Conditional Hardness of Precedence Constrained Scheduling on Identical Machines

Ola Svensson
KTH – Royal Institute of Technology
Stockholm, Sweden
osven@kth.se

ABSTRACT

Already in 1966, Graham showed that a simple procedure called list scheduling yields a 2-approximation algorithm for the central problem of scheduling precedence constrained jobs on identical machines to minimize makespan. Till this date it has remained the best algorithm and whether it can or cannot be improved has become a major open problem in scheduling theory.

We address this problem by establishing a quite surprising relation between the approximability of the considered problem and that of scheduling precedence constrained jobs on a single machine to minimize weighted completion time. More specifically, we prove that if the single machine problem is hard to approximate within a factor of $2-\epsilon$ then the considered parallel machine problem, even in the case of unit processing times, is hard to approximate within a factor of $2-\zeta$, where ζ tends to 0 as ϵ tends to 0. Combining this with Bansal & Khot's recent hardness result for the single machine problem gives that it is NP-hard to improve upon the approximation ratio obtained by Graham, assuming a new variant of the unique games conjecture.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Sequencing and scheduling*

General Terms

Theory

1. INTRODUCTION

One of the first approximation algorithms with a worst-case analysis can be traced back to 1966, when Graham [20] studied the following central scheduling problem (known as $P|prec|C_{max}$ in standard scheduling notation [19]): There is a set N of n jobs to be scheduled on m identical parallel machines. Each machine can process at most one job at a time and each job $j \in N$ requires p_j uninterrupted units of processing on one of the machines. Jobs also have precedence constraints between them that are specified by a partial order P on N, where $(i,j) \in P$ implies that job i must be completed before job j can be started. The goal is to find a schedule that minimizes the makespan $C_{max} = \max_j C_j$, where C_j is the time at which job j completes in the given schedule.

Arguably the simplest algorithm for many scheduling problems is the so called *list scheduling* procedure, where we order the jobs in a preference list and schedule the first available job(s) from the list whenever a machine falls idle. In his seminal paper [20], Graham showed that for $P|prec|C_{max}$, the list scheduling procedure has a worst-case performance guarantee of $2-1/m$.

Considering the special case with no precedence constraints, denoted by $P||C_{max}$, Graham [21] later refined his analysis and showed that the list scheduling procedure has a worst-case performance guarantee of $4/3-1/(3m)$, assuming the priority list is obtained by arranging jobs according to non-increasing processing times. Using more complex techniques, Hochbaum & Shmoys [22] improved upon the $4/3-1/(3m)$ approximation guarantee by giving a polynomial time approximation scheme for $P||C_{max}$. This gives a tight result since $P||C_{max}$ is known to be strongly NP-hard and hence does not admit a fully polynomial time approximation scheme [15]. We remark that the problem where the number of machines is fixed, i.e., not part of the input, is also resolved in terms of approximability: Sahni [33] gave a fully polynomial approximation scheme and it is easily seen to be weakly NP-hard by a reduction from partition. In summary, the considered problem is well understood in the absence of precedence constraints.

Unfortunately, understanding the complexity in the presence of precedence constraints has turned out to be much more challenging and remains limited. The mentioned $(2-1/m)$-approximation algorithm obtained by Graham in 1966, remains the algorithm of choice for this problem. The last progress in understanding whether it can or cannot be improved, was made over 30 years ago by Lenstra & Rinnooy Kan [29] who showed that it is NP-hard to approximate $P|prec|C_{max}$ within a factor less than 4/3. Closing this gap is mentioned as "Open Problem 1" in Woeginger and Schurman's list of ten open problems in scheduling theory [35]. Furthermore, the computational complexity of finding an exact solution to $Pm|prec, p_j = 1|C_{max}$ (the problem with a fixed number m of machines and unit processing times),

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.

open problem "OPEN8" from the original list of Garey and Johnson [16], is still open.

We remark that Du, Leung & Young [11] showed that $P2|prec|C_{max}$ is strongly NP-hard. Whereas, $P2|prec, p_j = 1|C_{max}$ is known to be polynomially solvable by techniques from matching theory (Fujii, Kasami & Ninomiya [13]). The best algorithm for problem $P3|prec, p_j = 1|C_{max}$ is a 4/3-approximation algorithm by Lam & Sethi [27][1] who analyzed an algorithm by Coffman & Graham [9]. In a recent paper [14], Gangal & Ranade showed that Graham's $(2 - 1/m)$-approximation algorithm can be improved to a $(2 - \frac{7}{3m+1})$-approximation algorithm in case of unit processing times and $m > 3$. Older results by Hu [23] and Garey, Johnson, Tarjan, & Yannakakis [17] give polynomial time algorithms in the case of unit processing times and special cases of precedence constraints. We refer the interested reader to the surveys [7] and [19] for further information on these and other algorithms for special cases of $P|prec|C_{max}$.

1.1 Our Results

We make the first progress in settling the approximability of $P|prec|C_{max}$ in the last 30 years: assuming a new possibly stronger version of the unique games conjecture (introduced by Bansal & Khot [4]), we show that it is NP-hard to approximate the scheduling problem $P|prec|C_{max}$ within any factor strictly less than 2, even in the case of unit processing times. The result is obtained by establishing a quite surprising connection between $P|prec|C_{max}$ and the scheduling problem of minimizing weighted completion time with precedence constraints ($1|prec|\sum w_jC_j$), studied in [4].

The problem $1|prec|\sum w_jC_j$ is another classical scheduling problem that has been studied since the 70's [36, 28, 30, 32]. Despite much interest, there was, until recently, a relatively large gap in our understanding of the approximability of $1|prec|\sum w_jC_j$. On the algorithmic side, several different techniques and linear programming formulations have been used to obtain 2-approximation algorithms for it (see e.g. [34, 6, 8, 31]). On the hardness side, before 2007, only NP-hardness for the exact problem was known [28, 30].

However, in contrast to $P|prec|C_{max}$, several recent results have lead to increased understanding of $1|prec|\sum w_jC_j$. In a series of papers [8, 10, 1] it was in fact established that $1|prec|\sum w_jC_j$ is a special case of weighted vertex cover; a result exploited to obtain improved approximation algorithms for special cases of precedence constraints [2]. Furthermore, new techniques have been used to obtain inapproximability results. Ambühl, Mastrolilli & the author [3] used the Quasi-random PCP due to Khot [25] to rule out the possibility of a polynomial time approximation scheme. More importantly, Bansal & Khot [4] settled the approximability of $1|prec|\sum w_jC_j$ assuming a certain conjecture. More specifically, they proved, assuming a new variant of the unique games conjecture (see Hypothesis 1 described in Appendix B), that it is NP-hard to approximate the value of an optimal schedule to $1|prec|\sum w_jC_j$ within a factor of $2 - \epsilon$ for any $\epsilon > 0$.

Our main result lets us profit from the hardness of $1|prec|\sum w_jC_j$ to obtain hardness of $P|prec|C_{max}$:

Theorem 1. *For any $\epsilon > 0$ and $\zeta \geq \zeta(\epsilon)$ where $\zeta(\epsilon)$ tends to 0 as ϵ tends to 0: if $1|prec|\sum w_jC_j$ has no $(2-\epsilon)$-approximation algorithm[2] then $P|prec|C_{max}$ has no $(2-\zeta)$-approximation algorithm, even in the case of unit processing times.*

The relation between the approximability of $1|prec|\sum w_jC_j$ and that of $P|prec|C_{max}$ is interesting on its own. In particular, even if the new variant of the unique games conjecture is false it might very well be that $1|prec|\sum w_jC_j$ (and thus $P|prec|C_{max}$ by the above theorem) is NP-hard to approximate within a factor $2 - \epsilon$, for any $\epsilon > 0$.

Proof overview 1. The proof is made in two steps. First in Section 2, we show that if $1|prec|\sum w_jC_j$ has no $(2-\epsilon)$-approximation algorithm then the following bipartite ordering problem is also hard: given an n by n bipartite graph $G(V, W, E)$ determine for a small $\zeta > 0$ whether

- (YES Case): G has a permutation $\pi : V \mapsto \{1, \ldots n\}$ such that for $i = 1, 2, \ldots, n$, the set
$$\{w \in W : \max_{\{v,w\}\in E} \pi(v) \leq i\}$$
has size at least $i - \zeta n$.
- (NO Case): For each permutation $\pi : V \mapsto \{1, \ldots n\}$, the set
$$\{w \in W : \max_{\{v,w\}\in E} \pi(v) \leq (1-\zeta)n\}$$
has size at most ζn.

The proof of the hardness of this problem is based on some new observations together with several older results regarding the structure of problem $1|prec|\sum w_jC_j$.

Second in Section 3, we provide a reduction from the bipartite graph problem to $P|prec|C_{max}$. The intuition of the reduction is the following. Let $\delta > \zeta$ be a small constant and Let $G(V, W, E)$ be an instance of the bipartite ordering problem. Consider the instance $\mathcal{I}$ of $P|prec|C_{max}$ with $(1-\delta)n$ machines, a job with processing time 1 for each vertex in G, and a precedence constraint from $v \in V$ to $w \in W$ if $\{v, w\} \in E$. Now suppose G is a YES instance, then there is a permutation π so that the set $\{w \in W : \max_{\{v,w\}\in E} \pi(v) \leq (1-\delta)n\}$ has size at least $(1-\delta)n - \zeta n \geq (1-2\delta)n$. This means that there is a schedule of $\mathcal{I}$ that completes the jobs corresponding to the first (according to π) $(1-\delta)n$ vertices in V at time 1; schedules the remaining jobs in V (at most δn many) and $(1-2\delta)n$ jobs in $\{w \in W : \max_{\{v,w\}\in E} \pi(v) \leq (1-\delta)n\}$ during time interval $[1, 2]$; and finally completes the remaining jobs in W at time 3. On the other hand if G is a NO instance, then it is easy to see that any schedule will have completed only a fraction ζ of the jobs in W at time 2. It follows that any schedule will need to schedule at least a fraction $1 - \zeta > 1 - \delta = m$ of the W-jobs from time 2 and will thus have makespan at least 4.

To amplify this "gap", we form a new instance $\mathcal{I}'$ that consists of several copies of $\mathcal{I}$ that are related by precedence constraints. For a hardness factor close to 2, a careful balance between the number of copies and the number of

[1] Also see [5] for a correction of an error in [27].

[2] Recall that an α-approximation algorithm is a polynomial time algorithm that approximates the value of an optimal solution within a factor of α.

machines is needed. With our current techniques ϵ must be significant smaller than ζ and as a result a strong hardness result (close to 2) is needed for $1|prec|\sum w_jC_j$ to obtain a good hardness result for $P|prec|C_{max}$. □

The above theorem together with the result of Bansal & Khot [4], gives us the following corollary.

COROLLARY 1. *Assuming a new variant of the unique games conjecture (Hypothesis 1), it is NP-hard to approximate $P|prec|C_{max}$ within a factor $(2-\zeta)$, for any $\zeta > 0$.*

2. BIPARTITE ORDERING AND SCHEDULING PROBLEM $1|\mathrm{PREC}|\sum \mathrm{W_J C_J}$

Problem $1|prec|\sum w_jC_j$ is the problem of scheduling a set N of n jobs on a single machine, which can process at most one job at a time. Each job $j \in N$ has a processing time p_j and a weight w_j. Jobs also have precedence constraints between them that are specified in the form of a partial order P on N. The goal is to find a schedule that minimizes $\sum_{j\in N} w_jC_j$, where C_j is the time at which job j completes in the given schedule. In this section we shall see that if $1|prec|\sum w_jC_j$ has no $(2-\epsilon)$-approximation algorithm for some $\epsilon > 0$, then there is no polynomial algorithm that distinguishes between certain bipartite graphs (see Theorem 2). This will then be used to prove our main result in Section 3.

Before presenting the result of this section we need the following definition.

Definition 1. For an n by n bipartite graph $G = (V, W, E)$ and a permutation $\pi : V \mapsto \{1, \dots, n\}$, define

$$V_i^\pi = \{v \in V : \pi(v) \le i\} \qquad \text{for } i = 1, \dots, n$$

$$W_i^\pi = \{w \in W : \max_{\{v,w\}\in E} \pi(v) \le i\} \quad \text{for } i = 0, 1, \dots, n.$$

THEOREM 2. *For any $\epsilon > 0$ and ζ such that $\zeta^2 > 2\epsilon$: if no polynomial algorithm approximates $1|prec|\sum w_jC_j$ within a factor of $(2-\epsilon)$, then there is no polynomial algorithm that distinguishes between n by n bipartite graphs*

- *that have a permutation π of V that satisfies*
$$|W_i^\pi| \ge i - \zeta n \qquad \text{for } i = 1, 2, \dots, n$$
- *and those for which each permutation π of V satisfies*
$$|W_{(1-\zeta)n}^\pi| \le \zeta n.$$

PROOF. Woeginger [37] proved that the general case of $1|prec|\sum w_jC_j$ is no harder to approximate than the following bipartite case:

1. The set N of jobs are partitioned into two sets A and B so that the set P of precedence constraints is a subset of $A \times B$.
2. The jobs in A have processing time 1 and weight 0 and the jobs in B have processing time 0 and weight 1.

We can thus restrict our attention to such bipartite instances. Structural results by Sidney [36] and algorithmic results by Lawler [28] imply that an α-approximation algorithm for bipartite instances that are so called non-Sidney-decomposable gives an α-approximation algorithm for all bipartite instances.[3] A bipartite instance is non-Sidney-decomposable if for each $A' \subseteq A$ the number of jobs in B with no predecessors in $A \backslash A'$ is at most $\frac{|A'|}{|A|}|B|$. We can thus further restrict ourselves to bipartite instances that are non-Sidney-decomposable.

When analyzing non-Sidney-decomposable bipartite instances it will be convenient to work with the so-called 2D Gantt chart, first introduced by Eastman et al. [12] and later revived by Goemans and Williamson [18] to give elegant proofs for various results related to $1|prec|\sum w_jC_j$. In a 2D Gantt chart, we have a horizontal axis of processing time and a vertical axis of weight. For a scheduling instance of the above form, the chart starts at point $(0, |B|)$ and ends at point $(|A|, 0)$. A job j is represented by a rectangle of length p_j and height w_j. Hence, a job of A is represented by a horizontal line of length 1 and a job of B is represented by a vertical line of length 1. Any schedule is represented in the 2D Gantt chart by placing the corresponding rectangles of the jobs in the order of the schedule such that the startpoint of a job is the endpoint of the previous job (or $(0, |B|)$ for the first job). The value $\sum_j w_jC_j$ of a schedule is then the area under the "work line" (see the shaded area in Figure 1-a).

Recall that a bipartite instance is non-Sidney-decomposable if for each $A' \subseteq A$ the number of jobs in B with no predecessors in $A \setminus A'$ is at most $\frac{|A'|}{|A|}|B|$. It follows that any schedule for such an instance will have its work line always above the diagonal in the 2D Gantt chart. Hence, any schedule of a non-Sidney-decomposable bipartite instance has value at least $|A||B|/2$. This was discovered independently in the general case by Chekuri and Motwani [6] and Margot et al. [31]; and later shown by Goemans and Williamson [18] using 2D Gantt charts.

As non-Sidney-decomposable bipartite instances are the hardest instances to approximate and any schedule of such an instance with job sets A and B has value at least $|A||B|/2$, we have the following. If there is no $(2-\epsilon)$-approximation algorithm for $1|prec|\sum w_jC_j$ then there is no polynomial algorithm that distinguishes between non-Sidney-decomposable bipartite instances with value $\frac{1+2\epsilon}{2}|A||B|$ and those with value $(1-\epsilon)|A||B|$, for any $\epsilon > 0$. Indeed, suppose there exists a polynomial time algorithm $\mathcal{A}$ that outputs "YES" if the scheduling instance has value at most $\frac{1+2\epsilon}{2}|A||B|$ and outputs "NO" if the scheduling instance has value at least $(1-\epsilon)|A||B|$. Then the following is a $(2-\epsilon)$-approximation algorithm: given a non-Sidney-decomposable bipartite instance with job sets A and B, run $\mathcal{A}$ on the instance and

- if $\mathcal{A}$ outputs "YES" then approximate the value of the instance to be $(1-\epsilon)|A||B|$ (since any schedule has value at least $|A||B|/2$ this is a $(2-2\epsilon)$-approximation);
- else approximate the value of the instance to be $|A||B|$ (since any schedule then has value at least $\frac{1+2\epsilon}{2}|A||B|$ this is a $\frac{2}{1+2\epsilon} \le (2-\epsilon)$-approximation).

Now let $\mathcal{I}$ be a non-Sidney-decomposable bipartite instance with the jobs partitioned into sets A and B and precedence constraints P. Let $n_A = |A|, n_B = |B|$, and $n = n_A n_B$. With $\mathcal{I}$ we associate an n by n bipartite graph $G(V, W, E)$ where there are n_B vertices in V for each $a \in A$ referred to as V_a, n_A vertices in W for each $b \in B$ referred to as W_b, and an edge between all vertices in V_a and all vertices in W_b if $(a, b) \in P$. In other words, G is the undirected graph of the precedence constraints where each job in A has

[3] We remark that we have restricted ourselves to bipartite instances of $1|prec|\sum w_jC_j$, but the results by Sidney and Lawler hold for general instances (see [18] for a nice exposition of these results).

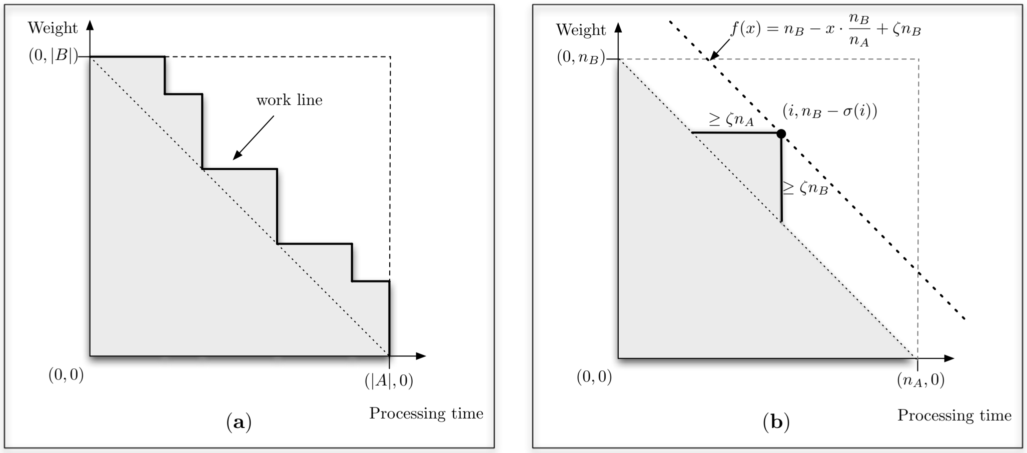

Figure 1: (a): An example of a 2D Gantt diagram. (b): An illustration of the argument used to prove Claim 1-a.

been copied n_B times and each job in B has been copied n_A times. The following claim completes the proof of the theorem.

CLAIM 1. *For any $\zeta > 0$ such that $\zeta^2 > 2\epsilon$:*

(a) If $\mathcal{I}$ has a schedule of value at most $\frac{1+2\epsilon}{2}n$ then G has a permutation π of V that satisfies

$$|W_i^\pi| \geq i - \zeta n \qquad \text{for } i = 1, 2, \ldots, n$$

(b) If all schedules of $\mathcal{I}$ have value at least $(1-\epsilon)n$ then each permutation π of V satisfies

$$|W_{(1-\zeta)n}^\pi| \leq \zeta n.$$

Proof of claim 1. **(a):** Let σ be a schedule of $\mathcal{I}$ with value at most $\frac{1+2\epsilon}{2}n$. For $i = 0, 1, 2, \ldots, n_A$, we let $\sigma(i)$ denote the number of jobs in B that have been scheduled when at most i jobs in A have completed. Now suppose toward contradiction that there exists an integer $i = 1, 2, \ldots, n_A$ such that $\sigma(i) \leq i\frac{n_B}{n_A} - \zeta n_B$ and consider the the 2D Gantt chart of σ. As σ will have its work line above both the diagonal and the point $(i, n_B - \sigma(i))$ on the dotted line(see Figure 1-b), the value of σ will be at least

$$n_B \cdot n_A/2 + \zeta n_A \cdot \zeta n_B/2 \quad = \quad \frac{1+\zeta^2}{2}n,$$

which is, by the choice of ζ, strictly greater than $\frac{1+2\epsilon}{2}n$. A contradiction since we assumed the value of σ to be at most $\frac{1+2\epsilon}{2}n$. Hence, for all $i = 1, 2, \ldots, n_A$ we have $\sigma(i) > i\frac{n_B}{n_A} - \zeta n_B$. Now let π be a permutation that orders all vertices in V_a before the vertices in $V_{a'}$ if σ schedules a before a'. Note that by the definition of G and π

$$|W_{i \cdot n_B}^\pi| \geq n_A \cdot \sigma(i) \qquad \text{for } i = 1, \ldots, n_A.$$

It follows that for $i = 1, 2, \ldots, n$

$$|W_i^\pi| \geq n_A \cdot \sigma(i/n_B) = i - \zeta n_A n_B = i - \zeta n.$$

We remarked that we assumed for simplicity that i/n_b evaluates to an integer. However, it is easy to see that a more careful analysis with $\lfloor i/n_b \rfloor$ will not have any impact on the final result.

(b): Assume all schedules of $\mathcal{I}$ have value at least $(1-\epsilon)n$ and suppose toward contradiction that there is a permutation π of V that satisfies $|W_{(1-\zeta)n}^\pi| > \zeta n$. Let σ be a schedule of $\mathcal{I}$ such that

1. for each $a, a' \in A$, σ schedules a before a' if $\max_{v \in V_a} \pi(v) \leq \max_{v' \in V_{a'}} \pi(v')$
2. and a job in B is scheduled as soon as its predecessors are finished.

Since $|W_{(1-\zeta)n}^\pi| > \zeta n$ at least ζn_B jobs in B are scheduled when at most $(1-\zeta)n_A$ jobs are completed in σ. It follows that the value of σ is at most

$$(1-\zeta)n_A \cdot \zeta n_B + n_A \cdot (1-\zeta)n_B = (1-\zeta^2)n.$$

A contradiction since $\zeta^2 > 2\epsilon$ and hence $(1-\zeta^2) < (1-\epsilon)$.

The proof of the above claim completed the proof of Theorem 2. □

3. PROOF OF MAIN RESULT

Here, we shall use Theorem 2 to establish the connection between $1|prec|\sum w_j C_j$ and $P|prec|C_{max}$. Given an n by n bipartite graph $G(V, W, E)$ and an integer d, we will construct an instance $\mathcal{I}(d)$ of $P|prec, p_j = 1|C_{max}$ such that for a small $\zeta > 0$ that only depends on d:

- *(Completeness)* If G has a permutation π of V that satisfies

  $$|W_i^\pi| \geq i - \zeta n \qquad \text{for } i = 1, 2, \ldots, n$$

  then $\mathcal{I}(d)$ has a schedule of makespan $d+1$.

- *(Soundness)* If each permutation π of V satisfies

  $$|W_{(1-\zeta)n}^\pi| \leq \zeta n.$$

  then any schedule of $\mathcal{I}(d)$ has makespan $2d$.

Theorem 1 then follows by combining Theorem 2 with the above reduction. We first present the reduction in Section 3.1 followed by the completeness and soundness analyses in Sections 3.2 and 3.3, respectively.

3.1 Construction

The final instance $\mathcal{I}(d)$ will consist of several copies of a set of precedence constrained jobs. This set will be referred to as the "building block" and is actually the only part of the construction that depends on G. The number of copies of the building block is carefully selected and precedence constraints between the copies are added to obtain the final instance $\mathcal{I}(d)$. We first present the building block, then we set the parameters that control the number of copies of the building block and the precedence constraints that are added between those copies. Finally, we present the instance $\mathcal{I}(d)$.

Building Block.

Let $\mathcal{B}$ denote the building block. The set of jobs is $V \cup W$ and there is a precedence constraint from $v \in V$ to $w \in W$ if $\{v, w\} \in E$. In addition the jobs in V, referred to as V-jobs, have processing time 1 and the jobs in W, referred to as W-jobs, have processing time 0. We remark that the jobs with processing time 0 — the W-jobs — are superfluous in the sense that removing them will not change the length of any schedule. Instead, they are included because they make the exposition much cleaner.

Parameters.

Before describing instance $\mathcal{I}(d)$ we need to define some parameters. Select $\gamma_d = \frac{1}{10}$ and let $\gamma_{\ell-1} = \gamma_\ell^{10}$ for $\ell = d, d-1 \ldots, 1$. Furthermore, let $\delta_{\ell+1} = 1/\gamma_\ell^5$ for $\ell = 1, 2, \ldots, d$. Note that $\delta_2, \ldots, \delta_{d+1}, 1/\gamma_1, \ldots, 1/\gamma_d$ are all integers. The parameters are also selected such that the following bounds hold (they will be used in the completeness and soundness analyses).

Lemma 1. *For each* $\ell = 1, \ldots, d$

$$(1 - 4\gamma_{\ell-1})^{\delta_{\ell+1}} \geq 1 - 2\gamma_\ell \tag{1}$$

$$\left(1 - \frac{\gamma_\ell}{4}\right)^{\delta_{\ell+1}} \leq \gamma_\ell/2 \tag{2}$$

The proof of the above bounds is straightforward and can be found in Appendix A. Finally, we select

$$\zeta = \gamma_0^{10}. \tag{3}$$

Instance.

Instance $\mathcal{I}(d)$ has m machines (selected below (4)) and consists of several copies of $\mathcal{B}$ related to each other by precedence constraints. Conceptually, it will be useful to partition these copies into d *layers*. The copies in layer $\ell = 1, 2, \ldots, d$ will then in turn be subdivided into several *groups*. We will denote the set of groups in layer ℓ by $\mathcal{L}_\ell$. We continue by a formal definition of the layers and will then define the precedence constraints between the layers (see also Figure 2 for an example of the construction):

- Layer 1 consists of a set
$$\mathcal{L}_1 = \{B^{(1,i)} : i \in \{1, \ldots, n\}\}$$
of $|\mathcal{L}_1| = n$ groups and each group $B^{(1,i)} \in \mathcal{L}_1$ contains $(1 + \gamma_1)\frac{m}{n \cdot |\mathcal{L}_1|}$ copies of $\mathcal{B}$.
- Layer $\ell = 2, 3, \ldots, d$ consists of a set
$$\mathcal{L}_\ell = \left\{ B^{(\ell,i,X)} : \begin{array}{c} i \in \{1, \ldots, n\}, \\ \text{X is a multi-set consisting of} \\ \delta_\ell \text{ groups from } \mathcal{L}_{\ell-1} \end{array} \right\}$$
of $|\mathcal{L}_\ell| = n \cdot |\mathcal{L}_{\ell-1}|^{\delta_\ell}$ groups and each group $B^{(\ell,i,X)} \in \mathcal{L}_\ell$ contains $(1 + \gamma_\ell)\frac{m}{n \cdot |\mathcal{L}_\ell|}$ copies of $\mathcal{B}$.

Note that the number of V-jobs and the number of W-jobs in layer $\ell = 1, \ldots, d$ are both $(1 + \gamma_\ell)m$.

We are now ready to define the precedence constraints between the different layers. For a group $B^{(\ell,i,X)} \in \mathcal{L}_\ell$ with $\ell \geq 2$, there are precedence constraints from all the jobs that correspond to w_i in the groups $\mathcal{L}_{\ell-1} \cap X$ to all the jobs in $B^{(\ell,i,X)}$.

Finally, we select the number of machines so as to guarantee that $(1 + \gamma_\ell)\frac{m}{n|\mathcal{L}_\ell|}$ is an integer for $\ell = 1, \ldots, d$. As $|\mathcal{L}_\ell|$ divides $|\mathcal{L}_d|$ for $\ell = 1, \ldots, d$, we let

$$m = \left(\prod_{i=0}^{d} \frac{1}{\gamma_i}\right) \cdot n \cdot |\mathcal{L}_d| = \left(\prod_{i=0}^{d} \frac{1}{\gamma_i}\right) \cdot n \cdot n^{1+\sum_{i=2}^{d} \prod_{j=i}^{d} \delta_j} \tag{4}$$

Note that m is a polynomial in n whose degree only depends on d and since layer $\ell = 1, 2, \ldots, d$ contains $(1+\gamma_\ell)m$ V-jobs and $(1 + \gamma_\ell)m$ W-jobs the construction is polynomial in n for any fixed d.

3.2 Completeness

Let π be a permutation of V that satisfies

$$|W_i^\pi| \geq i - \zeta n \qquad \text{for } i = 1, 2, \ldots, n \tag{5}$$

and let

$$t_i = \lfloor (1 - \gamma_i)n \rfloor \text{ for } i = 1, 2, \ldots, d.$$

The key lemma for the completeness is the following:

Lemma 2. *For each* $\ell = 1, \ldots, d$ *there exists a schedule* σ_ℓ *of all the jobs in layers* $1, \ldots, \ell$ *satisfying:*
for each $k = 1, \ldots, \ell$

(i) in at least a fraction $1 - 4\gamma_{k-1}$ *of layer* k*'s groups, all jobs corresponding to the vertices in* $V_{t_k}^\pi \cup W_{t_k}^\pi$ *are scheduled during time interval* $[k - 1, k]$*;*

(ii) the set of remaining jobs of layer k *that were not scheduled during* $[k-1, k]$ *contains at most* $4\gamma_k m$ V*-jobs and are scheduled during time interval* $[k, k + 1]$*.*

As σ_d will schedule all the jobs and have makespan $d + 1$, this will imply the completeness analysis.

Proof. We proceed by induction on ℓ.

Base Case.

For $\ell = 1$, consider schedule σ_1 defined as follows. During time interval

$[0, 1]$: σ_1 first schedules the jobs in layer 1 that correspond to the vertices in $V_{t_1}^\pi$ followed (at time 1) by the jobs that correspond to the vertices in $W_{t_1}^\pi$.

$[1, 2]$: σ_1 first schedules the remaining V-jobs of layer 1 followed (at time 2) by the remaining W-jobs of layer 1.

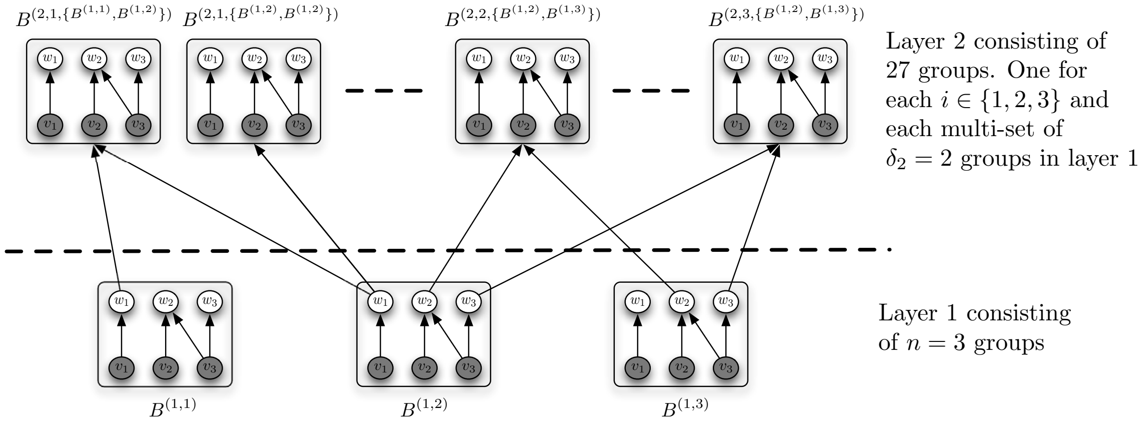
$B^{(2,1,\{B^{(1,1)},B^{(1,2)}\})}$
$B^{(2,1,\{B^{(1,2)},B^{(1,2)}\})}$
$B^{(2,2,\{B^{(1,2)},B^{(1,3)}\})}$
$B^{(2,3,\{B^{(1,2)},B^{(1,3)}\})}$
w_1
w_2
w_3
v_1
v_2
v_3
Layer 2 consisting of 27 groups. One for each $i \in \{1,2,3\}$ and each multi-set of $\delta_2 = 2$ groups in layer 1
Layer 1 consisting of $n = 3$ groups
$B^{(1,1)}$
$B^{(1,2)}$
$B^{(1,3)}$

Figure 2: An example of the groups in $\mathcal{I}(2)$ with $\delta_2 = 2$ arising from the bipartite graph with $V = \{v_1, v_2, v_3\}$, $W = \{w_1, w_2, w_3\}$, and $E = \{\{v_1, w_1\}, \{v_2, w_2\}, \{v_3, w_3\}, \{v_3, w_2\}\}$. Each rectangle represents a group that contains a layer dependent number of copies of the building block. The V-jobs and W-jobs are depicted in gray and white, respectively.

Before showing that σ_1 satisfies conditions (i) and (ii), we prove that σ_1 is a feasible schedule, i.e., no precedence constraints are violated and σ_1 does not schedule more than m jobs at any time. Note that there might only be a precedence constraint between two jobs in layer 1 if they correspond to two vertices $v \in V$ and $w \in W$ with $\{v, w\} \in E$. By Definition 1 there are no edges between the vertices in $V \setminus V^{\pi}_{t_1}$ and the vertices in $W^{\pi}_{t_1}$. It follows that no precedence constraints are violated by σ_1. To show σ_1 feasible, it remains to verify that σ_1 does not schedule more jobs than machines at any time. As the W-jobs have 0 processing time, they can be scheduled sequentially without increasing the makespan. It is thus sufficient to verify that no more than m V-jobs are scheduled at any time. During time interval $[0, 1]$, σ_1 schedules a fraction $t_1/n \leq (1 - \gamma_1)$ of the V-jobs. The total number of V-jobs scheduled during $[0, 1]$ is thus at most $(1 - \gamma_1)(1 + \gamma_1)m = (1 - \gamma_1^2)m \leq m$. Similarly, during time interval $[1, 2]$, σ_1 schedules the remaining $1 - t_1/n \leq 2\gamma_1$ fraction of the V-jobs. The total number of V-jobs scheduled during $[1, 2]$ is thus at most $2\gamma_1 m$.

We shall now see that σ_1 satisfies conditions (i) and (ii). Since we have proved that the remaining jobs scheduled by σ_1 during time interval $[1, 2]$ contains at most $2\gamma_1 m$ V-jobs, we know that σ_1 satisfies condition (ii) in the induction hypothesis. To see that σ_1 satisfies condition (i), it is sufficient to observe that in all layer 1's groups, all jobs corresponding to the vertices in $V^{\pi}_{t_1} \cup W^{\pi}_{t_1}$ are scheduled during time interval $[0, 1]$. This completes the proof of the base case.

Inductive Step.

Suppose we already defined schedules $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ that satisfy conditions (i) and (ii), for $\ell < d$. We shall now define a schedule $\sigma_{\ell+1}$ that schedules all jobs in layers $1, 2, \ldots, \ell+1$ and satisfies conditions (i) and (ii).

Before defining schedule $\sigma_{\ell+1}$ we need to introduce some new notation. A group $B \in \mathcal{L}_\ell$ is called *early* if σ_ℓ has completed all the jobs in B that correspond to the vertices in $V^{\pi}_{t_\ell} \cup W^{\pi}_{t_\ell}$ at time ℓ. Let

$$\mathcal{L}^{no\text{-}pre}_{\ell+1} = \left\{ B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1} : \begin{array}{l} w_i \in W^{\pi}_{t_\ell} \text{ and} \\ \text{each } B \in X \text{ is early} \end{array} \right\}.$$

Note that $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ is defined so that σ_ℓ has scheduled all the predecessors of the V-jobs in $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ at time ℓ.

We are now ready to define $\sigma_{\ell+1}$. Schedule $\sigma_{\ell+1}$ is an extension of σ_ℓ in the sense that $\sigma_{\ell+1}$ schedules all the jobs in layers $1, 2, \ldots, \ell$ in the same way as σ_ℓ. The jobs in layer $\ell + 1$ are scheduled by $\sigma_{\ell+1}$ as follows. During time interval

$[\ell, \ell + 1]$: $\sigma_{\ell+1}$ first schedules the jobs in $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ that correspond to the vertices in $V^{\pi}_{t_{\ell+1}}$ followed (at time $\ell + 1$) by the jobs in $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ that correspond to the vertices in $W^{\pi}_{t_{\ell+1}}$.

$[\ell + 1, \ell + 2]$: $\sigma_{\ell+1}$ first schedules the remaining V-jobs of layer $\ell + 1$ followed (at time $\ell + 2$) by the remaining W-jobs of layer 1.

As in the base case we start by showing that $\sigma_{\ell+1}$ is feasible. The proof that $\sigma_{\ell+1}$ satisfies conditions (i) and (ii) will then follow in a straightforward manner.

To show that $\sigma_{\ell+1}$ is feasible we need to prove that no precedence constraints are violated and the number of scheduled jobs does not exceed the number of machines m at any time.

We start by showing that no precedence constraints are violated. Since σ_ℓ does not violate any precedence constraints between jobs of layers $1, 2, \ldots, \ell$ neither does $\sigma_{\ell+1}$. Now consider the precedence constraints where at least one job is in layer $\ell + 1$. As already noted, σ_ℓ and thus $\sigma_{\ell+1}$ has finished all the predecessors of the V-jobs in $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ at time ℓ. Similarly, by condition (ii) in the induction hypothesis σ_ℓ and thus $\sigma_{\ell+1}$ has finished all the predecessor of all V-jobs in layer $\ell + 1$ at time $\ell + 1$. It follows that no precedence constraint from a job in a different layer than $\ell + 1$ to a job in layer $\ell + 1$ is violated. Now consider the precedence constraints between jobs in layer $\ell+1$. As in the base case, there might only be a precedence constraint between two jobs in the same layer if they correspond to two vertices $v \in V$ and

$w \in W$ with $\{v, w\} \in E$. By Definition 1 there are no edges between the vertices in $V \setminus V^{\pi}_{t_{\ell+1}}$ and the vertices in $W^{\pi}_{t_{\ell+1}}$. Hence, no precedence constraints are violated.

We continue by verifying that $\sigma_{\ell+1}$ does not schedule more than m jobs at any time. As noted in the base case, it is sufficient to verify that no more than m V-jobs are scheduled at any time. Since $\sigma_{\ell+1}$ and σ_ℓ do not differ during time interval $[0, \ell]$, we only need to verify $\sigma_{\ell+1}$ during time intervals $[\ell, \ell+1]$ and $[\ell+1, \ell+2]$. By condition (ii) in the induction hypothesis, σ_ℓ and thus $\sigma_{\ell+1}$ schedules at most $4\gamma_\ell m$ V-jobs from layer ℓ during time interval $[\ell, \ell+1]$. In addition $\sigma_{\ell+1}$ schedules a fraction $\frac{t_{\ell+1}}{n} \cdot \frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|}$ of the V-jobs from layer $\ell+1$ during $[\ell, \ell+1]$. The total number of V-jobs scheduled by $\sigma_{\ell+1}$ during $[\ell, \ell+1]$ is thus at most

$$4\gamma_\ell m + \frac{t_{\ell+1}}{n} \cdot \frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|} \cdot (1 + \gamma_{\ell+1})m. \tag{6}$$

During interval $[\ell+1, \ell+2]$, $\sigma_{\ell+1}$ schedules the remaining V-jobs from layer $\ell+1$. The number of such jobs is thus

$$\left(1 - \frac{t_{\ell+1}}{n} \cdot \frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|}\right)(1 + \gamma_{\ell+1})m. \tag{7}$$

The following claim completes the proof that $\sigma_{\ell+1}$ is feasible.

Claim 2. *We have that* (6) $\leq m$ *and* (7) $\leq 4\gamma_{\ell+1} m$.

Proof of claim 2. We start with proving (6) $\leq m$. As $\frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|} \leq 1$ and $\frac{t_{\ell+1}}{n} \leq 1 - \gamma_{\ell+1}$, we have that

$$(6) \leq 4\gamma_\ell + (1 - \gamma_{\ell+1})(1 + \gamma_{\ell+1})m \leq m,$$

where we used that $4\gamma_\ell = 4\gamma^{10}_{\ell+1} \leq \gamma^2_{\ell+1}$ for the last inequality.

To prove (7) $\leq 4\gamma_{\ell+1} m$ we need to bound the size of $\frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|}$. Recall that

$$\mathcal{L}^{no\text{-}pre}_{\ell+1} = \left\{ B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1} : \begin{array}{c} w_i \in W^{\pi}_{t_\ell} \text{ and} \\ \text{each } B \in X \text{ is early} \end{array} \right\}.$$

We can thus write $\frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|}$ as

$$\Pr_{B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1}} [w_i \in W^{\pi}_{t_\ell} \text{ and each } B \in X \text{ is early}],$$

which in turn equals

$$\frac{|W^{\pi}_{t_\ell}|}{|W|} \cdot \left(\Pr_{B \in \mathcal{L}_\ell}[B \text{ is early}]\right)^{\delta_{\ell+1}}.$$

By condition (i) in the induction hypothesis,

$$\Pr_{B \in \mathcal{L}_\ell}[B \text{ is early}] \geq 1 - 4\gamma_{\ell-1}$$

and by assumption (5) and the fact that $\zeta = \gamma_0^{10} << \gamma_\ell$ we have

$$\frac{|W^{\pi}_{t_\ell}|}{|W|} \geq \frac{\lfloor (1 - \gamma_\ell)n \rfloor - \zeta n}{n} \geq 1 - 2\gamma_\ell.$$

Substituting in these bounds gives us that

$$\begin{aligned} \frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|} &\geq (1 - 2\gamma_\ell) \cdot (1 - 4\gamma_{\ell-1})^{\delta_{\ell+1}} \\ &\geq (1 - 2\gamma_\ell)^2 \qquad \text{(by (1) in Lemma 1)} \\ &\geq 1 - 4\gamma_\ell \end{aligned}$$

We are now ready to prove (7) $\leq 4\gamma_{\ell+1} m$:

$$\begin{aligned} (7) &= \left(1 - \frac{t_{\ell+1}}{n} \cdot \frac{|\mathcal{L}^{no\text{-}pre}_{\ell+1}|}{|\mathcal{L}_{\ell+1}|}\right)(1 + \gamma_{\ell+1})m \\ &\leq (1 - (1 - 2\gamma_{\ell+1}) \cdot (1 - 4\gamma_\ell))(1 + \gamma_{\ell+1})m \\ &= (4\gamma_\ell + 2\gamma_{\ell+1} - 8\gamma_\ell\gamma_{\ell+1})(1 + \gamma_{\ell+1})m \\ &\leq (4\gamma_\ell + 2\gamma_{\ell+1})(1 + \gamma_{\ell+1})m \\ &= (4\gamma^{10}_{\ell+1} + 2\gamma_{\ell+1})(1 + \gamma_{\ell+1})m \\ &\leq 4\gamma_{\ell+1} m \qquad (\text{since } \gamma_{\ell+1} \leq 1/10) \end{aligned}$$

We have thus proved that (6) $\leq m$ and (7) $\leq 4\gamma_{\ell+1} m$ as required. □

We continue by proving that $\sigma_{\ell+1}$ satisfies conditions (i) and (ii) of the induction hypothesis. Since $\sigma_{\ell+1}$ is an extension of σ_ℓ, conditions (i) and (ii) are satisfied for $k = 1, 2, \ldots, \ell$. We continue by verifying the induction hypothesis for $k = \ell + 1$. To see that $\sigma_{\ell+1}$ satisfies condition (i), note that $\sigma_{\ell+1}$ schedules all jobs in $\mathcal{L}^{no\text{-}pre}_{\ell+1}$ that correspond to the vertices in $V^{\pi}_{t_{\ell+1}} \cup W^{\pi}_{t_{\ell+1}}$ during time interval $[\ell, \ell+1]$. As $\Pr_{B \in \mathcal{L}_{\ell+1}}[B \in \mathcal{L}^{no\text{-}pre}_{\ell+1}] \geq 1 - 4\gamma_\ell$ (see proof of the claim above), $\sigma_{\ell+1}$ satisfies condition (i). Now consider condition (ii). Obviously $\sigma_{\ell+1}$ schedules all the remaining jobs of layer $\ell+1$ during time interval $[\ell+1, \ell+2]$. Furthermore, as proved above, the set of remaining jobs contains (7) V-jobs with (7) $\leq 4\gamma_{\ell+1} m$.

By the above arguments the schedule $\sigma_{\ell+1}$ is feasible and satisfies conditions (i) and (ii). This concludes the inductive step and the completeness analysis. □

3.3 Soundness

By assumption we have that each permutation π of V satisfies

$$|W^{\pi}_{(1-\zeta)n}| \leq \zeta n. \tag{8}$$

Fix any feasible schedule σ. The key lemma for soundness is the following:

Lemma 3. *For each $\ell = 1, \ldots, d$, schedule σ has at time $2 \cdot (\ell - 1)$ completed at most a fraction $\gamma_{\ell-1}$ of the V-jobs in layer ℓ.*

As there are $(1 + \gamma_d)m$ V-jobs in layer d and $\gamma_d > \gamma_{d-1}$, this implies that the schedule must have makespan at least $2(d-1) + 2 = 2d$.

Proof. We proceed by induction on ℓ.

Base case.

For $\ell = 1$ the induction hypothesis is obviously true, because σ has completed no jobs at time 0.

Inductive Step.

Suppose we already proved the induction hypothesis for $1, 2, \ldots, \ell$ for $\ell < d$. We shall now prove that at time 2ℓ, σ

has completed at most a fraction γ_ℓ of the V-jobs in layer $\ell + 1$.

Let $t = 2(\ell - 1)$. For each $i = 1, \dots, n$ and each $B \in \mathcal{L}_\ell$, let $X_i^{(B)}$ be the indicator variable defined

$$X_i^{(B)} = \begin{cases} 1, & \text{If at time } t+1,\ \sigma \text{ has completed all the jobs} \\ & \text{in group } B \text{ that correspond to } w_i \in W. \\ 0, & \text{Otherwise.} \end{cases}$$

Moreover, for $i \in \{1, \dots, n\}$, let X_i denote the indicator variable that takes value 1 if $\mathbb{E}_{B \in \mathcal{L}_\ell}[X_i^{(B)}] \geq 1 - \frac{\gamma_\ell}{4}$ and 0 otherwise.

Claim 3. *We have* $\mathbb{E}_i[X_i] \leq 4\zeta$.

Proof of claim 3. Suppose toward contradiction that $\mathbb{E}_i[X_i] > 4\zeta$ and hence there are more than a fraction 4ζ of the indicator variables that equals 1. Let $a = \lceil 4\zeta n \rceil$ and assume without loss of generality that $X_1 = X_2 = \cdots = X_a = 1$. By the definition of X_i's,

$$\mathbb{E}_{i \in [a], B \in \mathcal{L}_\ell}[X_i^{(B)}] \geq 1 - \frac{\gamma_\ell}{4}.$$

It follows that

$$\Pr_{B \in \mathcal{L}_\ell}[E_{i \in [a]}[X_i^{(B)}] \geq 1/2] \geq 1 - \frac{\gamma_\ell}{2}.$$

By letting the expectation be over $[n]$ instead of $[a]$,

$$\Pr_{B \in \mathcal{L}_\ell}[E_{i \in [n]}[X_i^{(B)}] \geq 2\zeta] \geq 1 - \frac{\gamma_\ell}{2}.$$

Call a group $B \in \mathcal{L}_\ell$ *good* if $\mathbb{E}_{i \in [n]}[X_i^{(B)}] \geq 2\zeta$. By the inequality above, at least a fraction $1 - \frac{\gamma_\ell}{2}$ of the groups in $\mathcal{L}_\ell$ are good.

Now consider a good group $B \in \mathcal{L}_\ell$. As $\mathbb{E}_{i \in [n]}[X_i^{(B)}] \geq 2\zeta$, σ schedules all jobs in B that correspond to a subset $W' \subseteq W$ of vertices with $|W'| \geq 2\zeta n$. By assumption (8), σ must have completed more than a fraction $1 - \zeta$ of the V-jobs in B before all jobs corresponding to W' are scheduled. Hence, we can conclude that for each good block σ has completed at least a fraction $1 - \zeta$ of the V-jobs. As layer ℓ consists of $(1 + \gamma_\ell)m$ V-jobs and at least a fraction $1 - \gamma_\ell/2$ of them belong to good groups, the total number of the V-jobs completed by σ at time $t + 1$ is at least $(1 + \gamma_\ell)(1 - \gamma_\ell/2)(1 - \zeta)m$. However this leads to the following contradiction. By the induction hypothesis, σ has completed at most a fraction $\gamma_{\ell-1}$ of the V-jobs in layer ℓ at time t. Thus during time interval $[t, t + 1]$, the number of V-jobs σ must schedule is at least

$$(1 + \gamma_\ell)(1 - \gamma_\ell/2)(1 - \zeta)m - \gamma_{\ell-1}(1 + \gamma_\ell)m,$$

which is strictly greater than m since $(1 + \gamma_\ell)(1 - \gamma_\ell/2) \geq 1 + \gamma_\ell/4$, $\zeta = \gamma_0^{10} << \gamma_\ell$, and $\gamma_{\ell-1} = \gamma_\ell^{10} << \gamma_\ell$. A contradiction. □

Without loss of generality, let $X_1, X_2, \dots, X_k$ be the indicator variables with value 1. By the claim above we have that $k \leq 4\zeta$. A group $B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1}$ can be scheduled at time $t + 1$ if all of its predecessors have been completed at that time. All predecessors of $B^{(\ell+1,i,X)}$ have been completed at time $t + 1$ if $X_i^{(B)} = 1$ for each $B \in X$. We can thus write the fraction of groups in $\mathcal{L}_{\ell+1}$ for which all predecessors have been completed at time $t + 1$ as

$$\Pr_{B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1}}[\forall B \in X, X_i^{(B)} = 1], \qquad (9)$$

which is at most the sum of $\Pr_{B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1}}[i \leq k]$ and $\Pr_{B^{(\ell+1,i,X)} \in \mathcal{L}_{\ell+1}}[i > k \text{ and } \forall B \in X, X_i^{(B)} = 1]$. As $k \leq 4\zeta$ and $\mathbb{E}_B[X_i^{(B)}] < 1 - \frac{\gamma_\ell}{4}$ for $i > k$, we have that (9) is bounded from above by

$$\begin{aligned} 4\zeta + \left(1 - \frac{\gamma_\ell}{4}\right)^{|X|} &= 4\zeta + \left(1 - \frac{\gamma_\ell}{4}\right)^{\delta_{\ell+1}} \\ &\leq 4\zeta + \gamma_\ell/2 \text{ (by (2) in Lemma 1)} \\ &\leq \gamma_\ell \qquad \text{(since } \zeta = \gamma_0^{10} < \gamma_\ell/8) \end{aligned}$$

Hence, σ has at time $t + 2 = 2\ell$ completed at most a fraction γ_ℓ of the V-jobs in layer $\ell + 1$. This concludes the inductive step and the soundness analysis.

4. ACKNOWLEDGEMENTS

I would like to thank Johan Håstad for many useful comments on an early version of this paper. I would also like to thank Monaldo Mastrolilli for suggesting to do the reduction to $P|prec|C_{max}$ from $1|prec|\sum w_j C_j$ instead of from the new variant of the unique games conjecture.

This research is supported by ERC Advanced investigator grant 226203 and the Swiss National Science Foundation project "Approximation Algorithms for Machine Scheduling Through Theory and Experiments III" Project N. 200020-122110/1.

5. REFERENCES

[1] C. Ambühl and M. Mastrolilli. Single machine precedence constrained scheduling is a vertex cover problem. *Algorithmica*, 53(4):488–503, 2009.

[2] C. Ambühl, M. Mastrolilli, N. Mutsanas, and O. Svensson. Scheduling with precedence constraints of low fractional dimension. In *Proceedings of the 12th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 130–144, 2007.

[3] C. Ambühl, M. Mastrolilli, and O. Svensson. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constrained scheduling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 329–337, 2007.

[4] N. Bansal and S. Khot. Optimal long code test with one free bit. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 453–462, 2009.

[5] B. Braschi and D. Trystram. A new insight into the coffman-graham algorithm. *SIAM Journal on Computing*, 23(3):662–669, 1994.

[6] C. Chekuri and R. Motwani. Precedence constrained scheduling to minimize sum of weighted completion times on a single machine. *Discrete Applied Mathematics*, 98(1-2):29–38, 1999.

[7] B. Chen, C. N. Potts, and G. J. Woeginger. A review of machine scheduling: Complexity, algorithms and approximability. *Handbook of Combinatorial Optimization*, 3:21–169, 1998.

[8] F. A. Chudak and D. S. Hochbaum. A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. *Operations Research Letters*, 25:199–204, 1999.

[9] E. Coffman and R. Graham. Optimal scheduling for two-processor systems. *Acta Informatica*, 1:200–213, 1972.

[10] J. R. Correa and A. S. Schulz. Single machine scheduling with precedence constraints. *Mathematics of Operations Research*, 30(4):1005–1021, 2005.

[11] J. Du, J. Y.-T. Leung, and G. H. Young. Scheduling chain-structured tasks to minimize makespan and mean flow time. *Information and Computation*, 92(2):219–236, 1991.

[12] W. L. Eastman, S. Even, and I. M. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11(2):268–279, 1964.

[13] M. Fujii, T. Kasami, and K. Ninomiya. Optimal sequencing of two equivalent processors. *SIAM Journal on Applied Mathematics*, 17(4):784–789, 1969.

[14] D. Gangal and A. Ranade. Precedence constrained scheduling in $(2 - 73p + 1)\cdot$optimal. *Journal of Computer and System Sciences*, 74(7):1139–1146, 2008.

[15] M. R. Garey and D. S. Johnson. “ strong ” np-completeness results: Motivation, examples, and implications. *Journal of the ACM*, 25(3):499–508, 1978.

[16] M. R. Garey and D. S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness*. W.H. Freeman, 1979.

[17] M. R. Garey, D. S. Johnson, R. E. Tarjan, and M. Yannakakis. Scheduling opposing forests. *SIAM Journal on Algebraic and Discrete Methods*, 1(4):72–93, 1983.

[18] M. X. Goemans and D. P. Williamson. Two-dimensional Gantt charts and a scheduling algorithm of Lawler. *SIAM Journal on Discrete Mathematics*, 13(3):281–294, 2000.

[19] R. Graham, E. Lawler, J. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[20] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal (BSTJ)*, 45:1563–1581, 1966.

[21] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969.

[22] D. S. Hochbaum, , and D. B. Shmoys. Using dual approximation algorithms for scheduling problems theoretical and practical results. *Journal of the ACM*, 34(1):144–162, 1987.

[23] T. C. Hu. Parallel Sequencing and Assembly Line Problems. *Operations Research*, 9(6):841–848, 1961.

[24] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th annual ACM symposium on Theory of computing (STOC)*, pages 767–775, 2002.

[25] S. Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

[26] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2 - \varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[27] S. Lam and R. Sethi. Worst case analysis of two scheduling algorithms. *SIAM Journal on Computing*, 6(3):518–536, 1977.

[28] E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2:75–90, 1978.

[29] J. K. Lenstra and A. H. G. R. Kan”. Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35, 1978.

[30] J. K. Lenstra and A. H. G. R. Kan. The complexity of scheduling under precedence constraints. *Operations Research*, 26:22–35, 1978.

[31] F. Margot, M. Queyranne, and Y. Wang. Decompositions, network flows and a precedence constrained single machine scheduling problem. *Operations Research*, 51(6):981–992, 2003.

[32] C. N. Potts. An algorithm for the single machine sequencing problem with precedence constraints. *Mathematical Programming Study*, 13:78–87, 1980.

[33] S. K. Sahni. Algorithms for scheduling independent tasks. *Journal of the ACM*, 23(1):116–127, 1976.

[34] A. S. Schulz. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. In *Proceedings of the 5th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 301–315, 1996.

[35] P. Schuurman and G. J. Woeginger. Polynomial time approximation algorithms for machine scheduling: ten open problems. *Journal of Scheduling*, 2(5):203–213, 1999.

[36] J. B. Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 23:283–298, 1975.

[37] G. J. Woeginger. On the approximability of average completion time scheduling under precedence constraints. *Discrete Applied Mathematics*, 131(1):237–252, 2003.

APPENDIX

A. PROOF OF LEMMA 1

Recall that the parameters were selected to be $\gamma_d = \frac{1}{10}$, $\gamma_{\ell-1} = \gamma_\ell^{10}$ and $\delta_{\ell+1} = 1/\gamma_\ell^5$ for $\ell = 1, \ldots, d$. To prove the bounds we shall use Bernoulli's inequality:

$$1+nx \leq (1+x)^n \qquad \text{for all } x \in [-1, \infty] \text{ and for all } n = 1, 2, \ldots$$

Bound (1),i.e., $(1-4\gamma_{\ell-1})^{\delta_{\ell+1}} \geq 1 - 2\gamma_\ell$ now follows trivially. Indeed by Bernoulli's inequality and the selection of parameters we have

$$\begin{aligned}(1-4\gamma_{\ell-1})^{\delta_{\ell+1}} &\geq 1 - 4\gamma_{\ell-1}\cdot\delta_{\ell+1}\\ &= 1 - 4\gamma_\ell^{10}\cdot 1/\gamma_\ell^5\\ &= 1-4\gamma_\ell^5\\ &\geq 1-2\gamma_\ell\end{aligned}$$

We continue by showing Bound (2), i.e., $\left(1-\frac{\gamma_\ell}{4}\right)^{\delta_{\ell+1}} \leq \gamma_\ell/2$:

$$\begin{aligned}\left(1-\frac{\gamma_\ell}{4}\right)^{\delta_{\ell+1}} &= \left(\frac{4/\gamma_\ell - 1}{4/\gamma_\ell}\right)^{\delta_{\ell+1}}\\ &= \frac{1}{\left(\frac{4/\gamma_\ell}{4/\gamma_\ell - 1}\right)^{\delta_{\ell+1}}}\\ &= \frac{1}{\left(1+\frac{1}{4/\gamma_\ell - 1}\right)^{\delta_{\ell+1}}}\\ &\leq \frac{1}{1+\frac{\delta_{\ell+1}}{(4/\gamma_\ell - 1)}} \qquad \text{(by Bernoulli's inequality)}\\ &\leq \frac{4}{\gamma_\ell\cdot\delta_{\ell+1}}\\ &= \frac{4}{\gamma_\ell\cdot 1/\gamma_\ell^5}\\ &\leq \gamma_\ell/2\end{aligned}$$

□

B. THE NEW STRONGER UNIQUE GAMES CONJECTURE

Although we do not directly use the new stronger version of the unique games conjecture, we define it here for the sake of completeness. An instance of unique games $\mathcal{L} = (G(V, W, E), [n], \{\pi_{v,w}\}_{(v,w)})$ consists of a regular bipartite graph $G(V, W, E)$ and a set $[n]$ of labels. For each edge $(v, w) \in E$ there is a constraint specified by a permutation $\pi_{v,w} : [n] \mapsto [n]$. The goal is to find a labeling $\ell : (V \cup W) \mapsto [n]$ so as to maximize $val(\ell) := \Pr_{e\in E}[\ell \text{ satisfies } e]$, where a labeling ℓ is said to satisfy an edge $e = (v, w)$ if $\ell(v) = \pi_{v,w}(\ell(w))$. For a unique game instance $\mathcal{L}$, we let $OPT(\mathcal{L}) = \max_{\ell:V\cup W\mapsto[n]} val(\ell)$. The now famous unique games conjecture that has been extensively used to prove strong hardness of approximation results can be stated as follows.

Conjecture 1 ([24]). *For any constants $\zeta, \gamma > 0$, there is a sufficiently large constant $n = n(\zeta, \gamma)$ such that, for unique game instances $\mathcal{L}$ with label set $[n]$, it is NP-hard to distinguish between $OPT(\mathcal{L}) \geq 1 - \zeta$ and $OPT(\mathcal{L}) \leq \gamma$.*

To address the scheduling problem $1|prec|\sum w_jC_j$, Bansal & Khot introduced the following variant of the unique games conjecture:

Hypothesis 1 ([4]). *For arbitrarily small constants $\zeta, \gamma, \delta > 0$, there exists an integer $n = n(\zeta, \gamma, \delta)$ such that for a unique games instance $\mathcal{L} = (G(V, W, E), [n], \{\pi_{v,w}\}_{(v,w)\in E})$, it is NP-hard to distinguish between:*

- *(YES Case:) There are sets $V' \subseteq V, W' \subseteq W$ such that $|V'| \geq (1-\zeta)|V|$ and $|W'| \geq (1-\zeta)|W|$ and an assignment to $\mathcal{L}$ such that all the edges between the sets (V', W') are satisfied.*
- *(NO Case:) No assignment to $\mathcal{L}$ satisfies even a γ fraction of the edges. Moreover, the instance satisfies the following expansion property. For every set $S \subseteq V, |S| = \delta|V|$, we have $\Gamma(S) \geq (1-\delta)|W|$, where $\Gamma(S) = \{w \in W | \exists v \in S, (v, w) \in E\}$.*

One can see that Hypothesis 1 differs from Conjecture 1 in two ways. In the YES case, Hypothesis 1 requires that there exists a labeling that satisfies all constraints between to large sets $V' \subseteq V, W' \subseteq W$. Whereas Conjecture 1 only requires that almost all constraints are satisfied by a labeling. Furthermore, in the NO case Hypothesis 1 has the additional assumption that the instance satisfy the (arguably weak) expansion property that any two sets of relative size δ has an edge between them.

As remarked in [4], no algorithmic results rule out Hypothesis 1. Furthermore, each of the two extra properties (one in the YES case and the other in the NO case) can be achieved by suitable transformations from the standard unique games conjecture. (The property in the NO case can be achieved by imposing a dummy expander and Khot & Regev [26] showed how to transform a unique game instance into one with the property in the YES case preserving low soundness.) However, it remains an open problem whether assuming both properties simultaneously is equivalent to the standard unique games conjecture.