Optimization Problems with Diseconomies of Scale via Decoupling

Konstantin Makarychev* Microsoft Research

Maxim Sviridenko† Yahoo! Labs

Abstract

We present a new framework for solving optimization problems with a diseconomy of scale. In such problems, our goal is to minimize the cost of resources used to perform a certain task. The cost of resources grows superlinearly, as x^q, $q \geq 1$, with the amount x of resources used. We define a novel linear programming relaxation for such problems, and then show that the integrality gap of the relaxation is A_q, where A_q is the q-th moment of the Poisson random variable with parameter 1. Using our framework, we obtain approximation algorithms for the Minimum Energy Efficient Routing, Minimum Degree Balanced Spanning Tree, Load Balancing on Unrelated Parallel Machines, and Unrelated Parallel Machine Scheduling with Nonlinear Functions of Completion Times problems.

Our analysis relies on the decoupling inequality for nonnegative random variables. The inequality states that

$$\Big\|\sum_{i=1}^{n} X_i\Big\|_q \leq C_q \Big\|\sum_{i=1}^{n} Y_i\Big\|_q,$$

where X_i are independent nonnegative random variables, Y_i are possibly dependent nonnegative random variable, and each Y_i has the same distribution as X_i. The inequality was proved by de la Peña in 1990. However, the optimal constant C_q was not known. We show that the optimal constant is $C_q = A_q^{1/q}$.

*email: komakary@microsoft.com

†email: sviri@yahoo-inc.com

1 Introduction

In this paper, we study combinatorial optimization problems with a diseconomy of scale. We consider problems in which we need to minimize the cost of resources used to accomplish a certain task. Often, the cost grows linearly with the amount of resources used. In some applications, the cost is sublinear e.g., if we can get a discount when we buy resources in bulk. Such phenomenon is known as "economy of scale". However, in many applications the cost is superlinear. In such cases, we say that the cost function exhibits a "diseconomy of scale". A good example of a diseconomy of scale is the cost of energy used for computing. Modern hardware can run at different processing speeds. As we increase the speed, the energy consumption grows superlinearly. It can be modeled as a function $P(s) = \mu s^q$ of the processing speed s, where μ and q are parameters that depend on the specific hardware. Typically, $q \in (1, 3]$ (see e.g., [2, 20, 35]).

As a running example, consider the Minimum Power Routing problem studied by Andrews, Anta, Zhang, and Zhao [3]. We are given a graph $G = (V, E)$ and a set of demands $\mathcal{D} = \{(d_i, s_i, t_i)\}$. Our goal is to route d_i ($d_i \in \mathbb{N}$) units of demand i from the source $s_i \in V$ to the destination $t_i \in V$ such that every demand i is routed along a single path p_i (i.e. we want to find an unsplittable multi-commodity flow). We want to minimize the energy cost. Every link (edge) $e \in E$ uses $f_e(x_e) = \mu_e x_e^q$ units of power, where μ_e is a scaling parameter depending on the link e, and x_e is the load on e.

The straightforward approach to solving this problem is as follows. We define a mathematical programming relaxation that routes demands fractionally. It sends $y_{i,p} d_i$ units of demand via the path p connecting s_i to t_i. We require that $\sum_p y_{i,p} = 1$ for every demand i. The objective function is to minimize

$$\min \sum_{e \in E} c_e x_e^q = \min \sum_{e \in E} c_e \Big(\sum_{p: e \in p} y_{i,p} d_i \Big)^q,$$

where $x_e = \sum_{p: e \in p} y_{i,p} d_i$ is the load on the link e. This relaxation can be solved in polynomial time, since the objective function is convex (for $q \geq 1$). But, unfortunately, the integrality gap of this relaxation is $\Omega(n^{q-1})$ [3]. Andrews et al. [3] gave the following integrality gap example. Consider two vertices s and t connected via n disjoint paths. Our goal is to route 1 unit of flow integrally from s to t. The optimal solution pays 1. The LP may cheat by routing $1/n$ units of flow via n disjoint paths. Then, it pays only $n \times (1/n)^q = n^{1-q}$.

For the case of uniform demands, i.e., for the case when all $d_i = d$, Andrews et al. [3] suggested a different objective function:

$$\min \sum_{e \in E} c_e \max\{x_e^q, d^{q-1} x_e\}.$$

The objective function is valid, because in the integral case, x_e must be a multiple of d, and thus $x_e^q \geq d^{q-1} x_e$. Andrews et al. [3] proved that the integrality gap of this relaxation is a constant. Bampis et al. [9] improved the bound to the *fractional Bell number* A_q that is defined as follows: A_q is the q-th moment of the Poisson random variable P_1 with parameter 1 (see Figure 1 in Appendix D). I.e.,

$$A_q = \mathbb{E}\left[P_1^q\right] = \sum_{t=1}^{+\infty} t^q \frac{e^{-1}}{t!}. \tag{1}$$

For the case of general demands no constant approximation was known. The best known approximation due to Andrews et al. [3] was $O(k + \log^{q-1} \Delta)$ where $k = |\mathcal{D}|$ is the number of demands and $\Delta = \max_i d_i$ is the size of the largest demand (Theorem 8 in [3]).

In this work, we give an A_q-approximation algorithm for the general case and thus close the gap between the case of uniform and non-uniform demands. Our approximation algorithm uses a general framework for solving problems with a diseconomy of scale which we present in this paper. We use this framework to obtain approximation algorithms for several other combinatorial optimization problems. We give

$A_q^{1/q}$-approximation algorithm for Load Balancing on Unrelated Parallel Machines (see Section 2.2), $2^q A_q$-approximation algorithm for Unrelated Parallel Machine Scheduling with Non-linear Functions of Completion Times (see Section 2.3) and A_q-approximation algorithm for the Minimum Degree Balanced Spanning Tree problem (see Section 2.4). The best previously known bound for the first problem with $q \in [1, 2]$ was $2^{1/q}$ (see Figure 3 for comparison). The bound is due to Kumar, Marathe, Parthasarathy and Srinivasan [21]. There were no known approximation guarantees for the latter problems.

In the analysis, we use the de la Peña decoupling inequality [25, 26].

Theorem 1.1 (de la Peña [25, 26])**.** *Let $Y_1, \dots, Y_n$ be jointly distributed nonnegative (non-independent) random variables, and let $X_1, \dots, X_n$ be independent random variables such that each X_i has the same distribution as Y_i. Then, for every $q \geq 1$,*

$$\Big\| \sum_{i=1}^{n} X_i \Big\|_q \leq C_q \Big\| \sum_{i=1}^{n} Y_i \Big\|_q, \tag{2}$$

for some universal constant C_q.

The optimal value of the constant C_q was not known. The original proof of de la Peña relies on more general inequalities and does not give specific constants. We give a direct proof of this inequality. We show that the inequality holds for $C_q = A_q^{1/q}$ and moreover this bound is tight.

Theorem 1.2. *Inequality (2) holds for $C_q = A_q^{1/q}$, where A_q is the fractional Bell number (see Equation (1) and Figure 1) Moreover, $A_q^{1/q}$ is the optimal upper bound on C_q.*

1.1 General Framework

We now describe the general framework for solving problems with a diseconomy of scale. We consider optimization problems with n decision variables $y_1, \dots, y_n \in \{0, 1\}$. We assume that the objective function equals the sum of k terms, where the j-th term is of the form

$$\Big(\sum_{i=1}^{n} d_{ij} y_i \Big)^{q_j},$$

here $d_{ij} \geq 0$ and $q_j \geq 1$ are parameters. The vector $y = (y_1, \dots, y_n)$ must satisfy the constraint $y \in \mathcal{P}$ for some polytope $\mathcal{P} \subset [0, 1]^n$. Therefore, the optimization problem can be written as the following boolean convex program (IP):

$$\begin{aligned} \min \quad & \sum_{j\in[k]} \Big(\sum_{i\in[n]} d_{ij} y_i \Big)^{q_j} && (3) \\ & y \in \mathcal{P} && (4) \\ & y \in \{0, 1\}^n. && (5) \end{aligned}$$

We assume that we can optimize any linear function over the polytope $\mathcal{P}$ in polynomial time (e.g., $\mathcal{P}$ is defined by polynomially many linear inequalities, or there exists a separation oracle for $\mathcal{P}$). Thus, if we replace the integrality constraint (5) with the relaxed constraint $y \in [0, 1]^n$ (which is redundant, since $\mathcal{P} \subset [0, 1]^n$), we will get a convex programming problem that can be solved in polynomial time (see [11]). However, as we have seen in the example of Minimum Power Routing, the integrality gap of the relaxation can be as large as $\Omega(n^{q-1})$.

In this work, we introduce a linear programming relaxation of (3) - (5) that has an integrality gap of A_q (where $q = \max_j q_j$) under certain assumptions on the polytope $\mathcal{P}$. We define auxiliary variables z_{jS} for all $S \subset [n]$. In the integral solution, $z_{jS} = 1$ if and only if $y_j = 1$ for $j \in S$ and $y_j = 0$ for $j \notin S$.

$$\min \sum_{j\in[k]} \sum_{S\subseteq[n]} \left(\sum_{i\in S} d_{ij}\right)^{q_j} z_{jS} \tag{6}$$

$$y \in \mathcal{P}, \tag{7}$$

$$\sum_{S\subseteq[n]} z_{jS} = 1, \qquad \forall j \in [k], \tag{8}$$

$$\sum_{S:i\in S} z_{jS} = y_i, \qquad \forall i \in [n], j \in [k], \tag{9}$$

$$z_{jS} \geq 0, \qquad \forall S \subseteq [n], j \in [k]. \tag{10}$$

The optimization problem (6) - (10) is a relaxation of the original problem (3) - (5). Note that the LP has exponentially many variables. We show, however, that the optimal solution to this LP can be found in polynomial time up to an arbitrary accuracy $(1 + \varepsilon)$. We say that y is a $(1 + \varepsilon)$-approximately optimal solution if the cost of the solution is at most $(1 + \varepsilon)OPT$, where OPT is the cost of the optimal solution.

Theorem 1.3. *Suppose that there exists a polynomial time separation oracle for the polytope $\mathcal{P}$. Then, for every ε and q, there exists a polynomial time algorithm that finds a $(1 + \varepsilon)$-approximately optimal solution to LP (6) - (10).*

We then prove the following theorem.

Theorem 1.4. *Let $D_j = \{i : d_{ij} \neq 0\}$. Assume that there exists a randomized algorithm R that given a $y \in \mathcal{P}$, returns a random integral point $R(y)$ in $\mathcal{P} \cap \{0, 1\}^n$ such that*

1. $\Pr(R_i(y) = 1) = y_i$ *for all i (where $R_i(y)$ is the i-th coordinate of $R(y)$);*
2. *Random variables $\{R_i(y)\}_{i\in D_j}$ are independent for every j.*

Then, for every feasible solution (y^*, z^*) to the LP (6) - (10), we have

$$\mathbb{E}\Big[\sum_{j\in[k]} \Big(\sum_{i\in[n]} d_{ij} R_i(y^*)\Big)^{q_j}\Big] \leq A_q \sum_{j\in[k]} \sum_{S\subseteq[n]} \left(\sum_{i\in S} d_{ij}\right)^{q_j} z^*_{jS}, \tag{11}$$

where $q = \max_j q_j$ and A_q is the fractional Bell number (see (1)). Particularly, since the LP (6) - (10) is a relaxation for the IP (3) - (5), if (y^*, z^*) is a $(1+\varepsilon)$-approximately optimal solution to the LP (6) - (10), then

$$\mathbb{E}\Big[\sum_{j\in[k]} \Big(\sum_{i\in[n]} d_{ij} R_j(y^*)\Big)^{q_j}\Big] \leq (1 + \varepsilon) A_q \, IP,$$

where IP is the optimal cost of the boolean convex program (3) - (5).

This theorem guarantees that an algorithm R satisfying conditions (1) and (2) has an approximation ratio of $(1 + \varepsilon)A_q$.

In the next section, Section 2, we show how to use the framework to obtain A_q approximation algorithms for four different combinatorial optimization problems. Then, in Section 3, we give an efficient algorithm for solving LP (6) - (10). In Section 4, we prove the main theorem – Theorem 1.4. The proof easily follows from the decoupling inequality, which we prove in Section 5. Finally, in Section 6, we describe some generalizations of our framework.

2 Applications

In this section, we show applications of our general technique. We start with the problem discussed in the introduction – Energy Efficient Routing.

2.1 Energy Efficient Routing

We are given a directed graph $G = (V, E)$ and a set of demands $\mathcal{D}$. The demand $i \in \mathcal{D}$ is associated with a source node s_i and a destination node t_i. It requests d_i integer units of bandwidth. If f units of demand are routed via an edge e, then the edge (link) e consumes $f = c_e f^{q_e}$ units of energy ($c_e \geq 0$ and $q_e \geq 1$ are parameters defined for each edge e). The objective is to route all the demands from their sources to their destinations so that the total energy consumption is minimized. We consider the unsplittable version of the problem where each demand has to be routed through a single path. Recall, that Andrews et al. [3] gave an $O(k + \log^{q-1} \Delta)$-approximation algorithm where $k = |\mathcal{D}|$ and $\Delta = \max_{i\in\mathcal{D}} d_i$ (Theorem 8 in [3]). We give A_q-approximation algorithm for this problem.

We write a standard integer program (below, $\Gamma^+(u)$ is the set of edges outgoing from u; $\Gamma^-(u)$ is the set of edges incoming to u). Each variable $y_{i,e} \in \{0, 1\}$ indicates whether the edge e is used to route the flow from s_i to t_i.

$$\min \sum_{e\in E} \left(\sum_{i\in\mathcal{D}} d_i y_{i,e}\right)^{q_e} \tag{12}$$

$$\sum_{e\in\Gamma^+(u)} y_{i,e} - \sum_{e\in\Gamma^-(u)} y_{i,e} = 0 \qquad \forall i \in \mathcal{D}, u \in V \setminus \{s_i, t_i\} \tag{13}$$

$$\sum_{e\in\Gamma^+(s_i)} y_{i,e} = 1 \qquad \forall i \in \mathcal{D} \tag{14}$$

$$\sum_{e\in\Gamma^-(t_i)} y_{i,e} = 1 \qquad \forall i \in \mathcal{D} \tag{15}$$

$$y_{i,e} \in \{0, 1\} \qquad \forall i \in \mathcal{D}, e \in E \tag{16}$$

Using Theorem 1.3, we obtain an almost optimal fractional solution (y, z) of the LP relaxation (6) - (10) of the IP (12) - (16). We apply randomized rounding in order to select a path for each demand. Specifically, for each demand $i \in \mathcal{D}$, we consider the standard flow decomposition into paths: In the decomposition, each path p connecting s_i to t_i has a weight $\lambda_{i,p} \in \mathbb{R}^+$. For every edge e, $\sum_{p:e\in p} \lambda_{i,p} = d_i y_{i,e}$; and $\sum_p \lambda_{i,p} = d_i$. For each i, the approximation algorithm picks one path p connecting s_i to t_i at random with probability $\lambda_{i,p}/d_i$, and routes all demands from s_i to p_i via p. Thus, the algorithm always obtains a feasible solution.

We verify that the integral solution corresponding to this combinatorial solution satisfies the conditions of Theorem 1.4. Let $R_{i,e}(y)$ be the integral solution, i.e., let $R_{i,e}(y) = 1$ if the edge e is chosen in the path connecting s_i and t_i. First, $R_{i,e}(y) = 1$ if the path connecting s_i and t_i contains e, thus

$$\Pr(R_{i,e}(y) = 1) = \sum_{p:e\in p} \lambda_{i,p}/d_i = y_{i,e}.$$

Second, the paths for all demands are chosen independently. Each $R_{i,e}(y)$ depends only on paths that connect s_i to t_i. Thus all random variables $R_{i,e}(y)$ (for a fixed e) are independent. Therefore, by Theorem 1.4, the cost of the solution obtained by the algorithm is bounded by $(1 + \varepsilon)A_q\, OPT$, where OPT is the cost of the optimal solution to the integer program which is exactly equivalent to the Minimum Energy Efficient Routing problem.

2.2 Load Balancing on Unrelated Parallel Machines

We are given n jobs and m machines. The processing time of the job $j \in [n]$ assigned to the machine $i \in [m]$ is $p_{ij} \geq 0$. The goal is to assign jobs to machines to minimize the ℓ_q-norm of machines loads. Formally, we partition the set of jobs into into m sets $S_1, \dots, S_m$ to minimize $\left(\sum_{i\in[m]}(\sum_{j\in S_i} p_{ij})^q\right)^{1/q}$. This is a classical scheduling problem which is used to model load balancing in practice[1]. It was previously studied by Azar and Epstein [6] and by Kumar, Marathe, Parthasarathy and Srinivasan [21]. Particular, for $q \in (1,2]$ the best known approximation algorithm has performance guarantee $2^{1/q}$ [21] (Theorem 4.4). We give $\sqrt[q]{A_q}$-approximation algorithm substantially improving upon previous results (see Figure 3).

We formulate the unrelated parallel machine scheduling problem as a boolean nonlinear program:

$$\min \sum_{i\in[m]} \left(\sum_{j\in[n]} p_{ij} x_{ij}\right)^q \tag{17}$$

$$\sum_{i\in[m]} x_{ij} = 1 \qquad \forall j \in [n] \tag{18}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in [m], j \in [n]. \tag{19}$$

Using Theorem 1.3, we obtain an almost optimal fractional solution (x, z) of the LP relaxation (6)–(10) corresponding to the IP (17)–(19). We use the straightforward randomized rounding: we assign each job j to machine i with probability x_{ij}. We claim that, by Theorem 1.4, the expected cost of our integral solution is upper bounded by A_q times the value of the fractional solution (x, z). Indeed, the probability that we assign a job j to machine i is exactly equal to x_{ij}; and we assign job j to machine i independently of other jobs. That implies that our approximation algorithm has a performance guarantee of $\sqrt[q]{A_q}$ for the ℓ_q-norm objective.

2.3 Unrelated Parallel Machine Scheduling with Nonlinear Functions of Completion Times

As in the previous problem, in Unrelated Parallel Machine Scheduling with Nonlinear Functions of Completion Times, we are given n jobs and m machines. The processing time of the job $j \in [n]$ assigned to the machine $i \in [m]$ is $p_{ij} \geq 0$. We need to assign jobs to machines and starting times such that job processing intervals do not overlap. The goal is to minimize $\sum_{s=1}^n w_j C_j^p$ where C_j is the completion time of job j in the schedule and $p \geq 1$. Using classical scheduling notation this problem can be denoted as $R||\sum_j w_j C_j^p$.

The problem $R||\sum_j w_j C_j^p$ is well studied for $p = 1$. It is known to be APX-hard [19] while the best known approximation algorithm has performance guarantee $3/2$ [30, 32]. For $p > 1$ even single machine scheduling problem is not understood. The $\mathcal{NP}$-hardness of $1||\sum_j w_j C_j^p$ is a well-known open problem in scheduling for all $p > 0, p \neq 1$. There is a long series of papers focusing on branch and bound algorithms for the problem (see [12, 17] for the recent work and references). The performance guarantees of the Smith's rule was analyzed in [18]. Constant factor approximation algorithms for more general functions of completion times in single machine setting were designed in [10, 34]. To the best of our knowledge there are no works designing approximation algorithms for such problems in the multiple machine setting.

We show how to use our framework for this problem in Section B. Our algorithm gives $2^p A_p$ approximation.

[1] A slight modification of the problem, where the objective is $\min \sum_{i\in[m]}(\sum_{j\in S_i} p_{ij})^q$, can be used for energy efficient scheduling. Imaging that we need to assign n jobs to m processors/cores so that all jobs are completed by a certain deadline D. We can run processors at different speeds s_i. To meet the deadlines we must set $s_i = D^{-1}\sum_{j\in S_i} p_{ij}$. The total power consumption is proportional to $D \times \sum_{i=1}^m s_i^q = D^{1-q} \times \sum_{i=1}^m \left(\sum_{j\in S_i} p_{ij}\right)^q$. For this problem, our algorithm gives A_q approximation.

2.4 Degree Balanced Spanning Tree Problem

We are given an undirected graph $G = (V, E)$ with edge weigths $w_e \geq 0$. The goal is to find a spanning tree T minimizing the objective function

$$f(T) = \sum_{v \in V} \left(\sum_{e \in \delta(v) \cap T} w_e \right)^q \tag{20}$$

where $\delta(v)$ is the set of edges in E incident to the vertex v. For $q = 2$, a more general problem was considered before in the Operations Research literature [5, 22, 24, 27] under the name of Adjacent Only Quadratic Spanning Tree Problem. A related problem, known as Degree Bounded Spanning Tree, recieved a lot of attention in Theoretical Computer Science [31, 16]. We are not aware of any previous work on Degree Balanced Spanning Tree Problem.

Let x_e be a boolean decision variable such that $x_e = 1$ if we choose edge $e \in E$ to be in our solution (tree) T. We formulate our problem as the following convex boolean optimization problem

$$\begin{aligned} \min \sum_{v \in V} \left(\sum_{e \in \delta(v)} w_e x_e \right)^q & \\ x \in \mathcal{B}(\mathcal{M}) & \\ x_e \in \{0, 1\}, & \qquad\qquad \forall e \in E, \end{aligned}$$

where $\mathcal{B}(\mathcal{M})$ is the base polymatroid polytope of the graphic matroid in graph G. We refer the reader to Schrijver's book [29] for the definition of the matroid. Using Theorem 1.3, we obtain an almost optimal fractional solution x^* of the LP relaxation (6)-(10) corresponding to the above integer problem.

Following Calinescu et al. [7], we define the continuous extension of the objective function (20) for any fractional solution x'

$$F(x') = \sum_{S \subseteq [n]} f(S) \prod_{e \in S} x'_e \prod_{e \notin S} (1 - x'_e),$$

i.e. $F(x')$ is equal to the expected value of the objective function (20) for the set of edges sampled independently at random with probabilities $x'_e, e \in E$. The function F cannot be computed exactly, but it can be approximated up to any factor $(1 + \varepsilon)$ via sampling. By Theorem 1.4, we derive $F(x^*) \leq A_q \cdot LP^*$, where LP^* is the value of the LP relaxation (6)–(10) on the fractional solution x^*.

The rounding phase of the algorithm implements the pipage rounding technique [1] adopted to polymatroid polytopes by Calinescu et al. [7]. Calinescu et al. [7] showed that given a matroid $\mathcal{M}$ and a fractional solution $x \in \mathcal{B}(\mathcal{M})$, one can efficiently find two elements, or two edges in our case, e' and e'' such that the new fractional solution $\tilde{x}(\varepsilon)$ defined as $\tilde{x}_{e'}(\varepsilon) = x_{e'} + \varepsilon$, $\tilde{x}_{e''}(\varepsilon) = x_{e''} - \varepsilon$ and $\tilde{x}_e(\varepsilon) = x_e$ for $e \notin \{e', e''\}$ is feasible in the base polymatroid polytope for small positive and for small negative values of ε.

They also showed that if the objective function $f(S)$ is submodular then the function of one variable $F(\tilde{x}(\varepsilon))$ is convex. In our case, the objective function $f(S)$ is supermodular which follows from a more general folklore statement.

Fact 2.1. *The function $f(S) = g(\sum_{i \in S} w_i)$ is supermodular if $w_i \geq 0$ for $i \in [n]$ and $g(x)$ is a convex function of one variable.*

Therefore, the function $F(\tilde{x}(\varepsilon))$ is concave. Hence, we can apply the pipage rounding directly: We start with the fractional solution x^*. At every step, we pick e' and e'' (using the algorithm from [7]) and move to $\tilde{x}(\varepsilon)$ with $\varepsilon = \varepsilon_1 = -\min\{x_{e'}, 1 - x_{e''}\}$ or $\varepsilon = \varepsilon_2 = \min\{1 - x_{e'}, x_{e''}\}$ whichever minimizes the concave function $F(\tilde{x}(\varepsilon))$ on the interval $[\varepsilon_1, \varepsilon_2]$. We stop when the current solution $\tilde{x}$ is integral.

At every step, we decrease the number of fractional variables x_e by at least 1. Thus, we terminate the algorithm in at most $|E|$ iterations. The value of the function $F(\tilde{x})$ never increases. So the cost of the final integral solution is at most the cost of the initial fractional solution x^*, which, in turn, is at most $A_q \cdot LP^*$.

Note, that we have not used any special properties of graphic matroids. The algorithm from [7] works for general matroids accessible through oracle calls. So we can apply our technique to more general problems where the objective is to minimize a function like (20) subject to base matroid constraints.

3 Proof of Theorem 1.3

We now give an efficient algorithm for finding $(1+\varepsilon)$ approximately optimal solution to the LP (6) - (10).

Proof of Theorem 1.3. We first transform our instance to make all d_{ij}'s integral and polynomially bounded in nk/ε. This can be done using a very standard technique: round down all d_{ij} to be multiples of $\varepsilon' = d\varepsilon/(3kqn)$, where $d = \max d_{ij}$. By doing so we may decrease the optimal value of the program by a factor of at most $(1+\varepsilon)$. Then, we rescale all d_{ij} by $1/\varepsilon'$ to make them integral. So, from now on, we will assume that all d_{ij} are integral and polynomially bounded.

Observe that for every $y \in \mathcal{P}$, there exists a z such that the pair (y,z) is a feasible solution to the LP (6) - (10). For example, one such z is defined as $z_{jS} = \prod_{i\in S} y_i \prod_{i\notin S}(1-y_i)$. Of course, this particular z may be suboptimal. However, it turns out, as we show below, that for every y, we can find the optimal z efficiently. Let us denote the minimal cost of the j-th term in (6) for a given $y \in \mathcal{P}$ by $F_j(y)$. That is, $F_j(y)$ is the cost of the following LP. The variables of the LP are z_{jS}. The parameters $y \in \mathcal{P}$ and $j \in [k]$ are fixed.

$$\min \sum_{S\subseteq[n]} \left(\sum_{i\in S} d_{ij}\right)^{q_j} z_{jS} \tag{21}$$

$$\sum_{S\subseteq[n]} z_{jS} = 1 \tag{22}$$

$$\sum_{S: i\in S} z_{jS} = y_i, \qquad \forall i \in [n] \tag{23}$$

$$z_{jS} \geq 0, \qquad \forall S \subseteq [n] \tag{24}$$

Now, the LP (6) - (10) can be equivalently rewritten as (below y is the variable).

$$\min \sum_{j\in[k]} F_j(y) \tag{25}$$

$$y \in \mathcal{P} \tag{26}$$

In Lemma 3.1 (see below), we prove that LP (21) - (24) can be solved in polynomial time, and thus the functions $F_j(x)$ can be computed efficiently. The functions $F_j(y)$ are convex (see Lemma C.1), so the minimum of the convex programming problem (25) - (25) can be found using the ellipsoid method. Once the optimal y^* is found, we find z^* by solving the LP (21) - (24) for y^*. $\square$

Lemma 3.1. *There exists a polynomial time algorithm for solving the LP (21) - (24).*

Proof. We write the dual LP. We introduce a variable ξ for constraint (22) and variables ν_i for constraints (23).

$$\max \ \ \xi + \sum_i \nu_i y_i \tag{27}$$

$$\xi + \sum_{i\in S} \nu_i \leq \Big(\sum_{i\in S} d_{ij}\Big)^{q_j} \qquad \forall S \subset [n] \tag{28}$$

The LP has exponentially many constraints. However, finding a violated constraint is easy. To do so, we guess $B^* = \sum_{i\in S^*} d_{ij}$ for the set S^* violating the constraint. That is possible, since all d_{ij} are polynomially bounded, and so is B^*. Then we solve the maximum knapsack problem

$$\max_{S\subseteq[n]} \sum_{i\in S} \nu_i y_i$$
$$\sum_{i\in S} d_{ij} \leq B^*$$

using the standard dynamic programming algorithm and obtain the optimal set S. The knapsack problem is polynomially solvable, since B^* is polynomially bounded. If $\xi + \sum_{i\in S} \nu_i y_i \leq B^*$, then constraint (28) is violated for the set S; otherwise all constraints (28) are satisfied. □

4 Proof of Theorem 1.4

In this section, we prove the main theorem – Theorem 1.4.

Proof of Theorem 1.4. The theorem easily follows from the decoupling inequality (Theorem 1.2). Consider a feasible solution (y^*, z^*) to the IP (3) - (5). We prove inequality (11) term by term. That is, for every j we show that

$$\mathbb{E}\Big[\Big(\sum_{i\in D_j} d_{ij} R_i(y^*)\Big)^{q_j}\Big] \leq A_{q_j} \sum_{S\subseteq[n]} \left(\sum_{i\in S\cap D_j} d_{ij}\right)^{q_j} z^*_{jS}. \tag{29}$$

Recall that $D_j = \{i : d_{ij} \neq 0\}$. Above, we dropped terms with $i \notin D_j$, since if $i \notin D_j$, then $d_{ij} = 0$.

Fix a $j \in [n]$. Define random variables Y_i for $i \in D_j$ as follows: Pick a random set $S \subset [n]$ with probability z_{jS}, and let $Y_i = d_{ij}$ if $i \in S$, and $Y_i = 0$ otherwise. Note that random variables Y_i are dependent. We have

$$\Pr(Y_i = d_{ij}) = \sum_{S:i\in S} \Pr(S) = \sum_{S:i\in S} z^*_{jS} = y^*_i.$$

It is easy to see that

$$\mathbb{E}\Big[\Big(\sum_{i\in[n]} Y_i\Big)^{q_j}\Big] = \sum_{S\subseteq[n]} \Big(\sum_{i\in S} d_{ij}\Big)^{q_j} z_{jS}.$$

The right hand side is simply the definition of the expectation on the left hand side. Now, let $X_i = d_{ij}\, R_i(y^*)$ for $i \in D_j$. Note that by conditions of the theorem, $\Pr(X_i = d_{ij}) = y^*_i = \Pr(Y_i = d_{ij})$ (by condition (1)). Thus, each X_i has the same distribution as Y_i. Furthermore, X_i's are independent (by condition (2)). Therefore, we can apply the decoupling inequality

$$\mathbb{E}\Big[\Big(\sum_{i\in D_j} X_i\Big)^{q_j}\Big] \leq A_{q_j} \mathbb{E}\Big[\Big(\sum_{i\in D_j} Y_i\Big)^{q_j}\Big].$$

The left hand side of the inequality equals the left hand side of (29), the right hand side of the inequality equals the right hand side of (29). Hence, inequality (29) holds. □

5 Decoupling Inequality

In this section, we prove the decoupling inequality (Theorem 1.2) with the optimal constant $C_q = A_q^{1/q}$.

Theorem 1.2. *Let $Y_1, \dots, Y_n$ be jointly distributed nonnegative (non-independent) random variables, and let $X_1, \dots, X_n$ be independent random variables such that each X_i has the same distribution as Y_i. Then, for every $q \geq 1$,*

$$\|\sum_{i=1}^n X_i\|_q \leq A_q^{1/q} \|\sum_{i=1}^n Y_i\|_q,$$

where A_q is the fractional Bell number which equals the q-th moment of the Poisson random variable with parameter 1 (see (1)). The constant $A_q^{1/q}$ is the optimal constant.

Proof. We first show that we cannot replace A_q with a smaller constant. Consider the following example. Let $Y_i^{(n)}, i \in [n]$ be random variables taking value 1 with probability $1/n$, and 0 with probability $1 - 1/n$. We generate $Y_i^{(n)}$'s as follows. We pick a random $j \in [n]$ and let $Y_j = 1$ and $Y_i = 0$ for $i \neq j$. Random variables $X_i^{(n)}$ are i.i.d. Bernoulli random variables with $\mathbb{E}[X_i^{(n)}] = 1/n$. Then, the sum $\sum_{i=1}^n Y_i^{(n)}$ always equals 1, and $\|\sum_{i=1}^n Y_i^{(n)}\|_q = 1$. As $n \to \infty$, the sum $\sum_{i=1}^n X_i^{(n)}$ converges in distribution to a Poisson random variable with parameter 1. Thus, $\|\sum_{i=1}^n Y_i^{(n)}\|_q \to A_q^{1/q}$, and, hence, the constant $A_q^{1/q}$ cannot be improved.

We prove this theorem for finite random variables. The general case can be obtained using a standard argument by approximating random variables $Y_1, \dots, Y_n$ with discrete random variables[2]. Consider the finite probability space $(\Omega, \Pr)$ on which the random variables $Y_1, \dots Y_n$ are defined. Without loss of generality we may assume that $\Omega = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n$, where $\mathcal{Y}_i$ is the range of the random variable Y_i. Then every elementary event is a vector $\omega = (y_1, \dots, y_n) \in \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n$. The probability of the event $\omega = (y_1, \dots, y_n)$ equals

$$\Pr(\omega) = \Pr(Y_1 = y_1, \dots, Y_n = y_n).$$

Let $f_i(\omega)$ be the i-th coordinate of ω (or, more generally, the value the random variable Y_i takes when the elementary event $\omega \in \Omega$ occurs). For each $\omega \in \Omega$, we define $n+1$ independent random variables $P_\omega^i, i \in [n]$ and P_ω on a new probability space Ω'. Each P_ω^i and P_ω is a Poisson random variable with parameter $\lambda_\omega = \Pr(\omega)$. Then, $\mathbb{E}[P_\omega^i] = \mathbb{E}[P_\omega] = \Pr(\omega)$.

We prove the following inequalities that imply the theorem.

1. $$\|\sum_{i=1}^n X_i\|_q \leq \|\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega^i\|_q;$$

2. $$\|\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega^i\|_q \leq \|\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega\|_q;$$

3. $$\|\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega\|_q \leq A_q^{1/q} \|\sum_{i=1}^n Y_i\|_q.$$

We split the proof into three main lemmas.

Lemma 5.1. *Inequality 1 holds.*

[2] In this paper, we only use the discrete version of this inequality. We give the details in the full version of the paper.

Proof. We prove by induction on n the following inequality: for every $B \geq 0$,

$$\left\|B + \sum_{i=1}^{n} X_i\right\|_q \leq \left\|B + \sum_{\omega\in\Omega} \sum_{i=1}^{n} f_i(\omega) P_\omega^i\right\|_q.$$

For $n = 0$ this inequality trivially holds. For $n \geq 1$, we write

$$\begin{aligned}\left\|B + \sum_{i=1}^{n} X_i\right\|_q^q = \mathbb{E}\Big[\big(B + \sum_{i=1}^{n} X_i\big)^q\Big] &= \mathbb{E}_{X_n}\, \mathbb{E}\Big[\big((B + X_n) + \sum_{i=1}^{n-1} X_i\big)^q \quad \big| \quad X_n\Big] \\ &\leq \mathbb{E}_{X_n}\, \mathbb{E}\Big[\big((B + X_n) + \sum_{i=1}^{n-1} \sum_{\omega\in\Omega} f_i(\omega) P_\omega^i\big)^q \quad \big| \quad X_n\Big] \\ &= \mathbb{E}\Big[\big((B + X_n) + \sum_{i=1}^{n-1} \sum_{\omega\in\Omega} f_i(\omega) P_\omega^i\big)^q\Big].\end{aligned}$$

Here, we used the inductive hypothesis with $B_* = B + X_n$. Let

$$S_{n-1} = \sum_{i=1}^{n-1} \sum_{\omega\in\Omega} f_i(\omega) P_\omega^i.$$

We need to show that

$$\mathbb{E}\Big[\big(B + S_{n-1} + X_n\big)^q\Big] \leq \mathbb{E}\Big[\big(B + S_{n-1} + \sum_{\omega\in\Omega} f_n(\omega) P_\omega^n\big)^q\Big]. \tag{30}$$

We condition on S_{n-1} and prove that this inequality holds for every fixed S_{n-1}. Note that X_n and P_ω^n are independent from S_{n-1}. Let $B_\circ = B + S_{n-1}$. Since X_n and Y_n are identically distributed, we can replace X_n with Y_n. Thus, inequality (30) is equivalent to

$$\mathbb{E}\Big[\big(B_\circ + Y_n\big)^q\Big] \leq \mathbb{E}\Big[B_\circ + \sum_{\omega\in\Omega} f_n(\omega) P_\omega^n\big)^q\Big].$$

Define a linear function $l : \mathbb{R}^\Omega \to \mathbb{R}$ and a non-linear function $g : \mathbb{R}^\Omega \to \mathbb{R}$ as follows: for $v \in \mathbb{R}^\Omega$ (the coordinates of v are indexed by $\omega \in \Omega$),

$$\begin{aligned}l(v) &= B_\circ^q + \sum_{\omega\in\Omega} \big((f_n(\omega) + B_\circ)^q - B_\circ^q\big) v_\omega; \\ g(v) &= (B_\circ + \sum_{\omega\in\Omega} f_n(\omega) v_\omega)^q.\end{aligned}$$

Note that if exactly one coordinate of v equals to 1, and all other coordinates are equal to 0, then $g(v) = l(v)$. By Lemma 5.2 (see below), if all coordinates of v are nonnegative integers then $g(v) \geq l(v)$. Let χ_ω be the indicator random variable of the elementary event ω; and $\chi \in \mathbb{R}^\Omega$ be the vector with coordinates χ_ω. Then,

$$(B_\circ + Y_n)^q \equiv (B_\circ + \sum_{\omega\in\Omega} f_n(\omega) \chi_\omega)^q = l(\chi).$$

For $\chi = (\chi_\omega)$ and $P^n = (P_\omega^n)$, we have

$$\begin{aligned}\mathbb{E}\Big[\big(B_\circ + X_n\big)^q\Big] &= \mathbb{E}\big[l(\chi)\big] = l(\mathbb{E}[\chi]); \\ \mathbb{E}\Big[\big(B_\circ + \sum_{\omega\in\Omega} f_n(\omega) P_\omega^n\big)^q\Big] &= \mathbb{E}[g(P^n)] \geq \mathbb{E}[l(P^n)] = l(\mathbb{E}[P^n]).\end{aligned}$$

However, $\mathbb{E}[\chi_\omega] = \Pr(\omega) = \mathbb{E}[P^n_\omega]$. Hence, $\mathbb{E}[\chi] = \mathbb{E}[P^n]$, and

$$\mathbb{E}\Big[\big(B_\circ + X_n\big)^q\Big] \le \mathbb{E}\Big[\big(B_\circ + \sum_{\omega\in\Omega} f_n(\omega) P^n_\omega\big)^q\Big].$$

This finishes proof. $\square$

Lemma 5.2. *If all coordinates of v are nonnegative integers then $g(v) \ge l(v)$.*

Proof. Suppose that $g(v) \ge l(v)$ for some v. Fix a coordinate $\omega^* \in \Omega$, and let $\tilde{v}_\omega = v_\omega$ for $\omega \ne \omega^*$ and $\tilde{v}_{\omega^*} = v_{\omega^*} + 1$. That is, $\tilde{v}$ alters from v only in the coordinate ω^*. We show that $g(\tilde{v}) \ge l(\tilde{v})$. Consider the function $h(t) = (t + f_n(\omega^*))^q - t^q$. This function is increasing for $t \ge 0$:

$$h'(t) = \big((t + f_n(\omega^*))^q - t^q\big)' = q\big((t + f_n(\omega^*))^{q-1} - t^{q-1}\big) > 0.$$

We have

$$g(\tilde{v}) - g(v) = h\big(B_\circ + \sum_{\omega\in\Omega} f_n(\omega) v_\omega\big) \ge h(B_\circ) = (B_\circ + f_n(\omega))^q - B_\circ^q = l(\tilde{v}) - l(v).$$

Hence, $g(\tilde{v}) \ge l(\tilde{v})$.

For the zero vector $v = 0$, $g(v) = l(v)$. We can reach any nonnegative integer vector v, if we start from the zero vector and increment by 1 one coordinate at a time. Thus, $g(v) \ge l(v)$ for all nonnegative integer v. $\square$

Lemma 5.3. *Inequality 2 holds.*

We prove a slightly more general statement.

Lemma 5.4. *Let P^i_j, P_j be independent nonnegative random variables (for $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$), let f^i_j be a sequence of nonnegative real numbers; and let $B \in \mathbb{R}^+$ be a nonnegative real number. Suppose that each P^i_j has the same distribution as P_j and $q \ge 1$. Then*

$$\mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f^i_j P^i_j\Big)^q\Big] \le \mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f^i_j P_j\Big)^q\Big]. \tag{31}$$

To get Lemma 5.3 we apply Lemma 5.4 with $f^i_j = f_i(\omega_j)$, $P^i_j = P^i_{\omega_j}$, $B = 0$, where $\omega_1, \dots, \omega_m$ is an arbitrary ordering of elementary events in Ω.

Proof. We prove Lemma 5.4 by induction on m. For $m = 0$, inequality (31) trivially holds. Denote $F_j = \sum_{i=1}^n f^i_j$. We need to prove that

$$\mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f^i_j P^i_j\Big)^q\Big] \le \mathbb{E}\Big[\Big(B + \sum_{j=1}^m F_j P_j\Big)^q\Big].$$

Write

$$\begin{aligned}
\mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f^i_j P^i_j\Big)^q\Big] &= \mathbb{E}_{P^1_m,\dots,P^n_m}\, \mathbb{E}\Big[\Big(\big(B + \sum_{i=1}^n f^i_m P^i_m\big) + \sum_{j=1}^{m-1}\sum_{i=1}^n f^i_j P^i_j\Big)^q \;\mid\; P^1_m, \dots, P^n_m\Big] \\
&\le \mathbb{E}_{P^1_m,\dots,P^n_m}\, \mathbb{E}\Big[\Big(\big(B + \sum_{i=1}^n f^i_m P^i_m\big) + \sum_{j=1}^{m-1} F_j P_j\Big)^q \;\mid\; P^1_m, \dots, P^n_m\Big] \\
&= \mathbb{E}\Big[\Big(\big(B + \sum_{i=1}^n f^i_m P^i_m\big) + \sum_{j=1}^{m-1} F_j P_j\Big)^q\Big].
\end{aligned}$$

Here, we used the inductive hypothesis with $B_* = \big(B + \sum_{i=1}^n f_m^i P_m^i\big)$. Denote by $B_\circ$ the random variable $B + \sum_{j=1}^{m-1} F_j P_j$. Then,

$$\mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f_j^i P_j^i\Big)^q\Big] \le \mathbb{E}\Big[\Big(B_\circ + \sum_{i=1}^n f_m^i P_m^i\Big)^q\Big].$$

Using convexity of the function $t \mapsto t^q$ for $q \ge 1$, we get

$$\Big(B_\circ + \sum_{i=1}^n f_m^i P_m^i\Big)^q = \Big(\sum_{i=1}^n \frac{f_m^i}{F_m}\big(B_\circ + F_m P_m^i\big)\Big)^q \le \sum_{i=1}^n \frac{f_m^i}{F_m}\Big(B_\circ + F_m P_m^i\Big)^q.$$

Each term $(B_\circ + F_m P_m^i)$ is distributed as $(B_\circ + F_m P_m)$, hence $\mathbb{E}[B_\circ + F_m P_m^i] = \mathbb{E}[B_\circ + F_m P_m]$, and

$$\mathbb{E}\Big[\Big(B + \sum_{j=1}^m \sum_{i=1}^n f_j^i P_j^i\Big)^q\Big] \le \mathbb{E}\Big[\Big(B_\circ + \sum_{i=1}^n f_m^i P_m\Big)^q\Big] = \mathbb{E}\Big[\Big(B + \sum_{j=1}^m F_j P_j\Big)^q\Big].$$

This concludes the proof. □

Lemma 5.5. *Inequality 3 holds.*

Proof. Let

$$P = \sum_{\omega \in \Omega} P_\omega.$$

The random variable P has the Poisson distribution with parameter 1, since $\sum_\omega \mathbb{E}[\omega] = \sum_\omega \Pr(\omega) = 1$. We have

$$\begin{aligned} \Big\| \sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega \Big\|_q^q &= \mathbb{E}\Big[\Big(\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega\Big)^q\Big] \\ &= \sum_{k=1}^\infty \mathbb{E}\Big[\Big(\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) P_\omega\Big)^q \;\Big|\; P = k\Big] \cdot \Pr(P = k) \\ &= \sum_{k=0}^\infty \mathbb{E}\Big[\Big(\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega) \frac{P_\omega}{k}\Big)^q \;\Big|\; P = k\Big] \cdot \big(k^q \cdot \Pr(P = k)\big) \end{aligned} \tag{32}$$

Using convexity of the function $t \mapsto t^q$ for $q \ge 1$, we upper bound each term in the sum as follows:

$$\begin{aligned} \mathbb{E}\Big[\Big(\sum_{\omega\in\Omega}\sum_{i=1}^n f_i(\omega)\frac{P_\omega}{k}\Big)^q \;\Big|\; P = k\Big] &\le \mathbb{E}\Big[\sum_{\omega\in\Omega}\frac{P_\omega}{k}\Big(\sum_{i=1}^n f_i(\omega)\Big)^q \;\Big|\; P = k\Big] \\ &= \sum_{\omega\in\Omega}\mathbb{E}\Big[\frac{P_\omega}{k} \;\Big|\; P = k\Big] \cdot \Big(\sum_{i=1}^n f_i(\omega)\Big)^q. \end{aligned}$$

We observe that $\mathbb{E}[P_\omega \mid P = k] = k \Pr(\omega)$, which follows from the following well known fact (see e.g., Feller [14], Section IX.9, Problem 6(b), p. 237).

Fact 5.6. *Suppose P_1 and P_2 are independent Poisson random variables with parameters λ_1 and λ_2. Then, for every $k \in \mathbb{N}$,*

$$\mathbb{E}\big[P_1 \mid P_1 + P_2 = k\big] = \frac{\lambda_1}{\lambda_1 + \lambda_2}\, k.$$

In our case, $P_1 = P_\omega$, $P_2 = \sum_{\omega' \neq \omega} P_\omega$, $P_1 + P_2 = P$. Therefore, we have

$$\mathbb{E}\Big[\Big(\sum_{\omega\in\Omega}\sum_{i=1}^{n} f_i(\omega)\frac{P_\omega}{k}\Big)^q \quad | \quad P = k\Big] \leq \sum_{\omega\in\Omega}\Pr(\omega)\Big(\sum_{i=1}^{n} f_i(\omega)\Big)^q.$$

Plugging this inequality in (32), we obtain the desired bound

$$\begin{aligned}\|\sum_{\omega\in\Omega}\sum_{i=1}^{n} f_i(\omega)P_\omega\|_q^q &\leq \sum_{\omega\in\Omega}\Pr(\omega)\Big(\sum_{i=1}^{n} f_i(\omega)\Big)^q \cdot \sum_{k=1}^{\infty} k^q \cdot \Pr(P = k)\\ &= \mathbb{E}\Big[\big(\sum_{i=1}^{n} Y_i\big)^q\Big] \cdot \mathbb{E}\big[P^q\big] = \|\sum_{i=1}^{n} Y_i\|_q^q \cdot \|P\|_q^q.\end{aligned}$$

□

□

6 Generalizations

We can extend our results to a more general class of objective functions. Using our framework, we can solve combinatorial optimization problems with the objective function

$$\sum_{j\in[k]} f_j\big(\sum_{i\in[n]} d_{ij} y_i\big), \tag{33}$$

where f_j's are arbitrary increasing *convex* functions satisfying $f_j(0) = 0$. In this case, the approximation ratio equals $A_{\{f\}} = \mathbb{E}[a_{\{f\}}(P_1)]$, where P_1 is a Poisson random variable with parameter 1, and the function $a_{\{f\}}(t)$ is defined as $a_{\{f\}}(t) = \max\{f_j(tx)/f_j(x) : x > 0,\ j \in [k]\}$ for $t \in \mathbb{N}$. Note, that for $f(s) = \mu s^q$, $a_f(t) = t^q$.

Similarly, we can solve maximization problems with the objective function (33) if f_j's are arbitrary non-decreasing *concave* functions satisfying $f_j(0) = 0$. The approximation ratio equals $B_{\{f\}} = \mathbb{E}[b_{\{f\}}(P_1)]$, where $b_{\{f\}}(t) = \min\{f_j(tx)/f_j(x) : x > 0,\ j \in [k]\}$ for $t \in \mathbb{N}$. It is not hard to see that $B_{\{f\}} \geq (e-1)/e$. Indeed, in the worst case, $b_{\{f\}}(t) = 1$ for $t \geq 1$ and $b_{\{f\}}(t) = 0$ for $t = 0$, then $\mathbb{E}[b_{\{f\}}(P_1)] = \Pr(P_1 \geq 1) = 1 - 1/e$. This happens e.g., for the function $f(s) = \min\{s, 1\}$. Note that the approximation ratio of $(e-1)/e \approx 0.632$ for maximization problems of this form was previously known (see Calinescu et al. [7]). However, for some functions f we get a better approximation. For example, for $f(s) = \sqrt{s}$, we get an approximation ratio of $B_{\sqrt{s}} \approx 0.773$.

References

[1] Alexander A. Ageev and Maxim Sviridenko, Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee, J. Comb. Optim. 8(3): 307-328, (2004).

[2] S. Albers, Energy-efficient algorithms, Commun. ACM 53(5), 86-96 (2010).

[3] Matthew Andrews, Antonio Fernandez Anta, Lisa Zhang, Wenbo Zhao: Routing for Power Minimization in the Speed Scaling Model. IEEE/ACM Trans. Netw. 20(1): 285-294 (2012).

[4] Matthew Andrews, Spyridon Antonakopoulos, Lisa Zhang: Minimum-Cost Network Design with (Dis)economies of Scale. FOCS 2010: 585-592.

[5] A. Assad and W. Xu. The quadratic minimum spanning tree problem. Naval Research Logistics v39. (1992), 399-417.

[6] Yossi Azar, Amir Epstein. Convex programming for scheduling unrelated parallel machines. STOC 2005, pp. 331-337.

[7] Gruia Calinescu, Chandra Chekuri, Martin Pal and Jan Vondrak, Maximizing a submodular set function subject to a matroid constraint, SIAM Journal on Computing 40:6 (2011), pp. 1740-1766.

[8] V. Chvatal, On certain polytopes associated with graphs, J. Combinatorial Theory Ser. B, 18 (1975), pp. 138?154.

[9] Evripidis Bampis, Alexander Kononov, Dimitrios Letsios, Giorgio Lucarelli and Maxim Sviridenko, Energy Efficient Scheduling and Routing via Randomized Rounding, in Proceedings of FSTTCS 2013, pp. 449-460.

[10] Nikhil Bansal and Kirk Pruhs, The Geometry of Scheduling. FOCS 2010, pp.407-414.

[11] Stephen Boyd and Lieven Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[12] C. Durr and O. Vasquez, Order constraints for single machine scheduling with non-linear cost, The 16th Workshop on Algorithm Engineering and Experiments (ALENEX), 2014.

[13] M. Grotschel, L. Lovasz and A. Schrijver. Geometric algorithms and combinatorial optimization. Springer-Verlag, Berlin, 1988.

[14] W. Feller. An Introduction to Probability Theory and its Applications. (3rd edition). John Wiley & Sons, New York (1968).

[15] D. R. Fulkerson. Blocking and anti-blocking pairs of polyhedra, Math. Programming, 1 (1971), pp. 168-194.

[16] Michel X. Goemans. Minimum Bounded Degree Spanning Trees. FOCS 2006, 273-282.

[17] W.Hohn and T. Jacobs. An experimental and analytical study of order constraints for single machine scheduling with quadratic cost. In Proc. of the 14th Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 103-117, SIAM, 2012.

[18] W. Hohn and T.Jacobs. On the performance of Smith's rule in single-machine scheduling with nonlinear cost. In Proc. of the 10th Latin American Theoretical Informatics Symposium (LATIN), LNCS, Vol. 7256, pp. 482-493, Springer, 2012.

[19] Han Hoogeveen, Petra Schuurman, Gerhard J. Woeginger. Non-Approximability Results for Scheduling Problems with Minsum Criteria, INFORMS Journal on Computing 13(2), pp. 157-168 (2001).

[20] S. Irani and K. Pruhs, Algorithmic problems in power management, ACM SIGACT News 36 (2), 63-76.

[21] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan. A Unified Approach to Scheduling on Unrelated Parallel Machines, Journal of the ACM, Vol. 56, 2009.

[22] S. Maia, E. Goldbarg, and M. Goldbarg, On the biobjective adjacent only quadratic spanning tree problem. Electronic Notes in Discrete Mathematics, 41, (2013), 535-542.

[23] Yurii Nesterov and Arkadi Nemirovski. Interior-point polynomial algorithms in convex programming. Society for Industrial and Applied Mathematics, 1987.

[24] T. Oncan and A. Punnen. The quadratic minimum spanning tree problem: A lower bounding procedure and an efficient search algorithm. Computers and Operations Research v.37, (2010), 1762–1773.

[25] Victor de la Peña. Bounds on the expectation of functions of martingales and sums of positive RVs in terms of norms of sums of independent random variables. Proceedings of the American Mathematical Society 108.1 (1990): 233-239.

[26] Victor de la Peña and E. Giné, Decoupling: from dependence to independence. Springer, 1999.

[27] D. Pereira, M. Gendreau, A. Cunha, Stronger lower bounds for the quadratic minimum spanning tree problem with adjacency costs. Electronic Notes in Discrete Mathematics, 41, (2013), 229-236.

[28] A. Schrijver. Theory of Linear and Integer Programming. Wiley, 1998.

[29] A. Schrijver. Combinatorial Optimization. Springer, 2002.

[30] Andreas S. Schulz and Martin Skutella, Scheduling Unrelated Machines by Randomized Rounding, SIAM J. Discrete Math, 15(4), pp. 450-469 (2002).

[31] Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. STOC 2007: 661-670.

[32] Martin Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. J. ACM 48(2), 206-242 (2001).

[33] Martin Skutella. Approximation and randomization in scheduling. Technische Universitt Berlin, Germany, 1998.

[34] S. Stiller and A. Wiese. Increasing Speed Scheduling and Flow Scheduling. ISAAC (2) 2010, pp.279-290.

[35] A. Wierman, L. L. H. Andrew, and A. Tang. Power-aware speed scaling in processor sharing systems. In INFOCOM, pages 20072015, 2009.

Appendix

A Corollary A.1

In this section, we prove a corollary of Theorem 1.2, which we will need in the next section.

Corollary A.1. *Let $Y_1, \ldots, Y_n$ be jointly distributed (non-independent) nonnegative integral random variables, and let $X_1, \ldots, X_n$ be independent Bernoulli random variables taking values 0 and 1 such that for each i, $\mathbb{E}[X_i] = \mathbb{E}[Y_i]$. Then, for every $q \geq 1$,*

$$\|\sum_{i=1}^{n} X_i\|_q \leq A_q^{1/q} \|\sum_{i=1}^{n} Y_i\|_q,$$

where A_q is the fractional Bell number which equals the q-th moment of the Poisson random variable with parameter 1 (see (1)).

Proof. Consider independent random variables X'_i such that each X'_i is distributed as Y_i. Then, by Theorem 1.2,

$$\big\|\sum_{i=1}^{n} X'_i\big\|_q \leq A_q^{1/q}\,\big\|\sum_{i=1}^{n} Y_i\big\|_q.$$

Now, we prove by induction n that for every $B \geq 0$,

$$\big\|B + \sum_{i=1}^{n} X_i\big\|_q \leq \big\|B + \sum_{i=1}^{n} X'_i\big\|_q.$$

For $n = 0$ the statement trivially holds. For $n > 0$, we have (similarly to the proof of Lemma 5.1),

$$\big\|B + \sum_{i=1}^{n} X_i\big\|_q = \big\|(B + X_n) + \sum_{i=1}^{n-1} X_i\big\|_q \leq \|(B + X_n) + \sum_{i=1}^{n-1} X'_i\big\|_q.$$

Let $S_{n-1} = B + \sum_{i=1}^{n-1} X'_i$. We show that $\|S_{n-1} + X_n\|_q \leq \|S_{n-1} + X'_n\|_q$ for any fixed S_{n-1} (note, that X_n and X'_n are independent from S_{n-1}). Define a linear function $l(t)$ as follows

$$l(t) = S_{n-1}^q + ((S_{n-1} + 1)^q - S_{n-1}^q)\,t.$$

Then, $l(t) = (S_{n-1} + t)^q$ for $t \in \{0, 1\}$ and $l(t) < (S_{n-1} + t)^q$ for $t \in \mathbb{N} \setminus \{0, 1\}$. Hence,

$$\|S_{n-1} + X_n\|_q^q = \mathbb{E}[(S_{n-1} + X_n)^q] = \mathbb{E}[l(X_n)] = \mathbb{E}[l(X'_n)] \leq \mathbb{E}[(S_{n-1} + X'_n)^q] = \|S_{n-1} + X_n\|_q^q.$$

□

B Unrelated Parallel Machine Scheduling with Nonlinear Functions of Completion Times – Technical Details

We consider the following linear programming relaxation of the scheduling problem $R||\sum_j w_j C_j^p$. The variable $x_{ijt} = 1$ if job j starts at time t on machine i. For convenience we assume that $x_{ijt} = 0$ for negative values of t.

$$\min \sum_{i\in[m]} \sum_{j\in[n]} \sum_{t\geq 0} w_j (t + p_{ij})^p x_{ijt} \tag{34}$$

$$\sum_{i\in[m], t\geq 0} x_{ijt} = 1 \qquad \forall j \in [n] \tag{35}$$

$$\sum_{j\in[n]} \sum_{\tau=t-p_{ij}+1}^{t} x_{ij\tau} \leq 1 \qquad \forall i \in [m], t \geq 0 \tag{36}$$

$$x_{ijt} \geq 0 \qquad \forall i \in [m], j \in [n], t \geq 0. \tag{37}$$

The constraints (35) say that each job must be assigned, the constraint (36) says that at most one job can be processed in a unit time interval on each machine. Such linear programming relaxation are known under the name of *strong time indexed formulations*. The standard issue with such relaxations is that they have pseudo-polynomially many variables due to potentially large number of indices t. One way to handle this issue is to partition the time interval into intervals $((1+\varepsilon)^k, (1+\varepsilon)^{k+1}]$ and round all completion times to the endpoints of such intervals. This method leads to polynomially sized linear programming relaxations with $(1 + O(\varepsilon))$-loss in the performance guarantee (see [33] for detailed description of the method). From now on we ignore this issue and assume that the planning horizon upper bound $\sum_{i,j} p_{ij}$ is polynomially bounded in the input size.

Algorithm. Our approximation algorithm solves the linear programming relaxation (34) - (37). Let x^* be the optimal fractional solution of the LP. Each job is tentatively assigned to machine i to start at time t with probability x^*_{ijt}, independently at random. Let t_j be the tentative start time assigned to job j by our randomized procedure. Finally, we process jobs assigned to each machine in the order of the tentative completion times $t_j + p_{ij}$.

Analysis. We estimate the expected cost of the approximate solution returned by the algorithm. We denote the expected cost by APX. For each machine-job-tentative time triple (i, j, t), let J_{ijt} be the set of triples (i, j', t') such that $t' + p_{ij'} \leq t + p_{ij}$. Let X^{ijt} be the random boolean variable such that $X^{ijt} = 1$ if job j is assigned to machine i with tentative start time t. In addition, let $Z^{ijt}_{j'}$ be the random boolean variable such that $Z^{ijt}_{j'} = 1$ if job j' is assigned to machine i with tentative start time t' for some $(i, j', t') \in J_{ijt}$ by our randomized rounding procedure. Then,

$$\Pr\left(Z^{ijt}_{j'} = 1\right) = \sum_{t':(i,j',t')\in J_{ijt}} x^*_{ij't'}.$$

Suppose that job j is tentative scheduled on machine i at time t i.e., $X^{ijt} = 1$. We start processing job j after all jobs j' tentative scheduled on machine i at time t' with $t' + p_{ij'} \leq t + p_{ij}$ are finished. Thus the weighted expected completion time to the power of p for j equals (given $X^{ijt} = 1$)

$$\begin{aligned} \mathbb{E}\left[w_j C_j^p \quad \vert \quad X^{ijt} = 1\right] &= \mathbb{E}\Big[\Big(\sum_{j'\in[n]\setminus\{j\}} p_{i,j'} Z^{ijt}_{j'} + p_{ij}\Big)^p \quad \vert \quad X^{ijt} = 1\Big] \\ &= \mathbb{E}\Big[\Big(\sum_{j'\in[n]\setminus\{j\}} p_{i,j'} Z^{ijt}_{j'} + p_{ij}\Big)^p\Big]. \end{aligned}$$

In the second equality, we used that random variables $Z^{ijt}_{j'}$ are independent from the random variable X^{ijt}. Then,

$$\begin{aligned} APX &= \sum_{j\in[n]}\sum_{i\in[m]}\sum_{t\geq 0} w_j \mathbb{E}\left[w_j C_j^p \quad \vert \quad X^{ijt} = 1\right] \Pr\left(X^{ijt} = 1\right) \\ &= \sum_{j\in[n]}\sum_{i\in[m]}\sum_{t\geq 0} w_j \mathbb{E}\Big[\Big(\sum_{j'\in[n]\setminus\{j\}} p_{i,j'} Z^{ijt}_{ij'} + p_{ij}\Big)^p\Big]\ x^*_{ijt}. \end{aligned} \tag{38}$$

Note, that for fixed $i \in [m], j \in [n], t \geq 0$ random variables $Z^{ijt}_{j'}$ are independent from each other. We claim that

$$\mathbb{E}\Big[\Big(\sum_{j'\in[n]\setminus\{j\}} p_{i,j'} Z^{ijt}_{j'} + p_{ij}\Big)^p\Big] \leq A_p (t + 2p_{ij})^p. \tag{39}$$

Combining (38) and (39), we derive that the performance guarantee of our approximation algorithm is at most $2^p A_p$. We now prove inequality (39).

Let G_{ijt} be the interval graph where the vertex set $V(G_{ijt})$ is the collection of intervals corresponding to triples in $(i, j', t') \in J_{ijt}$ such that $x^*_{ij't'} > 0$. More precisely, every triple $(i, j', t') \in J_{ijt}$ corresponds to the interval $I_{ij't'} = [t', t' + p_{ij'})$ with corresponding weight $x^*_{ij't'} > 0$. Let $\mathcal{I}$ be the collection of all independent sets in G_{ijt}. The interval graph G_{ijt} is perfect and the weights $x^*_{ij't'}$ satisfy the constraints (36), so there is a collection of weights $\lambda_C \geq 0$, $C \in \mathcal{I}$ (for more formal argument see below) such that

$$\sum_{C\in\mathcal{I}} \lambda_C = 1,$$

$$\sum_{C\in\mathcal{I}:I_{ij't'}\in C} \lambda_C = x^*_{ij't'}, \qquad \forall (i, j', t') \in J_{ijt}$$

Formally, the claim above follows from the polyhedral characterization of perfect graphs proved by Fulkerson [15] and Chvatal [8] (see also Schrijver's book [28], Section 9, Application 9.2 on p. 118) that a graph G is perfect if and only if its stable set polytope is defined by the system below:

$$\begin{aligned} \sum_{v \in C} x_v &\le 1, && \text{for each clique } C, \\ x(v) &\ge 0, && \text{for each } v \in V. \end{aligned}$$

In the interval graph G_{ijt} all clique inequalities are included in the constraints (36) and therefore any set of weights $x^*_{ij't'}$ can be decomposed into a convex combination of independent sets in G_{ijt}.

We define a random variable $Y^{ijt}_{j'}$ as follows: Sample an independent set $C \in \mathcal{I}$ with probability λ_C and let

$$Y^{ijt}_{j'} = \left|\{I_{ij't'} \in C\}\right| .$$

Note that one job j' may have more than one interval $I_{ij't'}$ in the set C (for different t'). Random variables $Y^{ijt}_{j'}$ may be dependent but

$$\mathbb{E}[Y^{ijt}_{j'}] = \sum_{(i,j',t') \in J_{ijt}} x^*_{ij't'} = \mathbb{E}[Z^{ijt}_{j'}].$$

Therefore, by Corollary A.1 we have

$$\mathbb{E}\Big[\Big(\sum_{j' \in [n] \setminus \{j\}} p_{i,j'} Z^{ijt}_{j'} + p_{ij}\Big)^p\Big] \le A_p \mathbb{E}\Big[\Big(\sum_{j' \in [n] \setminus \{j\}} p_{i,j'} Y^{ijt}_{j'} + p_{ij}\Big)^p\Big]. \tag{40}$$

Now, observe, that $\sum_{j' \in [n] \setminus \{j\}} p_{i,j'} Y^{ijt}_{j'}$ is always bounded by $t + p_{ij}$, because all intervals in C are disjoint (C is an independent set) and all intervals are subsets of $[0, t + p_{ij}]$. Hence,

$$\mathbb{E}\Big[\Big(\sum_{j' \in [n] \setminus \{j\}} p_{i,j'} Z^{ijt}_{j'} + p_{ij}\Big)^p\Big] \le A_p \mathbb{E}[((t + p_{ij}) + p_{ij})^p] \le A_p (t + 2p_{ij})^p,$$

which concludes the proof.

C Convexity of F_j

We show that functions F_j defined in Section 3 are convex.

Lemma C.1. *Fix real numbers $q \ge 1$ and $d_1, \ldots, d_n \ge 0$. Define a function $F : [0,1]^n \to \mathbb{R}^+$ as follows: $F(y)$ equals the optimal value of the following LP:*

$$\begin{aligned} \min \sum_{S \subseteq [n]} &\big(\sum_{i \in S} d_i\big)^q z_S \\ \sum_{S \subseteq [n]} z_S &= 1 \\ \sum_{S : i \in S} z_S &= y_i, && \forall i \in [n] \\ z_S &\ge 0, && \forall S \subseteq [n] \end{aligned}$$

Then, F is a convex function.

Proof. Consider two vectors $y^*, y^{**} \in [0,1]^n$. Pick an arbitrary $\lambda \in [0,1]$. We need to show that

$$F(\lambda y^* + (1-\lambda)y^{**}) \le \lambda F(y^*) + (1-\lambda)F(y^{**}).$$

Consider the optimal LP solutions z^* and z^{**} for y^* and y^{**}. Then, by the definition of F,

$$F(y^*) = \sum_{S\subseteq[n]} \big(\sum_{i\in S} d_i\big)^q z_S^* \text{ and } F(y^{**}) = \sum_{S\subseteq[n]} \big(\sum_{i\in S} d_i\big)^q z_S^{**}.$$

Observe, that $\lambda z^* + (1-\lambda)z^{**}$ is a feasible solution for $\lambda y^* + (1-\lambda)y^{**}$ (since all LP constraints are linear). Hence, $F(\lambda y^* + (1-\lambda)y^{**})$ is at most the LP cost of $\lambda z^* + (1-\lambda)z^{**}$, which equals

$$\sum_{S\subseteq[n]} \Big(\sum_{i\in S} d_i\Big)^q (\lambda z_S^* + (1-\lambda)z_S^{**}) = \lambda F(y^*) + (1-\lambda)F(y^{**}).$$

□

D Figures

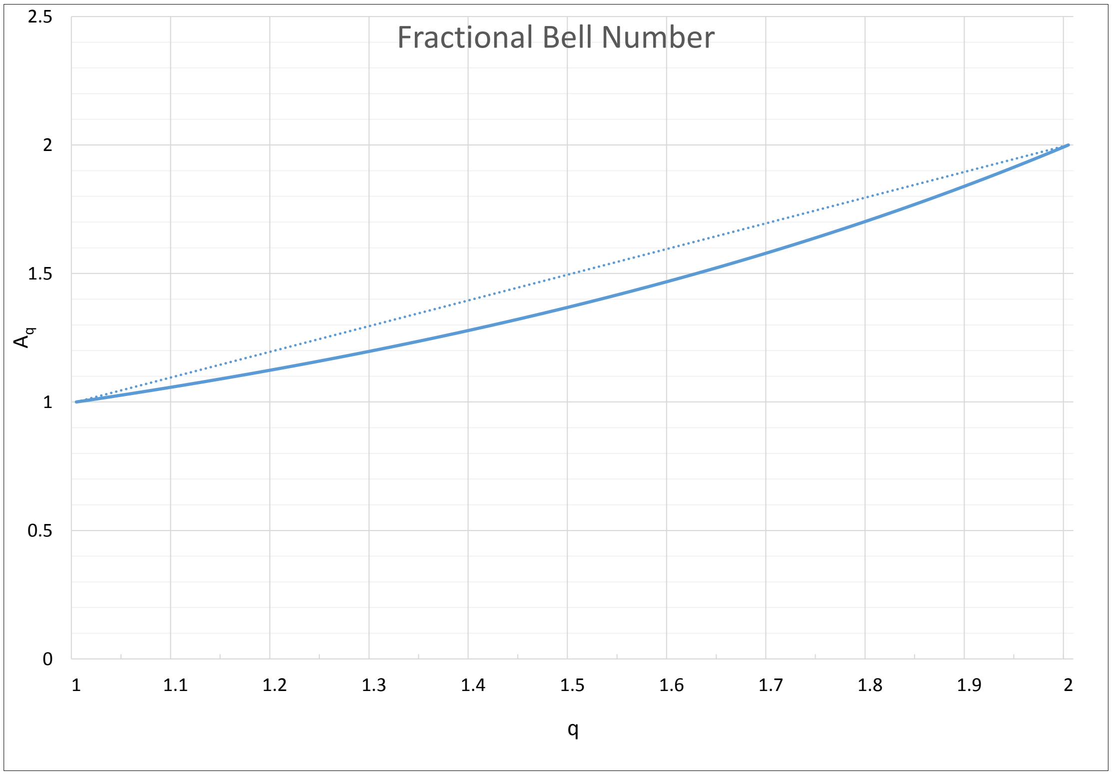

Figure 1: Graph of A_q for $q \in [1, 2]$. Note that the function $q \mapsto A_q$ is convex; $A_1 = 1$ and $A_2 = 2$. Thus, $A_q \leq q$ for $q \in [1, 2]$.

$q =$	1	1.125	1.25	1.375	1.5	1.625	1.75	1.875	2
$A_q =$	1	1.077	1.163	1.262	1.373	1.500	1.645	1.811	2

Figure 2: The values of A_q for some $q \in [1, 2]$. All values are rounded up to three decimal places.

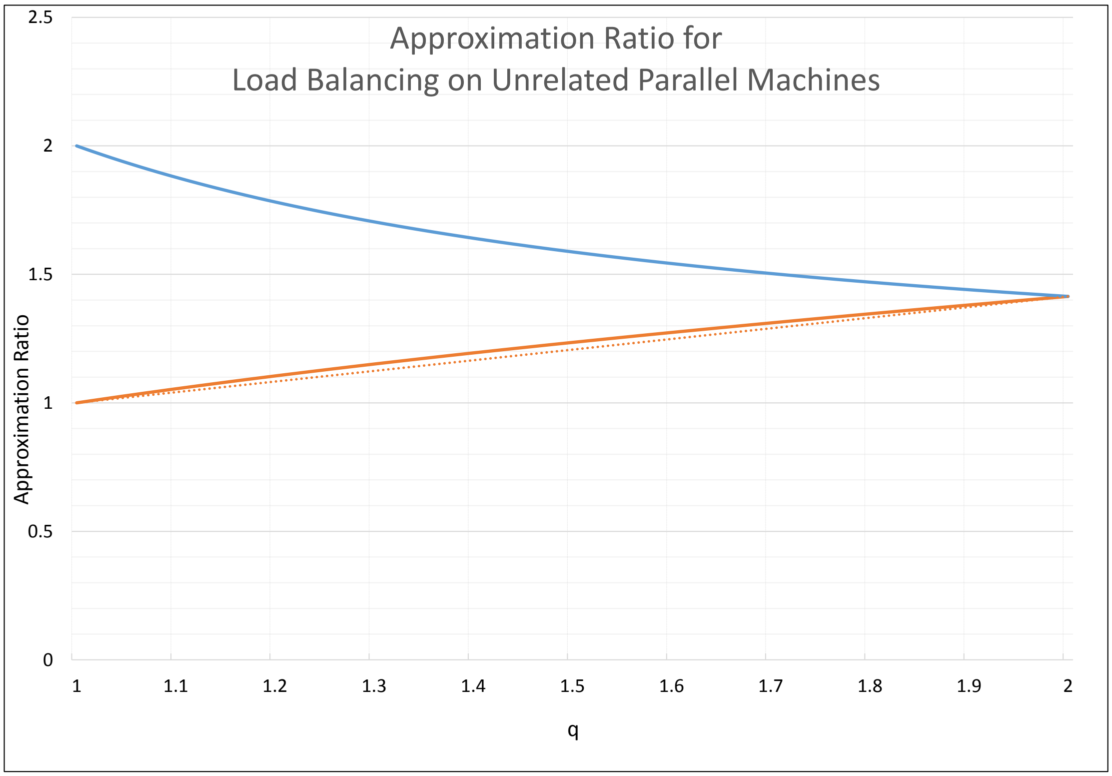

Figure 3: Approximation factor of our algorithm for Load Balancing on Unrelated Parallel Machines – $A_q^{1/q}$ – is plotted in red (below). Approximation factor of the algorithm due to Kumar, Marathe, Parthasarathy and Srinivasan [21] – $2^{1/q}$ – is plotted in blue (above). The function $A_q^{1/q}$ can be well approximated by the linear function $1 + (\sqrt{2} - 1)(q - 1)$ in the interval $q \in [1, 2]$.