



HAL
open science

Order constraints for single machine scheduling with non-linear cost

Christoph Dürr, Oscar Carlos Vasquez Perez

► **To cite this version:**

Christoph Dürr, Oscar Carlos Vasquez Perez. Order constraints for single machine scheduling with non-linear cost. 16th Workshop on Algorithm Engineering and Experiments (ALENEX 2014), Jan 2014, Portland, United States. pp.98-111, 10.1137/1.9781611973198.10 . hal-00908203

HAL Id: hal-00908203

<https://hal.science/hal-00908203>

Submitted on 22 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Order constraints for single machine scheduling with non-linear cost

Christoph Dürr^{*†}

Oscar C. Vásquez^{*‡}

Abstract

Typically in a scheduling problem we are given jobs of different processing times p_j and different priority weights w_j , and need to schedule them on a single machine in order to minimize a specific cost function. In this paper we consider the non-linear objective function $\sum w_j C_j^\beta$, where C_j is the completion time of job j and $\beta > 0$ is some arbitrary real constant. Except for $\beta = 1$ the complexity status of this problem is open. Past research mainly focused on the quadratic case ($\beta = 2$) and proposed different techniques to speed up exact algorithms. This paper proposes new pruning rules and generalizations of existing rules to non-integral β . An experimental study evaluates the impact of the proposed rules on the exact algorithm A*.

1 Introduction

In a scheduling problem the goal is to produce a schedule for the given jobs, that minimizes some objective function, usually depending on the job completion times, under problem specific constraints. One natural goal is to minimize the maximum completion time among all jobs, aka *makespan*. Another natural goal is to minimize the average completion time. A common way to combine these two objectives is to minimize the L_β -norm of the completion times, for some constant $\beta > 1$. As an example, the \TeX -typesetting system uses the ℓ_3 metric to compute optimal line breaks.

In this paper we consider the more general objective function $\sum_j w_j C_j^\beta$ where w_j is the given priority weight of job j , C_j its completion time and β some fixed positive constant. For convenience we introduce the function $f : t \mapsto t^\beta$, and call it the *penalty function*. We consider the most simple setting, where we are given n jobs, each job j has a processing time p_j and a priority weight w_j , and a schedule consists of an ordering of these jobs on a single machine. Here the completion time of job j is simply p_j plus the sum of the processing times over all jobs that are scheduled before j . There are several motivations to this scheduling problem. Some machines have a learning effect, and

their efficiency is increasing during the execution, while some other machines have a wear and tear effect and their efficiency is decreasing during the execution. This can be expressed with values of β respectively smaller or greater than one. Moreover in [17, 11] a particular dynamic voltage scheduling problem optimizing quality of service and energy consumption is reduced to this same objective function for some $0 < \beta < 1$.

Embarrassingly very little is known about the computational complexity of this problem, except for the special case $\beta = 1$, where scheduling jobs in order of decreasing Smith ratio w_j/p_j leads to the optimal schedule, as has been found out 60 years ago [14]. Most research focused on the quadratic objective function, i.e. for $\beta = 2$, and exact algorithms have been proposed [12, 5, 13, 2, 16].

Two research directions were applied to this problem, approximation algorithms and branch and bound algorithms. Approximation algorithms have been proposed for the more general problem $1||\sum f_j(C_j)$, where every job j is given an increasing penalty function $f_j(t)$, that does not need to be of the form $w_j t^\beta$. Bansal and Pruhs [3] provided a constant ratio approximation algorithm based on a geometric interpretation. This factor has been improved from 16 to $2 + \epsilon$ via a primal-dual approach by Cheung and Shmoys [4]. Epstein et al. considered the problem $1||\sum w_j f(C_j)$, and provided a $4 + \epsilon$ approximation algorithm for the setting where f is an arbitrary increasing differentiable penalty function chosen by the adversary *after* the schedule has been produced. A polynomial time approximation scheme has been provided by Megow and Verschae [11] for general monotone penalty functions f .

Finally, Höhn and Jacobs [9] derived a method to compute the tight approximation factor of the Smith-ratio-schedule for any particular monotone increasing convex or concave cost function. In particular for $f(t) = t^\beta$ they obtained for example the ratio 1.07 when $\beta = 0.5$ and the ratio 1.31 when $\beta = 2$.

For branch-and-bound algorithms pruning rules were proposed which reduce the number of nodes in the search graph, having a direct effect on the running time. For example, if we knew, that without loss of generality in an optimal schedule, job i is never scheduled after job j — which we denote by $i \prec_g j$ — then we could

^{*}Supported by ANR-Netoc

[†]CNRS, LIP6, Université Pierre et Marie Curie, Paris, France

[‡]LIP6 and Industrial Engineering Department, University of Santiago of Chile

eliminate roughly half of the potential orderings, and reduce the number of explored nodes.

So an extensive literature was devoted to finding stronger rules, which are weaker conditions on the job characteristics that would still imply $i \prec_g j$. The contribution of our paper is to provide new rules and generalize existing ones to arbitrary values of $\beta > 0$. We conclude this paper with an experimental study of the impact of our proposed pruning rules to the performance of an exhaustive search procedure.

2 Pruning rules

Consider a schedule where i, j are scheduled one after the other, with job i starting at time t . Then the exchange of jobs i, j has some effect to the objective value of the schedule, which depends on jobs i, j and on time t . We denote by $i \prec_{\ell(t)} j$ the property that the order i, j generates a strictly smaller cost than the order j, i . If it holds for all time points $t \in [a, b]$, then we denote this property by $i \prec_{\ell[a,b]} j$, and if it holds everywhere we denote it simply by $i \prec_{\ell} j$.

The previously introduced properties generalize to situations where i and j are not scheduled adjacently. We say that i, j satisfy the *global order property* for interval $[a, b]$ if whenever in some schedule S , the completion times of jobs i, j satisfy

$$a \leq C_j - p_j \leq C_i - p_i - p_j \leq b,$$

then S is sub-optimal.¹

Again if this property holds for all intervals, then we use simply the notation $i \prec_g j$.

We state the following conjecture, motivated by partial results and experiments.

CONJECTURE 1. *For all jobs i, j , $i \prec_{\ell} j$ implies $i \prec_g j$.*

A proof for this conjecture is given in this paper, for the case $\beta = 2$. Also during the experimental study, described in the last section of this paper, we successfully verified the conjecture on all the instances of our tests cases.

Interestingly the stronger implication $i \prec_{\ell[a,b]} j \Rightarrow i \prec_{g[a,b]} j$ does not hold. A counter example consists of the instance for $\beta = 2$ with $p_i = 13, w_i = 7, p_j = 8, w_j = 5, p_k = 1, w_k = 1$. For $t = 19/18$, we have $i \prec_{\ell[0,t]} j$ and $j \prec_{\ell(t,\infty)} i$. But the unique optimal solution is the sequence $jk i$, meaning that we don't have $i \prec_{g[0,t]} j$. This contrasts with Theorem 5.1, which states that the implication holds whenever $p_i \leq p_j$.

¹From the proof of Theorem 5.1 it will become clear why we require this lower bound on C_j .

3 Related work

Previous research focused mainly on the quadratic penalty function, i.e. $\beta = 2$. Branch-and-bound approaches with pruning rules implying order properties have been proposed, see [16, 2, 13, 1, 5, 15]. In 2000, Mondal and Sen [12] conjectured that $\beta = 2, (w_i \geq w_j) \wedge (w_i/p_i > w_j/p_j)$ implies the global order property $i \prec_g j$, and provided experimental evidence that this constraint would significantly improve the runtime of a branch-and-prune search. Recently, Höhn and Jacobs [7] succeeded to prove this conjecture. In addition they improved local and global order conditions and generalized them to integer constants $\beta \geq 2$. An extensive experimental study analyzed the effect of these rules to the performance of the branch-and-prune search.

We distinguish the following known rules.

Sen-Dileepan-Ruparel [13] for any $\beta > 0$, if $w_i > w_j$ and $p_i \leq p_j$, then $i \prec_g j$.

Höhn-Jacobs-1 [7] for $\beta \in \mathbb{N}, \beta \geq 3$, if $w_i/p_i \geq \beta w_j/p_j$ then $i \prec_{\ell} j$.

Höhn-Jacobs-2 [7] for $\beta \in \mathbb{N}, \beta \geq 3$, if $w_i \geq w_j$ and $w_i/p_i > w_j/p_j$ then $i \prec_{\ell} j$.

Mondal-Sen-Höhn-Jacobs Conjectured in [12], proved in [7]. For $\beta = 2$ the two previous rules are enforced by the stronger implication $i \prec_g j$.

In this paper we characterize the condition $i \prec_{\ell} j$, and provide new sufficient conditions for the property $i \prec_g j$. For the special case $\beta = 2$ actually $i \prec_{\ell} j$ implies $i \prec_g j$.

4 Preliminaries

To simplify notation, throughout the paper we assume that no two jobs have the same processing time, weight or Smith-ratio (weight over processing time). For convenience we extend the notation of the penalty function f to the makespan of schedule S as $f(S) := f(\sum_{i \in S} p_i)$. Also we denote by $F(S)$ the cost of schedule S .

We define the following function on $t \geq 0$

$$\phi_{ij}(t) := \frac{f(t + p_i + p_j) - f(t + p_j)}{f(t + p_i + p_j) - f(t + p_i)},$$

and

$$\Delta_{ij}(t) := w_j f(t + p_j) + w_i f(t + p_i + p_j) - w_i f(t + p_i) - w_j f(t + p_i + p_j),$$

Note that $\phi_{ij}(t)$ is well defined since f is strictly increasing by assumption and the durations p_i, p_j are

non-zero. It is this function ϕ_{ij} that permits us to analyze algebraically the local order property, since

$$i \prec_{\ell(t)} j \Leftrightarrow \phi_{ij}(t) < \frac{w_i}{w_j} \Leftrightarrow \Delta_{ij}(t) > 0.$$

The following technical lemmas show a connection between the properties of function $f : t \mapsto t^\beta$ and properties of function ϕ_{ij} , and show properties of f .

LEMMA 4.1. *If $p_i \neq p_j$ then ϕ_{ij} is strictly monotone, in particular:*

- If $p_i > p_j$ and $\beta > 1$, then ϕ_{ij} is strictly increasing.
- If $p_i < p_j$ and $\beta > 1$, then ϕ_{ij} is strictly decreasing.
- If $p_i > p_j$ and $\beta < 1$, then ϕ_{ij} is strictly decreasing.
- If $p_i < p_j$ and $\beta < 1$, then ϕ_{ij} is strictly increasing.

Proof. See Figure 1 for an illustration of the claimed properties. Since $\phi_{ij}(t) = 1/\phi_{ji}(t)$, it suffices to consider the case $p_i > p_j$.

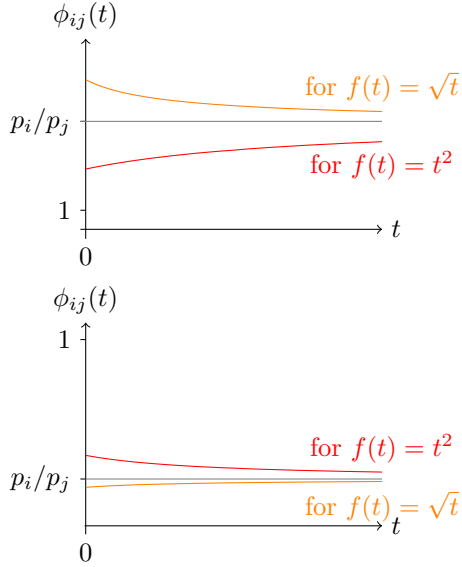


Figure 1: Examples of the function $\phi_{ij}(t)$ for $\beta = 0.5$ and $\beta = 2$, as well as for the cases $p_i > p_j$ and $p_i < p_j$.

Fix an arbitrary $t \geq 0$. For convenience let T be a fictive job of processing time t . We will show that

$$(4.1) \quad \phi'_{ij}(t) = \frac{[f'(Tij) - f'(Tj)][f(Tij) - f(Ti)] - [f'(Tij) - f'(Ti)][f(Tij) - f(Tj)]}{[f(Tij) - f(Ti)]^2} > 0.$$

Since the denominator of this fraction is positive, we can focus on the numerator:

$$\begin{aligned} & [f'(Tij) - f'(Tj)][f(Tij) - f(Ti)] \\ & - [f'(Tij) - f'(Ti)][f(Tij) - f(Tj)] \\ = & f'(Tij)f(Tij) - f'(Tj)f(Tij) \\ & - f'(Tij)f(Ti) + f'(Tj)f(Ti) \\ & - f'(Tij)f(Tij) + f'(Ti)f(Tij) \\ & + f'(Tij)f(Tj) - f'(Ti)f(Ti) \\ = & f'(Tij)f(Tj) - f'(Tj)f(Tij) \\ & + f'(Ti)f(Tij) - f'(Tij)f(Ti) \\ & + f'(Tj)f(Ti) - f'(Ti)f(Tj). \end{aligned}$$

Up to factor β inequality (4.1) is equivalent to

$$\begin{aligned} & (t + p_j)^{\beta-1} t^{\beta-1} ((t + p_j) - t) \\ & + (t + p_i)^{\beta-1} t^{\beta-1} (t - (t + p_i)) \\ & + (t + p_j)^{\beta-1} (t + p_i)^{\beta-1} ((t + p_i) - (t + p_j)) > 0 \\ \equiv & (t + p_j)^{\beta-1} t^{\beta-1} p_j \\ & + (t + p_j)^{\beta-1} (t + p_i)^{\beta-1} (p_i - p_j) > (t + p_i)^{\beta-1} t^{\beta-1} p_i \\ \equiv & (p_j/p_i)(1/(t + p_i))^{\beta-1} \\ & + (1 - p_j/p_i)(1/t)^{\beta-1} > (1/(t + p_j))^{\beta-1} \end{aligned}$$

Using a function $h : x \mapsto (1/x)^{\beta-1}$ we reformulate this inequality as

$$(p_j/p_i)h(t + p_i) + (1 - p_j/p_i)h(t) > h(t + p_j).$$

Note that $h(x)$ is a strictly convex function for positive x and $\beta > 1$, which implies

$$\lambda h(x_1) + (1 - \lambda)h(x_2) > h(\lambda x_1 + (1 - \lambda)x_2)$$

for any $0 < \lambda < 1$ and $x_1, x_2 > 0$.

We choose $\lambda = p_j/p_i$, $x_1 = t + p_i$, $x_2 = t$ and obtain

$$\begin{aligned} \lambda x_1 + (1 - \lambda)x_2 & = (p_j/p_i)(t + p_i) + (1 - p_j/p_i)t \\ & = tp_j/p_i + p_j + t - tp_j/p_i \\ & = t + p_j. \end{aligned}$$

In summary we obtain the inequalities

$$\begin{aligned} & (p_j/p_i)h(t + p_i) + (1 - p_j/p_i)h(t) \\ & > h((p_j/p_i)(t + p_i) + (1 - p_j/p_i)t) \\ & = h(t + p_j), \end{aligned}$$

completing the proof. \square

LEMMA 4.2. *For any jobs i, j , we have*

$$\lim_{t \rightarrow \infty} \phi_{ij}(t) = p_i/p_j.$$

Proof. We prove this claim using the generalized binomial theorem. Here $\binom{\beta}{k}$ is defined for any real positive valued β as $\beta(\beta-1)\cdots(\beta-k+1)/k!$. By definition its value is zero iff β is integral and $0 \leq \beta \leq k-1$. Then we have

$$(4.2) \quad (t+p_i+p_j)^\beta = (t+p_j)^\beta + \beta(t+p_j)^{\beta-1}p_i + \sum_{k=2}^{\infty} \binom{\beta}{k} (t+p_j)^{\beta-k} (p_i)^k.$$

Therefore the limit can be expressed as (where the sum ranges for k from 2 to infinity)

$$\begin{aligned} & \lim_{t \rightarrow \infty} \phi_{ij}(t) \\ &= \lim_{t \rightarrow \infty} \frac{(t+p_i+p_j)^\beta - (t+p_j)^\beta}{(t+p_i+p_j)^\beta - (t+p_i)^\beta} \\ &= \lim_{t \rightarrow \infty} \frac{\beta(t+p_j)^{\beta-1}p_i + \sum \binom{\beta}{k} (t+p_j)^{\beta-k} (p_i)^k}{\beta(t+p_i)^{\beta-1}p_j + \sum \binom{\beta}{k} (t+p_i)^{\beta-k} (p_j)^k} \\ &= \lim_{t \rightarrow \infty} \frac{\beta(1+p_j/t)^{\beta-1}p_i + \sum \binom{\beta}{k} \frac{1}{t^{k-1}} (1+p_j/t)^{\beta-k} (p_i)^k}{\beta(1+p_i/t)^{\beta-1}p_j + \sum \binom{\beta}{k} \frac{1}{t^{k-1}} (1+p_i/t)^{\beta-k} (p_j)^k} \\ &= \frac{p_i}{p_j}. \end{aligned}$$

which is positive as required. This concludes the proof. \square

LEMMA 4.4. For $t \geq 0$ let the function q be defined as

$$q(t) := \frac{f(t+p_j) - f(t)}{f(t+p_i) - f(t)} = \frac{(t+p_j)^\beta - t^\beta}{(t+p_i)^\beta - t^\beta}.$$

For $p_i > p_j$, if $\beta > 1$ then q is increasing and if $0 < \beta < 1$ then q is decreasing.

Proof. We show only the convex case, the concave case is analogous. First, we compute the first derivative of q

$$q'(t) = \beta \frac{((t+p_i)^\beta - t^\beta)((t+p_j)^{\beta-1} - t^{\beta-1}) - ((t+p_j)^\beta - t^\beta)((t+p_i)^{\beta-1} - t^{\beta-1})}{((t+p_i)^\beta - t^\beta)^2}$$

We now show that q' is strictly increasing. Since the denominator of this fraction is positive, we can focus on the numerator. Up to factor β this is equivalent to:

$$(4.3) \quad \begin{aligned} & (t+p_i)^\beta(t+p_j)^{\beta-1} - (t+p_i)^\beta t^{\beta-1} \\ & - t^\beta(t+p_j)^{\beta-1} + t^\beta t^{\beta-1} \\ & - (t+p_j)^\beta(t+p_i)^{\beta-1} + (t+p_j)^\beta t^{\beta-1} \\ & + t^\beta(t+p_i)^{\beta-1} - t^\beta t^{\beta-1}. \end{aligned}$$

\square

LEMMA 4.3. For $a < b$ and $p_i > p_j$,

$$\frac{p_i}{p_j} \cdot \frac{f(b+p_i) - f(a+p_i)}{f(b+p_j) - f(a+p_j)} \geq 1.$$

Proof. When f is convex, the second fraction is clearly greater than 1. So we focus on the concave case. For this purpose we define the function

$$g(x) := x(f(b+x) - f(a+x))$$

and show that g is increasing, implying $g(p_i)/g(p_j) \geq 1$ as required. So we have to show $g'(x) > 0$ in other words

$$f(b+x) - f(a+x) + x(f'(b+x) - f'(a+x)) \geq 0$$

or

$$(b+x)^\beta + x\beta(b+x)^{\beta-1} \geq (a+x)^\beta + x\beta(a+x)^{\beta-1}.$$

To establish the last inequality, we introduce another function

$$r(z) := (z+x)^\beta + x\beta(z+x)^{\beta-1}$$

and show that r is increasing, implying $r(b) \geq r(a)$. By analyzing its derivative we obtain

$$\begin{aligned} r'(z) &= \beta(z+x)^{\beta-1} + x\beta(\beta-1)(z+x)^{\beta-2} \\ &= (z+x)^{\beta-2}(\beta(z+x) + x\beta(\beta-1)) \\ &= (z+x)^{\beta-2}(\beta z + x\beta^2), \end{aligned}$$

We use the transformation

$$\begin{aligned} & (t+p_i)^\beta(t+p_j)^{\beta-1} - (t+p_j)^\beta(t+p_i)^{\beta-1} \\ &= (t+p_i)(t+p_i)^{\beta-1}(t+p_j)^{\beta-1} \\ & - (t+p_j)(t+p_j)^{\beta-1}(t+p_i)^{\beta-1} \\ &= (t+p_i)^{\beta-1}(t+p_j)^{\beta-1}(p_i - p_j) \end{aligned}$$

to transform (4.3) into

$$(t+p_i)^{\beta-1}(t+p_j)^{\beta-1}(p_i - p_j) - (t+p_i)^{\beta-1}t^{\beta-1}p_i + (t+p_j)^{\beta-1}t^{\beta-1}p_j$$

which is positive if and only if

$$(p_j/p_i)(1/(t+p_i))^{\beta-1} + (1-p_j/p_i)(1/t)^{\beta-1} > (1/(t+p_j))^{\beta-1}.$$

Using function $h : x \mapsto (1/x)^{\beta-1}$ we reformulate this inequality as

$$(p_j/p_i)h(t+p_i) + (1-p_j/p_i)h(t) > h(t+p_j).$$

Note that $h(x)$ is a strictly convex function for positive x and $\beta > 1$, which implies

$$\lambda h(x_1) + (1-\lambda)h(x_2) > h(\lambda x_1 + (1-\lambda)x_2)$$

for any $0 < \lambda < 1$ and $x_1, x_2 > 0$.

We choose $\lambda = p_j/p_i$, $x_1 = t + p_i$, $x_2 = t$ and obtain

$$\begin{aligned} \lambda x_1 + (1 - \lambda)x_2 &= (p_j/p_i)(t + p_i) + (1 - p_j/p_i)(t) \\ &= tp_j/p_i + p_j + t - tp_j/p_i \\ &= t + p_j. \end{aligned}$$

In summary we obtain the required inequality

$$\begin{aligned} &(p_j/p_i)h(t + p_i) + (1 - p_j/p_i)h(t) \\ &> h((p_j/p_i)(t + p_i) + (1 - p_j/p_i)t) \\ &= h(t + p_j). \end{aligned}$$

This concludes the proof. \square

5 Main Results

In [13] it has been shown that if job j has both longer processing time than i and smaller weight than i , then $i \prec_g j$, whereas in [7, 12] some conditions on jobs i, j imply the global order property were given for the quadratic penalty function. We enforce these statements using properties of $\phi_{ij}(t)$.

THEOREM 5.1. (RULE 1) *Let f be an arbitrary strictly increasing penalty function. Fix two jobs i, j , an interval $[a, b]$ and suppose $i \prec_{\ell[a, b]} j$. If $p_i \leq p_j$ then $i \prec_{g[a, b]} j$.*

Proof. Suppose that $p_i \leq p_j$. Let I be an instance containing jobs i, j and S a schedule on I of the form

$$S = AjBiD,$$

for some job sequences A, B, D . Let a be the total processing time of A and b the total processing time of AB . Then we have $a = C_j - p_j \leq C_i - p_i - p_j = b$ where C_i, C_j are the respective completion times in S . We show that exchanging the jobs i and j decreases the cost of the schedule. In particular we show the following inequality, where we dropped the suffix D of the schedules, since those jobs cancel in the difference anyway,

$$(5.4) \quad \max_{t \in [a, b]} \phi_{ij}(t) [F(AijB) - F(AiBj)] > F(AjiB) - F(AjBi).$$

This inequality would conclude the proof, for the following reason. First we claim

$$\max_{t \in [a, b]} \phi_{ij}(t) \leq 1,$$

which holds by definition of f in case $p_i \leq p_j$. This implies the stronger inequality

$$F(AijB) - F(AiBj) > F(AjiB) - F(AjBi),$$

or equivalently

$$(5.5) \quad F(AjBi) - F(AiBj) > F(AjiB) - F(AijB).$$

Since $i \prec_{\ell(a)} j$ implies that the right hand side is non-negative, this would conclude the proof.

In order to show inequality (5.4), for every job $k \in B$ we denote by t_k the completion time of k in the schedule $ABij$. In (5.4) we distinguish the contributions of jobs i, j and all jobs $k \in B$. In particular for every job k in B we have

$$(5.6) \quad \begin{aligned} &\max_{t \in [a, b]} \phi_{ij}(t) w_k [f(t_k + p_i + p_j) - f(t_k + p_i)] \\ &\geq w_k [f(t_k + p_i + p_j) - f(t_k + p_j)] \end{aligned}$$

since

$$\begin{aligned} &\phi_{ij}(t_k) w_k [f(t_k + p_i + p_j) - f(t_k + p_i)] \\ &= w_k [f(t_k + p_i + p_j) - f(t_k + p_j)]. \end{aligned}$$

We denote by p_B the total processing time over all jobs in B . Then since $f(C_i - p_B) < f(C_i)$ and since $\max_{t \in [a, b]} \phi_{ij}(t) < w_i/w_j$ we have

$$(5.7) \quad \begin{aligned} &\max_{t \in [a, b]} \phi_{ij}(t) w_j [f(C_i - p_B) - f(C_i)] \\ &> \frac{w_i}{w_j} \cdot w_j [f(C_i - p_B) - f(C_i)]. \end{aligned}$$

Adding (5.6) and (5.7) establishes the required inequality (5.4). \square

The following statement permits us to enumerate some conditions on job pairs and their Smith-ratios, which imply either a local or a global order property.

THEOREM 5.2. (RULE 2) *If f is an increasing strictly convex function, $w_i/p_i > w_j/p_j$ and $w_i > w_j$, then $i \prec_{\ell} j$. On the other hand if f is an increasing strictly concave function, $w_i/p_i > w_j/p_j$ and $w_i < w_j$, then $i \prec_g j$.*

Proof. First, we consider the case of a convex function f .

For $0 < \lambda < 1$, we have

$$(1 - \lambda)f(x_1) + \lambda f(x_2) \geq f((1 - \lambda)x_1 + \lambda x_2).$$

We choose $\lambda = w_j/w_i$, $x_1 = t + p_i + p_j$, $x_2 = t + p_j$ and have:

$$\begin{aligned} &(1 - w_j/w_i)f(t + p_i + p_j) + w_j/w_i f(t + p_j) \\ &\geq f(t + p_i + p_j - w_j p_i/w_i) > f(t + p_i). \end{aligned}$$

The latter inequality holds by case assumption on Smith-ratios of jobs i and j , and the strict monotonicity of f . Thus, we have:

$$(1 - w_j/w_i)f(t + p_i + p_j) + w_j/w_i f(t + p_j) - f(t + p_i) = \Delta_{ij}(t)/w_i > 0,$$

which implies $i \prec_\ell j$.

For the concave case, we have the implication $i \prec_\ell j$ by a similar argument. To prove the implication global, we observe that $p_i < p_j$ follows from the case assumption, and applying Theorem 5.1 permits to conclude $i \prec_g j$. \square

Finally, we refine the above statements for the function of the form $f(t) : t \mapsto t^\beta$, with $\beta \in \mathbb{R}^+$.

THEOREM 5.3. (RULE 3) *Let i, j be two jobs with $p_j < p_i$. If*

$$\begin{aligned} \beta > 1 \quad \text{and} \quad \frac{w_i}{w_j} &\geq \left(\frac{p_i}{p_j}\right)^\beta \\ \text{or } 0 < \beta < 1 \quad \text{and} \quad \frac{w_i}{w_j} &\geq \left(\frac{p_i}{p_j}\right)^2 \end{aligned}$$

then

$$i \prec_g j.$$

Proof. Let i, j be two jobs with the required properties. Let A, B be two arbitrary job sequences. We will show that the schedule $AjBi$ is suboptimal, thus showing that $i \prec_g j$.

First if $F(AjBi) \geq F(ABji)$, then by $i \prec_\ell j$ we have $F(AjBi) \geq F(ABji) > F(AiBj)$. So from now on we assume $F(AjBi) < F(ABji)$. Note that we could have assumed $F(AjBi) < F(AjiB)$ as well, but do not have use for it.

To conclude the proof, we will show $F(AjBi) > F(AiBj)$. In particular we will show the inequality

$$F(ABji) - F(AiBj) < F(AjBi) - F(AiBj)$$

or equivalently

$$(5.8) \quad F(ABji) - F(AjBi) < F(ABij) - F(AiBj).$$

Let a be the total processing time of A and b the total processing time of AB . For every job $k \in B$ we denote by t_k the completion time of k in the schedule AB .

By hypothesis the left hand side is positive, and by $w_i/w_j > p_i/p_j$ and Lemma 4.3 we can bound it as

(denoting $W_B(z) := \sum_{k \in B} w_k(f(t_k + z) - f(t_k))$)

$$\begin{aligned} &F(ABji) - F(AjBi) \\ &= w_j(f(b + p_j) - f(a + p_j)) - W_B(p_j) \\ &< w_i(f(b + p_i) - f(a + p_i)) \\ &\quad - \frac{w_i f(b + p_i) - f(a + p_i)}{w_j f(b + p_j) - f(a + p_j)} W_B(p_j) \\ &< w_i(f(b + p_i) - f(a + p_i)) \\ &\quad - \frac{w_i f(b + p_i) - f(a + p_i)}{w_j f(b + p_j) - f(a + p_j)} \min_{t \geq 0} \frac{f(t + p_j) - f(t)}{f(t + p_i) - f(t)} W_B(p_i). \end{aligned}$$

In order to upper bound the later expression by

$$\begin{aligned} &\leq w_i(f(b + p_i) - f(a + p_i)) - W_B(p_i) \\ &= F(ABij) - F(AiBj) \end{aligned}$$

as required, it suffices to show

$$\frac{w_i f(b + p_i) - f(a + p_i)}{w_j f(b + p_j) - f(a + p_j)} \min_{t \geq 0} \frac{f(t + p_j) - f(t)}{f(t + p_i) - f(t)} \geq 1.$$

This last step distinguishes two cases.

Case $\beta > 1$. By Lemma 4.4 the last fraction is minimum at $t = 0$, where it has the value $(p_j/p_i)^\beta$. By assumption $w_i/w_j \geq (p_i/p_j)^\beta$, the product of the first and last fraction is at least 1. By convexity of f , the second fraction is lower bounded by 1 as well, and we are done.

Case $0 < \beta < 1$. By Lemma 4.4 the last fraction is minimum at the limit $t \rightarrow \infty$, where it has the value p_j/p_i . By assumption $w_i/w_j \geq (p_i/p_j)^2$, we have that

$$\begin{aligned} &\frac{w_i f(b + p_i) - f(a + p_i)}{w_j f(b + p_j) - f(a + p_j)} \min_{t \geq 0} \frac{f(t + p_j) - f(t)}{f(t + p_i) - f(t)} \\ &\geq \left(\frac{p_i f(b + p_i) - f(a + p_i)}{p_j f(b + p_j) - f(a + p_j)}\right) \left(\frac{p_i p_j}{p_j p_i}\right) \end{aligned}$$

which is at least 1 by Lemma 4.3. This concludes the proof of the theorem. \square

We provide a proof of the special case $\beta = 2$ of our conjecture.

THEOREM 5.4. *Consider the penalty quadratic monomial function and fix two arbitrary jobs i, j . If $i \prec_\ell j$ then $i \prec_g j$.*

Proof. The case $p_i \leq p_j$, is covered by Theorem 5.1 for the interval $[0, \infty)$.

For the case $p_i > p_j$, by Lemma 4.1, ϕ_{ij} is strictly increasing and by Lemma 4.2 we have for any $t \geq 0$

$$\phi_{ij}(t) < p_i/p_j.$$

Thus, $\phi_{ij}(t) < w_i/w_j$ for any $t \geq 0$ (or equivalently $i \prec_\ell j$) if only if $p_i/p_j \geq w_i/w_j$. Finally, from the Mondal-Sen-Höhn-Jacobs rule we obtain $i \prec_g j$ (see [7]). \square

In summary we obtain the following rules

COROLLARY 5.1. (OUR RULES) *Let i, j be two jobs, with $p_i > p_j$.*

If $\beta > 1$

if $(p_i/p_j)^\beta \leq w_i/w_j$	then $i \prec_g j$
if $p_i/p_j < w_i/w_j$	then $i \prec_\ell j$ (*)
if $w_i/w_j < \phi_{ij}(0)$	then $j \prec_g i$
else $\exists t^* : w_i/w_j = \phi_{ij}(t^*)$ and	$i \prec_{\ell(0, t^*)} j$ and $j \prec_{g(t^*, \infty)} i$

If $\beta < 1$

if $(p_i/p_j)^2 \leq w_i/w_j$	then $i \prec_g j$
if $\phi_{ij}(0) < w_i/w_j$	then $i \prec_\ell j$
if $w_i/w_j < p_i/p_j$	then $j \prec_{g[0, t_1]} i$
else $\exists t^* : w_i/w_j = \phi_{ij}(t^*)$ and	$i \prec_{g(0, t^*)} j$ and $j \prec_{\ell(t^*, \infty)} i$,

where in case $\beta = 2$ (*) is enforced by the implication $i \prec_g j$.

Proof. We consider the case $\beta > 1$. The case $\beta < 1$ is symmetric, and the case $\beta = 2$ follows from the previous theorem. The first is Rule 3. For the second condition, ϕ_{ij} is strictly increasing by Lemma 4.1, and has limit p_i/p_j . Therefore $p_i/p_j < w_i/w_j$ implies $i \prec_\ell j$.

Monotonicity of ϕ_{ij} together with $\phi_{ij}(0) > w_i/w_j$ implies $\phi_{ij}(t) > w_i/w_j$ for all $t \geq 0$ as well, that is $j \prec_\ell i$. The implication $j \prec_g i$ follows by Theorem 5.1.

If none of the inequalities holds, by Lemma 4.1 and continuity of ϕ_{ij} there must be a unique time t^* such that $\phi_{ij}(t^*) = w_i/w_j$. In addition we have $i \prec_{\ell(0, t^*)} j$ and $j \prec_{\ell(t^*, \infty)} i$. We apply Theorem 5.1 to conclude $j \prec_{g(t^*, \infty)} i$. \square

Figure 2 illustrates the contribution of our rules for a penalty function $f : t \mapsto t^\beta$.

6 Experimental study

We conclude this paper with an experimental study, evaluating the impact of the proposed rules on the performance of a search procedure. Following the approach described in [7], we consider the Algorithm A* [6]. The search space is the directed acyclic graph consisting of all subsets $S \subseteq \{1, \dots, n\}$. Note that the potential search space has size 2^n which is already less than the space of the $n!$ different schedules. In this graph for every vertex S there is an arc to $S \setminus \{j\}$ for any $j \in S$. It is labeled with j , and has cost $w_j t^\beta$ for $t = \sum_{i \in S} p_i$. Every directed path from the root $\{1, \dots, n\}$ to $\{\}$ corresponds to a schedule of an objective value being the total arc cost.

So the goal is to compute the minimum distance between these two vertices. We use the algorithm A* for this purpose, which explores the graph using a priority queue containing arcs pointing to vertices that still need to be visited. An arc (S', S) has a weight corresponding to the distance from the root to S through this arc plus a basic lower bound of the optimum cost of scheduling S , which we choose to be simply $\sum_{i \in S} w_i p_i^\beta$.

Pruning is done when constructing the list of outgoing arcs at some vertex S . Potentially every job $i \in S$ can generate an arc, but ordering constraints might prevent that. Let j be the label of the arc leading to S (assuming S is not the root). Let $t_1 = \sum_{k \in S} p_k$. Now if $j \prec_{\ell(t_1 - p_i)} i$, then no arc is generated for job $i \in S$. The same thing happens when there is a job $k \in S$ with $i \prec_{g[0, t_1]} k$. In a search tree such a pruning would cut the whole subtree attached to that arc, but in a directed acyclic graph the improvement is not so dramatic, as the typical indegree of a vertex is linear in n .

6.1 Random instances We adopt the model of random instances described by Höhn and Jacobs. All previous experimental results were made by generating processing times and weights uniformly from some interval, which leads to easy instances, since any job pair i, j satisfies with probability 1/2 global precedence, i.e. $i \prec_g j$ or $j \prec_g i$. As an alternative, Höhn and Jacobs [7] proposed a random model, where the Smith-ratio of a job is selected according to $2^{N(0, \sigma^2)}$ with N being the normal distribution centered at 0 with variance σ . Therefore for $\beta = 2$ the probability that two jobs satisfy global precedence depends on σ , since the Höhn-Jacobs-1 rule compares the Smith-ratio among the jobs.

We adopted their model for other values of β as follows. When $\beta > 1$, the condition for $i \prec_g j$ of our rules can be approximated, when p_j/p_i tends to infinity, by the relation $w_i/p_i \geq \beta w_j/p_j$. Therefore in order to obtain a similar “hardness” of the random instances for the same parameter σ for different values of $\beta > 1$, we choose the Smith-ratio according to $2^{N(0, \beta^2 \sigma^2)}$. This way the ratio between the Smith-ratios of two jobs is a random variable from the distribution $2^{2N(0, \beta^2 \sigma^2)}$, and the probability that this value is at least β depends only on σ .

However when β is between 0 and 1, the condition for $i \prec_g j$ of our rule can be approximated when p_j/p_i tends to infinity by the relation $w_i/p_i \geq 2w_j/p_j$, and therefore we choose the Smith-ratio of the jobs according to the β -independent distribution $2^{N(0, 4\sigma^2)}$.

The instances of our main test sets are generated as follows. For each choice of $\sigma \in \{0.1, 0.2, \dots, 1\}$ and $\beta \in \{0.5, 0.8, 1.1, \dots, 3.2\}$. We generated 25 instances of 20 jobs each. The processing time of every job

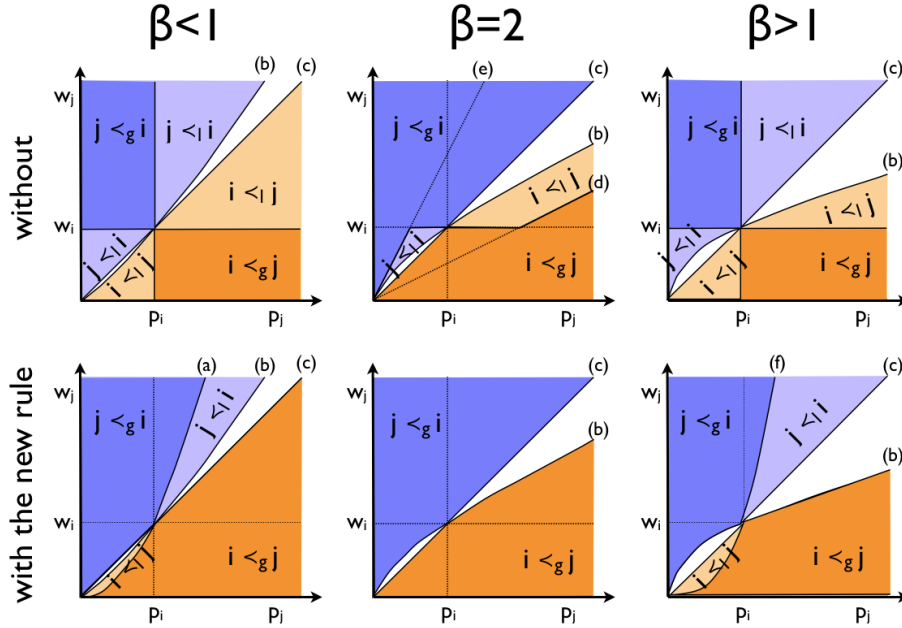


Figure 2: Job j compared to a fixed job i . Labels of particular functions: (a) $w_j = w_i(p_j/p_i)^2$, (b) $w_j = w_i((p_i + p_j)^\beta - p_i^\beta)/((p_i + p_j)^\beta - p_j^\beta)$, (c) $w_j = w_i p_j/p_i$, (d) $w_j = w_i p_j/2p_i$, (e) $w_j = 2w_i p_j/p_i$ and (f) $w_j = w_i(p_j/p_i)^\beta$.

is uniformly generated in $\{1, 2, \dots, 100\}$. Then the weight is generated according to the above described distribution. Note that the problem is independent on scaling of processing time or weights, motivating the arbitrary choice of the constant 100.

6.2 Hardness of instances As a measure of the hardness of instances, we consider the portion of job pairs i, j which satisfy global precedence. By this we mean that we have either $i \prec_{g[0, t_1]} j$ or $j \prec_{g[0, t_1]} i$ for t_1 being the total processing time over all jobs excepting jobs i, j . Figure 4 shows this measure for various choices of β .

The results depicted in Figure 4 confirm the choice of the model of random instances. Indeed the hardness of the instances seems to depend only little on β , except for $\beta = 2$ where particular strong precedence rules have been established. In addition the impact of our new rules is significant, and further experiments show how this improvement influences the number of generated nodes, and therefore the running time. Moreover it is quite visible from the measures that the instances are more difficult to solve when they are generated with a small σ value.

6.3 Comparison between forward and backward variant In this section, we consider two variants

of the above mentioned algorithm. In the *forward* approach, a partial schedule describes a prefix of length t of a complete schedule and is extended to its right along an edge of the search tree, and in this variant the basic lower bound is $\sum_{i \in S} w_i(t + p_i)^\beta$. However in the *backward* approach, a partial schedule S describes a suffix of a complete schedule and is extended to its left. Kaindl, Kainz and Radda [10] give experimental evidence that the backward variant generates for some problems less nodes in the search tree, and this fact has also been observed by Höhn and Jacobs [7].

We conducted an experimental study in order to find out which variant is most likely to be more efficient. The results are shown in Figure 5. The values are most significant for small σ values, since for large values the instances are easy anyway and the choice of the variant is not very important. The results indicate that without our rules the forward variant should be used whenever $\beta < 1$ or $\beta = 2$, while with our rules the forward variant should be used when $\beta > 1$.

Later on, when we measured the impact of our rules in the subsequent experiments, we compared the behavior of the algorithm using the most favorable variant dependent on the value of β as described above.

6.4 Timeout During the resolution a timeout was set, aborting executions that needed more than a million

nodes. In Figure 3 we show the fraction of instances that could be solved within the limited number of nodes. From these experiments we measure the instance sizes that can be efficiently solved, and observe that this limit is of course smaller when σ is small, as the instances become harder. But we also observe that with the usage of our rules much larger instances can be solved.

When β is close to 1, and instances consist of jobs of almost equal Smith-ratio, the different schedules diverge only slightly in cost, and intuitively one has to develop a schedule prefix close to the makespan, in order to find out that it cannot lead to the optimum. However for $\beta = 2$, the Mondal-Sen-Höhn-Jacobs rule make the instances easier to solve than for other values of β , even close to 2. Note that we had to consider different instance sizes, in order to obtain comparable results, as with our rules all 20 job instances could be solved.

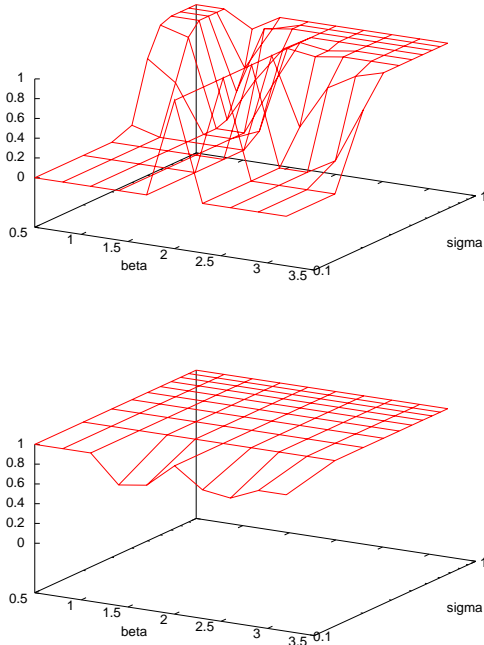


Figure 3: Proportion of instances which could be solved within the imposed time limit of a million nodes, with (below) and without (above) the new rules.

6.5 Improvement factor In this section we measure the influence on the number of nodes generated during a resolution when our rules are used. For $\beta = 2$ we compare our performance with the Mondal-Sen-Höhn-Jacobs rule, while for other values of β we compare with the Sen-Dileepan-Ruparel rule. For fairness

we excluded instances where the timeout was reached without the use of our rules. Figure 6 shows the ratio between the average number of generated nodes when the algorithm is run with our rules, and when it is run without our rules. Clearly this factor is smaller for $\beta = 2$, since the Mondal-Sen-Höhn-Jacobs rules apply here.

We observe that the improvement factor is more important for hard instances, i.e. when σ is small. From the figures it seems that this behavior is not monotone, for $\beta = 1.1$ the factor is less important with $\sigma = 0.1$ than with $\sigma = 0.3$. However this is an artifact of our pessimistic measurements, since we average only over instances which could be solved within the time limit, so in the statistics we filtered out the really hard instances.

7 Performance measurements for $\beta = 2$

For $\beta = 2$, the authors of [7] provide several test sets to measure the impact of their rules in different variants, see [8]. For completeness we selected two data sets from their collection to compare our rules with theirs.

The first set called `set-n` contains for every number of jobs $n = 1, 2, \dots, 35$, 10 instances generated with parameter $\sigma = 0.5$. This file permits to measure the impact of our rules as a function on the instance size.

The second test set that we considered is called `set-T` and contains for every parameter $\sigma = 0.100, 0.101, 0.102, \dots, 1.000$ 3 instances of 25 jobs. Results are depicted in figure 7.

For a general analysis, we generated instances described in section 6.1, and compared the average number of nodes generated with and without our rules.

8 Performance depending on input size

In addition we show the performance of the algorithm with our rules, in dependence on the number of jobs. Figure 8 shows for different number of jobs the number of generated nodes averaged over 100 instances generated with different σ parameters, exposing an expected running time which strongly depends on the hardness of the instances.

9 Acknowledgements

We are grateful to the anonymous referees who spotted errors in previous versions of this paper.

References

- [1] B. Alidaee. Numerical methods for single machine scheduling with non-linear cost functions to minimize total cost. *Journal of the Operational Research Society*, pages 125–132, 1993.
- [2] P. Bagga and K. Karlra. A node elimination procedure for Townsend’s algorithm for solving the single

- machine quadratic penalty function scheduling problem. *Management Science*, pages 633–636, 1980.
- [3] N. Bansal and K. Pruhs. The geometry of scheduling. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 407–414. IEEE Computer Society, 2010.
- [4] M. Cheung and D. Shmoys. A primal-dual approximation algorithm for min-sum single-machine scheduling problems. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 135–146, 2011.
- [5] F. Croce, R. Tadei, P. Baracco, and R. Di Tullio. On minimizing the weighted sum of quadratic completion times on a single machine. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 816–820. IEEE, 1993.
- [6] P. E. Hart, N. J. Nilsson, and B. Raphael. Correction to a formal basis for the heuristic determination of minimum cost paths. *ACM SIGART Bulletin*, (37):28–29, 1972.
- [7] W. Höhn and T. Jacobs. 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.
- [8] W. Höhn and T. Jacobs. Generalized min sum scheduling instance library. http://www.coga.tu-berlin.de/v-menue/projekte/complex_scheduling/generalized_min-sum_scheduling_instance_library/, 2012.
- [9] W. Höhn and T. Jacobs. On the performance of Smith’s rule in single-machine scheduling with nonlinear cost. In D. Fernandez-Baca, editor, *Proceedings of the 10th Latin American Theoretical Informatics Symposium (LATIN ’12)*, volume 7256 of *Lecture Notes in Computer Science*, pages 482–493. Springer Berlin Heidelberg, 2012.
- [10] H. Kaindl, G. Kainz, and K. Radda. Asymmetry in search. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(5):791–796, 2001.
- [11] N. Megow and J. Verschae. Dual techniques for scheduling on a machine with varying speed. In *Proc. of the 40th International Colloquium on Automata, Languages and Programming (ICALP)*, 2013.
- [12] S. Mondal and A. Sen. An improved precedence rule for single machine sequencing problems with quadratic penalty. *European Journal of Operational Research*, 125(2):425–428, 2000.
- [13] T. Sen, P. Dileepan, and B. Ruparel. Minimizing a generalized quadratic penalty function of job completion times: an improved branch-and-bound approach. *Engineering costs and production economics*, 18(3):197–202, 1990.
- [14] W. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- [15] W. Szwarc. Decomposition in single-machine scheduling. *Annals of Operations Research*, 83:271–287, 1998.
- [16] W. Townsend. The single machine problem with quadratic penalty function of completion times: a branch-and-bound solution. *Management Science*, pages 530–534, 1978.
- [17] O. C. Vasquez. Energy in computing systems with speed scaling: optimization and mechanisms design. Technical Report arXiv:1212.6375, arxiv.org, 2012.

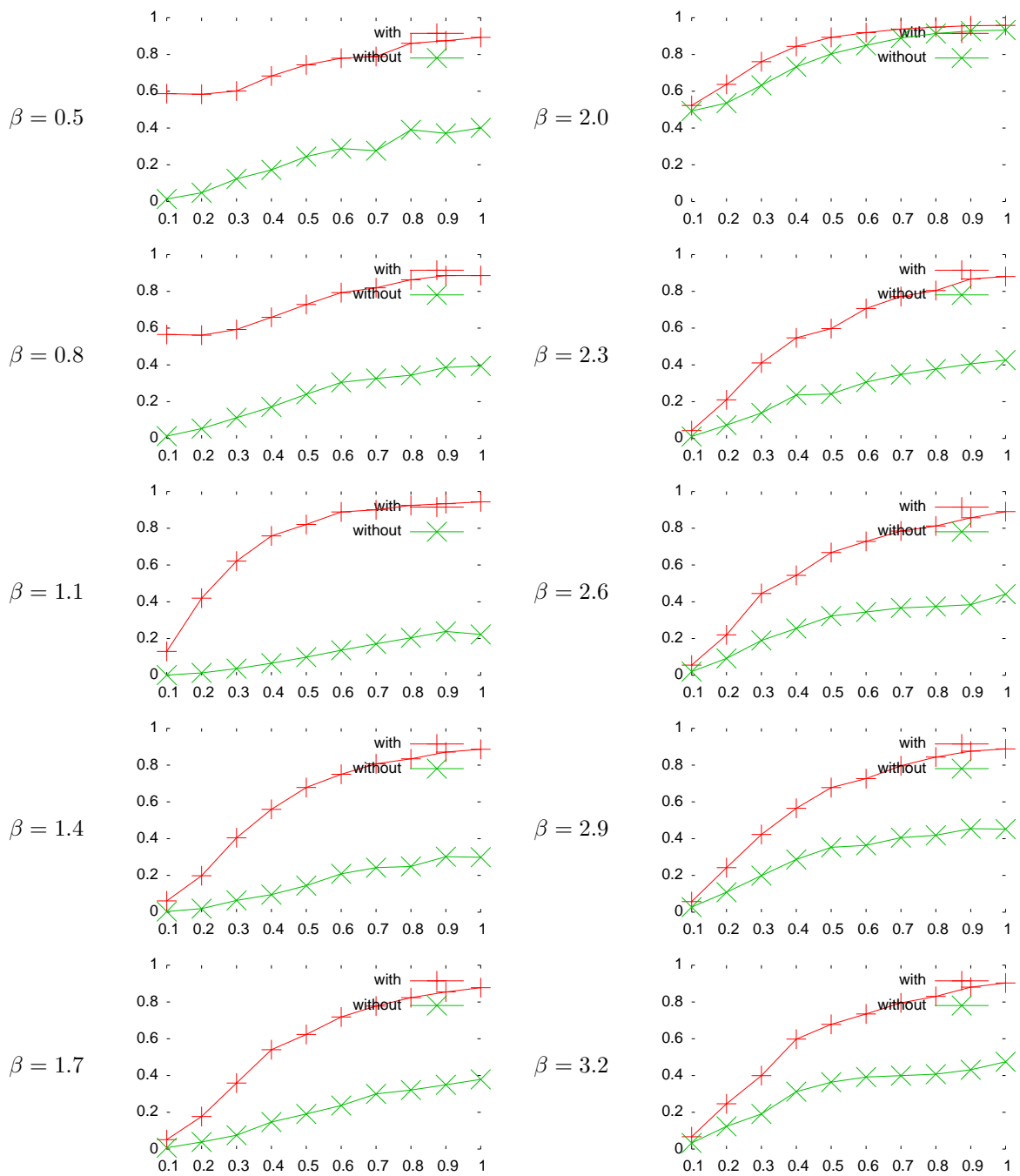


Figure 4: Proportion of job pairs that satisfy a global precedence relation as function of the parameter σ used in the random generation of the instances.

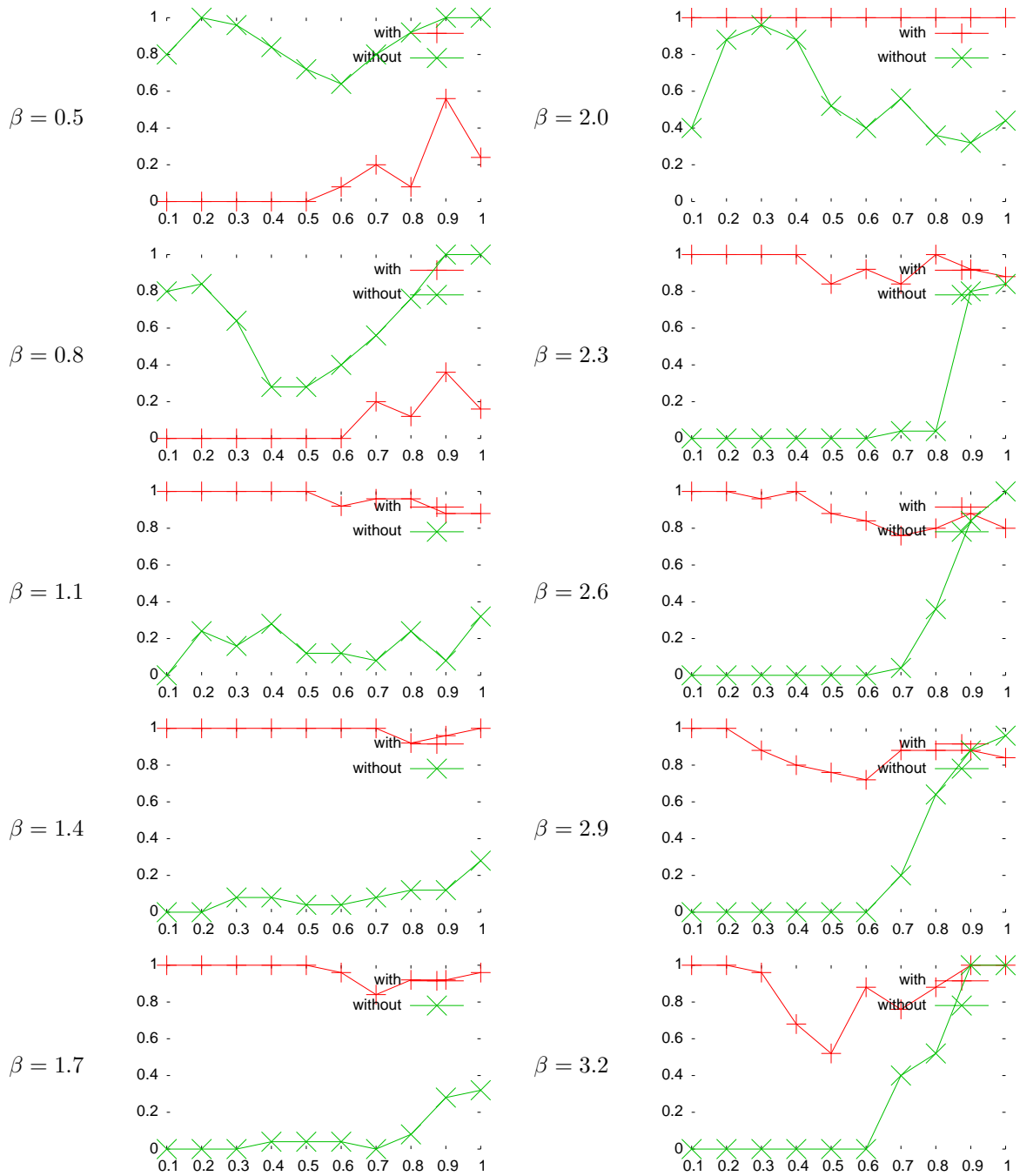


Figure 5: Proportion of instances for which the forward variant generated less nodes than the backward variant. The values are plotted as function of σ , both for the resolution with our new rules and without.

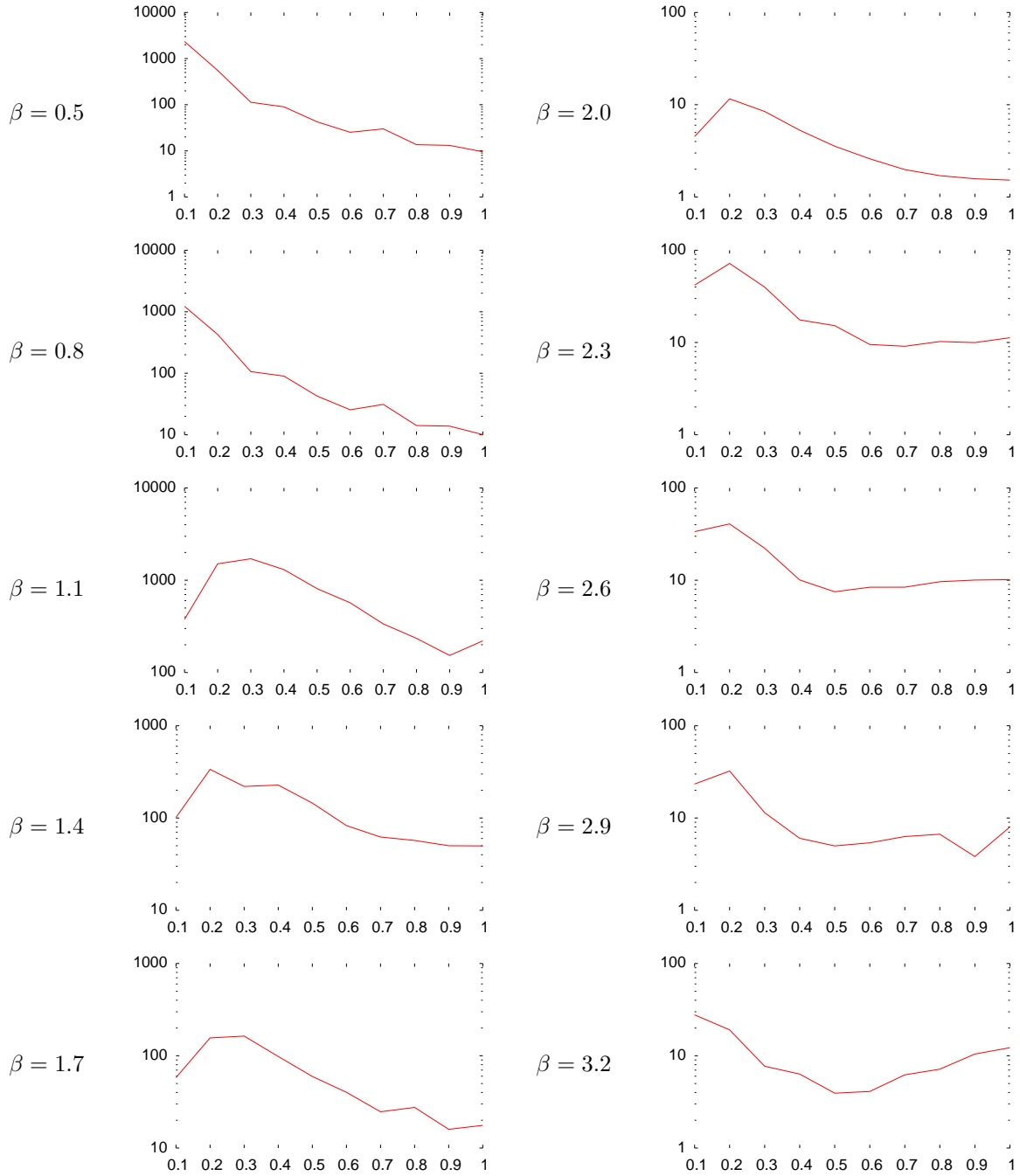


Figure 6: Average improvement factor as function of β and σ

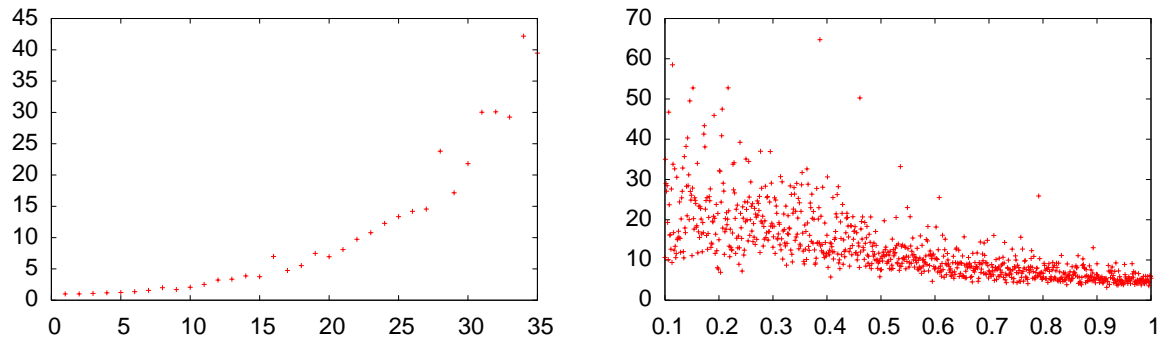


Figure 7: Improvement ratio for test sets `set-n` (left) and `set-T` (right)

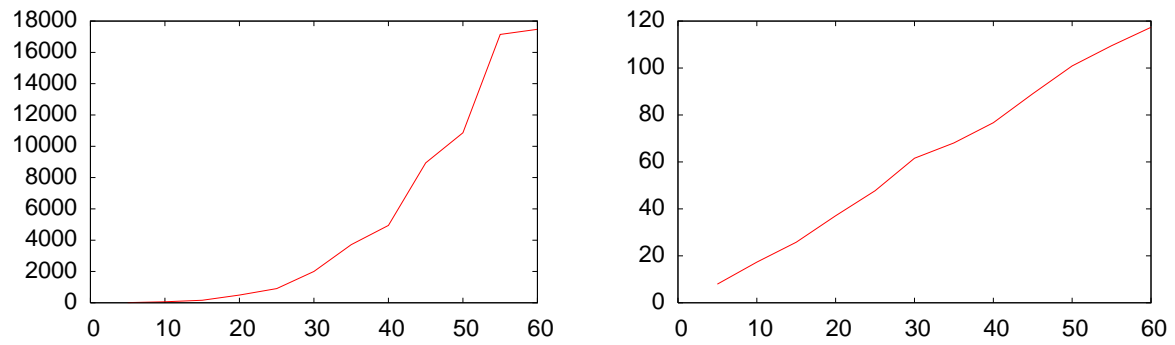


Figure 8: Average number of nodes in dependence on the size of the instances, generated with $\sigma = 0.1$ on the left and $\sigma = 0.5$ on the right.