Probabilistic Analysis for Scheduling with Conflicts

Sandy Irani*　　Vitus Leung †

Abstract

In this paper, we consider scheduling jobs that may be competing for mutually exclusive resources. We model the conflicts between jobs with a *conflict graph*, so that all concurrently running jobs must form an independent set in the graph. Our goal is to bound the maximum response time of any job in the system. We adopt a discrete model of time and assume that each job requires one time unit to be completed once it is started. It has been previously shown [6] that the best competitive ratio achievable by any online algorithm is $\Omega(n)$, where n is the number of nodes in the graph. As a result, we study scheduling with conflicts under probabilistic assumptions about the input. Each node i has a value p_i such that a job arrives at node i in any given time unit with probability p_i. Arrivals at different nodes and during different time periods are independent. Under reasonable assumptions on the value for the p_i's, we are able to obtain a bounded competitive ratio for an arbitrary conflict graph. In addition, if the conflict graph is a perfect graph, we give an algorithm whose competitive ratio converges to 1.

*Supported in part by NSF grants CCR-0105498 and CCR-9625844 and by ONR Award N00014-00-1-0617. Address: University of California, Irvine, Information and Computer Science, 444 Computer Science Building, Irvine, CA 92697-3425. Email: `irani@ics.uci.edu` .

†Address: Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185-1110. Email: `vjleung@cs.sandia.gov`. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000. Part of this work was done while at the University of California, Irvine and supported in part by Caltrans Contract 65Q090.

1 Introduction

In this paper, we consider scheduling jobs which are competing for limited resources. Jobs arrive in the system through time and require a certain set of resources to be completed. Any two jobs which require the same resource can not be executed simultaneously. We model the conflicts between jobs by a *conflict graph* where each node in the graph represents a type of job. Jobs of the same type have the same requirements. If two types of jobs demand a common resource, there is an edge between those nodes in the graph. Thus, at all times, the set of jobs currently being executed must belong to nodes which form an independent set in the graph. Note that if there are two jobs of the same type in the system, one must wait until the other is completed. We adopt a discrete model of time and assume that each job requires one time unit to be completed once it is started. At the beginning of a time unit, jobs may arrive on any subset of the nodes in G. The algorithm then chooses any independent set of nodes from which to schedule a job. At the end of the time unit, the scheduled jobs are gone from the graph. Then at the beginning of the next time unit, another set of jobs may arrive.

There are two natural optimization problems that arise from this model. The first is to minimize the *total response time* of all jobs in the system. The second is to minimize the *maximum response time* of any job which enters the system. We focus on bounding the maximum response time of any job which is the maximum, over all jobs j, of $d_j - a_j$, where d_j is the time when j is completed and a_j is the time when j arrives. This is appropriate for applications in which it is important to guarantee the best turnaround time to any job entering the system. See [6] for a discussion of how this model applies to traffic intersection control and scheduling in high-speed local-area networks with spacial reuse.

1.1 Our Results.

In a previous paper [6], it was shown that deterministic online algorithms have substantial limitations in this model. Specifically, it was shown that the competitive ratio can be as bad as $\Omega(n)$. This lower bound is shown both on a conflict graph which is a bipartite graph and on a conflict graph which is an interval graph. Furthermore, there is no known algorithm that achieves a competitive ratio that can be bounded by any function of n for bipartite or interval graphs. As a result, we are led to consider probabilistic assumptions over input sequences. The class of distributions that we consider can be defined by a vector $P = \langle p_1, p_2, \ldots, p_n \rangle$. At each time unit, a job arrives on node i with probability p_i. Arrivals in successive time units and on different nodes are completely independent. Any such vector P induces a distribution over arrival sequences which we will call $\mathcal{D}(P)$. For a given algorithm A, we will be interested in finding an algorithm which minimizes $E_{\sigma \in \mathcal{D}(P)}[\mathtt{cost}_A(\sigma)]$, where $\mathtt{cost}_A(\sigma)$ is the maximum response time of any job when algorithm A schedules input sequence σ. For the remainder of this paper, we will omit the subscript $\sigma \in \mathcal{D}(P)$. We are also interested in determining how good our algorithm is in comparison to the optimal algorithm (which knows σ in advance). In the style of analyzing online algorithms against a *diffuse adversary* as defined by Koutsoupias and Papadimitriou [9], we determine

$$\frac{E[\mathtt{cost}_A(\sigma)]}{E[\mathtt{cost}_{OPT}(\sigma)]}.$$

The first set of results we obtain apply to general conflict graphs. Then we show how to significantly improve those bounds in the case that the conflict graph is a perfect graph.

It is reasonable to restrict the set of distributions to stable distributions where it is possible to schedule jobs in such a way that the number of jobs in the system returns to 0 with probability 1. Let G^l denote the *extended graph* induced by the job arrivals in a sequence of l consecutive time units. This graph is obtained by replacing each node in the graph by a clique whose size is the number of jobs which arrive at that node during the l time units. If two nodes are adjacent in the conflict graph, then the two corresponding cliques are completely connected. The chromatic number of G^l is the number of time units necessary to schedule the set of jobs arriving in the l time units. (The chromatic number of a graph G, denoted $\chi(G)$, is the minimum number of colors required in a valid coloring of G. A valid coloring of a graph is an assignment of colors to nodes such that no two adjacent nodes receive the same color.)

Let $\epsilon_l = l - E[\chi(G^l)]$. ϵ_l represents the expected 'slack' if l time units are allocated to schedule the set of jobs which arrive during l consecutive time units. Note that the distribution over ϵ_l does not depend on which l consecutive time units are chosen. Certainly if $\epsilon_l < 0$ for all l, then even the optimal algorithm will accumulate a continually growing backlog of jobs. Thus, the only distributions of interest are those where there exists an l such that $\epsilon_l > 0$.

In all cases, the algorithm that we analyze is the simple algorithm which for a given l, gathers all the jobs that arrive in each block of l consecutive time units and optimally schedules them before any of the jobs in the next block of l time units. Note that the problem of scheduling the set of jobs optimally is equivalent to coloring the corresponding extended graph optimally which is known to be NP-hard [8]. We are assuming that the conflict graph is small or simple enough that it is feasible to color any induced extended graph either in real time or with some pre-processing.

We will call a period of l consecutive time units an *l-block*. Let G_i^l be the extended subgraph induced by the jobs that arrive in the i^{th} l-block.

Algorithm l-block: Optimally schedule the jobs from the i^{th} l-block starting at the first time unit after the i^{th} l-block finishes and after all jobs from the $i-1^{st}$ l-block have been scheduled.

The first theorem bounds the competitive ratio of l-block on any conflict graph:

Theorem 1 *Let G be any conflict graph and P be any distribution vector. Let (a, b) be the edge such that $p_a p_b$ is maximized. If $\epsilon_l > 0$,*

$$\frac{E[\texttt{cost}_{l-block}]}{E[\texttt{cost}_{OPT}]} \leq \frac{l\chi \ln\left(\frac{1}{p_a p_b}\right)}{\ln\left(1 + \frac{\epsilon_l}{l\chi}\right)}(1 + o(1)),$$

where χ is the chromatic number of the conflict graph and $o(1)$ is a function which tends towards 0 as the length of the sequence increases.

When $\epsilon_l/\chi l$ is small, this ratio is closely approximated by

$$\frac{(l\chi)^2 \ln\left(\frac{1}{p_a p_b}\right)}{\epsilon_l}(1 + o(1)).$$

It is reasonable to think that by combining jobs from consecutive l-blocks (i.e. considering larger l), some improvement in the performance can be gained. It is possible, for example, that $E[\chi(G_1^2)] < E[\chi(G_1^1)] + E[\chi(G_2^1)]$, in which case, the 2-block algorithm will perform better than the 1-block algorithm. If this is not the case, then even the adversary can not combine jobs from consecutive blocks which should lead to a stronger lower bound on the optimal cost. We are able to formalize this intuition in the case where the conflict graph is a perfect graph. In this case, we can prove that as l grows, the competitive ratio of l-block converges to 1. That is, the performance of l-block converges to the optimal offline algorithm.

Theorem 2 *Let G be a conflict graph which is a perfect graph and let P be any distribution vector. Suppose that for some r, $\epsilon_r > 0$. Then there is an algorithm A such that*

$$\frac{E[\mathtt{cost}_A]}{E[\mathtt{cost}_{OPT}]} = 1 + o(1),$$

where the $o(1)$ is a function which tends towards 0 as the length of the sequence grows.

The algorithm A uses the l-block algorithm, periodically increasing l. The above algorithm has the property that it has to know a value r such that $\epsilon_r > 0$ since the length of the blocks must be a multiple of r. The following theorem indicates that one only has to find a block size which is large enough.

Theorem 3 *If G is a perfect graph, and there exists r such that $E[\chi(G^r)] < r$, then for all $l > r$, $E[\chi(G^l)] < l$.*

1.2 Previous Work.

In previous work, a more traditional worst-case analysis was used to examine scheduling with conflicts under this model [6]. These results focus on two special classes of applications graphs: interval graphs and bipartite graphs. A simple algorithm is considered which works on a graph arising from a problem in traffic intersection control. The algorithm obtains an optimal competitive ratio of 2. This result is then generalized for arbitrary bipartite and interval graphs. Scheduling on conflict graphs which are interval graphs model the problem of scheduling connections on a bus. It is proven that for both bipartite and interval graphs, there is an algorithm which can complete every job in time $O(n^3A^2)$, where A is the maximum response time over all jobs in the optimal schedule and n is the number of nodes in the conflict graph. Note that to achieve a bound on the competitive ratio, it is necessary to upper bound the cost of the algorithm by a function which is linear in A.

A lower bound of $\Omega(n)$ is obtained on the competitive ratio of any deterministic online algorithm scheduling on an n-node path. Since a path is both bipartite and an interval graph, this gives a lower bound for both classes.

Over the past twenty-five years, most of the work done on scheduling with conflicts between certain pairs of jobs has been based on the well-known *Dining Philosophers* paradigm which is inherently a problem in decentralized control. More recently, Motwani, Phillips, and Torng [10] and Bar-Noy, Mayer, Schieber, and Sudan [2] considered problems with centralized control. Motwani *et al.* consider the problem where each vertex in the conflict graph represents a single job and two vertices are adjacent when their corresponding jobs are in conflict [10]. The goal in this work is to minimize the makespan which is the time to complete all the jobs which arrive in a given sequence. For the case where jobs arrive at integral times and have unit execution times, they give a 2-competitive algorithm. When jobs arrive at arbitrary times and have arbitrary execution times, they give a 3-competitive algorithm using a model which allows for preemption. Bar-Noy *et al.* consider a problem where each vertex in the conflict graph represents a task that is to be scheduled as often as possible, and again, two vertices are adjacent when their corresponding tasks cannot be scheduled concurrently [2]. Their interest is in maximizing a measure of fairness among competing jobs. A number of authors have also considered the static problem in which conflicting jobs must be scheduled so as to minimize the makespan [1, 3, 4, 7]. Schaffter [11] considers a closely related offline problem which seeks to find a schedule for a set of jobs such that the makespan of the schedule is minimized subject to the condition that certain sets of jobs can not be computed in parallel.

2 A Useful Result from Queuing Theory

Throughout this paper, we will use the following fact:

Lemma 4 *Consider a distribution over the integers whose expectation is negative. That is, we have a sequence of values $p_x > 0$, where x can range over the integers. Furthermore,*

$\sum_x p_x = 1$ and $\sum_x xp_x < 0$. Then there are two solutions to the following equation:

$$\sum_x \gamma^x p_x = 1.$$

One solution is $\gamma = 1$ and the other is a value γ_0 which is greater than 1. Furthermore, for any γ in the range $1 < \gamma < \gamma_0$, $\sum_x \gamma^x p_x < 1$.

Proof. Define $f(\gamma) = \sum_x \gamma^x p_x$. It is clear that $f(1) = 1$. We will prove that the derivative of $f(\gamma)$ is negative at $\gamma = 1$ and that the second derivative of $f(\gamma)$ is always positive for $\gamma > 0$. For the first fact, note that

$$\frac{df}{d\gamma}(\gamma) = \sum_x x\gamma^{x-1}p_x.$$

Thus,

$$\frac{df}{d\gamma}(1) = \sum_x xp_x < 0.$$

For the second fact, note that

$$\frac{d^2 f}{d\gamma} = \sum_x x(x-1)\gamma^{x-2}p_x.$$

$x(x-1)$ is at least 0 for any integer x and is greater than 0 for any integer besides 0 and 1. Also, $p_x \geq 0$ for any x, and p_x must be greater than 0 for some x besides 0 and 1 (since $\sum_x xp_x < 0$). In addition, $\gamma^{x-2} > 0$ for any x as long as $\gamma > 0$. □

Figure 1 depicts an example of a function of the form $\sum_x p_x\gamma^x = f(\gamma)$, where $\sum_x p_x = 1$ and $\sum_x xp_x < 0$. When we refer to the *solution* to the equation $f(\gamma) = 1$, we will always be referring to the unique solution γ_0 which is greater than 1. If X is a random variable whose expectation is less than 0, then the solution to $\sum_x prob[X = x]\gamma^x = 1$ will be referred to as $\gamma(X)$. Note that if for some $\gamma > 1$, $f(\gamma) > 1$, then $\gamma > \gamma(X)$ and if $f(\gamma) < 1$, then $\gamma < \gamma(X)$.

The following lemmas will be useful:

Lemma 5 *Consider a finite set of random variables $X_1, \ldots, X_k$ defined over the integers. Consider a random variable X whose distribution is a weighted average of the the distributions of the X_i's. That is, there is a distribution $\langle p_1, \ldots, p_k\rangle$ over the X_i's such that $\sum_{i=1}^k p_i = 1$ and X can be generated by first selecting X_i with probability p_i and then generating X according to the distribution governing X_i. If $\gamma(X_i) \geq \gamma$ for all i, then $\gamma(X) \geq \gamma$.*

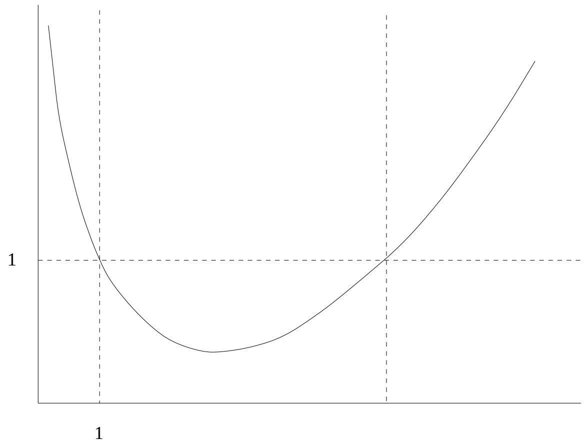

Figure 1: The graph of $\sum_x p_x \gamma^x = f(\gamma)$.

Proof. Let p_{ij} be the probability that X_i is j. We know that $\sum_x \gamma^x p_{ix} < 1$ for all i. The probability that X is equal to j is $\sum_{i=1}^k p_i p_{ij}$. Thus we have that

$$\sum_x \gamma^x \sum_{i=1}^k p_i p_{ix} = \sum_{i=1}^k p_i \sum_x \gamma^x p_{ix} \leq \sum_{i=1}^k p_i = 1.$$

$\square$

Lemma 6 *If X is the sum of any finite number of i.i.d. random variables $X_1, \ldots, X_n$, then for all $i \in \{1, \ldots, n\}$, $\gamma(X_i) = \gamma(X)$.*

Proof. Let p_x be the probability that any X_i is x. Since all the X_i's are identically distributed, this probability will be the same for all X_i's. Let γ be such that $\sum_x \gamma^x p_x = 1$ Then it is also the case that

$$\left(\sum_x \gamma^x p_x\right)^n = 1.$$

Regrouping we get that

$$\left(\sum_x \gamma^x p_x\right)^n = \sum_x \gamma^x \sum_{x_1,\ldots,x_n | x_1+\cdots+x_n=x} p_1 p_2 \cdots p_n.$$

The probability that $X = x$ is exactly $\sum_{x_1,\ldots,x_n|x_1+\cdots+x_n=x} p_1 p_2 \cdots p_n$. $\square$

All of our bounds on the maximum response time make use of the following result from queuing theory which follows from a result due to Iglehart [5]. The proof of Lemma 7 appears in the Appendix.

Lemma 7 *Suppose we have a sequence of identical independently distributed (i.i.d.) random variables* $\alpha_1, \alpha_2, \ldots$ *such that for all* $i \geq 1$, $\alpha_i \in \mathcal{Z}$, $E[\alpha_i] < 0$, *and* $\alpha_i \leq c$ *for some constant* c. *We will denote the* $Prob[\alpha_i = x]$ *by* p_x. *Then let* γ *be the constant greater than* 1 *which satisfies:*

$$\sum_{x \in \mathcal{Z}} \gamma^x p_x = 1.$$

Define the sequence $\hat{\alpha}_0 = 0$ *and* $\hat{\alpha}_i = \max\{0, \alpha_i + \hat{\alpha}_{i-1}\}$ *for* $i \geq 1$. *Let*

$$M_m = \max_{1 \leq i \leq m} \hat{\alpha}_i.$$

Then

$$\frac{\ln m}{\ln \gamma}(1 - o(1)) \leq E[M_m] \leq \frac{\ln m}{\ln \gamma}(1 + o(1)),$$

where the $o(1)$ *is taken to be a function of* m.

Lemma 7 is used to obtain an upper bounds for the l-block algorithm as follows:

Theorem 8 *Consider a graph, a distribution vector and a positive integer* l *such that* $E[\chi(G_i^l)] < l$. *Let* γ *be the solution to*

$$\sum_x Prob[\chi(G_i^l) - l = x]\gamma^x = 1.$$

The expected maximum latency of a job under the l*-block algorithm for a sequence lasting* ml *consecutive time units is at most* $2l + (\ln m / \ln \gamma)(1 + o(1))$.

Proof. We will use Lemma 7 with $\alpha_i = \chi(G_i^l) - l$. We will establish by induction on i that the jobs in the i^{th} l-block will be completed by time

$$il + \hat{\alpha}_{i-1} + \chi(G_i^l) = (i+1)l + \hat{\alpha}_{i-1} + \alpha_i \leq (i+1)l + \hat{\alpha}_i.$$

We will first see how this fact leads to the proof of the lemma. Since the jobs from the i^{th} l-block arrive after time $(i-1)l$, their latency is at most $2l + \hat{\alpha}_i$. M_m is the maximum over all i from 1 to m of $\hat{\alpha}_i$. Thus, $2l + M_m$ is an upper bound on the latency of any job in the first m time steps. By Lemma 7 the expectation of this value is at most $2l + (\ln m / \ln \gamma)(1 + o(1))$.

Now to establish that the jobs in the i^{th} l-block will be completed by time $il + \hat{\alpha}_{i-1} + \chi(G_i^l)$, we will assume that this is true for i and prove the claim for $i + 1$. (Note that it is trivially true for $i = 1$). The jobs in the i^{th} l-block will be completed by time $il + \hat{\alpha}_{i-1} + \chi(G_i^l) = (i+1)l + \hat{\alpha}_{i-1} + \alpha_i$. The jobs from the $i+1^{st}$ i-block can not begin before time $(i+1)l$, so they will begin at time $\max\{(i+1)l, (i+1)l+\hat{\alpha}_{i-1}+\alpha_i\} = (i+1)l+\hat{\alpha}_i$ and will be completed by time $(i+1)l + \hat{\alpha}_i + \chi(G_{i+1}^l)$. □

Theorem 9 *Consider a conflict graph G. Let l be any positive integer. Let C be any clique in G and let C_i^l be the number of jobs which arrive on nodes in C during the i^{th} l-block. Define γ to be the solution to*

$$\sum_x Prob[C_i^l - l = x]\gamma^x = 1.$$

The expected maximum latency of a job for any algorithm for a sequence which lasts m consecutive time units is at least $(\ln m/\ln\gamma)(1-o(1))$.

Proof. We will assume that $E[C_i^l - l] < 0$. Otherwise the expected backlog of jobs on the clique will increase linearly with l and the theorem follows. First we give the algorithm the advantage that it only has to serve jobs which arrive on nodes in clique C. We will use Lemma 7 with $\alpha_i = C_i^l - l$. We will establish by induction on i that the jobs which arrive on clique C in the i^{th} l-block will not be completed before time

$$(i-1)l + \hat{\alpha}_{i-1} + C_i^l = il + \hat{\alpha}_{i-1} + \alpha_i.$$

We will first see how this fact leads to the proof of the lemma. Since the the jobs from the i^{th} l-block have all arrived by time il, their latency is at least $\hat{\alpha}_{i-1} + \alpha_i$. Since the latency is always at least 0, this means that the latency is at least $\max\{0, \hat{\alpha}_{i-1} + \alpha_i\} = \hat{\alpha}_i$. M_m is the maximum over all i from 1 to m of $\hat{\alpha}_i$. Thus, M_m is a lower bound on the latency of any job in the first m time steps. By Lemma 7, the expectation of this value is at least $(\ln m/\ln\gamma)(1-o(1))$.

Now to establish that the jobs which arrive on clique C in the i^{th} l-block will not be completed before time $(i-1)l + \hat{\alpha}_{i-1} + C_i^l$, we will assume that this is true for i and prove the claim for $i+1$. (Note that it is trivially true for $i = 1$). The jobs in the i^{th} l-block will not be completed before time $(i-1)l + \hat{\alpha}_{i-1} + C_i^l = il + \hat{\alpha}_{i-1} + \alpha_i$. The jobs from the $i+1^{st}$ i-block can not begin before time il, so they can not begin before time $\max\{il, il + \hat{\alpha}_{i-1} + \alpha_i\} = il + \hat{\alpha}_i$ and can not be completed before time $il + \hat{\alpha}_i + C_{i+1}^l$. □

3 General Graphs

Before we begin, we need the following technical lemma.

Lemma 10 *Consider a random variable X distributed over a finite set of integers $\{-l, -l+1, \ldots, u-1, u\}$. Let p_x denote the probability that $X = x$. Suppose also that $p_0 = 0$. We can find a set of probabilities p_{jk} such that $\sum_{k=1}^{l} \sum_{j=1}^{u} p_{jk} = 1$ and for any pair (j, k) such that $p_{jk} > 0$, we have that $1 \leq k \leq l$, $1 \leq j \leq u$ and $-k \leq E[X] \leq j$. Furthermore, the distribution over X can be simulated as follows: pick a pair of integers (j, k) with probability p_{jk}. Now define X_{jk} to be the random variable which is either $-k$ or j and whose expectation is $E[X]$. Pick the value for X according to X_{jk}.*

Proof. For each $x \in \{-l, -l+1, \ldots, u-1, u\}$, we will separate p_x into two portions, a *used* portion and an *unused* portion. We denote the unused portion by $\hat{p}_x$. We proceed in a series of iterations. In each step, the number of values for which $\hat{p}_x = 0$ increases by one. The process is complete when $\hat{p}_x = 0$ for all x, Let X_r be the random variable which is x with probability $\frac{\hat{p}_x}{\sum_x \hat{p}_x}$. We maintain the property that $E[X_r] = E[X]$. We also maintain the property that for any k,

$$\hat{p}_{-k} + \sum_j p_{jk} Prob[X_{jk} = -k] = p_k,$$

and for any j,

$$\hat{p}_j + \sum_k p_{jk} Prob[X_{jk} = j] = p_j.$$

Let k be the largest integer such that $\hat{p}_{-k} > 0$. Let j be the largest integer such that $\hat{p}_j > 0$. It must be the case that $-k \leq E[X] \leq j$ because $E[X_r] = E[X]$.

Consider the quantity

$$-k\left(\frac{\hat{p}_{-k}}{\hat{p}_{-k} + \hat{p}_j}\right) + j\left(\frac{\hat{p}_j}{\hat{p}_{-k} + \hat{p}_j}\right).$$

If this quantity is less than $E[X]$ then let q_{-k} be the value for q which satisfies

$$-k\left(\frac{q}{q + \hat{p}_j}\right) + j\left(\frac{\hat{p}_j}{q + \hat{p}_j}\right) = E[X].$$

Otherwise let $q_{-k} \leftarrow \hat{p}_{-k}$. If this quantity is greater than $E[X]$, then let q_j be the solution to

$$-k\left(\frac{\hat{p}_{-k}}{\hat{p}_{-k}+q}\right)+j\left(\frac{q}{\hat{p}_{-k}+q}\right)=E[X].$$

Otherwise let $q_j \leftarrow \hat{p}_j$. Note that for at least one of $-k$ or j, it's q value will be equal to its $\hat{p}$ value. Now set $p_{jk} = q_{-k} + q_j$ and $\hat{p}_j \leftarrow \hat{p}_j - q_j$ and $\hat{p}_{-k} \leftarrow \hat{p}_{-k} - q_{-k}$.

Let X'_r denote the new value for X_r. Since $E[X_r]$ is a weighted average of $E[X_{jk}]$ and $E[X'_r]$ and $E[X_r] = E[X_{jk}]$, it must be the case that the $E[X_r]$ is equal to $E[X'_r]$. Furthermore, in the expression $\hat{p}_{-k} + \sum_j p_{jk} Prob[X_{jk} = k]$, $\hat{p}_{-k}$ has gone down by q_{-k} but the summation has increased by q_{-k}, so the expression is still equivalent to p_{-k}. The same argument holds for p_j. □

We will now consider the case where $E[\chi(G_i^l)] < l$ for some l and examine the performance of the l-block algorithm. Let α_i be the random variable $\chi(G_i^l) - l$. Define $a_k = Prob[\alpha_i = k]$. Now let γ_{alg} be the value greater than 1 such that

$$\sum_{x\in\mathcal{Z}}(\gamma_{alg})^x a_x = 1.$$

Defined $\epsilon_l = l - E[\chi(G_i^l)]$.

Lemma 11

$$\gamma_{alg} \geq \left(1+\frac{\epsilon_l}{\chi l}\right)^{\frac{1}{l\chi}},$$

Proof. For simplicity of notation, we will denote ϵ_l by ϵ in this proof. Since the α_i's are i.i.d., we will omit the subscript i. We know that α is an integer which is at least $-l$ and is at most $l(\chi-1)$. Thus, γ_{alg} is the solution to

$$\sum_{-l\leq j\leq l(\chi-1)} a_j\gamma^j = 1.$$

This is the same value which satisfies:

$$\sum_{0<j\leq l(\chi-1)}\left(\frac{a_j}{1-a_0}\right)\gamma^j + \sum_{-l\leq j<0}\left(\frac{a_j}{1-a_0}\right)\gamma^j = 1.$$

Thus, we can just consider the random variable α' which is j with probability

$$a'_j = \frac{a_j}{1-a_0},$$

for $j = -l, \ldots, -1, 1, 2, \ldots, l(\chi - 1)$. Define

$$E[\alpha'] = -\epsilon' = \frac{-\epsilon}{1 - a_0}.$$

For $1 \leq j \leq l(\chi - 1)$ and $1 \leq k \leq l$, define p_{jk} to be the solution to $k(1 - p_{jk}) - jp_{jk} = \epsilon'$. Let X_{jk} be the random variable which is j with probability p_{jk} and $-k$ with probability $1 - p_{jk}$. The expectation of X_{jk} is $-\epsilon'$. By Lemma 10, the distribution over α' can be expressed as a weighted average of the distributions over the X_{jk}'s. Let γ_{jk} be the solution to

$$p_{jk}(\gamma_{jk})^j + \frac{1 - p_{jk}}{(\gamma_{jk})^k} = 1.$$

If we can lower bound all of the γ_{jk}'s by the same value, then by Lemma 5, that value will also be a lower bound for γ_{alg}. We will show that

$$\gamma_{jk} \geq \left(1 + \frac{\epsilon}{j + k}\right)^{\frac{1}{j+k}} \geq \left(1 + \frac{\epsilon}{l\chi}\right)^{\frac{1}{l\chi}}.$$

Solving for p_{jk}, we get that $p_{jk} = (k - \epsilon')/(k + j)$. Thus, we will show that

$$\left(\frac{k - \epsilon'}{k + j}\right)\left(1 + \frac{\epsilon}{j + k}\right)^{\frac{j}{j+k}} + \left(\frac{j + \epsilon'}{k + j}\right)\left(1 + \frac{\epsilon}{j + k}\right)^{\frac{-k}{j+k}} \leq 1.$$

For simplicity of notation, let $\alpha = k/(k + j)$ and let $c = k + j$. We must show that

$$\left(\alpha - \frac{\epsilon'}{c}\right)\left(1 + \frac{\epsilon}{c}\right)^{1-\alpha} + \frac{\left(1 - \alpha + \frac{\epsilon'}{c}\right)}{\left(1 + \frac{\epsilon}{c}\right)^{\alpha}} \leq 1.$$

This is the same as

$$\left(\alpha - \frac{\epsilon'}{c}\right)\left(1 + \frac{\epsilon}{c}\right) + \left(1 - \alpha + \frac{\epsilon'}{c}\right) \leq \left(1 + \frac{\epsilon}{c}\right)^{\alpha}.$$

In order to lower bound the right hand side of the inequality above, we will use the Taylor expansion:

$$\left(1 + \frac{\epsilon}{c}\right)^{\alpha} = \sum_{i=0}^{\infty} \frac{\alpha(\alpha - 1)\cdots(\alpha - i + 1)}{i!}\left(\frac{\epsilon}{c}\right)^{i}.$$

Note that since $0 < \alpha < 1$, the terms with i odd are positive and the terms with i even are negative. Using this fact and the fact that $\epsilon/c < 1$, it can be verified that for $i \geq 3$ and i odd, the i^{th} plus the $i+1^{st}$ term is positive. This gives the following lower bound:

$$1 + \frac{\alpha\epsilon}{c} + \frac{\alpha(\alpha-1)\epsilon^2}{2c^2} \quad \leq \quad \left(1 + \frac{\epsilon}{c}\right)^{\alpha}$$

It can then be verified using the fact that $\epsilon' \geq \epsilon$ that

$$\left(\alpha - \frac{\epsilon'}{c}\right)\left(1 + \frac{\epsilon}{c}\right) + \left(1 - \alpha + \frac{\epsilon'}{c}\right) \leq 1 + \frac{\alpha\epsilon}{c} + \frac{\alpha(\alpha-1)\epsilon^2}{2c^2}.$$

□

For the lower bound on the optimal algorithm, we will invoke Theorem 9 and choose the clique to be a single edge (a, b) and l to be 1. Thus, $\alpha_i = C_i^1 - 1$ is a sequence of random variables which are distributed as follows:

$$\alpha_i = \begin{cases} 1 & \text{with probability } p_a p_b \\ 0 & \text{with probability } p_a(1-p_b) + p_b(1-p_a) \\ -1 & \text{with probability } (1-p_a)(1-p_b). \end{cases}$$

In order to invoke Theorem 9, we need to determine γ_{opt} which is the solution to

$$(\gamma)p_a p_b + [p_a(1-p_b) + p_b(1-p_a)] + \frac{(1-p_a)(1-p_b)}{\gamma} = 1. \tag{1}$$

Lemma 12

$$\gamma_{opt} \leq \frac{1}{p_a p_b}.$$

Proof. The left hand side of Equation 1 is greater than 1 when $\gamma = 1/p_a p_b$. □

We are now ready to prove Theorem 1 which is an upper bound on the ratio of the expected cost of the algorithm over the expected cost of the optimal.

Proof of Theorem 1. Let m be the number of l-blocks in the sequence. We use Theorems 8 and 9 and put the bounds from Lemma 11 and Lemma 12 together as follows:

$$\frac{E[\texttt{cost}_{l-block}]}{E[\texttt{cost}_{OPT}]} \quad \leq \quad \frac{2l + \ln m / \ln \gamma_{alg}(1 + o(1))}{\ln m / \ln \gamma_{opt}(1 - o(1))}$$

$$\leq \frac{\ln \gamma_{opt}}{\ln \gamma_{alg}}(1 + o(1))$$
$$\leq \frac{l\chi \ln\left(\frac{1}{p_a p_b}\right)}{\ln\left(1 + \frac{\epsilon}{l\chi}\right)}(1 + o(1))$$

□

4 Perfect Graphs

We now turn to the special case where the conflict graph is a perfect graph. We are assuming that it is the case that for some r, $E[\chi(G^r)] < r$. We will consider the lr-block algorithm. That is, the block size will always be a multiple of r. Let $\gamma_{on}(l) > 1$ satisfy

$$\sum_x Prob[\chi(G^{lr}) - lr = x](\gamma)^x = 1.$$

Let C be a clique in the graph. Let C_i^r be the random variable denoting number of jobs arriving on nodes in the clique in the i^{th} r-block. Define γ_C to be the solution to the equation:

$$1 = \sum_x (\gamma)^x Prob[C^r - r = x].$$

We prove the following theorem:

Theorem 13 *There is a clique C such that*

$$\gamma_{on}(l) \geq \gamma_C^{1-o(1)}.$$

The $o(1)$ above is taken to be as l grows. All other parameters of the system are constants.

The proof of Theorem 2 is straightforward once Theorem 13 has been established:

Proof of Theorem 2. The algorithm implements the lr-block algorithm, periodically increasing l so that l grows with the length of the sequence. The cost to the online algorithm is $2lr$ plus the backlog of jobs that accumulates after each block. We can ensure that the additional cost of $2lr$ grows more slowly than the backlog of jobs by doubling l every time the cost of the algorithm increases by a factor of 4. Using Theorems 8 and 9, we have that

$$\begin{aligned}\frac{E[\texttt{cost}_{online}]}{E[\texttt{cost}_{OPT}]} &= \frac{\ln \gamma_{opt}}{\ln \gamma_{on}(l)}(1+o(1)) \\ &\leq \frac{\ln \gamma_C}{\ln \gamma_C^{1-o(1)}}(1+o(1)) \\ &= (1+o(1)).\end{aligned}$$

□

Before proving Theorem 13, we require some definitions. Let C_i^{lr} be the number of jobs arriving in the i^{th} lr-block to clique C. Fix a value for l. For each clique C, let

$$\begin{aligned} p_{C,x} &= Prob[C^{lr} - lr = x] \\ \mu_C &= r - E[C^r] \\ \mu_{min} &= \min_C \mu_C \\ B &= \frac{-\mu_{min} l}{2} \end{aligned}$$

$C^{lr} - lr$ is the sum of l i.i.d. random variables which are distributed according to $C^r - r$. Thus, by Lemma 6, γ_C satisfies both of the following equations:

$$\begin{aligned} 1 &= \sum_x (\gamma)^x Prob[C^r - r = x] \\ 1 &= \sum_x (\gamma)^x p_{C,x} \end{aligned}$$

Note that γ_C is a constant that depends only on the parameters of the graph and is independent of l.

Define $\hat{\gamma}_C$ to be the solution to

$$\left[1 - N\sum_{x>B} p_{C,x}\right](\gamma)^B + N\sum_{x>B} p_{C,x}(\gamma)^x = 1,$$

where N is the number of distinct cliques in the graph.

Lemma 14 *For any clique C, $(\gamma_C)^{1-o(1)} \leq \hat{\gamma}_C$, where the $o(1)$ tends towards 0 as l gets large.*

We first give the proof of the theorem modulo Lemma 14.

Proof of Theorem 13. Consider the solutions to the following series of equations. The solution to the first two equations is the same. This follows from the fact that the graph G is a perfect graph which means that $\chi(G^{lr}) = \max_C C^{lr}$. In each of the subsequent steps, weight is shifted from smaller values of x to larger values of x which means the value of γ which satisfies the equation does not increase.

$$\begin{aligned}
1 &= \sum_x Prob[\chi(G^{lr}) - lr = x]\gamma^x \\
1 &= \sum_x Prob[\max_C C^{lr} - lr = x]\gamma^x \\
1 &= \sum_{x \leq B} Prob[\max_C C^{lr} - lr = x]\gamma^B + \sum_{x>B} Prob[\max_C C^{lr} - lr = x]\gamma^x \\
1 &= \left[1 - \sum_{x>B}\sum_C p_{C,x}\right]\gamma^B + \sum_{x>B}\sum_C p_{C,x}\gamma^x
\end{aligned}$$

Let $\bar{\gamma}$ be the solution to the last equation. Thus, $\gamma_{on}(l) \geq \bar{\gamma}$. Let

$$\begin{aligned}
\bar{f}(\gamma) &= \left[1 - \sum_{x>B}\sum_C\right] p_{C,x}\gamma^B + \sum_{x>B}\sum_C p_{C,x}\gamma^x \\
\hat{f}_C(\gamma) &= \left[1 - \sum_{x>B} Np_{C,x}\right]\gamma^B + \sum_{x>B} Np_{C,x}\gamma^x
\end{aligned}$$

$\bar{\gamma}$ is the solution to $\bar{f}(\gamma) = 1$. For each clique C, $\hat{\gamma}_C$ is the solution to $\hat{f}_C(\gamma) = 1$. Let D be the clique with the smallest $\hat{\gamma}_D$. This means that for any clique C, $\hat{f}_C(\hat{\gamma}_D) < 1$. Notice that for any γ,

$$\bar{f}(\gamma) = \frac{1}{N}\sum_C \hat{f}_C(\gamma),$$

where the sum is taken over all cliques C in the graph. This means that $\bar{f}(\hat{\gamma}_D) < 1$ which establishes that $\bar{\gamma} \geq \hat{\gamma}_D$. Thus, we know that $\gamma_{on}(l) \geq \bar{\gamma} \geq \hat{\gamma}_D$. We can invoke Lemma 14 and the theorem follows. $\square$

We will use the following technical lemma in the proof of Lemma 14 which we will prove later:

Lemma 15 *Suppose that $\hat{\gamma} > 1$ and satisfies the following inequality*

$$\sum_{s \le x \le b} q_x \hat{\gamma}^x \le 1 + A,$$

where $s < 0$ and $b > 0$. Suppose that $\bar{\gamma}$ is the solution to

$$\sum_{s \le x \le b} q_x \bar{\gamma}^x = 1.$$

Furthermore, suppose that there exists $X < 0$ and P such that the following conditions are satisfied.

1. $\sum_{s \le x \le b} q_x = 1$.

2. $\sum_{s \le x \le b} q_x x < 0$.

3. $\sum_{s \le x \le X} q_x \ge 1 - P$.

4. $\hat{\gamma}^{-X} \ge 6s^2$.

5. $P \le \frac{1}{4}$.

6. $0 < A \le \frac{1}{16s^2}$.

Then $\bar{\gamma} \ge \hat{\gamma}^{1-\epsilon}$ for $\epsilon = \frac{\ln\left(1+\frac{6A}{1-P}\right)}{\ln \hat{\gamma}}$.

Proof of Lemma 14. Throughout this proof, we refer to one clique C, so we drop the subscript C from the parameters. That is μ is used instead of μ_C and p_x is used instead of $p_{C,x}$.

The bound is proven in two steps. In each step we consider the solution to a new equation. The solution to the first equation is γ_C and the solution to the last equation is $\hat{\gamma}_C$. At each step we use Lemma 15 to prove that the equation changes by a sufficiently small amount that the solution decreases by at most a power of $1 - o(1)$.

First we would like to place a bound on $Prob[C^{lr} - lr > B]$. Define X_i to be the number of jobs to arrive to clique C on the i^{th} r-block minus r. X_i is a random variable with expectation $-\mu$ that lies in the range from $-r$ to $(\chi - 1)r$. $C^{lr} - r$ is $X_1 + \cdots + X_l$. Define Y_i to be $(X_i + \mu)/\chi r$. The expectation of Y_i is 0. Furthermore, Y_i lies in the range $[-1, 1]$. Recall that $B = -\mu_{min} l/2$. We first observe that

$$Prob[C^{lr} - lr > B] \leq Prob[C^{lr} - lr > -\mu l/2] = Prob\left[\sum_{i=1}^{l} Y_i > \mu l/2\chi r\right].$$

Chernoff's inequality says that

$$Prob\left[\sum_{i=1}^{l} Y_i > \lambda\right] \leq e^{\frac{-\lambda^2}{2l}}.$$

Plugging in $\lambda = \mu l/2\chi r$, we get that

$$Prob[C^{lr} - lr > B] \leq e^{\frac{-\mu^2 l}{8\chi^2 r^2}} \leq e^{-dl}, \tag{2}$$

for some constant d which depends only on r and the parameters of the graph and not on l. Recall that r is a fixed constant.

Now consider the following three equations:

$$\begin{aligned}
\gamma_0 \text{ satisfies } 1 &= \sum_x p_x \gamma^x \\
\gamma_1 \text{ satisfies } 1 &= (1 - Prob[C^{lr} - lr > B])\gamma^B + \sum_{x>B} p_x \gamma^x \\
\gamma_2 \text{ satisfies } 1 &= (1 - N \cdot Prob[C^{lr} - lr > B])\gamma^B + \sum_{x>B} N\gamma^x p_x
\end{aligned}$$

Note that $\gamma_0 = \gamma_C$ and $\gamma_2 = \hat{\gamma}_C$.

Step 1: $\gamma_1 \geq \gamma_0^{1-o(1)}$.

We know that

$$\sum_x p_x(\gamma_0)^x = 1$$

which means that

$$\sum_{x \leq B} p_x(\gamma_0)^B + \sum_{x>B} p_x(\gamma_0)^x \leq 1 + (\gamma_0)^B.$$

Let γ_1 be the solution to

$$\sum_{x \le B} p_x \gamma^B + \sum_{x > B} p_x \gamma^x = 1.$$

We will use Lemma 15 with $\hat{\gamma} \leftarrow \gamma_0$, $\bar{\gamma} \leftarrow \gamma_1$ and

$$\begin{aligned} q_x &= 0 \text{ for } s \le x < B \\ q_B &= 1 - \sum_{B < x \le b} p_x \\ q_x &= p_x \text{ for } B < x \le b \end{aligned}$$

We will let $A \leftarrow (\gamma_0)^B$, $P \leftarrow Prob[C^{lr} - lr > B]$, $X \leftarrow B$, $s \leftarrow -lr$, and $b \leftarrow \chi lr$. We must establish that for sufficiently large l, all the conditions of Lemma 15 are satisfied in order to lower bound γ_1.

Condition 1 follows from the fact that

$$\sum_{s \le x \le b} q_x = \sum_x p_x = 1.$$

To establish Condition 2, observe that $\sum_x xq_x < B + P\chi lr$. B is a negative number which is linear in l. P is a positive number which is exponentially small in l, thus the latter term goes to zero as l grows. This means that for l large enough, $\sum_x xq_x < 0$. Condition 3 follows from the fact that $X = B$. To establish condition 4, we need that $(\gamma_0)^{\mu_{min} l/2} \ge 6(lr)^2$. Since $(\gamma_0)^{\mu_{min} l/2}$ is exponential in l, for sufficiently large l, this condition will hold. Recall that γ_0 is a constant in l.

Condition 5 is satisfied using the bound in (2) which establishes that P is exponentially small in l, so there is an l large enough so that $P \le 1/4$. Condition 6 is satisfied since $A = (\gamma_0)^B = (\gamma_0)^{-\mu_{min} l/2} \le 1/16(lr)^2$ for sufficiently large l. Thus, by Lemma 15, $\gamma_1 \ge \gamma_0^{1-\epsilon}$ for

$$\epsilon = \frac{\ln\left(1 + \frac{6A}{(1-P)}\right)}{\ln \gamma_0}.$$

γ_0 is a constant as l grows. Also, $6A/(1-P)$ and thus $\ln(1 + (6A/(1-P)))$ go to zero as l grows. Thus, $\gamma_1 \ge \gamma_0^{1-o(1)}$.

Step 2: $\gamma_2 \geq \gamma_1^{1-o(1)}$.

Let $d = \ln l$. This step will in turn be proven in a series of steps:

$$\begin{array}{rcl} \gamma_{1a} & \geq & \gamma_1^{1-o(1)} \\ \gamma_2 & \geq & \gamma_{1b}^{1-o(1)} \end{array}$$

γ_{1a} will be defined shortly and γ_{1b} is defined to be

$$\gamma_{1b} = \gamma_{1a}^{1-\frac{1}{d}}.$$

Pick the largest S so that $(\gamma_1)^S \leq N^{2d}$. Recall that γ_1 satisfies

$$\left(1 - \sum_{x>B} p_x\right)(\gamma_1)^B + \sum_{x>B} (\gamma_1)^x p_x = 1.$$

Thus, we know that

$$\left[1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} p_x\right](\gamma_1)^B + \sum_{B<x\leq S} N(\gamma_1)^x p_x + \sum_{x>S} p_x \gamma_1^x \leq \left(N^{2d+1} \sum_{x>B} p_x\right) + 1.$$

Define γ_{1a} to be the solution to:

$$\left[1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} p_x\right]\gamma^B + \sum_{B<x\leq S} N\gamma^x p_x + \sum_{x>S} p_x \gamma^x = 1, \tag{3}$$

We will now apply Lemma 15 with $\hat{\gamma} = \gamma_1$, $\bar{\gamma} = \gamma_{1a}$,

$$\begin{array}{rcl} q_B & = & 1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} p_x \\ q_x & = & Np_x \ \text{ for } B < x \leq S \\ q_x & = & p_x \ \text{ for } \chi l \geq x > S \end{array}$$

and $A \leftarrow N^{2d+1} Prob[C^{lr} - lr > B]$, $P \leftarrow N \cdot Prob[C^{lr} - lr > B]$, $X \leftarrow B$, $s \leftarrow -lr$, and $b \leftarrow \chi lr$.

Now to establish that the conditions of Lemma 15 are satisfied. Condition 1 can be easily verified. Condition 2 states that

$$\sum_x q_x x < q_B B + \sum_{\chi lr \geq x > B} N p_x x \leq B + Nl\chi \sum_{x>B} p_x.$$

B is a negative number which is linear in l $(-\mu l/2)$. From Equation (2), $\sum_{x>B} p_x$ is exponentially decreasing in l, and N is a constant which depends only on parameter of the graph (not on l). This means that the second term goes to zero as l grows, and Condition 2 is satisfied. To verify Condition 3,

$$\sum_{x \leq B} q_x = q_B = 1 - \sum_{B<x\leq S} N p_x - \sum_{x>S} p_x \geq 1 - N \sum_{x>B} p_x = 1 - N Prob[C^{lr} - lr > B].$$

Condition 4 is the same as in Step 1. For condition 5, $Prob[C^{lr} - lr > B]$ is exponentially decreasing in l, and N is constant in l. Thus as l grows, P will tend towards 0. Finally for Condition 6, by definition, $A = N^{2d+1} Prob[C^{lr} - lr > B]$. $Prob[C^{lr} - lr > B]$ is exponentially decreasing in l. $d = \ln l$ and N is a constant in l which means that N^{2d+1} is polynomial in l. Thus, when l is large enough, $N^{2d}P$ will be at most $1/16(lr)^2$.

Since all the conditions of Lemma 15 are satisfied, we have that $\gamma_{1a} \geq \gamma_1^{1-\epsilon}$, where

$$\epsilon = \frac{\ln\left(1 + \frac{6A}{1-P}\right)}{\ln \gamma_1} \leq (1 + o(1)) \frac{\ln\left(1 + \frac{6A}{1-P}\right)}{\ln \gamma_0}.$$

γ_0 is a constant as l grows. Also, $6A/(1-P)$ and thus $\ln(1 + (6A/(1-P)))$ go to zero as l grows. Thus, $\gamma_{1a} \geq \gamma_1^{1-o(1)}$.

Now let $\gamma_{1b} = (\gamma_{1a})^{1-\frac{1}{d}}$. By the definition of S, we know that for $x \geq S$,

$$(\gamma_{1b})^x = \left(\frac{\gamma_{1a}}{(\gamma_{1a})^{\frac{1}{d}}}\right)^x \leq \frac{(\gamma_{1a})^x}{(\gamma_1)^{(1-o(1))\frac{x}{d}}} \leq \frac{(\gamma_{1a})^x}{(N^{2d})^{(1-o(1))\frac{1}{d}}} \leq \frac{(\gamma_{1a})^x}{N^{2-o(1)}} \leq \frac{(\gamma_{1a})^x}{N}. \quad (4)$$

The definition of S is used in the second inequality above. We will now apply Lemma 15 again with $\hat{\gamma} = \gamma_{1b}$, $\bar{\gamma} = \gamma_2$,

$$\begin{aligned} q_B &= 1 - \sum_{B<x} N p_x \\ q_x &= N p_x \text{ for } x > B \end{aligned}$$

$P \leftarrow N \cdot Prob[C^{lr} - lr \geq B]$, $X \leftarrow B$, $s \leftarrow -lr$, and $b \leftarrow \chi lr$. The arguments that Conditions 1 through 5 are satisfied are almost the same as above.

It remains to show that

$$\left[1 - \sum_{B<x} Np_x\right](\gamma_{1b})^B + \sum_{x>B} Np_x(\gamma_{1b})^x = 1 + A,$$

for some small value of A so that we can apply Lemma 15 again. We will rewrite the equation as:

$$\left[1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} Np_x\right](\gamma_{1b})^B + \sum_{B<x\leq S} Np_x(\gamma_{1b})^x + \sum_{x>S} Np_x(\gamma_{1b})^x, \qquad (5)$$

and bound the difference between (5) and (3). We will do this term by term. Using the bound from (4),

$$\sum_{x>S}[Np_x(\gamma_{1b})^x - p_x(\gamma_{1a})^x] \leq \sum_{x>S}\left[Np_x\frac{(\gamma_{1a})^x}{N} - p_x(\gamma_{1a})^x\right] = 0.$$

Then we have that:

$$\begin{aligned}\sum_{B<x\leq S}[N(\gamma_{1b})^x p_x - N(\gamma_{1a})^x p_x] \leq \sum_{B<x\leq S} N(\gamma_{1b})^x p_x &\leq Prob[C^{lr} - lr > B]N(\gamma_{1b})^S \\ &\leq Prob[C^{lr} - lr > B]N(\gamma_1)^S \\ &\leq Prob[C^{lr} - lr > B]N^{2d+1}\end{aligned}$$

And finally:

$$\begin{aligned}\left[1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} Np_x\right](\gamma_{1b})^B - \left[1 - \sum_{B<x\leq S} Np_x - \sum_{x>S} p_x\right](\gamma_{1a})^B &\leq (\gamma_{1b})^B \\ &\leq (\gamma_0)^B.\end{aligned}$$

Thus, we have that

$$\left[1 - \sum_{B\leq x} Np_x\right](\gamma_{1b})^B + \sum_{x>B} Np_x(\gamma_{1b})^x \leq 1 + Prob[C^{lr} - lr > B]N^{2d+1} + (\gamma_0)^B.$$

We have argued that $Prob[C^{lr} - lr > B]N^{2d+1}$ is exponentially small in l. γ_0 is a constant and B is a negative number which is linear in l. So for $A = Prob[C^{lr} - lr > B]N^{2d+1} + (\gamma_0)^B$, for large enough l, Condition 6 will be satisfied. We can apply Lemma 15 again to get that $\gamma_2 \geq \gamma_{1b}^{1-\epsilon}$ where

$$\epsilon = \frac{\ln\left(1 + \frac{A}{1-P}\right)}{\ln \gamma_{1b}} \leq (1 + o(1)) \frac{\ln\left(1 + \frac{A}{1-P}\right)}{\ln \gamma_0}.$$

γ_0 is a constant as l grows. Also, $6A/(1-P)$ and thus $\ln(1 + (6A/(1-P)))$ go to zero as l grows. Thus, $\gamma_2 \geq \gamma_{1b}^{1-o(1)}$.

□

Proof of Lemma 15. First we will set $p_0 = 0$ and renormalize by dividing all the probabilities by $1 - p_0$. This gives a new distribution q_x, where $q_0 = 0$. Note that the $\bar{\gamma}$ for the new distribution is the same as the $\bar{\gamma}$ for the old distribution. We also have that

$$\sum_{s \leq x \leq b} q_x \hat{\gamma}^x \leq \frac{1 + A - p_0}{1 - p_0}.$$

Since $p_0 \leq P \leq 1/4$, $(1 + A - p_0)/(1 - p_0) \leq 1 + 2A$. All the rest of the conditions for Lemma 15 still hold. From this point on, we will assume an adjusted distribution where $p_0 = 0$. The only fact that has changed is that $\sum_x p_x \hat{\gamma}^x \leq 1 + 2A$.

We will make use of the following lemma:

Lemma 16 *Consider $\gamma > 1$ which satisfies*

$$\sum_x p_x \gamma^x = d,$$

where $d > 1$, and the x's are integral. Suppose that $p_0 = 0$ and

$$\sum_{s \leq x \leq b} p_x = \sum_x p_x = 1.$$

Let r be the smallest integer such that $\gamma^r > d$. (Note that r is positive). For each pair $(i, j) \in [s, \ldots, r-1] \times [r, \ldots, b]$, we can find α_{ij} and β_{ij} such that

$$\alpha_{ij}\gamma^i + \beta_{ij}\gamma^j = d(\alpha_{ij} + \beta_{ij})$$

$$\begin{aligned}\sum_{r\le j\le b}\alpha_{ij} &= p_i\\ \sum_{s\le i<r}\beta_{ij} &= p_j\end{aligned}$$

Furthermore, if $p_i = 0$, then $\alpha_{ij} = 0$ for all j and if $p_j = 0$, then $\beta_{ij} = 0$ for all i.

We first prove Lemma 15 modulo Lemma 16. We use Lemma 16 to get a set of α's and β's such that for any pair (a, c),

$$\alpha_{ac}\hat{\gamma}^a + \beta_{ac}\hat{\gamma}^c = (1+2A)(\alpha_{ac}+\beta_{ac}). \tag{6}$$

Note that since $r > 0$ and $X < 0$, then $X < r$. We would like to find a $\bar{\gamma} > 1$ such that for any $a \in \{s, \ldots, X\}$ and any $c \in \{r, \ldots, b\}$,

$$[\alpha_{ac}\hat{\gamma}^a + \beta_{ac}\hat{\gamma}^c] - [\alpha_{ac}\bar{\gamma}^a + \beta_{ac}\bar{\gamma}^c] \ge \frac{2A}{1-P}(\alpha_{ac}+\beta_{ac}). \tag{7}$$

If we succeed, then we will have

$$\begin{aligned}
\sum_x p_x(\bar{\gamma})^x &= \sum_{s\le i<r}\sum_{r\le j\le b}[\alpha_{ij}\bar{\gamma}^i + \beta_{ij}\bar{\gamma}^j]\\
&\le \sum_{s\le i\le X}\sum_{r\le j\le b}[\alpha_{ij}\bar{\gamma}^i + \beta_{ij}\bar{\gamma}^j] + \sum_{X<i<r}\sum_{r\le j\le b}[\alpha_{ij}\bar{\gamma}^i + \beta_{ij}\bar{\gamma}^j]\\
&= \sum_{s\le i\le X}\sum_{r\le j\le b}[(\alpha_{ij}\hat{\gamma}^i + \beta_{ij}\hat{\gamma}^j) - ((\alpha_{ij}\hat{\gamma}^i + \beta_{ij}\hat{\gamma}^j) - (\alpha_{ij}\bar{\gamma}^i + \beta_{ij}\bar{\gamma}^j))]\\
&+ \sum_{X<i<r}\sum_{r\le j\le b}[(\alpha_{ij}\hat{\gamma}^i + \beta_{ij}\hat{\gamma}^j) - ((\alpha_{ij}\hat{\gamma}^i + \beta_{ij}\hat{\gamma}^j) - (\alpha_{ij}\bar{\gamma}^i + \beta_{ij}\bar{\gamma}^j))]\\
&\le \sum_{s\le i\le X}\sum_{r\le j\le b}\left(1+2A-\frac{2A}{1-P}\right)(\alpha_{ij}+\beta_{ij}) + \sum_{X<i<r}\sum_{r\le j\le b}(1+2A)(\alpha_{ij}+\beta_{ij})\\
&\le \sum_{s\le i\le X}\sum_{r\le j\le b}\left(1+2A-\frac{2A}{1-P}\right)\alpha_{ij} + \sum_{s\le i<r}\sum_{r\le j\le b}(1+2A)\beta_{ij}\\
&+ \sum_{X<i<r}\sum_{r\le j\le b}(1+2A)\alpha_{ij}\\
&\le \sum_{s\le i\le X}p_i\left(1+2A-\frac{2A}{1-P}\right) + \sum_{r\le j\le b}(1+2A)p_j + \sum_{X<i<r}p_i(1+2A)\\
&\le (1-P)\left(1+2A-\frac{2A}{1-P}\right) + (1+2A)P\\
&= 1.
\end{aligned}$$

The fifth line in the equations above uses (7), (6) and the fact that for $\hat{\gamma} > \bar{\gamma} > 1$,

$$[\alpha_{ac}\hat{\gamma}^a + \beta_{ac}\hat{\gamma}^c] - [\alpha_{ac}\bar{\gamma}^a + \beta_{ac}\bar{\gamma}^c] \geq 0.$$

This follows from the fact that for the function $f(\gamma) = \alpha_{ac}\gamma^a + \beta_{ac}\gamma^c$, $f(1) = \alpha_{ac} + \beta_{ac}$, $f(\hat{\gamma}) > \alpha_{ac} + \beta_{ac}$ and $f''(\gamma) > 0$ for all $\gamma > 0$.

After normalizing Equation 6, we get that for

$$p = \frac{\beta_{ac}}{\alpha_{ac} + \beta_{ac}},$$

$$(1-p)\hat{\gamma}^a + p\hat{\gamma}^c = (1+2A).$$

Thus, we would like to find a $\bar{\gamma}$ such that for any $a \leq X$ and $c \geq r$,

$$[(1-p)\hat{\gamma}^a + p\hat{\gamma}^c] - [(1-p)\bar{\gamma}^a + p\bar{\gamma}^c] \geq \frac{2A}{1-P}. \tag{8}$$

Using $(1-p)\hat{\gamma}^a + p\hat{\gamma}^c = (1+2A)$ to solve for p, we get that

$$p = \frac{1 + 2A - \hat{\gamma}^a}{\hat{\gamma}^c - \hat{\gamma}^a}.$$

We use this value to plug into (8). We also make the substitutions $\bar{\gamma} \leftarrow \hat{\gamma}^{1-\epsilon}$, where

$$\epsilon = \frac{\ln(1+3A')}{c \ln \hat{\gamma}} \leq \frac{\ln(1+3A')}{\ln \hat{\gamma}}$$

and $A' = 2A/1 - P$. This results in the following inequality which we would like to establish:

$$\left(\frac{\hat{\gamma}^{c+a} - \hat{\gamma}^a(1+2A)}{\hat{\gamma}^c - \hat{\gamma}^a}\right)[1 - (1+3A')^{\frac{-a}{c}}] + \left(\frac{(1+2A)\hat{\gamma}^c - \hat{\gamma}^{c+a}}{\hat{\gamma}^c - \hat{\gamma}^a}\right)\left(\frac{3A'}{1+3A'}\right) \geq A'.$$

Since $[1 - (1+3A')^{\frac{-a}{c}}] < 0$ and $3A'/(1+3A') > 0$, it is sufficient to establish

$$\left(\frac{\hat{\gamma}^{c+a} - \hat{\gamma}^a}{\hat{\gamma}^c - \hat{\gamma}^a}\right)[1 - (1+3A')^{\frac{-a}{c}}] + \left(\frac{\hat{\gamma}^c - \hat{\gamma}^{c+a}}{\hat{\gamma}^c - \hat{\gamma}^a}\right)\left(\frac{3A'}{1+3A'}\right) \geq A'. \tag{9}$$

Recall that $a \leq X < 0$ and $c \geq r > 0$. Keep in mind the assumptions of the lemma (specifically Condition 4) which tell us that

$$\hat{\gamma}^{-a} \geq \hat{\gamma}^{-X} \geq 6s^2 \geq 6 \max\left\{1, \left(\frac{-a}{c}\right)^2\right\}. \tag{10}$$

We will also make use of the fact that

$$A' = \frac{2A}{1-P} \leq \frac{8}{3}A \leq \frac{1}{6s^2} \tag{11}$$

The first inequality uses Condition 5 and the second uses Condition 6. We divide the proof of (9) into two cases:

Case 1: $c \geq -a$: Using the fact that $0 < -a/c < 1$, it can be verified that $1-(1+3A')^{\frac{-a}{c}} \geq -3A'$. Using the fact that $A' \leq 1/6$ from (11), it can be verified that $3A'/(1+3A') \geq 2A'$. Substituting these facts in to (9), it is only necessary to verify that

$$-3A'\left(\frac{\hat{\gamma}^{c+a}-\hat{\gamma}^{a}}{\hat{\gamma}^{c}-\hat{\gamma}^{a}}\right) + 2A'\left(\frac{\hat{\gamma}^{c}-\hat{\gamma}^{c+a}}{\hat{\gamma}^{c}-\hat{\gamma}^{a}}\right) \geq A',$$

which can be done easily using the fact that $\hat{\gamma}^{c} \geq \hat{\gamma}^{-a} \geq 6$.

Case 2: $c < -a$: For ease of notation, we will make the substitution $\beta = \hat{\gamma}^{-a}$ and $k = -c/a$. According to the assumption of this case, $0 < k < 1$. We want to prove the following:

$$\left(\frac{\beta^{k-1}-\beta^{-1}}{\beta^{k}-\beta^{-1}}\right)[1-(1+3A')^{\frac{1}{k}}] + \left(\frac{\beta^{k}-\beta^{k-1}}{\beta^{k}-\beta^{-1}}\right)\left(\frac{3A'}{1+3A'}\right) \geq A'.$$

As in the previous case, we can use the fact that $3A'/(1+3A') \geq 2A'$. We will also use the fact that

$$1-(1+3A')^{\frac{1}{k}} \geq \frac{-3A'}{k^2}, \tag{12}$$

which we will establish at the end of this case. Modulo (12), we just need to establish that

$$-\frac{3A'}{k^2}\left(\frac{\beta^{k-1}-\beta^{-1}}{\beta^{k}-\beta^{-1}}\right) \quad + \quad 2A'\left(\frac{\beta^{k}-\beta^{k-1}}{\beta^{k}-\beta^{-1}}\right) \geq A'.$$

It is sufficient to show that

$$\frac{3}{k^2\beta}(1-\beta^k) + \beta^k - 2\beta^{k-1} + \frac{1}{\beta} \geq 0.$$

From (10), we know that $k^2\beta \geq 6$. Also $\hat{\gamma}^c = \beta^k > 1$. Thus, it is sufficient to verify that

$$\frac{1}{2}(1-\beta^k) + \beta^k - 2\beta^{k-1} + \frac{1}{\beta} \geq 0.$$

Since $\beta \geq 4$, the inequality holds.

It now remains to establish Inequality 12 which is the same as proving

$$\left(1 + \frac{3A'}{k^2}\right)^k \geq 1 + 3A'.$$

Using the Taylor expansion for the left hand side, we will show that

$$\left(1 + \frac{3A'}{k^2}\right)^k \geq 1 + \frac{3A'}{k} - \frac{k(k-1)}{2}\left(\frac{3A'}{k^2}\right)^2. \tag{13}$$

This amounts to showing that for odd $i \geq 3$, the i^{th} plus the $i+1^{st}$ term is greater than 0. Note that the i^{th} term is positive and the $i+1^{st}$ term is negative.

$$\frac{k(k-1)\cdots(k-i+1)}{i!}\left(\frac{3A'}{k^2}\right)^i \geq -\frac{k(k-1)\cdots(k-i+1)(k-i)}{(i+1)!}\left(\frac{3A'}{k^2}\right)^{i+1}.$$

Rearranging, this is the same as

$$\frac{k^2}{3A'} \geq \frac{i-k}{i+1}.$$

From (11) we know that $A' \leq 1/6s^2 \leq k^2/6$, so the above inequality holds.

Now that Inequality 13 has been established, it is only necessary to verify that

$$1 + \frac{3A'}{k} - \frac{1}{2}\left(1 - \frac{1}{k}\right)\left(\frac{3A'}{k}\right)^2 \geq 1 + 3A'.$$

□

Proof of Lemma 16. We prove the theorem by induction on k, the number of values for x such that $p_x \neq 0$. If $k \leq 2$, then the lemma is trivial.

Now suppose that we have k values for x such that $p_x \neq 0$. Let s and b be the smallest and largest, respectively, such that $p_s \neq 0$ and $p_b \neq 0$. Note that $\gamma^s < d < \gamma^b$. If $p_s\gamma^s + p_b\gamma^b > d$, then let $\alpha_{sb} = p_s$ and $\beta_{sb} = (d(\alpha_{sb} + \beta_{sb}) - p_s\gamma^s)/\gamma^b$. If $p_s\gamma^s + p_b\gamma^b < d$, then let $\beta_{sb} = p_b$ and $\alpha_{sb} = (d(\alpha_{sb} + \beta_{sb}) - p_b\gamma^b)/\gamma^s$. And we have that

$$\alpha_{sb}\gamma^s + \beta_{sb}\gamma^b = d(\alpha_{sb} + \beta_{sb}).$$

Now define the distribution $q_x = p_x/(1 - \alpha_{sb} - \beta_{sb})$ for $x \neq s, b$. $q_s = (p_s - \alpha_{sb})/(1 - \alpha_{sb} - \beta_{sb})$, and $q_b = (p_b - \beta_{sb})/(1 - \alpha_{sb} - \beta_{sb})$. Note that $\sum_x q_x = 1$, and there are at most $k-1$ values for x such that $q_x \neq 0$. Furthermore, it can be verified that $\sum_x q_x \gamma^x = d$.

We can apply the inductive hypothesis to get $\bar{\alpha}_{ij}$ and $\bar{\beta}_{ij}$ such that

$$\bar{\alpha}_{ij}\gamma^i + \bar{\beta}_{ij}\gamma^j = d(\bar{\alpha}_{ij} + \bar{\beta}_{ij}),$$

$\sum_j \bar{\alpha}_{ij} = q_i$, and $\sum_i \bar{\beta}_{ij} = q_j$. Also, $\bar{\beta}_{sb} = \bar{\alpha}_{sb} = 0$ because we know that $q_b = 0$ or $q_s = 0$.

Now for $(i,j) \in [s, \ldots, r-1] \times [r, \ldots, b]$, such that $i \neq s$ or $j \neq b$, let $\alpha_{ij} = \bar{\alpha}_{ij}(1 - \alpha_{sb} - \beta_{sb})$, and $\beta_{ij} = \bar{\beta}_{ij}(1 - \alpha_{sb} - \beta_{sb})$. All the α's and β's have been determined and it can be verified that the conditions are met. $\square$

4.1 Finding a good block size

The algorithm described in the above section requires knowing a value for r such that $E[\chi(G^r)] < r$. The following lemma implies Theorem 3 and shows that it is only necessary to find an r that is large enough. That is, if $E[\chi(G^r)] < r$, then $E[\chi(G^l)] < l$ for all $l \geq r$.

Lemma 17 *If G is a perfect graph, then*

$$\frac{E[\chi(G^l)]}{E[\chi(G^{l+1})]} \geq \frac{l}{l+1}.$$

Proof. Let $G_{n_1,n_2,\ldots,n_k}$ be the extended graph formed by the arrival of n_i jobs on node i. Recall that p_i is the probability a job arrives at node i in any given time unit. Let $q_i = 1 - p_i$.

$$E[\chi(G^l)] = \sum_{n_1=0}^{l}\sum_{n_2=0}^{l}\cdots\sum_{n_k=0}^{l}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{n_i}{q_i}^{l-n_i}. \tag{14}$$

$$E[\chi(G^{l+1})] = \sum_{n_1=0}^{l+1}\sum_{n_2=0}^{l+1}\cdots\sum_{n_k=0}^{l+1}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l+1}{n_i}{p_i}^{n_i}{q_i}^{l+1-n_i}. \tag{15}$$

We will use the identity below

$$\sum_{S\subseteq\{1,2,\ldots,k\}}\prod_{i\in S}p_i\prod_{i\notin S}q_i = 1.$$

to re-express each term

$$\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{n_i}{q_i}^{l-n_i} \tag{16}$$

in $E[\chi(G^l)]$ as follows:

$$\begin{aligned}
&\sum_{S\subseteq\{1,2,\ldots,k\}}\prod_{i\in S}p_i\prod_{i\notin S}q_i\cdot\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{n_i}{q_i}^{l-n_i}\\
=\;&\sum_{u_1=0}^{1}\sum_{u_2=0}^{1}\cdots\sum_{u_k=0}^{1}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}p_i^{u_i}q_i^{1-u_i}\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{n_i}{q_i}^{l-n_i}\\
=\;&\sum_{u_1=0}^{1}\sum_{u_2=0}^{1}\cdots\sum_{u_k=0}^{1}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{n_i+u_i}{q_i}^{l+1-n_i-u_i}\\
=\;&\sum_{u_1=n_1}^{n_1+1}\sum_{u_2=n_2}^{n_2+1}\cdots\sum_{u_k=n_k}^{n_k+1}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{u_i}{q_i}^{l+1-u_i}. \qquad (17)
\end{aligned}$$

Now we can express $E[\chi(G^l)]$ by substituting the term (17) for the term (16) in (14):

$$E[\chi(G^l)]=\sum_{n_1=0}^{l}\sum_{n_2=0}^{l}\cdots\sum_{n_k=0}^{l}\sum_{u_1=n_1}^{n_1+1}\sum_{u_2=n_2}^{n_2+1}\cdots\sum_{u_k=n_k}^{n_k+1}\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l}{n_i}{p_i}^{u_i}{q_i}^{l+1-u_i}.$$

We would like to rearrange the expression above by grouping terms with the same factor

$$\prod_{i=1}^{k}{p_i}^{n_i}{q_i}^{l+1-n_i}.$$

Let $\hat{n}_i=\max\{0,n_i-1\}$ and $\check{n}_i=\min\{l,n_i\}$.

$$E[\chi(G^l)]=\sum_{n_1=0}^{l+1}\sum_{n_2=0}^{l+1}\cdots\sum_{n_k=0}^{l+1}\sum_{u_1=\hat{n}_1}^{\check{n}_1}\sum_{u_2=\hat{n}_2}^{\check{n}_2}\cdots\sum_{u_k=\hat{n}_k}^{\check{n}_k}\chi(G_{u_1,u_2,\ldots,u_k})\prod_{i=1}^{k}\binom{l}{u_i}{p_i}^{n_i}{q_i}^{l+1-n_i}. \tag{18}$$

By comparing (18) and (15), it can be seen that it is sufficient to show that for each choice of $n_1,\ldots,n_k$,

$$\begin{aligned}
&\frac{1}{l}\left[\sum_{u_1=\hat{n}_1}^{\check{n}_1}\sum_{u_2=\hat{n}_2}^{\check{n}_2}\cdots\sum_{u_k=\hat{n}_k}^{\check{n}_k}\chi(G_{u_1,u_2,\ldots,u_k})\prod_{i=1}^{k}\binom{l}{u_i}\right] && (19)\\
\geq\;&\frac{1}{l+1}\left[\chi(G_{n_1,n_2,\ldots,n_k})\prod_{i=1}^{k}\binom{l+1}{n_i}\right]. && (20)
\end{aligned}$$

Since $G_{n_1,n_2,\ldots,n_k}$ is a perfect graph, there is a subset S such that

$$\chi(G_{n_1,n_2,\ldots,n_k}) = \sum_{j\in S} n_j.$$

Note that we never need to include j in S if $n_j = 0$. If we substitute

$$\sum_{j\in S} n_j$$

into (20), we get

$$\frac{1}{l+1}\sum_{j\in S} n_j \left[\prod_{i=1}^{k}\binom{l+1}{n_i}\right] \;=\; \frac{1}{l+1}\sum_{j\in S}(l+1)\binom{l}{n_j-1}\prod_{i\neq j,1\leq i\leq k}\binom{l+1}{n_i}.$$

We know that

$$\chi(G_{u_1,u_2,\ldots,u_k}) \geq \sum_{j\in S} u_j.$$

Thus, (19) is at least

$$\frac{1}{l}\sum_{u_1=\hat{n}_1}^{\check{n}_1}\sum_{u_2=\hat{n}_2}^{\check{n}_2}\cdots\sum_{u_k=\hat{n}_k}^{\check{n}_k}\sum_{j\in S} u_j\prod_{i=1}^{k}\binom{l}{u_i} = \frac{1}{l}\sum_{j\in S}\sum_{u_1=\hat{n}_1}^{\check{n}_1}\sum_{u_2=\hat{n}_2}^{\check{n}_2}\cdots\sum_{u_k=\hat{n}_k}^{\check{n}_k} l\binom{l-1}{u_j-1}\prod_{i\neq j,1\leq i\leq k}\binom{l}{u_i}.$$

The theorem follows from the identity: for each $j \in \{1,\ldots,k\}$,

$$\binom{l}{n_j-1}\prod_{i\neq j,1\leq i\leq k}\binom{l+1}{n_i} = \sum_{u_1=\hat{n}_1}^{\check{n}_1}\sum_{u_2=\hat{n}_2}^{\check{n}_2}\cdots\sum_{u_k=\hat{n}_k}^{\check{n}_k}\binom{l-1}{u_j-1}\prod_{i\neq j,1\leq i\leq k}\binom{l}{u_i}.$$

This identity can be proven by induction on k. If $k = 1$, then the identity is

$$\binom{l}{n_j-1} = \sum_{u_j=\hat{n}_j}^{\check{n}_j}\binom{l-1}{u_j-1}.$$

For $n_j = l+1$, then $\hat{n}_j = \check{n}_j = l$, and

$$\binom{l}{n_j-1} = \binom{l-1}{\hat{n}_j-1}.$$

For $1 \leq n_j \leq l$, we use the fact that

$$\binom{l}{n_j - 1} = \binom{l-1}{\hat{n}_j - 1} + \binom{l-1}{\check{n}_j - 1}.$$

(Recall that it will never be the case that $n_j = 0$ for any $j \in S$.)

Now suppose that we have k values for i. We can apply the inductive hypothesis on $k-1$ of the values for i including $i = j$ to reduce the identity to

$$\binom{l+1}{n_i} = \sum_{u_i = \hat{n}_i}^{\check{n}_i} \binom{l}{u_i}.$$

For $n_i = l+1$ or $n_i = 0$, then $\hat{n}_i = \check{n}_i$, and

$$\binom{l+1}{n_i} = \binom{l}{\hat{n}_i}.$$

Then for $1 \leq n_i \leq l$, we use the fact that

$$\binom{l+1}{n_i} = \binom{l}{\hat{n}_i} + \binom{l}{\check{n}_i}.$$

□

References

[1] B.S. Baker and E.G. Coffman, Jr., Mutual exclusion scheduling, *Theoret. Comput. Sci.*, **162** (1996), 225–243.

[2] A. Bar-Noy, A. Mayer, B. Schieber and M. Sudan, Guaranteeing fair service to persistent dependent tasks, *SIAM J. on Computing*, **27** (1998), 1168–1189.

[3] H.L. Bodlaender and K. Jansen, On the complexity of scheduling incompatible jobs with unit-times, in *Mathematical Foundations of Computer Science* (A.M. Borzyszkowski and S. Sokolowski, eds.). Springer-Verlag. Berlin, 1993, pp. 291–300.

[4] H.L. Bodlaender, K. Jansen and G.J. Woeginger, Scheduling with incompatible jobs, *Discrete Appl. Math.*, **55** (1994), 219–232.

[5] D.L. Iglehart, Extreme values in the GI/G/1 queue, *The Annals of Mathematical Statistics*, 43 (1972), pp. 627-635.

[6] S. Irani and V. Leung, Scheduling with conflicts, and applications to traffic signal control, in *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 1996.

[7] K. Jansen, Scheduling of incompatible jobs on unrelated machines, *Internat. J. of Found. of Comput. Sci.*, **4** (1993), 275–291.

[8] R.M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations* (R.E. Miller and J.W. Thatcher, eds.). Plenum. New York, 1972, pp. 85–103.

[9] E. Koutsoupias and C.H. Papadimitriou, Beyond competitive analysis, in *35th Symposium on Foundation of Computer Science*, IEEE Computer Society Press, 1994.

[10] R. Motwani, S. Phillips and E. Torng, Nonclairvoyant scheduling, *Theoret. Comput. Sci.*, **130** (1994), 17–47.

[11] M.W. Schaffter, Scheduling with forbidden sets, *Discrete Appl. Math.*, **72** (1997), 155–166.

A Appendix: Proofs of Queuing Theory Results

A.1 Iglehart's Lemma

Iglehart proves a continuous time version of the lemma below in which arrival and departure rates from a queue are i.i.d. random variables. Iglehart's proof goes through, practically unchanged, for discrete time intervals to yield the following tight characterization of the distribution over M_m.

Lemma 18 *Suppose we have a sequence of identical independently distributed random variables $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that $\alpha_i \in \mathcal{Z}$, $E[\alpha_i] < 0$, and $\alpha_i \leq c$ for some constant c. We* *will denote the* $Prob[\alpha_i = x]$ *by* p_x. *Then let* γ *be the positive constant which satisfies*

$$\sum_{x \in \mathcal{Z}} \gamma^x p_x = 1.$$

Define the sequence $\hat{\alpha}_i = \max\{0, \alpha_i + \hat{\alpha}_{i-1}\}$, *and let*

$$M_m = \max_{1 \le i \le m} \hat{\alpha}_i.$$

Then

$$\lim_{m \to \infty} Pr[\ln \gamma M_m - \log km \le x] = \Lambda^{\frac{1}{\phi}}(x),$$

where $\Lambda(x) = e^{-e^{-x}}$, k *is a constant which depends only on* α_i, *and* $\phi = E[\min\{j \mid j > 0, \hat{\alpha}_j = 0\}]$.

A.2 Proof of Lemma 7

Let γ,ϕ and M_m be defined as in §A.1. We will show that there is a constant k such that for large enough m,

$$\frac{\ln km - \ln \phi}{\ln \gamma} - \frac{\gamma}{\gamma - 1} \le E[M_m] \le \frac{\ln km}{\ln \gamma} + \frac{2\gamma}{\phi(\gamma - 1)}.$$

Using Lemma 18, for m large enough we have that

$$\begin{aligned} Prob\left[M_m - \frac{\ln km}{\ln \gamma} \ge y\right] &= 1 - \frac{1}{e^{\frac{e^{-y \ln \gamma}}{\phi}}} \\ &\le 1 - \frac{1}{1 + \frac{2}{\phi \cdot e^{y \ln \gamma}}} \\ &\le \frac{2}{\phi \cdot e^{y \ln \gamma}}. \end{aligned}$$

Thus,

$$\begin{aligned} E[M_m] - \frac{\ln km}{\ln \gamma} &\le \sum_{y=0}^{\infty} \frac{2 \cdot e^{-y \ln \gamma}}{\phi} \\ &= \frac{2}{\phi} \sum_{y=0}^{\infty} \gamma^{-y} = \frac{2\gamma}{\phi(\gamma - 1)}. \end{aligned}$$

For the lower bound, we will use the fact that for m large enough,

$$Prob\left[\frac{\ln km - \ln \phi}{\ln \gamma} - M_m \geq y\right] = e^{-\gamma^y}.$$

Thus,

$$\begin{aligned}\frac{\ln km - \ln \phi}{\ln \gamma} - E[M_m] &\leq \sum_{y=0}^{\infty} \frac{1}{e^{\gamma^y}} \\ &\leq \sum_{y=0}^{\infty} \frac{1}{\gamma^y} = \frac{\gamma}{\gamma - 1}.\end{aligned}$$

□