On the Performance of Smith's Rule in Single-Machine Scheduling with Nonlinear Cost

Wiebke Höhn[1,⋆] and Tobias Jacobs[2,⋆⋆]

[1] Technische Universität Berlin, Germany
hoehn@math.tu-berlin.de
[2] NEC Laboratories Europe, Heidelberg, Germany
tobias.jacobs@neclab.eu

Abstract. We consider the problem of scheduling jobs on a single machine. Given some continuous cost function, we aim to compute a schedule minimizing the weighted total cost, where the cost of each individual job is determined by the cost function value at the job's completion time. This problem is closely related to scheduling a single machine with nonuniform processing speed. We show that for piecewise linear cost functions it is strongly NP-hard.

The main contribution of this article is a tight analysis of the approximation factor of Smith's rule under any particular convex or concave cost function. More specifically, for these wide classes of cost functions we reduce the task of determining a worst case problem instance to a continuous optimization problem, which can be solved by standard algebraic or numerical methods. For polynomial cost functions with positive coefficients it turns out that the tight approximation ratio can be calculated as the root of a univariate polynomial. To overcome unrealistic worst case instances, we also give tight bounds that are parameterized by the minimum, maximum, and total processing time.

1 Introduction

We address the problem of scheduling jobs on a single machine so as to minimize the weighted sum of completion costs. The input consists of a set of jobs $j = 1, \ldots, n$, where each job j has an individual weight $w_j \geq 0$ and processing time $p_j \geq 0$, and the goal is to find a one-machine schedule minimizing $\sum_{j=1}^{n} w_j f(C_j)$, where C_j denotes the completion time of job j in the schedule. The only assumption we make about the cost function $f : \mathbb{R} \to \mathbb{R}$ at this point is that it is continuous and monotone. In the classic three-field notation [6], the problem we consider reads as $1 \,||\, \sum w_j f(C_j)$. Note that the question of allowing preemption does not play a role here, because the jobs do not have release times and so the possibility of preemption never leads to a cheaper optimal schedule.

⋆ Supported by the German Research Foundation (DFG) as part of the priority programme "SPP 1307: Algorithm Engineering".
⋆⋆ Work supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD).

D. Fernández-Baca (Ed.): LATIN 2012, LNCS 7256, pp. 482–493, 2012.

An important alternative interpretation of problem $1 \,|\,|\, \sum w_j f(C_j)$ is the scenario of linear cost and nonuniform processor speed. Assume that the processor speed at any time t is given by a nonnegative function $g : \mathbb{R} \to \mathbb{R}$, and the processing times (or workloads) p_j of the jobs are given with respect to a unit speed processor. The total workload processed until time t is $G(t) := \int_0^t g(t)dt$. Conversely, if the total workload of job j and all jobs processed before it is t', then the cost of j in the schedule is $G^{-1}(t')$. Therefore, the problem is equivalent to $1 \,|\,|\, \sum w_j G^{-1}(C_j)$. Note that G^{-1} is always monotone, and it is continuous even if g is not. Moreover if g is increasing or decreasing then G^{-1} is convex and concave, respectively. The case of cost function f and processor speed function g is equivalent to problem $1 \,|\,|\, \sum w_j f(G^{-1}(C_j))$.

Related Work. The problem $1 \,|\,|\, \sum w_j f(C_j)$ with nonlinear cost function f has been studied for half a century. For quadratic cost functions there is a long series of articles on branch-and-bound schemes; see e.g., [12,9]. In a companion paper we combine and improve the methods of these articles, and compare them in an extensive computational study [7]. Further references can be found therein.

The problem of minimizing the total weighted flowtime on one or multiple machines with or without preemption is a well studied problem, and efficient approximation schemes are known for many variants [1,3]. In [2], Bansal and Pruhs motivate the usage of monomial cost functions in the context of processor scheduling, where jobs have nonuniform release dates. They show that even in the case of uniform weights there is no $n^{o(1)}$-competitive online algorithm, and they analyze a number of scheduling strategies using resource augmentation.

A more general problem version, where each job has its individual cost function, has recently attracted attention. Bansal and Pruhs have given a geometric interpretation that yields a $O(\log\log nP)$-approximation in the presence of release dates and preemption. In the special case of uniform release dates, their method achieves the constant factor of 16. That factor has recently been improved to $2 + \epsilon$ via a primal-dual approach by Cheung and Shmoys [4].

For $1 \,|\,|\, \sum w_j f(C_j)$ with arbitrary concave f, Stiller and Wiese [11] show that Smith's rule (see below for a definition) guarantees an approximation factor of $(\sqrt{3}+1)/2$. The result is tight in the sense that for a certain cost function f there is a problem instance where this factor is reached by Smith's rule. Epstein et al. provide an approximation algorithm for the problem variant with release dates by generalizing their results on scheduling unreliable machines [5]. Their method generates a schedule which has approximation guarantee $4 + \varepsilon$ for *any* cost function. Both the algorithm by Epstein et al. as well as Smith's rule analyzed by Stiller and Wiese yield schedules that are *universal* in the sense of being generated without knowledge of the cost function.

Our Contribution. The computational complexity of problem $1 \,|\,|\, \sum w_i f(C_i)$ is a long standing open question [9,11]. In Section 4 we give a first result in that direction by showing that for piecewise linear and monotone cost functions the problem is NP-hard in the strong sense. The instances we reduce to can be interpreted as a processor that alternates between two different speeds. Such

Table 1. The first table shows the tight approximation factor of Smith's rule for various cost functions. The factors for polynomials hold under the assumption of non-negative coefficients. In the second table, examples of the parameterized analysis are shown.

cost function	approx. factor
square root	1.07
degree 2 polynomials	1.31
degree 3 polynomials	1.76
degree 4 polynomials	2.31
degree 5 polynomials	2.93
degree 6 polynomials	3.60
degree 10 polynomials	6.58
degree 20 polynomials	15.04
exponential	∞

cost fct.	$p_{\min}$	$p_{\max}$	P	approx. factor
x^2	1	20	500	1.028
x^2	1	20	1000	1.014
x^2	1	20	5000	1.003
x^2	1	100	500	1.136
x^2	1	100	1000	1.071
x^2	1	100	5000	1.015
x^3	1	100	1000	1.149
x^5	1	100	1000	1.296
x^{10}	1	100	1000	1.630

scenarios are likely to occur in practice, e.g., when some extra computational power becomes available at nighttime.

Our main result is a tight analysis of the approximation factor of Smith's rule [10] also known as WSPT (Weighted-Shortest-Processing-Time-First). This well known strategy first computes the *WSPT ratio* $q_j := w_j/p_j$ for each job and then sorts the jobs by descending q_j, which is optimal in the linear cost case. In Section 2, we show that for all convex and all concave cost functions tight bounds for the approximation factor can be obtained as the solution of a continuous optimization problem with at most two degrees of freedom. In the case of cost functions that are polynomials with positive coefficients, it will turn out that the approximation factor can be calculated simply by determining the root of a univariate polynomial. An overview of approximation factors with respect to a number of cost functions are depicted in Table 1, showing that WSPT achieves the best known approximation factor for cost functions that are polynomials of degree up to three. Regarding universal scheduling methods, WSPT provides the best known approximation factor for up to degree six.

The worst case approximation factors are established by extreme instances that consist of one large job and an infinite number of infinitesimally small jobs. In order to analyze the performance of WSPT for realistic instances, we introduce three parameters that restrict the problem instances under consideration. These parameters are the minimum, maximum, and total job length $p_{\min}, p_{\max}$ and P, respectively. In Section 3 we show how to obtain tight bounds for the approximation ratio of Smith's rule under any parameter configuration. Some examples of this analysis are given in Table 1.

2 Tight Analysis of Smith's Rule

In this section we analyze the worst case approximation factor obtained by Smith's rule in the case of any convex or concave cost function. The following simple observation will be used a number of times.

Observation 1. *Problem* $1\,||\sum w_j f(C_j)$ *is invariant to weight scaling, i.e., if I is a problem instance and I' is obtained from I by multiplying all job weights with a constant c, then the cost of any schedule for I' is c times the cost of the same schedule for I.*

We denote by $\mathrm{WSPT}(I)$ the schedule computed for instance I by Smith's rule, and by $\mathrm{OPT}(I)$ an optimal schedule for I. Slightly abusing notation, the cost of these schedules will also be denoted by $\mathrm{WSPT}(I)$ and $\mathrm{OPT}(I)$.

Theorem 1. *Let f be a convex cost function. Then the tight approximation ratio of Smith's rule can be calculated as*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \max\left\{\frac{\int_0^q f(t)dt + p\cdot f(q+p)}{p\cdot f(p) + \int_p^{p+q} f(t)dt} \mid p\geq 0, q\geq 0\right\}\ . \tag{1}$$

When f is concave, the tight ratio is

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \max\left\{\frac{p\cdot f(p) + \int_p^{p+q} f(t)dt}{\int_0^q f(t)dt + p\cdot f(q+p)} \mid p\geq 0, q\geq 0\right\}\ . \tag{2}$$

These equalities hold regardless of the tie breaking strategy used by Smith's rule.

In what follows, we prove a number of lemmas which successively narrow the space of instances we need to consider when searching for a worst case problem instance for Smith's rule. Determining the worst case solution in the final instance space will then be shown to be equivalent to the continuous optimization problem described in the above theorem.

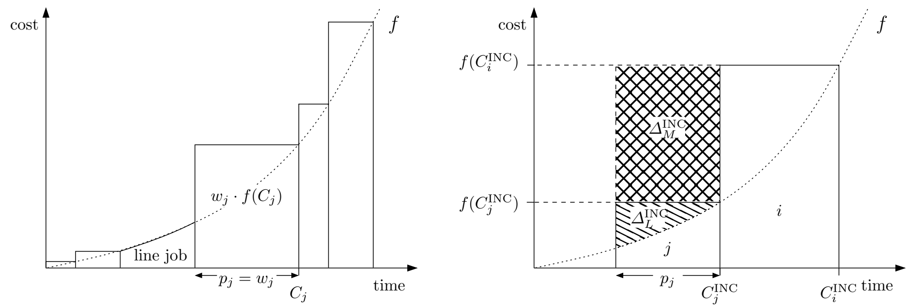
Very similar to the analysis of Stiller and Wiese [11], we first show that it is sufficient to consider instances with constant WSPT ratio, and that a most expensive schedule is obtained by inverting the optimal job order. Thereafter, again as Stiller and Wiese, we restrict to instances with several small jobs and one large job. However, their proof of this property is based on a modification of the cost function which makes it invalid for our problem setting. The remainder of our proof follows a completely different line of argumentation.

The following observation can be shown by continuity considerations, see the full version of this paper for a formal proof or [8] for an explanation of the general principle behind this argumentation.

Observation 2. *If the cost function f is continuous, then the approximation factor* $\sup\{\mathrm{WSPT}(I)/\mathrm{OPT}(I)\}$ *is independent of the tie breaking policy employed by* WSPT.

As a consequence, we can assume that WSPT always breaks ties in the worst possible way. In terms of an adversary model, we can assume that the adversary not only chooses the problem instance, but also the way WSPT breaks ties.

The next lemma shows that we can restrict our attention to problem instances where Smith's ratio is the same for all jobs. By Observation 2, we can further



(a) A job j's cost is represented by a rectangle. Line jobs are a collection of infinitesimally small jobs. Their total cost is given by the area under the graph of f.

(b) Jobs i and j in schedule $\mathrm{INC}(I)$ from Lemma 3. The marked areas represent the change in cost when merging the jobs or when making job j a line job.

Fig. 1. Geometric interpretation of a schedule for instances with $w_j = p_j$ for all jobs j

assume that WSPT schedules the jobs in the worst possible order while OPT schedules them in the best possible order. When the cost function is convex or concave, there is a very simple characterization of these special orders, as shown in Lemma 2. Due to the analogy to Stiller and Wiese [11], the proofs of these two lemmas are omitted in this extended abstract.

Lemma 1. *For the worst case ratio of Smith's rule we can assume that the* WSPT *ratio* q_j *is* 1 *for all jobs* j*. More formally,*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)} \mid w_j = p_j \text{ for each job } j \in I\right\}.$$

Lemma 2. *If the cost function* f *is convex, then*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \sup\left\{\frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \mid w_j = p_j \text{ for each job } j \in I\right\}$$

where $\mathrm{INC}(I)$ *and* $\mathrm{DEC}(I)$ *denotes the cost of the schedule where the jobs in* I *are processed in order of their increasing and decreasing processing time, respectively.*

If f *is concave, then* $\sup\{\mathrm{WSPT}(I)/\mathrm{OPT}(I)\}$ *is obtained analogously with the reciprocal of* $\mathrm{INC}(I)/\mathrm{DEC}(I)$*.*

At this point we introduce a geometric interpretation of our scheduling problem. In this interpretation each job j is represented by a rectangle having width w_j and height $f(C_j)$. As we can restrict our attention to unit ratio jobs, the width equals p_j. Hence, by arranging the rectangles along the x-axis in the order in which the corresponding jobs appear in some schedule S, each rectangle ends at the x-axis at its completion time in S. When drawing the graph of the cost function f into the same graphic, all upper right corners of the rectangles lie on this graph. The total cost of S results as the area of all rectangles. Note that the area below the graph of f, i.e., $\int_0^{\sum w_j} f(x)dx$ is a lower bound on the cost any schedule. An example is depicted in Figure 1(a).

We now introduce the notion of so called *line jobs*. A line job represents an infinite set of infinitesimally small jobs having finite total processing time. More formally, for some fixed p and $\epsilon > 0$ consider the multiset consisting of p/ϵ identical jobs, each having processing time and weight ϵ. Then the line job of length p represents the job multiset obtained for $\epsilon \to 0$. Instead of being represented by a rectangle, a line job corresponds to a strip of width p whose upper boundary is given by the graph of f; see again Figure 1(a).

In the presence of line jobs we still calculate the cost of a schedule by summing up the area of all rectangles and stripes. The correctness of that approach follows from the continuity of the cost function f. A difference between regular jobs and line jobs is that line jobs can be preempted—because they actually consist of many small jobs. However, for problem instances I that consist of both regular jobs and line jobs, observe the schedule $\mathrm{INC}(I)$ first processes the line jobs and then continues with the regular jobs. Symmetrically, $\mathrm{DEC}(I)$ processes the line jobs at the very end of the schedule. So because of Lemma 2 the possibility of preemption does not play any role in our analysis.

Throughout this section, the proofs are given for convex cost functions only, even though everything holds for concave functions as well. However, by syntactically replacing all terms marked with * with their opposite, one obtains the proof for the concave case. Examples of such opposite pairs are convex/concave, best/worst, nonpositive/nonnegative, $\geq$ / $\leq$ or $>$ / $<$.

Lemma 3. *For determining the worst case ratio of Smith's rule when the cost function is concave or convex, we can restrict our attention to instances that consist of one regular job and one line job. More formally,*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \sup\left\{\frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \;\middle|\; \begin{array}{l} I \textit{ consists of one regular job} \\ \textit{and one line job} \end{array}\right\}$$

in the case of convex cost functions. For concave cost functions, the supremum $\sup\{\mathrm{WSPT}(I)/\mathrm{OPT}(I)\}$ *is obtained analogously with the reciprocal ratio.*

Proof. Based on the knowledge we have from Lemma 2 we start with some instance I consisting of regular jobs and show how to transform it into an instance I' having the property stated in the lemma, such that $\mathrm{INC}(I')/\mathrm{DEC}(I')$ $\geq^* \mathrm{INC}(I)/\mathrm{DEC}(I)$ in the case of convex* cost functions.

The transformation of I proceeds as follows: while the instance contains more than one regular job, consider the first regular job j that is processed by schedule $\mathrm{INC}(I)$. Either transform j into a line job with processing time p_j, or replace j and its successor i in $\mathrm{INC}(I)$ by a new job k having processing time and weight $p_j + p_i$. We will show below that at least one of these two possibilities always effectuates that the approximation factor does not improve. We finally obtain a problem instance with one regular job and multiple line jobs. The line jobs can then be merged into a single line job without changing the cost of INC and DEC, which holds because this final operation does not change the total area of stripes in the geometric interpretation.

Let the jobs j, i be defined as in the preceding paragraph. Let I be the instance before the corresponding transformation step, let I_L be the instance obtained by replacing j with a line job, and let I_M be the instance obtained by merging j and i. What we want to show is

$$\frac{\mathrm{INC}(I_L)}{\mathrm{DEC}(I_L)} \geq^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \quad \vee \quad \frac{\mathrm{INC}(I_M)}{\mathrm{DEC}(I_M)} \geq^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \tag{3}$$

in the case of convex* cost functions. Throughout the proof we assume that the schedules INC and DEC are fixed in the sense that $\mathrm{INC}(I_M)$ and $\mathrm{DEC}(I_M)$ is simply obtained by inserting the merged job k at the place where the pair i, j has been in $\mathrm{INC}(I)$ and $\mathrm{DEC}(I)$, respectively. From Lemma 2 we know that the true $\mathrm{INC}(I_M)$ and $\mathrm{DEC}(I_M)$ will make the factor even larger. Let

$$\begin{aligned} \Delta_L^{\mathrm{INC}} &:= \mathrm{INC}(I) - \mathrm{INC}(I_L), \quad & \Delta_M^{\mathrm{INC}} &:= \mathrm{INC}(I_M) - \mathrm{INC}(I) \ , \\ \Delta_L^{\mathrm{DEC}} &:= \mathrm{DEC}(I) - \mathrm{DEC}(I_L), \quad & \Delta_M^{\mathrm{DEC}} &:= \mathrm{DEC}(I_M) - \mathrm{DEC}(I) \ . \end{aligned}$$

Note that the definitions are such that for nondecreasing cost functions all four Δ-values will be nonnegative (although the assumption of nondecreasingness is not necessary for the result to hold); see also Figure 1(b) for a geometric interpretation of the Δ-values. We are going to show that

$$\frac{\Delta_L^{\mathrm{INC}}}{\Delta_L^{\mathrm{DEC}}} \leq^* \frac{\Delta_M^{\mathrm{INC}}}{\Delta_M^{\mathrm{DEC}}} \ . \tag{4}$$

Using (4) and the fact that $A \vee B$ is logically equivalent to $\neg A \Rightarrow B$, one can show (3) for convex* cost functions via the implication chain:

$$\frac{\mathrm{INC}(I_L)}{\mathrm{DEC}(I_L)} = \frac{\mathrm{INC}(I) - \Delta_L^{\mathrm{INC}}}{\mathrm{DEC}(I) - \Delta_L^{\mathrm{DEC}}} <^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \quad \Rightarrow \quad \frac{\Delta_L^{\mathrm{INC}}}{\Delta_L^{\mathrm{DEC}}} >^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \quad \Rightarrow \quad \frac{\Delta_M^{\mathrm{INC}}}{\Delta_M^{\mathrm{DEC}}} >^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)},$$

where Equation 4 is used for the last implication, and this finally implies

$$\frac{\mathrm{INC}(I_M)}{\mathrm{DEC}(I_M)} = \frac{\mathrm{INC}(I) + \Delta_M^{\mathrm{INC}}}{\mathrm{DEC}(I) + \Delta_M^{\mathrm{DEC}}} >^* \frac{\mathrm{INC}(I)}{\mathrm{DEC}(I)} \ .$$

For proving Equation 4 we need explicit formulae for the four Δ-values. Let C_j^{INC} be the completion time of job j in $\mathrm{INC}(I)$, and let $C_j^{\mathrm{DEC}}, C_i^{\mathrm{INC}}, C_i^{\mathrm{DEC}}$ be defined accordingly. Δ_L^{INC} is the difference of the contribution of regular job j to the cost of $\mathrm{INC}(I)$ and the contribution of line job j to the cost of $\mathrm{INC}(I_L)$, so

$$\Delta_L^{\mathrm{INC}} \ = \ p_j \cdot f(C_j^{\mathrm{INC}}) - \int_{C_j^{\mathrm{INC}} - p_j}^{C_j^{\mathrm{INC}}} f(t) dt \ = \ \int_0^{p_j} \big(f(C_j^{\mathrm{INC}}) - f(C_j^{\mathrm{INC}} - p_j + x) \big) dx \ .$$

Δ_L^{DEC} calculates analogously, but with each occurrence of INC replaced by DEC.

Δ_M^{INC} is the difference between the cost contribution of the merged job k to $\mathrm{INC}(I_M)$ and the contributions of i, j to the cost of $\mathrm{INC}(I)$. Also, in $\mathrm{INC}(I_M)$ the completion time of k equals the former completion time of i in $\mathrm{INC}(I)$.

$$\Delta_M^{\mathrm{INC}} = (p_j + p_i) f(C_i^{\mathrm{INC}}) - p_j f(C_j^{\mathrm{INC}}) - p_i f(C_i^{\mathrm{INC}}) = p_j \big(f(C_j^{\mathrm{INC}} + p_i) - f(C_j^{\mathrm{INC}}) \big) \ .$$

For calculating Δ_L^{DEC}, observe that in $\mathrm{DEC}(I_M)$ the merged job k completes at time C_j^{DEC}, so

$$\Delta_M^{\mathrm{DEC}} = (p_i + p_j) f(C_j^{\mathrm{DEC}}) - p_i f(C_i^{\mathrm{DEC}}) - p_j f(C_j^{\mathrm{DEC}}) = p_i \big(f(C_j^{\mathrm{DEC}}) - f(C_j^{\mathrm{DEC}} - p_i) \big) \ .$$

We relate Δ_L^{DEC} and Δ_M^{DEC} as

$$\begin{aligned} \Delta_L^{\mathrm{DEC}} &= \int_0^{p_j} \big(f(C_j^{\mathrm{DEC}}) - f(C_j^{\mathrm{DEC}} - p_j + x)\big)dx \\ &\geq^* \frac{p_j}{2}\big(f(C_j^{\mathrm{DEC}}) - f(C_j^{\mathrm{DEC}} - p_j)\big) \qquad = \frac{p_j \Delta_M^{\mathrm{DEC}}}{2p_i} \end{aligned} \tag{5}$$

in the case of convex* cost functions. The inequality holds because if f is convex* then the expression in the integral is concave*.

For obtaining a similar relation between Δ_L^{INC} and Δ_M^{INC}, observe that

$$\frac{f(C_j^{\mathrm{INC}}) - f(C_j^{\mathrm{INC}} - p_j + x)}{p_j - x} \leq^* \frac{f(C_j^{\mathrm{INC}} + p_i) - f(C_j^{\mathrm{INC}})}{p_i} \qquad \forall x \in [0, p_j) \tag{6}$$

if f is convex*. As the right hand side of (6) is independent of x, it follows that

$$\frac{\Delta_L^{\mathrm{INC}}}{1/2 \cdot p_j^2} = \frac{\int_0^{p_j} \big(f(C_j^{\mathrm{INC}}) - f(C_j^{\mathrm{INC}} - p_j + x)\big)dx}{\int_0^{p_j} (p_j - x)dx} \leq^* \frac{f(C_j^{\mathrm{INC}} + p_i) - f(C_j^{\mathrm{INC}})}{p_i} = \frac{\Delta_M^{\mathrm{INC}}}{p_i p_j} . \tag{7}$$

Equation 4 now follows directly from (5) and (7). □

Theorem 1 now is a direct consequence of Lemma 3. On the right hand side of Equation 1, the parameters p and q respectively correspond to the length of the regular job and the line job in the problem instance I. The expressions in the numerator and denominator are exactly the cost of $\mathrm{INC}(I)$ and $\mathrm{DEC}(I)$, respectively. The correctness of Equation 2 can be verified analogously.

In the remainder of this section we show that for an important class of cost functions Theorem 1 can be further simplified. We have already exploited the fact that problem $1 \,||\, \sum w_j f(C_j)$ is invariant to weight scaling. Similarly, we say that f is invariant to time scaling if there is an function $\phi : \mathbb{R} \to \mathbb{R}$ such that when instance I' is obtained from I by scaling the processing times by some factor c, then $S(I') = \phi(c)S(I)$ for any schedule S. Note that while invariance to weight scaling holds regardless of the cost function, not every cost function is invariant to time scaling, consider e.g. $f : x \mapsto x^2 + x$. Assuming time scalability we can normalize the total processing time to 1, and Theorem 1 yields:

Corollary 1. *Let f be a cost function that is invariant to time scaling and convex or concave. Then the tight bound for the approximation ratio of Smith's rule can be determined as*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \max\left\{\frac{\int_0^{1-p} f(t)dt + p \cdot f(1)}{pf(p) + \int_p^1 f(t)dt} \mid 0 \leq p \leq 1\right\}$$

in the case of convex cost functions, and for concave f, $\sup\{\mathrm{WSPT}(I)/\mathrm{OPT}(I)\}$ *is obtained analogously when maximizing over the reciprocal.*

For monomials $f : t \mapsto t^k$ Corollary 1 reduces the determination of the approximation ratio of Smith's rule to the calculation of the root of a univariate polynomial. Although polynomial cost functions are not invariant to time scaling in general, an important subclass of polynomials can be analyzed as monomials.

Theorem 2. *For cost functions that are polynomials with positive coefficients and degree k, the approximation factor of Smith's Rule is the same as for $t \mapsto t^k$.*

Proof. Let $f = c_1 f_1 + \ldots + c_m f_m$ be the polynomial cost function, where $f_1, \ldots, f_m$ are monomials, and let f_1 be the monomial with the highest degree k. For any schedule S for problem instance I, let $S_i(I)$ denote the cost of S with respect to cost function f_i. If $S^{\mathrm{OPT}} = \mathrm{OPT}(I)$ is an optimal schedule for I, then

$$\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)} = \frac{\mathrm{WSPT}_1(I)+\ldots+\mathrm{WSPT}_m(I)}{S_1^{\mathrm{OPT}}(I)+\ldots+S_m^{\mathrm{OPT}}(I)} \leq \frac{\sum_{i=1}^m \mathrm{WSPT}_i(I)}{\sum_{i=1}^m \mathrm{OPT}_i(I)} \leq \max_{i=1\ldots m} \frac{WSPT_i(I)}{OPT_i(I)} \leq a_1,$$

where $\mathrm{OPT}_i(I)$ is the optimal schedule for I under cost function f_i and a_1 is the tight approximation ratio of Smith's rule with respect to f_1. The last inequality is a consequence of the following lemma which we prove in the full version.

Lemma 4. *Let a_k be the tight approximation factor of Smith's rule for the cost function $f : t \mapsto t^k$. Then a_k is monotone in k for $k \geq 1$ and $\lim_{k\to\infty} a_k = \infty$.*

In order to show that the above inequality is tight, fix I as a problem instance where the worst case approximation factor of WSPT with respect to f_1 is reached. As f_1 is invariant to time scaling, the same approximation factor is reached for each instance $c \cdot I$, which is obtained from I by multiplying all processing times by constant c. As f_1 is the monomial with the largest degree, for $c \to \infty$ the optimal solution $\mathrm{OPT}(c \cdot I)$ with respect to f converges against the optimal solution $\mathrm{OPT}_1(c \cdot I)$ with respect to f_1. As the summand with f_1 also dominates the numerator and denominator of $\mathrm{WSPT}(c \cdot I)/\mathrm{OPT}(c \cdot I)$, we have $\lim_{c\to\infty} \mathrm{WSPT}(c \cdot I)/\mathrm{OPT}(c \cdot I) = \mathrm{WSPT}_1(c \cdot I)/\mathrm{OPT}_1(c \cdot I) = a_1$. □

3 Parameterized Analysis

In this section we refine the analysis of Smith's rule in order to make it more suitable to realistic problem instances. To this end, we introduce parameters $p_{\min}, p_{\max} > 0$, the minimum and maximum job length, and P the total length of all jobs, assuming that $p_{\max}$ and P are multiples of $p_{\min}$. These parameters allow us to ban infinitesimally small and very large jobs as they appear in the unparameterized analysis. In the case of cost functions that are invariant to time scaling, $p_{\min}$ can be assumed w.l.o.g. to be 1. Throughout the analysis, the three parameters will be assumed to be fixed.

Due to this discretization, the tie breaking policy of WSPT is becoming a relevant issue. The proof of Observation 2 exploits the fact that problem instances with ties can be approximated arbitrarily close by instances without ties, but such continuity arguments are not possible in the presence of a $p_{\min}$. In what follows we continue to analyze the version of WSPT having the worst possible tie breaking rule, and remark here that the approximation factors can become smaller if better tie breaking rules are employed.

The analysis is similar to the unparameterized case above. Also here we can show that in worst case instances all jobs have a WSPT ratio of 1, and the largest ratio is obtained when comparing the schedules that sort the jobs in increasing and decreasing order of the job's weight, respectively.

Observation 3. *Lemma 1 and Lemma 2 also hold in the presence of the parameters p_{min}, p_{max}, and P, without any modification of the proofs.*

Lemma 3 in the unparameterized analysis has stated that worst case instances consist of one regular job and one line job. The refined analysis will be similar. Instead of a regular job of length p we will have a sequence of $\lfloor p/p_{\max}\rfloor$ jobs each having a length of $p_{\max}$, plus one length $p \bmod p_{\max}$ job, where p is a multiple of $p_{\min}$ between 0 and P. Instead of a line job we will have $(P-p)/p_{\min}$ jobs each having length $p_{\min}$. So given the parameters $p_{\min}$, $p_{\max}$, P one can determine the tight approximation factor of Smith's rule by finding the value of p maximizing the ratio between INC and DEC. Denote by $\mathrm{INC}(p, p_{\min}, p_{\max}, P)$ and $\mathrm{DEC}(p, p_{\min}, p_{\max}, P)$ the schedule where the jobs of the instance determined by $p, p_{\min}, p_{\max}, P$ are scheduled by increasing and decreasing weight, respectively. The tight approximation factor is given in the next theorem. Due to the similarity to the analysis in Section 2, its proof is omitted here.

Theorem 3. *Given the minimum, maximum and total processing times p_{min}, p_{max} and P, the tight approximation ratio of Smith's rule can be calculated as*

$$\sup\left\{\frac{\mathrm{WSPT}(I)}{\mathrm{OPT}(I)}\right\} = \left\{\frac{\mathrm{INC}(p, p_{min}, p_{max}, P)}{\mathrm{DEC}(p, p_{min}, p_{max}, P)} \mid p = 0, p_{min}, 2p_{min}, \ldots, P\right\}$$

in the case of convex cost functions. If f is concave $\sup\{\mathrm{WSPT}(I)/\mathrm{OPT}(I)\}$ *is obtained analogously with the reciprocal of* INC/DEC.

4 Hardness for Piecewise Linear Cost Functions

In this section we show that problem $1\,||\sum w_j f(C_j)$ is strongly NP-hard in general. The complexity is proven via reduction from strongly NP-complete 3-Partition, and the scheduling instance reduced to has a piecewise linear monotone cost function. In particular, it suffices for NP-hardness that f alternates between two different slopes that can be chosen arbitrarily.

In 3-Partition, one needs to decide whether a given set A of $3\,m$ elements from $\mathbb{N}^+$ with $B/4 < a < B/2$ for all $a \in A$, where $B := \frac{1}{m}\sum_{a\in A} a$, can be partitioned into m disjoint sets $A_1, \ldots, A_m$ with $\sum_{a\in A_i} a = B$ for $i = 1, \ldots, m$.

Theorem 4. *The problem $1\,||\sum w_j f(C_j)$ is strongly NP-hard for piecewise linear monotone cost functions f.*

Proof. Given an instance of 3-Partition, an equivalent scheduling instance is constructed as follows. For each element $a_\ell \in A$, $\ell = 1, \ldots, 3m$, we add a job j_ℓ having processing time $p_\ell = a_\ell$ and weight $w_\ell = a_\ell$. The cost function f is defined to be piecewise linear. It alternates between two different slopes r and s with $r > s \geq 0$ that can be chosen arbitrarily. For each $i \in \mathbb{N}^+$ the slope during time interval $[(i-1)\cdot B, (i-1)\cdot B+1]$ is r, and the slope during $[(i-1)\cdot B+1, i\cdot B]$ is s. The cost threshold is set to

$$\alpha \;:=\; s \cdot \sum_{1\leq k\leq \ell\leq 3m} a_\ell a_k \;+\; \frac{(r-s)Bm(m+1)}{2}\,.$$

The equivalence of the problems is established by showing that any schedule where some new job begins at time $(i-1)\cdot B$ for each $i=1,\ldots,m$ has cost α, and any other schedule has larger cost. This will complete the proof.

As the job lengths are integers, no job ever ends inside a slope r interval. Therefore we can as well assume that the slope is s everywhere, and at each time $(i-1)\cdot B$, $i\in\mathbb{N}^+$ there is a point of discontinuity where the constant $(r-s)$ is added to the cost function. So f can be expressed as $f(t)=s\cdot t+(r-s)\left\lceil\frac{t}{B}\right\rceil$. Let $f=f_1+f_2$ with $f_1: t\mapsto st$ and $f_2: t\mapsto (r-s)\lceil t/B\rceil$. As $w_\ell=p_\ell$, the cost of a schedule σ w.r.t. f_1 is

$$\sum_{\ell=1}^{3m} w_{\sigma(\ell)}\cdot s\sum_{k=1}^{\ell} w_{\sigma(k)} = s\sum_{1\le k\le \ell\le 3m} a_\ell a_k\ .$$

This expression is independent of the order in which the jobs are scheduled, and it is equal to the first summand of α. Thus, for minimizing the cost w.r.t. f we can ignore f_1 and determine a schedule minimizing the cost w.r.t. f_2.

Function f_2 can be further split up into $f_2=f_2^1+f_2^2+\ldots$, where $f_2^i(t)=0$ for $t\le(i-1)B$ and $f_2^i(t)=(r-s)$ for $t>(i-1)B$. For $i=1,\ldots,m$, let W_i be the total weight of all jobs with completion time greater than $(i-1)B$. As the total processing time and weight of all jobs is mB, it clearly holds that $W_i\ge(m-i+1)B$ and so the cost of any schedule with respect to f_2^i is $(r-s)W_i\ge(r-s)(m-i+1)B$. Furthermore, this holds with equality if and only if a new job starts at time $(i-1)B$. Therefore, the total cost w.r.t. f_2 is at least

$$\sum_{i=1}^{m}(r-s)(m-i+1)B \;=\; \sum_{i=1}^{m} B\cdot i\cdot(r-s) \;=\; \frac{(r-s)\cdot B\cdot m(m+1)}{2}\ ,$$

which is exactly the second summand of α, and this cost is only reached if a new job starts at each time $(i-1)B$ for $i=1,\ldots,m$. □

5 Conclusions

We have shown that for monotone and piecewise linear cost function f problem $1\,||\sum w_j f(C_j)$ is strongly NP-hard, and we have given a tight analysis of Smith's rule that can be applied for arbitrary convex or concave cost functions.

We remark that the cost function of the instances reduced to in the hardness proof is neither convex nor concave, so the computational complexity of the problem for convex/concave cost functions remains open. We believe that a proof of NP-hardness for these cases must have a fundamentally different structure than the proof given in this work, because here the hard instances cannot consist of only jobs with WSPT ratio 1.

For low degree polynomial cost functions WSPT achieves the best known approximation factors. Provided that these problems do not turn out to be in P, another natural question for future research is whether better factors can be achieved in polynomial time in general, and by universal algorithms in particular.

References

1. Afrati, F.N., Bampis, E., Chekuri, C., Karger, D.R., Kenyon, C., Khanna, S., Milis, I., Queyranne, M., Skutella, M., Stein, C., Sviridenko, M.: Approximation schemes for minimizing average weighted completion time with release dates. In: Proc. of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999), pp. 32–44 (1999)
2. Bansal, N., Pruhs, K.R.: Server Scheduling to Balance Priorities, Fairness, and Average Quality of Service. SIAM Journal on Computing 39(7), 3311–3335 (2010)
3. Becchettia, L., Leonardia, S., Marchetti-Spaccamela, A., Pruhs, K.: Online weighted flow time and deadline scheduling. J. Discrete Algorithms 4, 339–352 (2006)
4. Cheung, M., Shmoys, D.B.: A Primal-Dual Approximation Algorithm for Min-Sum Single-Machine Scheduling Problems. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) APPROX/RANDOM 2011. LNCS, vol. 6845, pp. 135–146. Springer, Heidelberg (2011)
5. Epstein, L., Levin, A., Marchetti-Spaccamela, A., Megow, N., Mestre, J., Skutella, M., Stougie, L.: Universal Sequencing on a Single Machine. In: Eisenbrand, F., Shepherd, F.B. (eds.) IPCO 2010. LNCS, vol. 6080, pp. 230–243. Springer, Heidelberg (2010)
6. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Ann. Discrete Math. 5, 287–326 (1979)
7. Höhn, W., Jacobs, T.: An experimental and analytical study of order constraints for single machine scheduling with quadratic cost. In: Proc. of the 14th Workshop on Algorithm Engineering and Experiments (ALENEX 2012). SIAM (to appear, 2012)
8. Jacobs, T.: Resistant algorithms: Obtaining simpler proofs by excluding corner cases. CoRR abs/0910.1387 (2009)
9. Mondal, S.A., Sen, A.K.: An improved precedence rule for single machine sequencing problems with quadratic penalty. Eur. J. Oper. Res. 125, 425–428 (2000)
10. Smith, W.E.: Various optimizers for single-stage production. Nav. Res. Logist. Q. 3, 59–66 (1956)
11. Stiller, S., Wiese, A.: Increasing speed scheduling and flow scheduling. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010, Part II. LNCS, vol. 6507, pp. 279–290. Springer, Heidelberg (2010)
12. Townsend, W.: The single machine problem with quadratic penalty function of completion times: a branch-and-bound solution. Manage. Sci. 24, 530–534 (1978)