average liability is defined as:

$$\bar{y} = \frac{\sum_{j=1}^{n} p_j y_j}{\sum_{j=1}^{n} p_j}. \tag{11}$$

It is the average amount of borrowing required to accomplish a schedule. For the schedule depicted in Figure 2, $\bar{y}$ = 77.2 (the area under the cash-time curve divided by 125). In many real-world situations, such as when the liability is financed via an interest-bearing loan, it may be more appropriate to minimize the average liability rather than its maximum value.

- Maximizing the average asset-liability ($\overline{al}$): average asset-liability is defined as:

$$\overline{al} = \frac{\int_0^{\sum_{j=1}^{n} p_j} al(t)\,dt}{\sum_{j=1}^{n} p_j} = \frac{\sum_{j=1}^{n} p_j al_j}{\sum_{j=1}^{n} p_j}. \tag{12}$$

It measures the average amount of cash available throughout the scheduling period and could be interpreted as an index of financial productivity. Low values of $\overline{al}$ means that more cash is engaged in the processing, which will cause more opportunity cost in turn. For the schedule of Figure 2, $\overline{al}$ = 49.2. Since $\sum_{j=1}^{n} p_j$ is constant, $\sum_{j=1}^{n} p_j al_j$ can be used instead of $\overline{al}$.

- Maximizing the minimum asset-liability ($al_{\min}$): minimum asset-liability is defined as:

$$al_{\min} = \min(al_1, al_2, ..., al_n). \quad (13)$$

$al_{\min}$ can be seen as a cash safety stock for confronting instability in financial parameters. High value of $al_{\min}$ provides a tool for handling unanticipated situations such as changes in costs or prices.

Figure 2: A typical cash-time diagram.

3. Maximizing average asset-liability

According to the classic display framework of scheduling, $1 \| -\overline{al}$ is the single machine scheduling problem with the objective of maximizing $\overline{al}$.$1 \| -\overline{al}$ gives rise to a very straightforward rule to find the optimal schedule, namely the most profit ratio first (MPRF). According to this rule, jobs are arranged in decreasing order of $\frac{pr_j - ct_j}{p_j}$. $pr_j - ct_j$ is the profit margin of job *j*. Hence, $\frac{pr_j - ct_j}{p_j}$ could be interpreted as its profit ratio (i.e. the profit gained from devoting one unit of time of machine to process job *j*).

Theorem 1. MPRF rule is optimal for $1 \| -\overline{al}$.

Proof. The assertion can be proved by contradiction. Suppose a schedule *Sc*, that is not MPRF, is optimal and $al(t)$ is its asset-liability function. In this schedule, there must be at least two adjacent jobs, say job *j* followed by *k*, such that $\frac{pr_j - ct_j}{p_j} < \frac{pr_k - ct_k}{p_k}$. Assume job *j* starts its processing at t_0. Perform a so-called Adjacent Pairwise Interchange on jobs *j* and *k*. Call the new schedule Sc' and denote it parameters by the sign " $'$ ". Since *Sc* is optimal, $\overline{al}' \leq \overline{al}$ which yields

$\sum_{i=1}^{n} p_i al'_j \leq \sum_{i=1}^{n} p_i al_j$ (Equation 12). For $t \in (0, t_0) \cup \left(t_0 + p_j + p_k, \sum_{i=1}^{n} p_i \right)$ the function $al(t)$ is not affected by the interchange (equations 5 and 8). Hence,

$$al'_j p_j + al'_k p_k \leq al_j p_j + al_k p_k \,. \tag{14}$$

Using Equation 5, one can conclude

$$al_j p_j + al_k p_k = (B - ct_j) p_j + (B - ct_j + pr_j - ct_k) p_k \tag{15}$$

and

$$al'_j p_j + al'_k p_k = (B - ct_k) p_k + (B - ct_k + pr_k - ct_j) p_j \tag{16}$$

Where $B = al_0 + \sum_{i \in A_j} (pr_i - ct_i)$. Substituting equations 15, 16 in equation 14 yields $\frac{pr_k - ct_k}{p_k} \leq \frac{pr_j - ct_j}{p_j}$. This contradicts the initial assumption that $\frac{pr_j - ct_j}{p_j} < \frac{pr_k - ct_k}{p_k}$ and completes the proof.

The objective of $1 || -\overline{al}$ is to keep the average available cash of the system as high as possible. At any time instance, some of the assets of system are engaged in processing the incumbent job and the rest of the assets are available. Hence, it can be stated that maximizing $\overline{al}$ focuses on releasing the cash engaged in inventory (i.e. work in process or WIP) as much as possible. In the realm of classic scheduling, the objective $\sum w_j C_j$ also focuses on minimizing the inventory cost. The weight w_j of job j may be regarded as an importance factor. For $1 || \sum w_j C_j$ Weighted Shortest Processing Time (WSPT) first rule yields the optimal schedule. According to WSPT, jobs are arranged in decreasing order of $\frac{w_j}{p_j}$ [4]. However, if one considers w_j to be $pr_j - ct_j$ WSPT and MPRF yield the same schedule. This approach provides a concrete method to decide on the values of w_j's.

The computation time required to arrange the jobs according to WSPT is the time required to sort the jobs according to the ratio of two parameters (w_j and p_j). A simple sort can be done in $O(n \log (n))$ [1]. The same argument is true about MPRF.

4. Minimizing maximum liability

As mentioned before, the objective y_{max} is important in practice since it involves the minimization of the value that should be borrowed in order to accomplish the schedule. A generalization of this objective is $-al_{min}$, the maximization of minimum available cash. Here, we first devote our attention to $1||-al_{\min}$ and then apply the results on $1||y_{\max}$.

Problem $1||-al_{\min}$ can be solved in polynomial time. To do this, a combination of two simple rules is used; least cost first (LCF) and most price first (MPF). According to LCF, jobs are sequenced in increasing order of ct_j. MPF demands jobs to be arranged in decreasing order of pr_j. As expressed in Theorem 2, LCF/ MPF yields the optimal solution for $1||-al_{\min}$. According to this rule, jobs are scheduled under LCF and ties are broken under MPF. However, let us first express the rationale that leads to the optimal solution.

Lemma 1. At least one of the optimal solutions of $1||-al_{\min}$ satisfies the LCF rule.

Proof. Consider an optimal schedule Sc for $1||-al_{\min}$. If Sc is LCF, the proof is complete. If not, there must be at least two adjacent jobs, say job j followed by job k, such that $ct_j > ct_k$. Perform the Adjacent Pairwise Interchange on jobs j and k and call the new schedule Sc' and denote its parameters by the symbol " $'$ ". It can be stated that

$$al_{\min} = \min(al_i) = \min(\min(al_i \mid i \neq j,k), \min(al_j, al_k)) \tag{17}$$

and

$$al'_{\min} = \min(al'_i) = \min(\min(al'_i \mid i \neq j,k), \min(al'_j, al'_k)). \tag{18}$$

Values of al_i's are not affected by the interchange, where $i = 1,2,\ldots,n$ and $i \neq j,k$. Applying $ct_j > ct_k$ in Equation 5 yields $\min(al'_j, al'_k) \geq \min(al_j, al_k)$. Consequently, equations 17 and 18 yield $al'_{\min} \geq al_{\min}$. Finally, since al_{min} is optimal (i.e. the maximum value), $al'_{\min}$ must also be optimal.

If there are no jobs with identical costs, only one LCF schedule exists which must be optimal according to Lemma 1. Otherwise, it would be intuitively easy to see that any LCF/MPF schedule is at least as good as any LCF schedule when considering maximizing $al_{\min}$. Hence, Theorem 2 is stated without proof.

Theorem 2. LCF/MPF rule is optimal for $1 || -al_{\min}$.

If LCF/MPF yields a positive value for $al_{\min}$, then the optimal value of $y_{\max}$ in $1 || y_{\max}$ is zero. Otherwise, the optimal value of $y_{\max}$ and $al_{\min}$ are the same. Hence, the following corollary can be written:

Corollary 1. LCF/MPF rule is optimal for $1 || y_{\max}$.

5. Minimizing average liability

Equation 11 means that $\sum_{j=1}^{n} p_j y_j = \bar{y} \sum_{j=1}^{n} p_j$. Since $\sum_{j=1}^{n} p_j$ is constant, minimizing $\sum_{j=1}^{n} p_j y_j$ and $\bar{y}$ are equivalent and $1 || \sum p_j y_j$ is the same as $1 || \bar{y}$. In the following lemma, it is demonstrated that $1 || \sum p_j y_j$ is equivalent to $1 | r_j = r | \sum w_j E_j$. In Theorem 4, the NP-hardness of the latter is proved.

Lemma 2. $1 || \sum p_j y_j$ is equivalent to $1 | r_j = r | \sum w_j E_j$.

Let us call $1 || \sum p_j y_j$ as Problem 1. Moreover, Let us define $1 | r_j = r | \sum w_j E_j$ as Problem 2 with the following parameters (denoted with asterisks):

- $w_j^* = p_j$,
- $d_j^* = pr_j$,

- $p_j^* = pr_j - ct_j$,
- $r_j^* = r^* = al_0$.

Notice that both problems have the same sets of jobs but with different parameters. Hence, the set of feasible solutions for both problems is the same. Moreover, for any given solution, Equation 6 yields $C_j^* = aL_j$ which means $C_j^* - d_j^* = aL_j - pr_j$. Therefore, $L_j^* = al_j$ which yields $\max\left(-L_j^*, 0\right) = \max\left(-al_j, 0\right)$. Thus, $E_j^* = y_j$ and consequently $\sum_{j=1}^{n} w_j^* E_j = \sum_{j=1}^{n} p_j y_j$. In short, problems 1 and 2 have identical feasible solutions and objective functions. This shows that problems 1 and 2 are the equivalent.

Theorem 4. $1 \| \bar{y}$ is strongly NP-hard.

Proof. Lemma 2 expresses $1 \| \sum p_j y_j \sim 1 | r_j = r | \sum w_j E_j$ where the sign "~" shows the equivalency. Hence,

$$1 \| \bar{y} \sim 1 \| \sum p_j y_j \sim 1 | r_j = r | \sum w_j E_j . \tag{19}$$

One can see that $1 | r_j = r | \sum w_j E_j$ is the general form of $1 \| \sum w_j E_j$ and solving $1 | r_j = r | \sum w_j E_j$ is at least as hard as $1 \| \sum w_j E_j$. Hence,

$$1 \| \sum w_j E_j \propto 1 | r_j = r | \sum w_j E_j . \tag{20}$$

According to Valente, $1 \| \sum w_j E_j$ in non-delay scheduling can be transformed into the equivalent weighted tardiness problem, which is a well-known strongly NP-hard problem [31]. Consequently, Equation 20 yields that $1 | r_j = r | \sum w_j E_j$ is strongly NP-hard which yields the strong NP-hardness of $1 \| \bar{y}$ according to Equation 19.

6. Maximizing average asset-liability subject to a maximum value for liability

A situations that might be the goal of every industrial manager, is maximizing $\overline{al}$ while the liability is not allowed to be more than a pre-defined value, say y_0. The problem can be displayed as $1|y_j \le y_0|-\overline{al}$. As proved in Section 3, the MPRF rule is optimal for $1||-\overline{al}$ and maximizes $\overline{al}$. However, it does not guarantee that no cash shortage happens. LCF/ MPF delivers a no-cash-shortage schedule but compromises $\overline{al}$. It might seem that a combination of the two rules may be helpful for $1|y_j = 0|-\overline{al}$. However $1|y_j \le y_0|-\overline{al}$ in general is proved to be NP-hard via conversion to a classic scheduling problem, namely $1|r_j|\sum w_j C_j$.

Lemma 3. $1|y_j \le y_0|-\sum p_j al_j$ is equivalent to $1|E_j \le y_0|\sum w_j L_j$.

Proof. Let's define a transformed problem via asterisked parameters so that $w_j^* = -p_j$, $d_j^* = pr_j$, $p_j^* = pr_j - ct_j$ and $r_j^* = r^* = al_0$. For any given solution, Equation 6 yields $C_j^* = aL_j$ which means $C_j^* - d_j^* = aL_j - pr_j$. Therefore, $L_j^* = al_j$ and $\sum w_j^* L_j^* = -\sum p_j al_j$. $L_j^* = al_j$ also means that $\max(-L_j^*, 0) = \max(-al_j, 0)$ which yields $E_j^* = y_j$. Hence, $y_j \le y_0$ is equivalent to $E_j^* \le y_0$ and the proof is complete.

Lemma 4. $1|E_j \le y_0|\sum w_j L_j$ is strongly NP-hard.

Proof. $E_j \le y_0$ means that job *j* should not start it processing sooner than $d_j - p_j - y_0$ which in turn could be interpreted as $r_j = d_j - p_j - y_0$ in non-delay scheduling. Moreover, $\sum w_j L_j = \sum w_j (C_j - d_j) = \sum w_j C_j - \sum w_j d_j$. As $\sum w_j d_j$ is constant, minimizing $\sum w_j L_j$ is equivalent to minimizing $\sum w_j C_j$. Consequently

$$1|E_j \le y_0|\sum w_j L_j \sim 1|r_j|\sum w_j C_j \; . \tag{21}$$

As $1/r_j/\sum w_jC_j$ is known to be strongly NP-hard [32], $1|E_j \le y_0|\sum w_jL_j$ is also strongly NP-hard according to Equation 21 and the proof is complete.

Now, Theorem 5 can be stated regarding the complexity of $1|y_j \le y_0|-\overline{al}$.

Theorem 5. $1|y_j \le y_0|-\overline{al}$ is strongly NP-hard.

Proof. $1|y_j \le y_0|-\overline{al}$ is equivalent to $1|y_j \le y_0|-\sum p_j al_j$ and the rest is straightforward according to lemmas 3 and 4.

7. Numerical experiments for revealing the effect of financial assumptions

SPT is arguably the most well-known rule of scheduling which is focused on minimizing the inventory costs and number of jobs in the workshop. In spite of these economic or financial implications, SPT does not consider the flow of cash introduced in this research. Since due dates are not a part of SPT and the models of this research as well, numerical comparison between SPT, MRPF and LCF/MPF will reveal to some extent the significance and importance of the assumption regarding the flow of cash.

For the aforementioned numerical study, some scenarios regarding three important scheduling characteristics have been considered here: size of scheduling system, financial power or solvency of scheduling system, and profitability of the line of work. These scenarios are described in Table 1. The combination of the scenarios results in 36 test problems.

For each test problem, 150 random instances were generated. Processing times, costs and prices were respectively chosen from *U[5, 20], U[10, 50], and U[1.01× cj,* $pm_j^{\max}$ *]* where *U[a, b]* stands for uniform distribution between *a* and *b* and $pm_j^{\max}$ is defined as the maximum possible value for profit margin of job *j*. Each random instance was solved by MPRF, LCF/MPF and SPT. The rules were embedded in a simulation framework and coded in Microsoft Visual Basic. Experiments were implemented on a computer with Intel Core i5-6500 CPU that works at 3.20GHz with four gigabytes of RAM. The whole simulation framework ended under 20 minutes. Average result regarding 150 instances of each test problem can be found in the appendix (Table A.1). However, Figure 3 is extracted from the results.

Table 1: Detail of scenarios considered for generating test problems

Consider respectively the three objective functions $-\overline{al}$, $-al_{\min}$ and $\sum C_j$. In Figure 3 ΔZ_{ij} is a compromise measure which shows the percent of deviation from optimality that is introduced in objective *i* as a result of optimizing objective *j*. Comparison of panels *a* and *b* with panels *c* and *d* clearly states that applying SPT completely disrupts $-\overline{al}$, $-al_{\min}$ whereas applying MRPF and LCF/MPF has a mild effect on $\sum C_j$. Panel *a* suggests that if SPT is applied, scenarios regarding the financial power of the system and the profitability of the work has a mixed effect on the deviation of $-\overline{al}$ from optimality. Nonetheless, this deviation is under 40%. Moreover, the effect of applying SPT on $-\overline{al}$ seems to be negligible for a financially powerful and solvent system (forth scenario for al_0).

Figure 3: Scenarios of deviation from optimality for $\sum C_j$, $-\overline{al}$ and $-al_{\min}$

According to panel *b,* the magnitude of the deviation from optimality for $-al_{\min}$ is much more if SPT is applied. It even reaches about 1200% under some scenarios. Nonetheless, three observations are apparent in panel *b*. Firstly, $-al_{\min}$ is very adversely sensitive to SPT for a line of work that is profitable (scenario 2 and 3 for $pm_j^{\max}$). Secondly, for the extreme scenarios where the scheduling system is very insolvent or very solvent (scenarios 1 and 4 for al_0), $-al_{\min}$ is not sensitive to SPT. Thirdly, scenario 3 al_0 for seems to maximize the sensitivity of $-al_{\min}$ to SPT. According to panel *c* scenarios of al_0 do not seem to affect $\sum C_j$ when MRPF is used. The effect of profitability of the line of work is also minor and limited to 15%. Finally, panel *d* suggests that if LCF/MPF is used, on average $\sum C_j$ deviated 23% from optimality regardless of scenarios of al_0 and $pm_j^{\max}$.

8. Conclusions

This paper discussed cash management in single machine scheduling. The worst-case scenario was assumed for the flow of cash caused by the processing of jobs. It was assumed that the price [cost] of each job is entirely received [paid] upon the end [start] of its processing. Altogether, five different single machine models were defined and analyzed. The goal was either to develop optimal scheduling rules in order to generally create a balance between payments and revenues over time or to analyze the complexity of the problems. The summary of results is as follows:

- If jobs are scheduled according to decreasing order of profit ratio, the average liquidity (available cash) is maximized, the inventory cost is minimized and the productivity of the system is maximized as well. Moreover, this rule deviates $\sum C_j$ from optimality by 15% at most (based on the test problems considered here).
- If jobs are scheduled according to increasing order of cost and ties are broken according to decreasing order of price, the total amount of borrowing needed to accomplish the schedule is minimized and a deviation no more that 23% from optimality is created for $\sum C_j$ (based on the test problems considered here).
- The above rule also maximizes the minimum available cash, which is a measure of protection against financial instability.
- The problem of minimizing the average value of liabilities (cash deficiency) is strongly NP-hard.
- If maximum cash deficiency is restricted to a predefined value, the problem of maximizing the average liquidity is strongly NP-hard.
- Based on the test problems considered here, if SPT is used to schedule jobs, average available cash deviates up to 40% from optimality and minimum available cash deviates drastically from optimality (up to 1200%).

Future studies for single machine scheduling could be focused on objective functions that involve due dates. Asset-liability management of more complex scheduling problems, such as job shops and flow shops, is of great practical value. Involving time value of money and loans with interest cost are also characteristics of many real world production environments.

References

[1] Pinedo, M. L., "Scheduling - Theory, Algorithms, and Systems", 5th ed.New York: Springer, (2016).

[2] Błażewicz, J., Ecker, K. H., Pesch, E., *et al.*, "Handbook on scheduling: From theory to practice", 2nd ed.Springer, (2019).

[3] Jackson, J. R., "Scheduling a production line to minimize maximum tardiness", "Scheduling a production line to minimize maximum tardiness", , Research Report 43, Management Science Research Project, University of California, Los Angeles, (1955). .

[4] Smith, W. E., "Various optimizers for single- stage production", *Nav. Res. Logist. Q.*, **3**(1- 2), pp. 59–66, (1956). https://doi.org/10.1002/nav.3800030106.

[5] Gafarov, E. R., Lazarev, A. A., and Werner, F., "Single machine scheduling problems with financial resource constraints: Some complexity results and properties", *Math. Soc. Sci.*, **62**(1), pp. 7–13, (2011). https://doi.org/10.1016/j.mathsocsci.2011.04.004.

[6] Xie, J., "Polynomial algorithms for single machine scheduling problems with financial constraints", *Oper. Res. Lett.*, **21**(1), pp. 39–42, (1997). https://doi.org/10.1016/S0167-6377(97)00007-2.

[7] Györgyi, P. and Kis, T., "Approximation schemes for parallel machine scheduling with non-renewable resources", *Eur. J. Oper. Res.*, **258**(1), pp. 113–123, (2017). https://doi.org/10.1016/j.ejor.2016.09.007.

[8] Györgyi, P. and Kis, T., "Reductions between scheduling problems with non-renewable resources and knapsack problems", *Theor. Comput. Sci.*, **565**(1), pp. 63–76, (2015). https://doi.org/10.1016/j.tcs.2014.11.007.

[9] Györgyi, P. and Kis, T., "Approximation schemes for single machine scheduling with non-renewable resource constraints", *J. Sched.*, **17**(2), pp. 135–144, (2014). https://doi.org/10.1007/s10951-013-0346-9.

[10] Györgyi, P., "A PTAS for a resource scheduling problem with arbitrary number of parallel machines", *Oper. Res. Lett.*, **45**(6), pp. 604–609, (2017). https://doi.org/10.1016/j.orl.2017.09.007.

[11] Kis, T., "Approximability of total weighted completion time with resource consuming jobs", *Oper. Res. Lett.*, **43**(6), pp. 595–598, (2015). https://doi.org/10.1016/j.orl.2015.09.004.

[12] Yedidsion, L., Shabtay, D., Korach, E., *et al.*, "A bicriteria approach to minimize number of tardy jobs and resource consumption in scheduling a single machine", *Int. J. Prod. Econ.*, **119**(2), pp. 298–307, (2009). https://doi.org/10.1016/j.ijpe.2009.03.003.

[13] Briskorn, D., Choi, B.-C., Lee, K., *et al.*, "Complexity of single machine scheduling subject to nonnegative inventory constraints", *Eur. J. Oper. Res.*, **207**(2), pp. 605–619, (2010). https://doi.org/10.1016/j.ejor.2010.05.036.

[14] Kellerer, H., Kotov, V., Rendl, F., *et al.*, "The stock size problem", *Oper. Res.*, **46**(3-supplement-3), pp. 1–12, (1998). https://doi.org/10.1287/opre.46.3.S1.

[15] Figielska, E., "A heuristic for scheduling in a two-stage hybrid flowshop with renewable resources shared among the stages", *Eur. J. Oper. Res.*, **236**(2), pp. 433–444, (2014). https://doi.org/10.1016/j.ejor.2013.12.003.

[16] Singh, G. and Ernst, A. T., "Resource constraint scheduling with a fractional shared resource", *Oper. Res. Lett.*, **39**(5), pp. 363–368, (2011). https://doi.org/10.1016/j.orl.2011.06.003.

[17] Gohareh, M. M., Gharneh, N. S., and Yaghin, R. G., "Techniques for cash management in scheduling manufacturing operations", *J. Ind. Eng. Int.*, **13**(2), pp. 265–273, (2017). https://doi.org/10.1007/s40092-016-0171-9.

[18] Morady Gohareh, M., Shams Gharneh, N., and Fatemi Ghomi, S. M. T., "Cash management in single and parallel machine scheduling", *Int. J. Prod. Res.*, **52**(9), pp. 2550–2564, (2014). https://doi.org/10.1080/00207543.2013.853888.

[19] Yang, W.-H., "Scheduling jobs on a single machine to maximize the total revenue of jobs", *Comput. Oper. Res.*, **36**(2), pp. 565–583, (2009). https://doi.org/10.1016/j.cor.2007.10.018.

[20] Szmerekovsky, J. G., "Single machine scheduling under market uncertainty", *Eur. J. Oper.*

Res., **177**(1), pp. 163–175, (2007). https://doi.org/10.1016/j.ejor.2005.09.047.
[21] Voutsinas, T. G. and Pappis, C. P., "Scheduling jobs with values exponentially deteriorating over time", *Int. J. Prod. Econ.*, **79**(3), pp. 163–169, (2002). https://doi.org/10.1016/S0925-5273(02)00110-X.
[22] Kannan, D., Khodaverdi, R., Olfat, L., *et al.*, "Integrated fuzzy multi criteria decision making method and multi-objective programming approach for supplier selection and order allocation in a green supply chain", *J. Clean. Prod.*, **47**pp. 355–367, (2013). https://doi.org/10.1016/j.jclepro.2013.02.010.
[23] Ghaleb, M., Taghipour, S., Sharifi, M., *et al.*, "Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures", *Comput. Ind. Eng.*, **143**p. 106432, (2020). https://doi.org/10.1016/j.cie.2020.106432.
[24] Tang, L., Zhao, X., Liu, J., *et al.*, "Competitive two-agent scheduling with deteriorating jobs on a single parallel-batching machine", *Eur. J. Oper. Res.*, **263**(2), pp. 401–411, (2017). https://doi.org/10.1016/j.ejor.2017.05.019.
[25] Woo, Y.-B., Kim, B. S., and Moon, I., "Column Generation Algorithms for a Single Machine Problem with Deteriorating Jobs and Deterioration Maintenance Activities", *Procedia Manuf.*, **39**pp. 1119–1128, (2019). https://doi.org/10.1016/j.promfg.2020.01.358.
[26] Wang, T., Baldacci, R., Lim, A., *et al.*, "A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine", *Eur. J. Oper. Res.*, **271**(3), pp. 826–838, (2018). https://doi.org/10.1016/j.ejor.2018.05.050.
[27] Asadujjaman, M., Rahman, H. F., Chakrabortty, R. K., *et al.*, "Resource Constrained Project Scheduling and Material Ordering Problem with Discounted Cash Flows", *Comput. Ind. Eng.*, p. 107427, (2021). https://doi.org/10.1016/j.cie.2021.107427.
[28] Cui, L., Liu, X., Lu, S., *et al.*, "A variable neighborhood search approach for the resource-constrained multi-project collaborative scheduling problem", *Appl. Soft Comput.*, p. 107480, (2021). https://doi.org/10.1016/j.asoc.2021.107480.
[29] Artigues, C., Demassey, S., and Neron, E., "Resource-constrained project scheduling: models, algorithms, extensions and applications", John Wiley & Sons, (2013).
[30] Hartmann, S. and Briskorn, D., "An Updated Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem", *Eur. J. Oper. Res.*, (2021). https://doi.org/10.1016/j.ejor.2021.05.004.
[31] Valente, J. M. S., "Local and global dominance conditions for the weighted earliness scheduling problem with no idle time", *Comput. Ind. Eng.*, **51**(4), pp. 765–780, (2006). https://doi.org/10.1016/j.cie.2006.08.019.
[32] Lenstra, J. K., Kan, A. H. G. R., and Brucker, P., "Complexity of machine scheduling problems", *Ann. Discret. Math.*, **1**(1), pp. 343–362, (1977). https://doi.org/10.1016/S0167-5060(08)70743-X.

Mehdy Morady Gohareh is an assistant professor at Department of Industrial Engineering of Arak University and graduated from Amirkabir University of Technology (Tehran Polytechnic). His research specialty lie in the areas computer simulation and production planning. In addition, he has served at many industries as a consultant in the fields of quality, maintenance and business process reengineering.

Ehsan Mansouri is a graduate of Industrial Engineering from Tarbiat Modares University and a faculty member at the Department of Industrial Engineering of Arak University. In addition to academic activities and publishing various articles and books, he has also achieved success in the field of technology and has been awarded the position of top technologist in Markazi province in 2019 and 2021 via the establishment of a knowledge-based company and patenting and commercialization of various products. His research interests include systems analysis, business models, artificial intelligence, and operations research.

R. Ghasemy Yaghin is an assistant professor at Department of Textile Engineering, Amirkabir University of Technology, Tehran, Iran. His research interests lie in the areas of supply chain tactical planning and pricing as well as production management of textile industry. His research is mainly focused on operations management problems of the textile industry that involves operations research (mathematical optimization) and statistical learning methods.

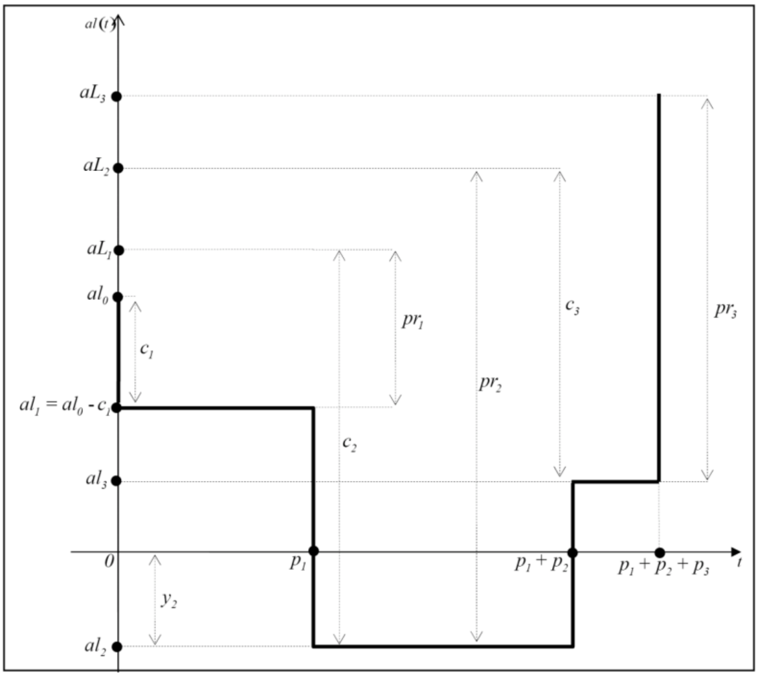

Figure 4: Cash-time diagram for a three-job single machine scheduling problem and the sequence j_1-j_2-j_3.

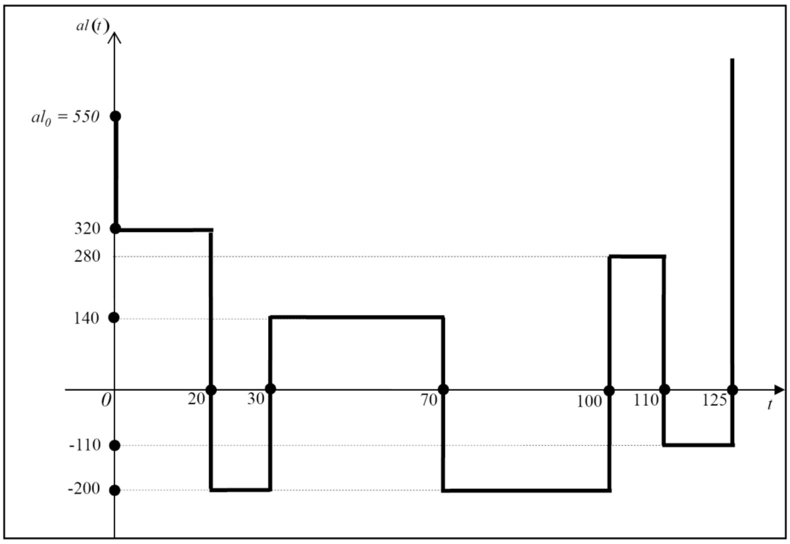

Figure 5: A typical cash-time diagram.

Topic of scenario	Scenario control parameter	Scenario number	control parameter value	Description
size of scheduling system	number of jobs (n)	1	n = 20	a small scheduling system
		2	n = 50	a medium scheduling system
		3	n = 100	a large scheduling system
financial power of scheduling system	Initial available cash (al_0)	1	$al_0 = -\left(\sum_{j=1}^{n} pr_j - c_j\right)$	a scheduling system that will be in debt throughout the entire scheduling horizon
		2	$al_0 = 0$	a scheduling system that struggles with asset-liability issues from the start of scheduling but has a positive balance at the end
		3	$al_0 = 14$	a scheduling system with 10% chance to have the ability to start scheduling with a positive asset-liability balance*
		4	$al_0 = \sum_{j=1}^{n} c_j$	a scheduling system that has no asset-liability problems and can even afford to pay all the costs at the start of scheduling
profitability of the line of work	maximum possible profit margin for each job ($pm_j^{\max}$)	1	$pm_j^{\max} = 1.05c_j$	processing a typical job is this environment generates at most a 5% profit
		2	$pm_j^{\max} = 1.5c_j$	processing a typical job is this environment generates at most a 50% profit (a rather profitable line of work)
		3	$pm_j^{\max} = 3c_j$	processing a typical job is this environment generates up to 200% of profit (a very profitable line of work)

* Since the costs are chosen from U[10, 50], the probability of the cost of a typical job to be less than 14 is 10%.

Table 1: Detail of scenarios considered for generating test problems

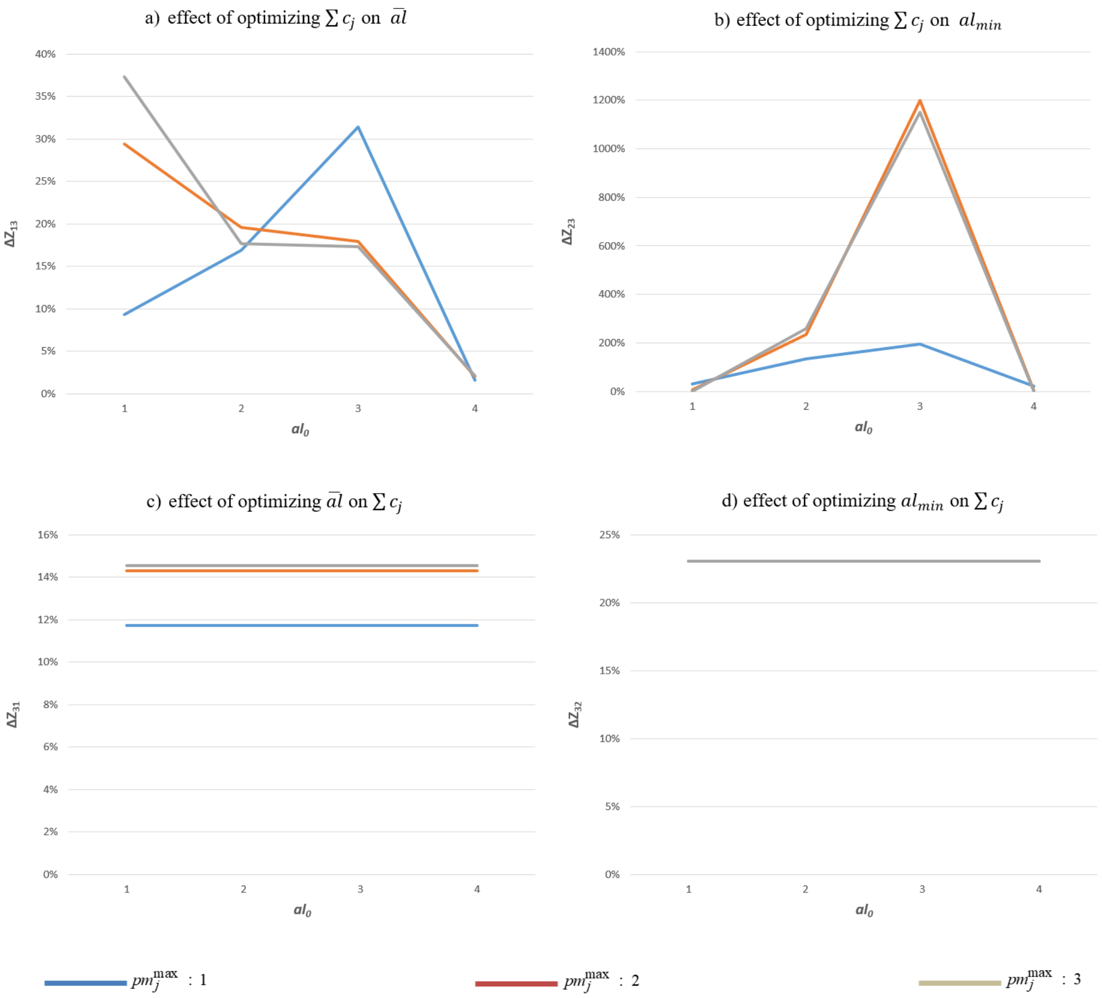

Figure 6: Scenarios of deviation from optimality for $\sum C_j$, $-\overline{al}$ and $-al_{\min}$

Appendix

Scenarios: n	al_0	$pr_j - c_j$	$1//-\overline{al}$ (MPRF): $-\overline{al}$	$-al_{min}$	$\sum C_j$	$1//-al_{\min}$ (LCF/MPF): $-\overline{al}$	$-al_{min}$	$\sum C_j$	$1//\sum C_j$ (SPT): $-\overline{al}$	$-al_{min}$	$\sum C_j$
1	1	1	36	71	2392	41	55	2614	38	69	2151
2	1	1	45	100	14371	58	75	15826	49	101	12832
3	1	1	59	150	56524	86	116	62776	67	149	50496
1	2	1	18	50	2392	23	37	2614	20	49	2151
2	2	1	0	50	14371	13	22	15826	4	49	12832
3	2	1	-31	50	56524	-5	14	62776	-24	49	50496
1	3	1	4	36	2392	9	23	2614	6	35	2151
2	3	1	-14	36	14371	-1	8	15826	-10	35	12832
3	3	1	-45	36	56524	-19	0	62776	-38	35	50496
1	4	1	-72	-16	2392	-67	-26	2614	-71	-16	2151
2	4	1	-226	-138	14371	-213	-166	15826	-222	-141	12832
3	4	1	-485	-359	56524	-458	-391	62776	-477	-363	50496
1	1	2	79	259	2443	127	228	2614	96	250	2151
2	1	2	145	533	14718	267	500	15826	190	534	12832
3	1	2	252	975	57880	505	939	62776	342	969	50496
1	2	2	-75	50	2443	-28	19	2614	-58	49	2151
2	2	2	-240	50	14718	-118	14	15826	-195	49	12832
3	2	2	-520	50	57880	-268	12	62776	-430	48	50496
1	3	2	-89	36	2443	-42	5	2614	-72	35	2151
2	3	2	-254	36	14718	-132	0	15826	-209	35	12832
3	3	2	-534	36	57880	-282	-2	62776	-444	34	50496
1	4	2	-848	-450	2443	-800	-486	2614	-831	-453	2151
2	4	2	-2166	-1439	14718	-2044	-1463	15826	-2121	-1439	12832
3	4	2	-4384	-3267	57880	-4131	-3296	62776	-4294	-3274	50496
1	1	3	221	890	2448	412	859	2614	292	877	2151
2	1	3	476	1979	14750	964	1947	15826	660	1980	12832
3	1	3	894	3724	58012	1903	3687	62776	1261	3717	50496
1	2	3	-388	50	2448	-198	17	2614	-317	49	2151
2	2	3	-1042	50	14750	-555	14	15826	-859	49	12832
3	2	3	-2152	50	58012	-1143	11	62776	-1785	48	50496
1	3	3	-402	36	2448	-212	3	2614	-331	35	2151
2	3	3	-1056	36	14750	-569	0	15826	-873	35	12832
3	3	3	-2166	36	58012	-1157	-3	62776	-1799	34	50496
1	4	3	-3433	-1892	2448	-3243	-1928	2614	-3363	-1909	2151
2	4	3	-8634	-5740	14750	-8147	-5762	15826	-8451	-5737	12832
3	4	3	-17382	-12944	58012	-16373	-12972	62776	-17015	-12962	50496

Rounded average value of objective functions*; Problems (rule) columns as headed above.

*150 random instances were averaged

Table A.1: Details of test problem results