Single Machine Scheduling with Weighted Nonlinear Cost

Wiebke Höhn * Tobias Jacobs †

March 2011

Abstract

We consider the problem of scheduling jobs on a single machine. Given a nonlinear cost function, we aim to compute a schedule minimizing the weighted total cost, where the cost of each job is defined as the cost function value at the job's completion time. Throughout the past decades, great effort has been made to develop fast optimal branch-and-bound algorithms for the case of quadratic costs. The practical efficiency of these methods heavily depends on the utilization of structural properties of optimal schedules such as order constraints, i.e., sufficient conditions for pairs of jobs to appear in a certain order. The first part of this paper substantially enhances and generalizes the known order constraints. We prove a stronger version of the global order conjecture by Mondal and Sen that has remained open since 2000, and we generalize the two main types of local order constraints to a large class of polynomial cost functions.

The new constraints directly give rise to branch-and-bound algorithms with improved efficiency. We take a different route in the second part of this paper and demonstrate the usefulness of order constraints as analytical tools. The WSPT rule, which is well-known to be optimal in the linear cost case, is proven to approximate optimal schedules up to a constant factor that equals the degree of the cost function when the latter is a polynomial with nonnegative coefficients. Furthermore, we give a slightly modified algorithm improving that factor; from 2 to 1.75 in the quadratic case. The previously best known approximation ratio achieved for all these problems is 16.

1 Introduction

We address the problem of scheduling jobs on a single machine in nonlinear cost scenarios. The input consists of n jobs, each having a nonnegative weight and processing time. We refer to the jobs as integers $1,\dots,n$, and denote the processing time and weight of a job $j \in \{1,\dots,n\}$ by p_j and w_j, respectively. The objective is to schedule these jobs nonpreemptively on a single machine such that the weighted sum of completion time costs $\sum_{j=1}^{n} w_j f(C_j)$ is minimized, where C_j is the completion time of job j and $f : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ a fixed nondecreasing function. Following the three-field notation [6], this problem is denoted by $1 \,||\, \sum w_j f(C_j)$.

The model under consideration is a natural generalization of the classic scheduling problem $1 \,||\, \sum w_j C_j$. That linear cost case is solved to optimality by the well-known Smith's rule or WEIGHTED SHORTEST PROCESSING TIME rule (WSPT), which schedules the jobs in nonincreasing order of their w_j/p_j-ratios [12]. Verifying optimality of this method is very easy. Still, in the next paragraphs, we use its discussion to exemplify the terminology of local and global order constraints.

*hoehn@math.tu-berlin.de. Technische Universität Berlin, Germany. Supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Priority Program "Algorithm Engineering" (1307).

†jacobs@nii.ac.jp. National Institute of Informatics, Tokyo, Japan.Supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD).

Whenever in an optimal schedule jobs i and j are processed consecutively, then $w_i/p_i > w_j/p_j$ is a sufficient condition for i being processed before j. This holds true because otherwise interchanging the jobs would reduce the cost of i by $w_i p_j$, and this quantity is larger than the amount $w_j p_i$ the cost of j would be increased by. The fact that job i must be processed before j whenever these jobs are adjacent is expressed by the notation $i \prec_\ell j$. We say that i and j are *locally comparable* when either $i \prec_\ell j$ or $j \prec_\ell i$ holds. A *local order constraint* is an implication of the form $C \Rightarrow i \prec_\ell j$, where C is a proposition that only depends on the characteristics of job i and j. In our example, the local order constraint reads as $w_i/p_i > w_j/p_j \Rightarrow i \prec_\ell j$.

In contrast, a *global order constraint* is an implication of the form $C \Rightarrow i \prec_g j$. Again, the sufficient condition C must only depend on jobs i and j. The relation $i \prec_g j$ expresses that job i is processed before j by any optimal schedule, even—and this is the difference to local order constraints—if there are other jobs scheduled between i and j. The jobs i and j are called *globally comparable* if $i \prec_g j$ or $j \prec_g i$.

In case of our linear cost example, the above local order constraint is actually a global one, i.e., $w_i/p_i > w_j/p_j \Rightarrow i \prec_g j$. This can be shown by induction on the number of jobs scheduled between i and j: If j is processed before i in an optimal schedule, then, by the induction hypothesis, for any job k between j and i it must hold $w_j/p_j \leq w_k/p_k$ and $w_k/p_k \leq w_i/p_i$, so $w_i/p_i > w_j/p_j$ cannot be true. The optimality of the WSPT rule is implied by the fact that any job pair i, j either is globally comparable, or $w_i/p_i = w_j/p_j$ and their processing order does not impact the objective function value if being processed consecutively.

When considering nonlinear cost functions, there are apparently no local or global order constraints which immediately imply an optimal schedule (cf. [9]). In particular, the above condition $w_i/p_i > w_j/p_j$ is not even sufficient for local comparability. Due to nonlinearity, the benefit of interchanging adjacent jobs i and j can be positive or negative, depending on the position of the job pair in the schedule. A prominent nonlinear special case of our problem are quadratic cost functions $1||\sum w_j C_j^2$. This problem has been studied to a great extent with a focus on optimal solution methods; see e. g., [1, 2, 4, 7, 8, 9, 10, 14]. Most of these works utilize branch-and-bound approaches with pruning rules based on order constraints. Order constraints will also be of fundamental importance for our approximability results.

Related work. Schild and Fredman [10] propose a complete enumeration to solve the problem for quadratic cost functions, taking into account a local order constraint to reduce the solution space. Townsend [14] was the first to solve problem $1||\sum w_j C_j^2$ by branch-and-bound. For pruning, he uses an upper bound on the optimal cost related to a local order constraint. Bagga and Kalra [2] add a further node elimination rule, which is a sufficient condition for sets of jobs appearing at the first r positions in an optimal schedule. Global order constraints in this context were first used by Gupta and Sen [7], who proved the sufficient condition $p_j > p_i \wedge w_j p_j > w_i p_i$. Order constraints that additionally depend on the time interval within which the job pair is processed are used in [11] and [4]. Finally, Mondal and Sen conjectured the global order constraint $w_i \geq w_j \wedge w_i/p_i \geq w_j/p_j \Rightarrow i \prec_g j$ and demonstrated in an experimental setup that this constraint would significantly further reduce the runtime required for computing optimal schedules. An overview of the order constraints known so far can be found in the left hand side of Figure 1.

It should be mentioned that in Townsend's paper [14] a generalized problem is also considered. In that model each job j carries two different kinds of weights w_j and w'_j and the cost of a schedule is $\sum(w_j C_j^2 + w'_j C_j)$. For further results see also [7, 10, 11, 13].

In the body of work mentioned so far, practically efficient methods for optimally solving problem $1||\sum w_j C_j^2$ have been given, but the worst-case runtime of all these algorithms is exponential. In fact, up to today it is not known if the problem is NP-hard or if there is hope to find a polynomial time algorithm for it. Polynomial time heuristics, whose solution quality has been evaluated experimentally, have been proposed by Alidaee [1]. Recently, Bansal and Pruhs [3] presented a factor 16 approximation for the more general problem where each job has its individual nondecreasing cost function. The constant factor is achieved by an algorithm that is based on a nontrivial geometric interpretation and uses randomization. A reformulation of the model as a maximization problem admits an approximation factor of $3/2$, as shown by Fisher and Krieger [5].

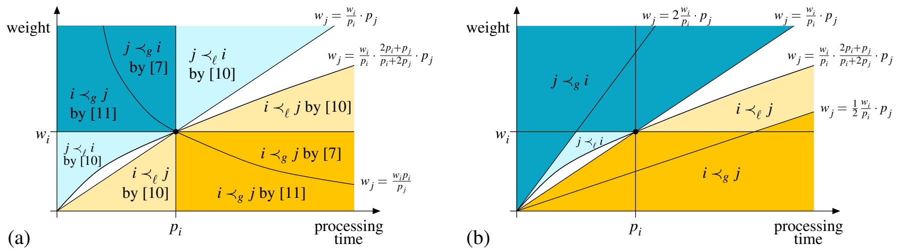


Figure 1: The figures show the known order constraints for the problem $1\,||\,\sum w_j C_j^2$ in comparability maps with respect to a job i. Fig (a) shows the results known from previous works, whereas Fig. (b) integrates the new insights of this paper. In a comparability map, each job is represented by a point in $\mathbb{R}^2$ where the first component is its processing time and the second its weight.

Our contribution. We present a number of new insights into both optimal and approximate solutions to problem $1\,||\,\sum w_j f(C_j)$ for rather general classes of cost functions f. In the following two sections, the landscape of known order constraints is significantly enhanced and generalized. Section 2 mainly generalizes known constraints for quadratic cost functions to polynomials with nonnegative coefficients or even wider function classes. Then, in Section 3, we concentrate on quadratic functions and prove an even stronger version of the global order conjecture by Mondal and Sen [8] that has remained open since 2000. An illustration of the existing and new constraints with respect to $1\,||\,\sum w_j C_j^2$ can be found in Figure 1.

In Section 4 and 5 the insights gained in Section 2 are employed for analyzing two simple heuristic scheduling methods, whose runtimes are dominated by the time required to sort n numbers. The results hold for any polynomial cost function with nonnegative coefficients. First, in Section 4, we focus on the well-known WSPT rule to sort the jobs by the ratio w_i/p_i. It turns out that the approximation ratio guaranteed by this simple and natural heuristic is upper bounded by the degree α of the cost function. Then, in Section 5, we give an alternative heuristic whose main difference to WSPT is an initial rounding step and a certain tie breaking rule. This heuristic guarantees an improved approximation ratio of $\alpha - \frac{\alpha-1}{2^\alpha}$. In the special case of $1\,||\,\sum w_j C_j^2$, the factors by the original and modified version of WSPT are 2 and 1.75, respectively. To the best of our knowledge, the previously best ratio for these problems is achieved by the randomized factor 16 algorithm [3] designed for a more general problem.

Further assumption. Throughout the paper, we assume for simplicity that both the weights and the processing times are strictly positive. This is without loss of generality since one can always compute a schedule for the nonzero jobs and then insert the processing time zero jobs at the beginning and the weight zero jobs at the end of the schedule without increasing or decreasing its total cost.

2 Discussion of order constraints

In this section we address order constraints for various classes of cost functions. In Section 2.1, we discuss constraints for nondecreasing (and convex) cost functions, whereas the focus of Section 2.2 is on a certain class of polynomials. In Section 3 thereafter, we address quadratic cost functions and prove an even stronger version of the conjecture of Mondal and Sen [8].

We start with the observation that the local comparability relation can be equivalently described by a formula which expresses that it is always cheaper to schedule i immediately before j than vice versa. We have $i \prec_\ell j$ if and only if

$$w_i \cdot f(t+p_i) + w_j \cdot f(t+p_i+p_j) < w_j \cdot f(t+p_j) + w_i \cdot f(t+p_j+p_i) \qquad \forall t \geq 0\,. \tag{1}$$

Moreover, we define two abstract classes of order constraints, *gap constraints* and *weight constraints*. With one exception, all constraints shown in this and the following section fall into one of these two categories.

Definition 1 (Gap and weight constraints). *A local order constraint is a local gap constraint if, for some gap $\beta > 1$, $w_i/p_i \geq \beta \cdot w_j/p_j$ implies $i \prec_\ell j$ for jobs i and j. We refer to it also as β-gap constraint. A cost function satisfies the local weight constraint if $w_i \geq w_j \wedge w_i/p_i \geq w_j/p_j$ implies $i \prec_\ell j$ for nonidentical jobs i and j. Global variants of both constraints are defined analogously.*

2.1 General order constraints

We begin with a global order constraint that generalizes a result first used in [11] in the context of quadratic cost functions. Thereafter, we argue that the local weight constraint is observed by any strictly convex cost function. In fact, the latter generalizes the local order constraint for quadratic cost functions, observed by Townsend [14]. The proofs of the following lemmas can be found in the appendix.

Lemma 1. *Consider problem $1||\sum w_j f(C_j)$ for a nondecreasing cost function f. Then for two nonidentical jobs i and j, $w_i \geq w_j$ and $p_i \leq p_j$ implies $i \prec_g j$.*

Lemma 2. *The problem $1||\sum w_j f(C_j)$ with nonlinear strictly convex cost function f observes the local weight constraint.*

2.2 Order constraints for polynomial cost functions

In this section, we consider cost functions of the type $f: t \mapsto a_1 t + a_2 t^2 + \ldots + a_\alpha t^\alpha$, where the coefficients $a_1, \ldots, a_\alpha$ are nonnegative real numbers. We denote this class of functions by $\mathscr{P}^+$. It contains all polynomials with nonnegative coefficients and no constant term or, equivalently, all nonnegative polynomials whose only root is zero.

Theorem 1. *A nonlinear cost function $f \in \mathscr{P}^+$ of degree α observes the local α-gap constraint.*

sketch. One can show that it suffices to prove the theorem for monomials $f: t \mapsto t^\alpha$. We then need to show that given two jobs i, j with $w_i/p_i \geq \alpha \cdot w_j/p_j$ it holds that $i \prec_\ell j$. Resolving the defining condition for $i \prec_\ell j$ from equation (1) for w_i, we get the necessary and sufficient condition

$$w_i > g_t(p_i) \quad \text{for each } t \geq 0\text{, where} \quad g_t : p_i \to \frac{(t+p_i+1)^\alpha - (t+1)^\alpha}{(t+p_i+1)^\alpha - (t+p_i)^\alpha}\ .$$

For any $t \geq 0$, the function g_t can be shown to be concave, to be 0 for $p_i = 0$, and to satisfy $g_t(0) \leq \alpha$. These properties show that each point $w_i \geq \alpha \cdot p_i$ lies above all curves g_t. A detailed proof can be found in the appendix. □

3 Proof of the conjecture by Mondal and Sen

In [8], Mondal and Sen propose a branch-and-bound algorithm for $1||\sum w_j C_j^2$ that uses a pruning rule based on the global weight constraint they conjecture.

Conjecture (Mondal and Sen [8]). Let $1, \ldots, n$ be an instance of $1||\sum w_j C_j^2$. There is an optimal schedule where for any pair i, j of nonidentical[1] jobs satisfying $w_i \geq w_j$ and $w_i/p_i \geq w_j/p_j$ the job i is processed before j.

As the conjecture has not been proven up to now, it has been unclear whether the branch-and-bound algorithm always computes the optimal solution, although a counterexample has never been found. In this section we show that Mondal and Sen were in fact right. The theorem we are going to show is a stronger

[1] Actually, the original statement of the conjecture by Mondal and Sen does not demand that the jobs are nonidentical. However, we believe that the above version is the right way to state the conjecture, because it avoids to claim the existence of a solution where i precedes j and vice versa when the jobs are identical.

version of their conjecture, because in addition to the global weight constraint we also prove that problem $1 \,||\, \sum w_j C_j^2$ observes the global 2-gap constraint.

Theorem 2. *Problem $1 \,||\, \sum w_j C_j^2$ observes the global weight constraint and the global 2-gap constraint: For any pair of jobs i, j, we have $i \prec_g j$ if one of the following sufficient conditions hold:*

(a) $w_i \geq w_j$, $\frac{w_i}{p_i} \geq \frac{w_j}{p_j}$, and i, j are not identical.

(b) $\frac{w_i}{p_i} \geq 2\frac{w_j}{p_j}$.

Proof. Both constraints are proven simultaneously by induction on the number of jobs that are scheduled between a job pair satisfying the respective constraint. More specifically, we show for increasing values of k that there is no optimal schedule where for a pair of jobs i, j satisfying (a) or (b) the job j is processed before i and there are at most k other jobs between them. The base case $k = 0$ are the local variants of both constraints, the local weight constraint (Lemma 2) and the local gap constraint (Theorem 1).

By the induction hypothesis, we assume that the theorem holds when, for some fixed k, there are less than k jobs scheduled between two jobs i and j satisfying property (a) or (b). Based on these assumptions, we first show the induction step for (a) in Lemma 3. This lemma is then used in the induction step for (b) in Lemma 4. The proof of these two lemmas will complete also this proof. □

Lemma 3. *Assume that for some fixed k Theorem 2 holds for any pair of jobs i, j which satisfy (a) or (b) and between which less than k other jobs are scheduled. Then constraint (a) holds also for the case when there are k jobs scheduled between i and j.*

Proof. Assume for contradiction that there is an optimal schedule where job pair i, j satisfies (a), j is processed before i, and there are k jobs in between. We make four simplifying assumptions. Firstly, we assume that $w_j = p_j = 1$. This is without loss of generality, because the optimality of a schedule is invariant to weight and cost scaling. Secondly, we can assume that i is the very last job, since after removing all jobs scheduled after i we obtain an optimal schedule for the remaining jobs. Thirdly, we assume that $1 < p_i = w_i$. The first inequality holds because otherwise the assumption of (a) being satisfied and the jobs not being identical would imply that $p_i = p_j = 1$ and $w_i > w_j = 1$, so the global order constraint (Lemma 1) immediately leads to a contradiction. Hence, we have $w_i \geq p_i > 1$. The reason why $p_i = w_i$ can be assumed without loss of generality is that lowering the weight of i preserves the optimality of the schedule under consideration. In fact, if there was a better schedule for the instance with w_i decreased, a straightforward calculation shows that this schedule would also be less costly for the original instance. Assumption number four is simply a relabeling: we refer to job j as $(1,1)$, to job i as (b,b), and the jobs between them are assumed to be $1, \ldots, k$. To make the notation even shorter we also write $J_{1\ldots k}$ instead of $1, \ldots, k$.

We now define a function Δ measuring the cost increase or decrease when the sub-schedule $J_{1\ldots k}$ is interchanged with a job (x,x) having identical processing time and weight x (the job (x,x) is used as an abstraction of the jobs $(1,1)$ and (a,a)). This function Δ depends on the jobs $J_{1\ldots k}$, the value of x, and the starting time t of (x,x) or $(1,1)$ (depending on which one is processed first). However, we suppress the dependence on everything but x by writing $\Delta(x) :=$

$$\sum_{\ell=1}^{k} w_\ell \left(t + \sum_{m=1}^{\ell} p_m \right)^2 + x \left(t + \sum_{m=1}^{k} p_m + x \right)^2 - x\,(t+x)^2 - \sum_{\ell=1}^{k} w_\ell \left(t + x + \sum_{m=1}^{\ell} p_m \right)^2 .$$

Whenever Δ is negative, it is strictly cheaper to schedule $J_{1\ldots n}$ before (x,x); when Δ is positive it is strictly better to first process (x,x). Δ being zero means that both possibilities have equal cost. Calculating the second derivative results in

$$\frac{d\Delta(x)}{dx^2} = \sum_{\ell=1}^{k} (2p_\ell - w_\ell) .$$

This is the point where we require the induction hypothesis. Property (b) and the characteristics of job $(1,1)$ imply that $w_\ell < 2p_\ell$ for $\ell = 1,\ldots,k$. Therefore, the second derivative of Δ is strictly positive and $\Delta(x)$ is strictly convex in x.

As Δ is strictly convex, it has at most two roots. One of those roots is at $x = 0$, so there is some $x_0 \geq 0$ such that $\Delta(x) < 0$ for $x \in (0,x_0)$, $\Delta(x_0) = 0$, and $\Delta(x)$ is strictly increasing for $x \geq x_0$. Note that possibly $x_0 = 0$ and Δ might be completely nonnegative.

The optimal schedule under consideration contains $(1,1), J_{1\ldots k}, (b,b)$ as a contiguous sub-schedule. If $\Delta(1) < 0$, then one obtains a cheaper schedule by interchanging $(1,1)$ and $J_{1\ldots k}$, a contradiction to optimality. If $\Delta(1) \geq 0$, then $\Delta(b) > \Delta(1)$. In that case consider the alternative schedule obtained by the following operations: (1) interchange $(1,1)$ and $J_{1\ldots k}$, (2) interchange $(1,1)$ and (b,b), (3) interchange $J_{1\ldots k}$ and (b,b). The second operation decreases the cost, due to the local weight constraint of Lemma 2. The first operation does not decrease the cost due to $\Delta(1) \geq 0$, but the increase is more than compensated for by the third operation because $\Delta(b) > \Delta(1)$. All in all we obtain a cheaper schedule, which contradicts the optimality of the original one. □

Lemma 4. *Assume that for some fixed k, Theorem 2 holds for any pair of jobs i, j which satisfy (a) or (b) and between which less than k other jobs are scheduled. Then constraint (b) holds also for the case when there are k jobs scheduled between i and j.*

Proof. Assume for contradiction that there is an optimal schedule where job pair i, j satisfies (b), j is processed before i, and there are k jobs in between. Like in the proof of Lemma 3, we assume w.l.o.g. that $w_j = p_j = 1$, and we rename the k jobs between j and i to $1,\ldots,k$ and use $J_{1,\ldots,k}$ as an abbreviation for that sequence, of course also assuming $i, j > k$. The job i satisfies $w_i \geq 2p_i$. We can also assume that $p_i < 1$ because otherwise the job pair i, j would satisfy property (a), which is impossible due to Lemma 3. Furthermore, is must hold that $w_i \leq 1$, because otherwise we would have $p_i < p_j$ and $w_i \geq w_j$, and the global order constraint stated in Lemma 1 would immediately prove suboptimality. Summarizing, it suffices to analyze the situation $w_j = p_j = 1 > w_i \geq 2p_i$.

In the following notation we ignore all jobs scheduled before and after the jobs $j, J_{1\ldots k}, i$; slightly abusing notation, we refer to the schedule and also to its cost by $[j, J_{1,\ldots,n}, i]$, analogously for other permutations of this job subset.

Claim: $\quad [j,i,J_{1\ldots k}] - [j,J_{1\ldots k},i] \; < \; [i,j,J_{1\ldots k}] - [i,J_{1\ldots k},j]$.

If that claim is true, then the suboptimality of $[i, J_{1\ldots k}, j]$ can be shown by calculating

$$[i,J_{1\ldots k},j] \;<\; [i,j,J_{1\ldots k}] - [j,i,J_{1\ldots k}] + [j,J_{1\ldots k},i] \;<\; [j,J_{1\ldots k},i]\,,$$

where the second inequality is due to the local weight constraint (Lemma 2).

To prove the claim, let T be the point of time when job i completes in the optimal schedule. Let further $p_{1\ldots k}$ be the total processing time of all jobs in $J_{1\ldots k}$. We regard the cost caused by these k jobs as a function F of the completion time of the last job k.

The left hand side of the claimed inequality accounts for the (positive or negative) cost gain of transforming $[j,i,J_{1\ldots k}]$ into $[j,J_{1\ldots k},i]$. By this transformation the jobs $J_{1\ldots k}$ become processed earlier, their cost decreasing by $\text{gain}_{\text{left}}$, and job i becomes more expensive, requiring to pay an additional amount of $\text{loss}_{\text{left}}$, where

$$\text{gain}_{\text{left}} := F(T) - F(T-p_i) \qquad \text{and} \qquad \text{loss}_{\text{left}} := w_i(T^2 - (T-p_{1\ldots k})^2)\,.$$

On the right hand side of the claimed inequality, $[i,j,J_{1\ldots k}]$ is transformed into $[i,J_{1\ldots k},j]$. Here we have

$$\text{gain}_{\text{right}} := F(T) - F(T-1) \qquad \text{and} \qquad \text{loss}_{\text{right}} := T^2 - (T-p_{1\ldots k})^2\,.$$

The loss terms satisfy $\text{loss}_{\text{left}} = w_i \cdot \text{loss}_{\text{right}}$. We are going to show below that $\text{gain}_{\text{left}} < w_i \cdot \text{gain}_{\text{right}}$, which implies the claim as follows:

$$\begin{aligned} 0 &\leq [j,i,J_{1\dots k}] - [j,J_{1\dots k},i] \;=\; \text{gain}_{\text{left}} - \text{loss}_{\text{left}} \\ &< w_i \cdot (\text{gain}_{\text{right}} - \text{loss}_{\text{right}}) \;\leq\; \text{gain}_{\text{right}} - \text{loss}_{\text{right}} \\ &= [i,j,J_{1\dots k}] - [i,J_{1\dots k},j] \; . \end{aligned}$$

The first inequality is due to the optimality of schedule $[j,J_{1\dots k},i]$, the second one follows from $\text{loss}_{\text{left}} = w_i \cdot \text{loss}_{\text{right}}$ and the above proposition that $\text{gain}_{\text{left}} < w_i \cdot \text{gain}_{\text{right}}$, and the third inequality follows from $w_i \leq 1$.

It remains to show the proposition $\text{gain}_{\text{left}} < w_i \cdot \text{gain}_{\text{right}}$, which can be written as

$$F(T) - F(T-p_i) \;<\; w_i \cdot (F(T) - F(T-1)) \; .$$

All we need to know about F is that it is a quadratic function in t that is nonnegative and strictly increasing for $t \geq p_{1\dots k}$. As we are reasoning about the difference of function values, the same reasoning will hold after the function curve has been shifted vertically. We can also shift the function horizontally if we shift the points of evaluation T,$T-p_i$,$T-1$ along with it. For the sake of simpler calculations, we do transformations of that kind, such that we obtain $T-1 = F(T-1) = 0$. Then F can be written as $F(t) = a \cdot t \cdot (t+b)$ with $a > 0$. Furthermore, F cannot have a positive root and therefore $b \geq 0$.

Utilizing $w_i \geq 2p_i$, we can write

$$F(T) - F(T-p_i) \;\leq\; F(1) - F(1-p_i) \;\leq\; a(1+b) - a(1-w_i/2)(1-w_i/2+b) \; ,$$

the right hand side of which can be reformulated to

$$a(1+b) - a(1-w_i + (1-w_i/2)b + w_i^2/4) \; .$$

This quantity is strictly smaller than

$$a(1+b) - a(1-w_i + (1-w_i)b) = w_i a(1+b) = w_i(F(1) - 0) \; ,$$

which is equal to $w(F(T) - F(T-1))$. □

4 Analysis of the WSPT rule

Ordering by nonincreasing w_i/p_i is known to be an optimal strategy in the linear cost case of $1||\sum w_j C_j$ [12], which can be shown by a rather simple interchange argument (cf. Section 1). Here we generalize that finding to higher degree cost functions. More specifically, we show that, when the cost function is from the class $\mathscr{P}^+$ of polynomials with nonnegative coefficients, the WSPT rule guarantees an approximation factor equal to the cost function's degree. This result follows by the combination of Theorem 1 from Section 2 with the following theorem.

Theorem 3. *Consider problem $1||\sum w_j f(C_j)$ for a nondecreasing function f. If observing a β-gap constraint, then the WSPT rule is a factor β-approximation algorithm for this problem, no matter how ties are broken.*

Proof. For the sake of simplicity we assume that the jobs are labeled in the order they are scheduled by the WSPT rule. We refer to that schedule simply by WSPT and also consider some optimal schedule OPT. The proof technique is to analyze the increase in cost that occurs during a transformation from OPT to WSPT. Slightly abusing notation, we refer by WSPT and OPT to also the cost of these schedules.

The transformation proceeds as follows: Starting from OPT, repeatedly interchange job n with its right neighbor until it has become the very last job. Then perform a corresponding series of local interchanges to move job $n-1$ to its position in WSPT, and so on. During this transformation process, each job i first moves to the left by being interchanged with jobs j with $j > i$, then it moves to the right by being interchanged with jobs j with $j < i$. Once i is at its WSPT position, it does not move anymore.

In each single local operation some pair of jobs j,i with $j > i$ is interchanged, such that afterwards it is scheduled in the order i,j. During the operation, the completion time of i decreases by p_j and the completion time of j increases by p_i, therefore the cost of i decreases by some $\Delta^-_{ij} > 0$ while the cost of j increases by some $\Delta^+_{ij} > 0$. We know that $w_i/p_i \geq w_j/p_j$, which implies that

$$\frac{\beta w_i}{p_i} \geq \beta \frac{w_j}{p_j}.$$

In other words, if w_i were by a factor of β larger, then the sufficient condition of the β-gap constraint would hold and the interchange operation would decrease the schedule's cost. Making w_i by a factor of β larger would cause also Δ^-_{ij} to be exactly by a factor of β larger. It follows that that $\beta\Delta^-_{ij} > \Delta^+_{ij}$ for each pair i,j of jobs that are interchanged at some point of the transformation process. Let M be the set of all (unordered) pairs of interchanged jobs. Summing up for all elements of M, we obtain

$$\sum_{\{i,j\}\in M} \Delta^+_{ij} < \beta \sum_{\{i,j\}\in M} \Delta^-_{ij}\,. \tag{2}$$

Furthermore, during the transformation process each job first moves a number of times left and then a number of times right in the schedule. Each time moving left, the job becomes responsible for some value Δ^-_{ij}. As no job can move to a position smaller than 0, the Δ^-_{ij}-values corresponding to any particular job sum up to at most the cost of that job in OPT. Hence, for the sum of all Δ^-_{ij}-values of all jobs we have

$$\sum_{\{i,j\}\in M} \Delta^-_{ij} \leq \mathrm{OPT}. \tag{3}$$

From equation (2) and (3) we obtain the approximation factor as follows:

$$\mathrm{WSPT} - \mathrm{OPT} = \sum_{\{i,j\}\in M} \Delta^+_{ij} - \sum_{\{i,j\}\in M} \Delta^-_{ij} < \qquad (\beta-1)\sum_{\{i,j\}\in M} \Delta^-_{ij} \leq (\beta-1)\mathrm{OPT}\,.$$

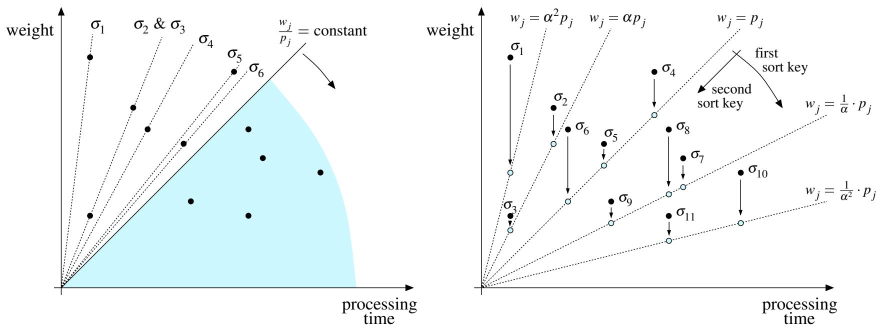
□

Corollary 1. *For problems $1\,||\sum w_j f(C_j)$ with cost function $f \in \mathscr{P}^+$ of degree α, the WSPT rule is an α-approximation algorithm.*

5 An improved approximation algorithm

In the previous section it has been shown that a local β-gap constraint yields an approximation guarantee of β for the WSPT rule. For polynomial cost functions from $\mathscr{P}^+$, β corresponds to the degree α. In this section we give an alternative ordering scheme with an improved approximation guarantee of $\alpha - \frac{\alpha-1}{2^\alpha}$ for cost functions from $\mathscr{P}^+$. For example, in the quadratic cost scenario one obtains a 1.75-approximation. While the analysis of WSPT has been based on the local β-gap constraint, the improved algorithm additionally exploits the local weight constraint stated in Lemma 2.

For the description of the algorithm we interpret jobs i as two-dimensional vectors $(p_i, w_i) \in \mathbb{R}^2$ as in Figure 1 or 2. Let α be the degree of the cost function $f \in \mathscr{P}^+$, and consider in $\mathbb{R}^2$ the family of straight lines $L := \{w = \alpha^z p \mid z \in \mathbb{Z}\}$.



(a) Classic WSPT. (b) Modified WSPT.

Figure 2: The WSPT rule in (a) simply sorts the jobs by their nonincreasing w_j/p_j-ratio and breaks ties arbitrarily. The modified version in (b) firstly rounds down the weights of the jobs such that they lie on straight lines of slope α^k, $k \in \mathbb{Z}$, where α is the degree of the cost function . On each such line, the jobs are sorted in nonincreasing order of their modified weights. By σ_j, we denote the jth job in a schedule computed with the respective rule. E.g., at the positions 4 and 6 the schedules differ.

In the first step, our algorithm modifies all jobs by moving the corresponding vectors vertically downwards to the next line from L; see Figure 2. Technically, this is achieved by defining the new weight of job $i \in \{1,\ldots,n\}$ as $w'_i := p_i \alpha^{\lfloor \log_\alpha (w_i/p_i) \rfloor}$. In the modified instance, any pair of nonidentical jobs is locally comparable, either because they lie on the same straight line of M and so the sufficient condition for the local weight constraint (Lemma 2) holds, or because they lie on different lines of M, so their w_i/p_i-ratio differs by a factor of at least α and the local α-gap constraint (Theorem 1) is applicable. In the second step, our algorithm orders the jobs by nonincreasing w_i/p_i, then by nonincreasing w_i for subsets of jobs with equal w_i/p_i. Since the pairwise local comparability of any two jobs with respect to the modified weights yields a complete order on the jobs, by this procedure, we obtain an optimal schedule for the modified instance.

For analyzing the performance of the computed solution, it is crucial to observe that for the modified weights w'_i we have $w'_i \le w_i < \alpha w'_i$. By a rather standard argument this implies an approximation factor of α, but here we use a slightly more involved analysis for showing the improved ratio.

Theorem 4. *For cost functions $f \in \mathscr{P}^+$, the algorithm given in this section has an approximation guarantee of factor $\alpha - \frac{\alpha-1}{2\alpha}$, where α is the degree of f.*

Proof. For proving the theorem we use a kind of reasoning that is in a way similar to the proof of Theorem 3. We start with an optimal schedule OPT and analyze the impact on the cost when transforming it into the schedule ALG computed by our algorithm. In our notation we do not distinguish between schedules like OPT and their cost.

We divide the set of all jobs into two subsets M^+ and M^-, where M^- consists of all those jobs that finish earlier in ALG than in OPT. When the jobs in M^- are moved to their position in ALG, the costs are reduced by a nonnegative amount Δ^-. Accordingly, moving the jobs in M^+ to their new position increases the cost by a certain amount Δ^+. Both Δ^- and Δ^+ are defined with respect to the original jobs.

The key point of the analysis is to give a better lower bound on $\mathrm{OPT} - \Delta^-$ than 0. To this end we consider a two-machine schedule S where the jobs in M^- are processed on the first machine in the time intervals where they would be processed by ALG, and the jobs from M^+ are processed on the second machine at the time intervals they would be processed by OPT. Clearly, $S = OPT - \Delta^-$. Lemma 5 given below states that the cost of any two-machine schedule is at most by a factor of $(\frac{1}{2})^\alpha$ smaller than the cost of the best one-machine schedule OPT, which implies $\mathrm{OPT} - \Delta^- \ge (\frac{1}{2})^\alpha \mathrm{OPT}$, or

$$\Delta^- \le \left(1 - \frac{1}{2^\alpha}\right) \mathrm{OPT}\,. \tag{4}$$

Let Δ^-_{mod} and Δ^+_{mod} be the corresponding values of Δ^- and Δ^+ with respect to the modified weights. As the modified weights differ from the original ones by factors at most α, it holds that

$$\Delta^-_{\text{mod}} \leq \Delta^- \leq \alpha\Delta^-_{\text{mod}} \qquad \text{and} \qquad \Delta^+_{\text{mod}} \leq \Delta^+ \leq \alpha\Delta^+_{\text{mod}} .$$

ALG is optimal with respect to the modified weights, so $\Delta^+_{\text{mod}} \leq \Delta^-_{\text{mod}}$ and thus

$$\Delta^+ \leq \alpha\Delta^+_{\text{mod}} \leq \alpha\Delta^-_{\text{mod}} \leq \alpha\Delta^- . \tag{5}$$

Using the upper bounds on Δ^+ and Δ^- from equation (4) and (5), we get

$$\begin{aligned} \text{ALG} &= \text{OPT} + \Delta^+ - \Delta^- \leq \text{OPT} + (\alpha-1)\Delta^- \\ &\leq \text{OPT} + (\alpha-1)\left(1 - \frac{1}{2^\alpha}\right)\text{OPT} = \left(\alpha - \frac{\alpha-1}{2^\alpha}\right)\text{OPT} . \end{aligned}$$

□

Lemma 5. *The cost of the optimal 1-machine schedule is at least by a factor of 2^α larger than the cost of any 2-machine schedule for the same set of jobs, where α is the degree of the cost function.* □

Proof. It suffices to prove that any 2-machine schedule S can be transformed into a 1-machine schedule such that the completion time of each job at most doubles, because then the cost of each job increases at most by factor 2^α.

We use induction on the number of jobs n. The base case $n = 1$ is trivial. For the induction step let T be the total processing time of all jobs and assume that job n has the latest completion time in S. Applying the induction hypothesis we can construct a 1-machine schedule for jobs $1, \ldots, n-1$ so that their completion time at most doubles. Then we append job n as the final job to that 1-machine schedule. The new completion time of n is T, whereas in S its completion time has been at least $T/2$. □

6 Conclusion

In this paper we have proven a number of new local and global order constraints for the quadratic cost problem $1||\sum w_j C_j^2$ and generalizations of it. An open question is whether the global order constraints shown for the quadratic case can be extended to a wider class of polynomial cost functions. Another task for future research is to test in an experimental setup to what extend the stronger constraints lead to faster optimal algorithms.

We have further shown that simple and natural heuristics for problem $1||\sum w_j f(C_j)$ lead to schedules whose approximation ratios are small constants when f is a low degree polynomial. We note that it is not clear whether the proven factors are tight for these heuristics. General bounds on the approximability of these problems by polynomial time algorithms also remain an open question, since it is not even known whether or not the computation of optimal solutions is NP-hard. While our approximation factors depend on the cost function's degree, the 16-approximation by Bansal and Pruhs [3] shows that lower bounds are unlikely to be related to this characteristic.

References

[1] B. Alidaee. Numerical methods for single machine scheduling with non-linear cost functions to minimize total cost. *J. Oper. Res. Soc.*, 44:125–132, 1993.

[2] P. C. Bagga and K. R. Kalra. A node elimination procedure for townsend's algorithm for solving the single machine quadratic penalty function scheduling problem. *Manage. Sci.*, 26:633–636, 1980.

[3] N. Bansal and K. Pruhs. The geometry of scheduling. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '10)*, pages 407–414, 2010.

[4] F. Della Croce, W. Szwarc, R. Tadei, P. Baracco, and R. di Tullio. Minimizing the weighted sum of quadratic completion times on a single machine. *Nav. Res. Logist.*, 42(8):1263–1270, 1995.

[5] M. L. Fisher and A. M. Krieger. Analysis of a linearization heuristic for single-machine scheduling to maximize profit. *Math. Program.*, 28:218–225, 1984.

[6] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discrete Math.*, 5:287–326, 1979.

[7] S. K. Gupta and T. Sen. On the single machine scheduling problem with quadratic penalty function of completion times: An improved branching procedure. *Manage. Sci.*, 30:644–647, 1984.

[8] S. A. Mondal and A. K. Sen. An improved precedence rule for single machine sequencing problems with quadratic penalty. *Eur. J. Oper. Res.*, 125:425–428, 2000.

[9] M. H. Rothkopf and S. A. Smith. There are no undiscovered priority index sequencing rules for minimizing total delay costs. *Oper. Res.*, 32:451–456, 1984.

[10] A. Schild and I. J. Fredman. Scheduling tasks with deadlines and non-linear loss functions. *Manage. Sci.*, 9:73–81, 1962.

[11] T. Sen, P. Dileepan, and B. Ruparel. Minimizing a generalized quadratic penalty function of job completion times: an improved branch-and-bound approach. *Eng. Cost. Prod. Econ.*, 18:197–202, 1990.

[12] W. E. Smith. Various optimizers for single-stage production. *Nav. Res. Logist. Q.*, 3(1–2):59–66, 1956.

[13] W. Szwarc. Decomposition in single-machine scheduling. *Ann. Oper. Res.*, 86:271–287, 1998.

[14] W. Townsend. The single machine problem with quadratic penalty function of completion times: a branch-and-bound solution. *Manage. Sci.*, 24:530–534, 1978.

A Proofs from Section 2

Proof of Lemma 1: If such a job j is scheduled before such i in an optimal schedule, the total cost can be decreased by interchanging the two jobs: The cost of i decreases more than the cost of j increases, and the completion times of the jobs between i and j can also only decrease. The mentioned jobs are the only ones whose processing interval is affected by the modification, so the total cost decreases, a contradiction to optimality.

Proof of Lemma 2: Whenever $i \prec_\ell j$ holds for jobs i and j, then it still holds after the processing time of i has been made shorter. Thus, we can assume without loss of generality that $w_i/p_i = w_j/p_j$. By the standard scaling argument we can also assume that $w_j = p_j = 1$ and $1 < w_i = p_i =: a$. Inserting these values into equation (1) gives

$$a \cdot \big(f(t+a) - f(t+a+1)\big) \ < \ f(t+1) - f(t+a+1)\ .$$

We define the function $g : x \mapsto f(-x+t+a+1) - f(t+a+1)$. As the subtracted term is a constant and f is strictly convex, g is strictly concave. Now the above inequality reads as $a \cdot g(1) \ < \ g(a)$, which holds true due to the strict concavity of g.

Proof of Theorem 1: In order to prove the theorem, we first show that it suffices to consider monomials $f : t \mapsto at^\alpha$. Let $f_1, \ldots, f_m$ be cost functions, each observing a local gap constraint with respect to some gap $\beta_i \geq 1$, $i = 1,\ldots,m$. Then, for any two jobs i, j with $w_i/p_i \geq \beta \cdot w_j/p_j$, inequality (1) is satisfied with respect to each cost function $f_1,\ldots,f_m$. Summing up these m inequalities shows that $i \prec_\ell j$ with respect to $f := f_1 + \ldots + f_m$. Thus, we can assume that the cost function is given as $f : t \mapsto t^\alpha$; note that the leading coefficient can be omitted due to the standard scaling argument.

We need to show that given two jobs i, j with $w_i/p_i \geq \alpha \cdot w_j/p_j$ it holds that $i \prec_\ell j$. Again we use the weight and cost scaling argument to assume that $w_j = p_j = 1$ and so $w_i/p_i \geq \alpha$. Furthermore, since f is nonlinear and thus strictly convex, the general local weight constraint in Lemma 2 already shows the claim for $p_i \geq 1$, since this implies $w_i > w_j$. Hence, we can assume here that $p_i < 1$. The defining condition for $i \prec_\ell j$ from equation (1) becomes

$$w_i \cdot (t + p_i)^\alpha + (t + p_i + 1)^\alpha < (t+1)^\alpha + w_i \cdot (t + 1 + p_i)^\alpha \quad \text{for each } t \geq 0 . \tag{6}$$

Resolving for w_i, we get the necessary and sufficient condition

$$w_i > g_t(p_i) \quad \text{for each } t \geq 0 \text{, where} \quad g_t : p_i \to \frac{(t + p_i + 1)^\alpha - (t+1)^\alpha}{(t + p_i + 1)^\alpha - (t + p_i)^\alpha} .$$

If we interpret job i as the point $(p_i, w_i) \in \mathbb{R}^2$, the function curve of g_t represents all realizations of job i where (6) is satisfied with equality for that specific value of t, and all values of (p_i, w_i) lying strictly above all these curves guarantee $i \prec_\ell j$. It is sufficient to show that the straight line $h : p_i \mapsto \alpha p_i$ lies completely above g_t for any t and $p_i < 1$, because any point (p_i, w_i) with $\frac{w_i}{p_i} \geq \alpha$ is located on or above h.

As we assume $p_i < 1$, we know that $g_t(p_i)$ is strictly decreasing in t. Therefore we can further simplify the analysis by setting $t = 0$ without loss of generality. The function g_t reduces to

$$g_0(p_i) = \frac{(p_i + 1)^\alpha - 1}{(p_i + 1)^\alpha - p_i^\alpha} = \frac{\sum_{k=1}^{\alpha} \binom{\alpha}{k} p_i^k}{\sum_{k=0}^{\alpha-1} \binom{\alpha}{k} p_i^k} = \frac{\sum_{k=1}^{\alpha-1} \binom{\alpha}{k} p_i^k}{1 + \sum_{k=1}^{\alpha-1} \binom{\alpha}{k} p_i^k} + \frac{p_i^\alpha}{\sum_{k=0}^{\alpha-1} \binom{\alpha}{k} p_i^k}.$$

As both summands are concave, so is g_o. It also holds that $g_0(0) = h(0) = 0$ and $g_0'(0) = \alpha = h'(p_i)$, so $g_0(p_i) < h(p_i)$ is satisfied for each $p_i > 0$.