THEORY OF COMPUTING, Volume 14 (14), 2018, pp. 1–29
www.theoryofcomputing.org

Small Extended Formulation for Knapsack Cover Inequalities from Monotone Circuits

Abbas Bazzi* Samuel Fiorini† Sangxia Huang‡ Ola Svensson§

Received December 9, 2016; Revised August 13, 2018; Published December 2, 2018

Abstract: Initially developed for the min-knapsack problem, the knapsack cover inequalities are used in the current best relaxations for numerous combinatorial optimization problems of covering type. In spite of their widespread use, these inequalities yield linear programming (LP) relaxations of exponential size, over which it is not known how to optimize exactly in polynomial time. In this paper we address this issue and obtain LP relaxations of quasi-polynomial size that are at least as strong as that given by the knapsack cover inequalities.

For the min-knapsack cover problem, our main result can be stated formally as follows: for any $\varepsilon > 0$, there is a $(1/\varepsilon)^{O(1)} n^{O(\log n)}$-size LP relaxation with an integrality gap of at most $2+\varepsilon$, where n is the number of items. Previously, there was no known relaxation of subexponential size with a constant upper bound on the integrality gap. Our techniques are also sufficiently versatile to give analogous results for the closely related flow cover inequalities that are used to strengthen relaxations for scheduling and facility location problems.

A conference version of this paper appeared in the Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA 2017) [7].

*Supported in part by a SNSF Grant.

†Supported by ERC Consolidator Grant 615640-ForEFront.

‡Supported by ERC Starting Grant 335288-OptApprox.

§Supported by ERC Starting Grant 335288-OptApprox.

ACM Classification: G.1.6

AMS Classification: 68Q25, 90C59

Key words and phrases: extended formulations, communication complexity, linear programming, knapsack

DOI: 10.4086/toc.2018.v014a014

Our construction is inspired by a connection between extended formulations and monotone circuit complexity via Karchmer-Wigderson games. In particular, our LP is based on $O(\log^2 n)$-depth monotone circuits with fan-in 2 for evaluating weighted threshold functions with n inputs, as constructed by Beimel and Weinreb. We believe that a further understanding of this connection may lead to more positive results complementing the numerous lower bounds recently proved for extended formulations.

1 Introduction

Capacitated covering problems[1] play a central role in combinatorial optimization. These are the problems modeled by Integer Programs (IPs) of the form

$$\min\left\{\sum_{i=1}^{n} c_i x_i \;\middle|\; Ax \geqslant b,\ x \in \{0,1\}^n\right\},$$

where A is a size-$m \times n$ nonnegative matrix and b and c are size-m and size-n nonnegative vectors, respectively. The *min-knapsack problem* is the special case arising when there is a single covering constraint, that is, when $m = 1$. This is arguably the simplest interesting capacitated covering problem.

In terms of complexity, the min-knapsack problem is well-understood: on the one hand it is weakly NP-hard [30] and on the other hand it admits an FPTAS [31, 37]. However, for its own sake and since it appears as a key substructure of numerous other IPs, improving our polyhedral understanding of the problem is important. By this, we mean finding "good" linear programming (LP) relaxations for the min-knapsack problem. Indeed, the polyhedral study of this problem has led to the development of important tools, such as the knapsack cover inequalities, for the strengthening of LP relaxations. These inequalities and generalizations thereof are now used in the current best known relaxations for several combinatorial optimization problems, such as single-machine scheduling [6] and capacitated facility location [2]. However, despite this important progress in the past, many fundamental questions remain open even in the most basic setting.

State of the art. The feasible region of a min-knapsack instance is specified by positive *item sizes* $s_1, \ldots, s_n$ and a positive *demand* D. In this context, a vector $x \in \{0,1\}^n$ is *feasible* if $\sum_{i=1}^{n} s_i x_i \geqslant D$. To specify completely an instance of the min-knapsack problem, we are further given nonnegative *item costs* $c_1, \ldots, c_n$. Solving the resulting instance then amounts to solving the IP

$$\min\left\{\sum_{i=1}^{n} c_i x_i \;\middle|\; \sum_{i=1}^{n} s_i x_i \geqslant D,\ x \in \{0,1\}^n\right\}.$$

The *basic* LP relaxation, i. e.,

$$\min\left\{\sum_{i=1}^{n} c_i x_i \;\middle|\; \sum_{i=1}^{n} s_i x_i \geqslant D,\ x \in [0,1]^n\right\},$$

[1] The term "capacitated" is used in the literature to emphasize that the entries of matrix A can take any non-negative value in contrast to the uncapacitated version where entries are Boolean.

provides a lower bound on the optimum value that can be quite bad. More precisely, defining the *integrality gap* as the supremum over all instances of the ratio of the optimum value of the IP to the optimum value of the LP relaxation, it is easy to see that the integrality gap is unbounded.

Several inequalities have been proposed for strengthening this basic LP relaxation. Already in the 1970s, Balas [3], Hammer, Johnson and Peled [27] and Wolsey [40] independently proposed to add the *uncapacitated* knapsack cover inequalities: for every subset $A \subseteq [n] := \{1,\dots,n\}$ of the items such that $\sum_{i\in A} s_i < D$, add the inequality $\sum_{i\notin A} x_i \geqslant 1$ (saying that at least one item in $[n]\setminus A$ needs to be picked in order to satisfy the demand). Unfortunately, these (exponentially many) inequalities are not sufficient for bringing down the integrality gap to a constant. A strengthening of these inequalities was therefore proposed more recently by Carr, Fleischer, Leung and Philipps [15]. They defined the following valid inequalities: for every set of items $A \subseteq [n]$ such that $\sum_{i\in A} s_i < D$, there is a corresponding (capacitated) *knapsack cover inequality*

$$\sum_{i\notin A} s'_i x_i \geqslant U \tag{1.1}$$

where $U = U(A) := D - \sum_{i\in A} s_i$ is the *residual demand* and $s'_i = s'_i(A) := \min\{s_i, U\}$. The validity of (1.1) is due to the fact that every feasible solution $x \in \{0,1\}^n$ has to contain some object $i \notin A$. This object can be *large*, that is, have $s_i \geqslant U$, and in this case the inequality is clearly satisfied. Otherwise, in case every object $i \notin A$ is *small*, the total size of the objects $i \notin A$ picked by x has to be at least the residual demand U.

Carr et al. [15] proved that whenever $x \in \mathbb{R}^n_{\geqslant 0}$ satisfies all knapsack cover inequalities, $2x$ dominates a convex combination of feasible solutions, that is, there exist feasible solutions $x^{(j)} \in \{0,1\}^n$ ($j \in [q]$) and coefficients $\lambda_j \geqslant 0$ summing up to 1 such that

$$2x \geqslant \sum_{j=1}^{q} \lambda_j x^{(j)}\,.$$

Given any nonnegative item costs, one of the $x^{(j)}$ will have a cost that is at most 2 times that of x. This implies that the integrality gap of the corresponding LP relaxation is at most 2.

The LP relaxation defined by the knapsack cover inequalities is "good" in the sense that it has a constant integrality gap. However, it has exponential *size*, that is, exponentially many inequalities, over which it is not known how to optimize exactly in polynomial time; in particular, it is not known how to employ the Ellipsoid algorithm because the problem of separating the knapsack cover inequalities reduces to another knapsack problem (which is NP-hard in general).

In contrast, for the *max-knapsack problem*, Bienstock [11] proved that for all $\varepsilon > 0$ there exists a size-$n^{O(1/\varepsilon)}$ LP relaxation whose integrality gap[2] is at most $1+\varepsilon$. That LP is defined by an extended formulation that uses $n^{O(1/\varepsilon)}$ extra variables besides the x-variables. We remark that it is a notorious open problem to prove or disprove the existence of a $f(1/\varepsilon)\cdot n^{O(1)}$-size LP relaxation for max-knapsack with integrality gap at most $1+\varepsilon$, see, e. g., Section 5.3 in the survey on extended formulations by Conforti, Cornuéjols and Zambelli [21]. Coming back to the min-knapsack problem, it is not known whether there exists a polynomial-size LP relaxation with constant integrality gap or not.[3]

[2]For maximization problems, one takes the supremum of the ratio of the optimum value of the LP relaxation to the optimum value of the IP.

[3]We remark that Bienstock and McClosky [12] considered the easier case when the relaxation is allowed to depend on the

Main result. We come close to resolving the question and show that min-knapsack admits a quasi-polynomial-size LP relaxation with integrality gap at most $2+\varepsilon$. Furthermore, we show that the LP relaxation can be constructed in quasi-polynomial time whenever the data is integer and quasi-polynomially bounded. The upper bound on the integrality gap originates from the fact that our LP relaxation is at least as strong as that provided by a slightly weakened form of the knapsack cover inequalities. We point out that, under some conditions, we can bound the size of our relaxation by a polynomial, see Section 3.2. A more precise statement of our main result is as follows.

Theorem 1.1. *For all given $\varepsilon \in (0,1)$, item sizes $s_1,\dots,s_n \in \mathbb{R}_+$ and demand $D \in \mathbb{R}_+$, there exists a size-$(1/\varepsilon)^{O(1)} n^{O(\log n)}$ extended formulation defining an LP relaxation of min-knapsack with integrality gap at most $2+\varepsilon$. Moreover, one can construct such an extended formulation in quasi-polynomial time whenever $1/\varepsilon$, s_1, ..., s_n and D are positive integers bounded by $n^{O(\log n)}$.*

Our techniques are sufficiently versatile to give analogous results for the closely related *flow cover inequalities*. Specifically, we show in Section 4 that adjusting our techniques to the flow cover inequalities yields that the latter can also be approximated by a quasi-polynomial number of inequalities of roughly the same strength. Notwithstanding their own combinatorial merit, these exponentially many inequalities have been successfully used to strengthen relaxations for many scheduling and facility location problems. As Theorem 1.1 is obtained by giving quasi-polynomially many inequalities of roughly the same strength as the exponentially many cover inequalities, our techniques also lead to relaxations of quasi-polynomial size for their numerous applications, as we discuss in the related work section.

Beyond the result itself, the novelty of our approach lies in the concepts we rely on and the techniques we develop. Our starting point is a connection between monotone circuits and extended formulations that we explain below. This connection was instrumental in the recent *lower bounds* of Göös, Jain and Watson on the extension complexity of independent set polytopes [26], and can be traced back to a paper of Hrubeš [28]. Here we use it for the first time to prove an *upper bound*.

From monotone circuits to extended formulations. Each choice of item sizes and demand gives rise to a *weighted threshold function* $f:\{0,1\}^n \to \{0,1\}$ defined as

$$f(x) := \begin{cases} 1 & \text{if } \sum_{i=1}^n s_i x_i \geqslant D\,, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

Since we assume that the item sizes and demand are nonnegative, f is *monotone* in the sense that $a \leqslant b$ implies $f(a) \leqslant f(b)$, for all $a,b \in \{0,1\}^n$.

Clearly, we have that $x \in \{0,1\}^n$ is feasible if and only if $x \in f^{-1}(1)$. Furthermore, for $a \in f^{-1}(0)$, we can rewrite the uncapacitated knapsack cover inequalities as $\sum_{i:a_i=0} x_i \geqslant 1$. Consider the *slack matrix* $S_{a,b} := \sum_{i:a_i=0} b_i - 1$ indexed by pairs $(a,b) \in f^{-1}(0) \times f^{-1}(1)$. By Yannakakis' factorization theorem [41], the existence of a size-r LP relaxation of min-knapsack that is at least as strong as that

objective function to be optimized (i. e., on the cost of the items). In this case, using techniques similar to those developed for polynomial-time approximation schemes, they obtained polynomial-size relaxations with integrality gap at most $1+\varepsilon$, for any fixed $\varepsilon > 0$. This is, however, a very different setting and, as the developed inequalities depend on the objective function, they do not generalize to other problems.

given by the uncapacitated knapsack cover inequalities is equivalent to the existence of a decomposition of the slack matrix S as a sum of r nonnegative rank-1 matrices.

Now suppose that there exists a depth-t monotone circuit (that is, using only AND gates and OR gates) of fan-in 2 for computing $f(x)$. A result of Karchmer and Wigderson [29] then implies a partition of the entries of S into at most 2^t rectangles[4] $R \subseteq f^{-1}(0) \times f^{-1}(1)$ such that in each of these rectangles R, there exists some index $i^* = i^*(R)$ such that $a_{i^*} = 0$ and $b_{i^*} = 1$ for all $(a,b) \in R$. Then we may write, for $(a,b) \in R$,

$$S_{a,b} = \sum_{i:a_i=0} b_i - 1 = \sum_{i:a_i=0,\ i \neq i^*} b_i = \sum_{i \neq i^*} (1-a_i) b_i \tag{1.3}$$

so that S restricted to the entries of R can be expressed as a sum of at most $n-1$ nonnegative rank-1 matrices of the form $((1-a_i)b_i)_{(a,b)\in R}$, where i is a fixed index distinct from i^*. This implies a decomposition of the whole slack matrix S as a sum of at most $2^t(n-1)$ nonnegative rank-1 matrices, and thus the existence of a $2^t(n-1)$-size LP relaxation of min-knapsack that captures the uncapacitated knapsack cover inequalities. Since f is a weighted threshold function, we can take $t = O(\log^2 n)$, as proved by Beimel and Weinreb [10]. Therefore, we obtain an $n^{O(\log n)}$-size extended formulation for the uncapacitated knapsack cover inequalities. Unfortunately, these inequalities do not suffice to guarantee a bounded integrality gap.

For the full-fledged knapsack cover inequalities (1.1), the simple idea described above breaks down. If the special index $i^* = i^*(R)$ for some rectangle R corresponds to a large object, we can write

$$\sum_{i:a_i=0} s_i' b_i - U = \sum_{i:a_i=0,\ i \neq i^*} s_i' b_i = \sum_{i \neq i^*} s_i'(1-a_i) b_i$$

where each matrix $(s_i'(1-a_i)b_i)_{(a,b)\in R}$ has rank at most 1 because $s_i'(1-a_i)$ depends on a only. However, i^* may correspond to a small object, in which case we cannot decompose the slack matrix as above.

Nevertheless, we prove that it is possible to overcome this difficulty. Two key ideas we use to achieve this are to discretize some of the quantities (which explains why we lose an ε in the integrality gap) and resort to several weighted threshold functions instead of just one. If all these functions admit $O(\log n)$-depth monotone circuits of fan-in 2, then we obtain a size-$n^{O(1)}$ LP relaxation. We remark that it is an intriguing open problem whether general threshold functions admit $O(\log n)$-depth monotone circuits of fan-in 2.

Relation with other work. Knapsack cover inequalities and their generalizations such as flow cover inequalities were used as a systematic way to *strengthen* LP formulations of other (seemingly unrelated) problems [15, 14, 33, 4, 5, 16, 6, 20, 34, 19, 22]. By strengthening we mean that one would start with a polynomial-size LP formulation with a potentially unbounded integrality gap for some problem of interest, and then show that adding (adaptations of) knapsack cover inequalities reduces this integrality gap. As a motivational example for the flow cover inequalities, we illustrate in Section 4 how this strengthening works for the Single Demand Facility Location problem, reducing the integrality gap to 2. This approach shares the same drawback that was identified in the case of min-knapsack, which is that the size of the resulting LP formulation becomes exponential. However, our result can be extended to show that it yields quasi-polynomial-size LP formulation for many such applications. To name a few:

[4] A *rectangle* is the Cartesian product of a set of row indices and a set of column indices.

- Carr et al. [15] applied these inequalities to the Generalized Vertex Cover problem, Multi-color Network Design problem and the Fixed Charge Flow problem, and showed how these inequalities reduce the integrality gap of the starting LP formulations.

- Efsandiari et al. [22] used a knapsack-cover-strengthened LP formulation to design an $O(\log k)$-approximation algorithm for Precedence-Constrained Single-Machine Deadline scheduling problem, where k is the number of distinct deadlines.

- Carnes and Shmoys [14] designed primal-dual algorithms for the Single-Demand Facility Location, where the primal LP formulation is strengthened by adding (generalizations) of knapsack cover inequalities.

- Bansal and Pruhs [6] studied the Generalized Scheduling Problem (GSP) that captures many interesting scheduling problems such as Weighted Flow Time, Flow Time Squared and Weighted Tardiness. In particular, they showed a connection between GSP and a certain geometric covering problem, and designed an LP-based approximation algorithm for the latter that yields an approximate solution for the GSP. The LP formulation that they use for the intermediate geometric cover problem is strengthened using knapsack cover inequalities, and yields an $O(\log\log nP)$-approximation for the GSP where n is the number of jobs, and P is the maximum job size. In the special case of identical release time of the jobs, their LP formulation yields a 16-approximation algorithm. This constant-factor approximation was later improved by Cheung et al. [19] to a $(4+\varepsilon)$-approximation, where the authors added the knapsack cover inequalities directly to the LP formulation of the scheduling problem, i. e., without resorting to the intermediate geometric cover problem as in [6]. For both the GSP and its special case, our method yields an LP formulation whose size is quasi-polynomial in n, and polynomial in both $\log P$ and $\log W$, where W is the maximum increase in the cost function of a job at any point in time. The integrality gaps achieved by our LPs are $O(\log\log nP)$ for the GSP and $4+\varepsilon$ in its special case.

Extended formulations have received a considerable amount of attention recently, mostly for proving impossibility results. Pokutta and Van Vyve [38] proved a worst-case $2^{\Omega(\sqrt{n})}$ size lower bound for extended formulations of the max-knapsack polytope, which directly implies a similar result for the min-knapsack polytope. Other recent articles include [25, 13, 17, 39, 32, 8].

Outline. We prove the first part of Theorem 1.1 in Section 3, after giving preliminaries in Section 2. This is the part of Theorem 1.1 that claims the existence of an extended formulation, which is our main contribution. In Section 3, instead of explicitly constructing our extended formulation, we provide a nonnegative factorization of the appropriate slack matrix. For this, we use the language of communication complexity—we give an $O(\log^2 n+\log(1/\varepsilon))$-complexity two-party communication protocol with private randomness and nonnegative outputs whose expected output is the slack of a given feasible solution with respect to a given (weakened) knapsack cover inequality.

Next, in Section 4, we extend our communication protocol to the flow cover inequalities for the Single-Demand Facility Location problem, and show how to approximate the exponentially many flow cover inequalities using a smaller LP formulation.

Finally, we use recent results of Fiorini, Huynh and Weltge [24] to show the second part of Theorem 1.1. That is, we show how to construct in quasi-polynomial time a quasi-polynomial-size LP formulation for min-knapsack with integrality gap at most $(2+\varepsilon)$. This is done in Section 5.

2 Preliminaries

In this section, we introduce some key notions related to our problem. We review extended formulations and extension complexity of pairs of polyhedra in Section 2.1. Next, we define randomized communication protocols with non-negative outputs that compute entries of matrices in expectation. Finally, in Section 2.3, we review some constructions of low-depth monotone circuits, and the Karchmer-Wigderson game that relates circuit complexity and communication complexity.

2.1 Polyhedral pairs, extended formulations and slack matrices

Let $P \subseteq \mathbb{R}^n$ be a polytope and $Q \subseteq \mathbb{R}^n$ be a polyhedron containing P. The complexity of the *polyhedral pair* (P,Q) can be measured by its extension complexity, which roughly measures how compactly we can represent a relaxation of P contained in Q. The formal definition is as follows.

Definition 2.1. Given a polyhedral pair (P,Q) where $P \subseteq Q \subseteq \mathbb{R}^n$, we say that a system $E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant}$, $E^{=}x + F^{=}y = g^{=}$ in $\mathbb{R}^{n+k}$ is an *extended formulation* of (P,Q) if the polyhedron

$$R := \left\{ x \in \mathbb{R}^n \;\middle|\; \exists y \in \mathbb{R}^k : E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant}, E^{=}x + F^{=}y = g^{=} \right\}$$

contains P and is contained in Q. The *size* of the extended formulation is the number of inequalities in the system. The *extension complexity* of (P,Q), denoted by $\mathrm{xc}(P,Q)$, is the minimum size of an extended formulation of (P,Q).

Although the case $P = Q$ is probably the most frequent, we will need polyhedral pairs here. In a seminal paper, Yannakakis [41] showed that one can study the extension complexity of a polytope P through the non-negative rank of a matrix associated with P, namely, its slack matrix.

Definition 2.2. Let (P,Q) be a polyhedral pair with $P \subseteq Q \subseteq \mathbb{R}^n$. Assume that $P = \mathrm{conv}(\{v_1,\dots,v_p\})$, where $v_1,\dots,v_p \in \mathbb{R}^n$ and $Q = \{x \in \mathbb{R}^n \mid Ax \geqslant b\}$, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. We now define the slack matrix S of the pair (P,Q) with respect to the given representations of P and Q. The ith row of S corresponds to the constraint $A_i x \geqslant b_i$, while the jth column of S corresponds to the point v_j. The value $S_{i,j}$ measures how close the constraint $A_i x \geqslant b_i$ is to being tight for point v_j. More specifically, the *slack matrix* $S \in \mathbb{R}^{m\times p}_{\geqslant 0}$ is defined as $S_{i,j} := A_i v_j - b_i$ for all $i \in [m]$, $j \in [p]$.

Note that the slack matrix is not unique as it depends on the choices of points $v_1,\dots,v_p$ and linear description $Ax \geqslant b$.

Definition 2.3. Given a non-negative matrix $M \in \mathbb{R}^{m\times n}_{\geqslant 0}$, we say that a pair of matrices T,U is a *rank-r non-negative factorization* of M if $T \in \mathbb{R}^{m\times r}_{\geqslant 0}$, $U \in \mathbb{R}^{r\times n}_{\geqslant 0}$, and $M = TU$. We define the *non-negative rank* of M as $\mathrm{rk}_+(M) := \min\{r : M \text{ has a rank-}r \text{ non-negative factorization}\}$. Notice that a non-negative factorization of M of rank at most r is equivalent to a decomposition of M as a sum of at most r non-negative rank-1 matrices.

Yannakakis [41] proved that for a polytope P of dimension at least 1 and any of its slack matrices S, the extension complexity of P is equal to the non-negative rank of S, i. e., $\mathrm{xc}(P) = \mathrm{rk}_+(S)$. In particular, all the slack matrices of P have the same nonnegative rank.

This *factorization theorem* can be extended to polyhedral pairs: we have $\mathrm{xc}(P,Q) \in \{\mathrm{rk}_+(S), \mathrm{rk}_+(S) - 1\}$ whenever S is a slack matrix of (P,Q), see, e. g., [13].

2.2 Randomized communication protocols

We now define a certain two-party communication problem and relate it to the non-negative rank discussed earlier, following the framework in Faenza, Fiorini, Grappe and Tiwary [23].

Definition 2.4. Let $S \in \mathbb{R}_{\geqslant 0}^{\mathcal{A}\times\mathcal{B}}$ be a non-negative matrix whose rows and columns are indexed by $\mathcal{A}$ and $\mathcal{B}$, respectively. Let Π be a communication protocol with private randomness between two players Alice and Bob. Alice gets an input $a \in \mathcal{A}$ and Bob gets an input $b \in \mathcal{B}$. They exchange bits in a pre-specified way according to Π. At the end, one of the players outputs some *non-negative* number $\xi \in \mathbb{R}_{\geqslant 0}$. Any of the two players is allowed to output a nonnegative number, and thus end the protocol. We say that Π *computes S in expectation* if for every a and b, the expectation of the output ξ equals $S_{a,b}$.

The *communication complexity of a protocol* Π is the maximum of the number of bits exchanged *between* Alice and Bob, over all pairs $(a,b) \in \mathcal{A}\times\mathcal{B}$ and the private randomness of the players. The size of the final output does not count towards the communication complexity of a protocol. The *communication complexity of S*, denoted $R^{\mathrm{cc}}_{\exp}(S)$ is the minimum communication complexity of a randomized protocol Π computing S in expectation.

Faenza et al. [23] relate the non-negative rank of a non-negative matrix S, to the communication complexity $R^{\mathrm{cc}}_{\exp}(S)$. In particular, they prove that if $\mathrm{rk}_+(S) \neq 0$, then $R^{\mathrm{cc}}_{\exp}(S) = \log_2 \mathrm{rk}_+(S) + \Theta(1)$. Combining this with the factorization theorem, we get $R^{\mathrm{cc}}_{\exp}(S) = \log_2 \mathrm{xc}(P,Q) + \Theta(1)$ whenever (P,Q) is a polyhedral pair with slack matrix S, provided that $\mathrm{xc}(P,Q) \neq 0$.

2.3 Weighted threshold functions and Karchmer-Widgerson Game

An important part of our protocol depends on the communication complexity of (monotone) weighted threshold functions. We start with the following result from [9, 10] which gives low-depth circuits for such functions. Another construction was given in [18]. The circuits as stated in [9, 10, 18] have logarithmic depth, polynomial size and unbounded fan-in, thus it is straightforward to convert them into circuits with fan-in 2 with a logarithmic increase in depth. Below we state the result for circuits of fan-in 2 as will be used later. Recall that a circuit is *monotone* if it uses only AND and OR gates, but no NOT gates.

Theorem 2.5 ([9, 10])**.** *Let $w_1,\dots,w_n \in \mathbb{Z}_{>0}$ be positive weights, and $T \in \mathbb{Z}_{\geqslant 0}$ be a threshold. Let $f : \{0,1\}^n \to \{0,1\}$ be the monotone function such that $f(x_1,\dots,x_n) = 1$ if and only if $\sum_{i=1}^n w_i x_i \geqslant T$. Then there is a depth-$O(\log^2 n)$ monotone circuit of fan-in 2 that computes the function f.*

The well-known *Karchmer-Wigderson game* [29] connects the depth of monotone circuits and communication complexity. Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, the *monotone Karchmer-Wigderson game* is the following: Alice receives $a \in f^{-1}(0)$, Bob receives $b \in f^{-1}(1)$, they communicate bits to each other, and the goal is to agree on a position $i \in \{1,\ldots,n\}$ such that $a_i = 0$ and $b_i = 1$. Let $D^{\mathrm{cc}}_{\mathrm{mon-KW}}(f)$ be the deterministic communication complexity of this game.

Theorem 2.6 ([29])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function, $D^{\mathrm{cc}}_{\mathrm{mon-KW}}(f)$ be the deterministic communication complexity of the Karchmer-Widgerson game, and* $\mathrm{depth}(f)$ *be the minimum depth of a fan-in* 2 *monotone circuit that computes f. Then* $\mathrm{depth}(f) = D^{\mathrm{cc}}_{\mathrm{mon-KW}}(f)$.

Combining Theorems 2.5 and 2.6, we immediately get that $D^{\mathrm{cc}}_{\mathrm{mon-KW}}(f) = O(\log^2 n)$ for every weighted threshold function f on n inputs.

3 Small LP relaxation for Min-Knapsack

In this section, we show the existence of a $(1/\varepsilon)^{O(1)} n^{O(\log n)}$-size LP relaxation of min-knapsack with integrality gap $2+\varepsilon$, proving Theorem 1.1. First, we give a high-level overview of the construction in Section 3.1. The actual protocol is described and analyzed in Section 3.2.

3.1 Overview

Consider the slack matrix S that has one row for each knapsack cover inequality and one column for each feasible solution of min-knapsack. More precisely, let $f : \{0,1\}^n \to \{0,1\}$ denote the weighted threshold function defined by the item sizes s_i ($i \in [n]$) and demand D as in (1.2). The rows and columns of S are indexed by $a \in f^{-1}(0)$ and $b \in f^{-1}(1)$ respectively. The entries of S are given by

$$S_{a,b} := \sum_{i:a_i=0} s'_i b_i - U\,,$$

where as before $U = U(a) := D - \sum_{i:a_i=1} s_i$, and $s'_i = s'_i(a) = \min\{s_i, U\}$. Geometrically, S is the slack matrix of the polyhedral pair (P,Q) in which P is the min-knapsack polytope and Q is the (unbounded) polyhedron defined by the knapsack cover inequalities.

Ideally, we would like to design a communication protocol for S, as those discussed in Section 2.2, with low communication complexity. This would imply a low-rank non-negative factorization of S. From the factorization theorem of Section 2.1, it would follow that there exists a small-size extended formulation yielding a polyhedron R containing the min-knapsack polytope P and contained in the knapsack-cover relaxation Q. Hence, we would get a small-size LP relaxation for min-knapsack that implies the exponentially many knapsack cover inequalities, and thus have integrality gap at most 2.

However, due to the fact that the quantities involved can be exponential in n, making them too expensive to communicate directly, we have to settle for showing the existence of a small-size extended formulation that *approximately* implies the knapsack cover inequalities. Before discussing further these complications, we give an idealized version of the protocol to help with the intuition. Assume for now that all item sizes and the demand are polynomial in n. Thus Alice and Bob can communicate them with $O(\log n)$ bits.

The goal of the two players is to compute the slack $S_{a,b} = \sum_{i:a_i=0} s'_i b_i - U$, when Alice is given an infeasible $a \in \{0,1\}^n$ and Bob is given a feasible $b \in \{0,1\}^n$. That is, after several rounds of communication, either one of them outputs some non-negative value ξ, such that the expectation of ξ equals $S_{a,b}$.

We define for a set of items $J \subseteq [n]$ the quantity $s(J) := \sum_{j\in J} s_j$, and $s'(J) := \sum_{j\in J} s'_j$. Let A and B be the subsets of $[n]$ corresponding to Alice's input a and Bob's input b, respectively. The slack we want to compute thus becomes $s'(B \setminus A) - U$.

At the beginning, Alice computes the residual demand U and sends it to Bob. Now observe that if there is some $i^* \in B \setminus A$, such that $s_{i^*} \geqslant U$, then we have $s'_{i^*} = U$, and we can easily write the slack as

$$s'\left(B \setminus (A \cup \{i^*\})\right) + (s'_{i^*} - U) = s'\left(B \setminus (A \cup \{i^*\})\right)$$

(similarly to the uncapacitated case discussed in the introduction). Recall that we call an item i *large* if $s_i \geqslant U$ and *small* otherwise. Let I_{large} be the set of large items and I_{small} be the set of small items.

The rest of the protocol is divided into two cases as follows, depending on whether Alice and Bob can easily find a large item $i^* \in B \setminus A$. To this end, Alice sends $s(I_{\text{large}} \cap A)$ to Bob. Note that now Bob can compute $s(I_{\text{small}} \cap A) = D - U - s(I_{\text{large}} \cap A)$. Bob computes the contribution of large items in B, that is, $s(I_{\text{large}} \cap B)$.

If $s(I_{\text{large}} \cap B) > s(I_{\text{large}} \cap A)$, then we are guaranteed that there is some $i^* \in I_{\text{large}} \cap (B \setminus A)$. Moreover, defining the threshold function

$$g(x) := \begin{cases} 1 & \text{if } \sum_{i\in I_{\text{large}}} s_i x_i \geqslant s(I_{\text{large}} \cap B), \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

we have $g(a) = 0$ and $g(b) = 1$. Hence, Alice and Bob can find such an item with $O(\log^2 n)$ bits of communication, see Section 2.3. With that, it is not hard to compute $s'\left(B \setminus (A \cup \{i^*\})\right)$ with $O(\log n)$ bits of communication: Alice samples a uniformly random item i and sends the index to Bob, Bob replies with b_i, Alice outputs $s'_i \cdot n$ if $b_i = 1$, $i \neq i^*$ and $i \notin A$, and outputs 0 otherwise. All her outputs are non-negative and their expectation is exactly the slack.

In the other case, $s(I_{\text{large}} \cap B) \leqslant s(I_{\text{large}} \cap A)$. Unlike in the previous case, we present a simple protocol that does not use monotone circuits. Note that

$$s(B) = s(I_{\text{large}} \cap B) + s(I_{\text{small}} \cap B) \geqslant D = s(I_{\text{large}} \cap A) + s(I_{\text{small}} \cap A) + U\,,$$

thus

$$s(I_{\text{small}} \cap B) - s(I_{\text{small}} \cap A) - U \geqslant s(I_{\text{large}} \cap A) - s(I_{\text{large}} \cap B) \geqslant 0\,.$$

We now write the slack as

$$\begin{aligned} s'(B \setminus A) - U &= s'(I_{\text{large}} \cap (B \setminus A)) + s(I_{\text{small}} \cap (B \setminus A)) - U \\ &= s'(I_{\text{large}} \cap (B \setminus A)) + s(I_{\text{small}} \cap B) - s(I_{\text{small}} \cap (A \cap B)) - U \\ &= s'(I_{\text{large}} \cap (B \setminus A)) + s(I_{\text{small}} \cap B) - s(I_{\text{small}} \cap A) + s(I_{\text{small}} \cap (A \setminus B)) - U \\ &= s'(I_{\text{large}} \cap (B \setminus A)) + s(I_{\text{small}} \cap (A \setminus B)) + \left(s(I_{\text{small}} \cap B) - s(I_{\text{small}} \cap A) - U\right)\,. \end{aligned}$$

Alice and Bob can compute the first and the second term in expectation using a protocol similar to that in the previous case. The last term can be computed by Bob with all the information he has at this stage. To conclude, in both cases, Alice and Bob can compute the exact slack $S_{a,b}$ with $O(\log^2 n)$ bits of communication.

3.2 The protocol

The actual slack matrix S^ε we work with is defined as

$$S^\varepsilon_{a,b} := \sum_{i:a_i=0} s'_i b_i - \frac{2}{2+\varepsilon} U\,, \tag{3.2}$$

where $\varepsilon > 0$ is any small constant, $a \in f^{-1}(0)$ and $b \in f^{-1}(1)$. S^ε is the slack matrix of the polyhedral pair (P, Q^ε) where P is the min-knapsack polytope and Q^ε is the polyhedron defined by a slight weakening of the knapsack cover inequalities obtained by replacing the right-hand side of (1.1) by $(2/(2+\varepsilon)) \cdot U < U$. For every $x \in \mathbb{R}^n_{\geqslant 0}$ that satisfies all weakened knapsack cover inequalities, we have that $((2+\varepsilon)/2)x$ satisfies all original knapsack cover inequalities, and thus, by virtue of [15], $(2+\varepsilon)x$ dominates a convex combination of feasible solutions. Therefore the integrality gap of the resulting LP relaxation (obtained from a non-negative factorization of S^ε) is at most $2+\varepsilon$.

In order to refer to the "derived" weighted threshold functions g as in (3.1), we need a last bit of terminology. We say that $g : \{0,1\}^n \to \{0,1\}$ is a *truncation* of f if there exist $U, T \in \mathbb{Z}_{>0}$ with $T \leqslant D$ such that $g(x) = 1$ iff $\sum_{i=1}^n w_i x_i \geqslant T$, where $w_i = s_i$ if $s_i \geqslant U$ and $w_i = 0$ otherwise. We are now ready to state our main technical lemma.

Lemma 3.1. *For all constants $\varepsilon \in (0,1)$, item sizes $s_i \in \mathbb{Z}_{>0}$ ($i \in [n]$), all smaller than $2^{\lceil n \log n \rceil}$, and demand $D \in \mathbb{Z}_{>0}$ with $\max\{s_i \mid i \in [n]\} \leqslant D \leqslant \sum_{i=1}^n s_i$, such that all truncations of the corresponding weighted threshold function admit depth-t monotone circuits of fan-in 2, there is a $O(\log(1/\varepsilon) + \log n + t)$-complexity randomized communication protocol with non-negative outputs that computes the slack matrix S^ε in expectation. Since we may always take $t = O(\log^2 n)$, this gives a $O(\log(1/\varepsilon) + \log^2 n)$-complexity protocol.*

Before giving the proof, let us remark that the first part of Theorem 1.1 follows directly from this lemma. Indeed, the extra assumptions in the lemma are without loss of generality: the fact that we may assume without loss of generality that the item sizes s_i are positive integers that can be written down with at most $\lceil n \log n \rceil$ bits, is due to a classic result from [36] (cf. [35]); and the fact that we may also assume that the demand D is a positive integer with $\max\{s_i \mid i \in [n]\} \leqslant D \leqslant \sum_{i=1}^n s_i$ should be clear.

Moreover, Lemma 3.1 implies that we can obtain a relaxation of polynomial size if all truncations of the weighted threshold function have monotone circuits of logarithmic depth. In particular, this is the case if all item sizes are polynomial in n. In that case the threshold function (and its truncations) can simply be written as the majority function on $O(\sum_i s_i)$ input bits by using the unary representation of numbers[5] and, as such functions have monotone circuits of fan-in 2 of logarithmic depth [1], i. e., depth $O(\log(\sum_i s_i))$. Thus, using majority functions instead of threshold functions in our communication protocol, we get that for all $\varepsilon \in (0,1)$, $c > 0$, item sizes $s_1, \ldots, s_n \in \{0, 1, \ldots, n^c\}$ and demand $D \in \mathbb{N}$, there exists a size-$(1/\varepsilon)^{O(1)} n^{O(c)}$ extended formulation defining an LP relaxation of min-knapsack with integrality gap at most $2+\varepsilon$. However, it is important to note here that when c is a constant (and hence

[5] To be precise, the majority circuit takes $\sum_{i=1}^n s_i + A$ bits as input. The first $\sum_i s_i$ bits are the representation of the numbers $s_1, \ldots, s_n$ in unary; the inputs that interest us are those where, for every $i \in [n]$, the s_i bits corresponding to s_i are all set to 0 or 1. Finally, the additional A bits are set to constants 0 or 1 so as to make sure that the majority outcome corresponds to the given threshold. Note that A is at most $\sum_i s_i$ and thus the majority function is on at most $O(\sum_i s_i)$ bits.

the sizes $s_1,\ldots,s_n$ and the demand D are polynomial in n), we can write down an exact polynomial-size LP formulation of the min-knapsack problem.[6]

We now proceed by proving our main technical result, i. e., Lemma 3.1.

Proof of Lemma 3.1. Let $\alpha = \alpha(\varepsilon) := 2/(2+\varepsilon)$ and $\delta > 0$ be such that $(1-2\delta)/(1+\delta) = \alpha$. Thus $\delta = \varepsilon/(6+2\varepsilon) = \Theta(\varepsilon)$. As above, we denote by $a \in f^{-1}(0)$ the input of Alice and by $b \in f^{-1}(1)$ that of Bob, and let A and B denote the corresponding subsets of $[n]$.

First, Alice tells Bob the identity of the set of large items $I_{\text{large}} = \{i \in [n] \mid s_i \geqslant U\}$ and its complement, the set of small items I_{small}. This costs $O(\log n)$ bits of communication. For instance, Alice can simply send the index of a smallest large item to Bob, or inform Bob that I_{large} is empty. Recall that

$$U = D - s(A) = D - s(I_{\text{large}} \cap A) - s(I_{\text{small}} \cap A)\,.$$

Then, Alice sends Bob the unique nonnegative integer k such that $(1+\delta)^k \leqslant U < (1+\delta)^{k+1}$. This sets the scale at which the protocol is operating. Since $U \leqslant n \cdot 2^{\lceil n \log n \rceil} \leqslant 2^{n^2}$, we have $(1+\delta)^k \leqslant 2^{n^2}$. This implies that $k = O((1/\varepsilon)n^2)$, thus k can be sent to Bob with $\log(1/\varepsilon) + 2\log n + O(1) = O(\log(1/\varepsilon) + \log n)$ bits. Let $\widetilde{U} := (1+\delta)^k$.

To efficiently communicate an approximate value of $s(I_{\text{large}} \cap A)$, Alice sends the unique nonnegative integer ℓ such that

$$(1+\ell\delta)\widetilde{U} < D - s(I_{\text{large}} \cap A) \leqslant (1+\ell\delta)\widetilde{U} + \delta\widetilde{U}.$$

Since small items have size at most U and we have at most n of them, we have $s(I_{\text{small}} \cap A) \leqslant Un$. Hence,

$$D - s(I_{\text{large}} \cap A) = U + s(I_{\text{small}} \cap A) \leqslant (n+1)U \leqslant (n+1)(1+\delta)\widetilde{U}\,.$$

Since $(1+\ell\delta)\widetilde{U} < (n+1)(1+\delta)\widetilde{U}$, we have $\ell = O((1/\varepsilon)n)$. This means that Alice can communicate ℓ to Bob with only $O(\log(1/\varepsilon) + \log n)$ bits. Let $\tilde{\Delta} = \tilde{\Delta}(\delta) := (1+\ell\delta)\widetilde{U}$. This is Bob's strict under-approximation of $D - s(I_{\text{large}} \cap A)$, so that $D - \tilde{\Delta}$ is a strict over-approximation of $s(I_{\text{large}} \cap A)$.

Bob checks if $s(I_{\text{large}} \cap B) \geqslant D - \tilde{\Delta}$. If this is the case, then the weighted threshold function g such that $g(x) = 1$ iff $\sum_{i \in I_{\text{large}}} s_i x_i \geqslant D - \tilde{\Delta}$ separates a from b in the sense that $g(a) = 0$ and $g(b) = 1$. Since g is a truncation of f, Alice and Bob can exchange t bits to find an index $i^* \in I_{\text{large}}$ such that $a_{i^*} = 0$ and $b_{i^*} = 1$.

We can rewrite the slack $S^{\varepsilon}_{a,b} = s'(B \setminus A) - \alpha U$ as

$$s'(B \setminus (A \cup \{i^*\})) + s'_{i^*} - \alpha U = s'(B \setminus (A \cup \{i^*\})) + (U - \alpha U) = \sum_{i: a_i = 0,\ i \neq i^*} s'_i b_i + (U - \alpha U)\,. \tag{3.3}$$

With the knowledge of i^*, Alice and Bob can compute the slack as follows:

[6]This can be done by casting the folklore *Dynamic Programming* algorithm for the min-knapsack problem, as a minimum s-t flow problem on a weighted graph G with polynomially many vertices, and arguing that the well-known exact LP relaxation of the latter is also an exact LP relaxation of the former. The reader familiar with the dynamic programming algorithm should notice that the edges G would only depend on the sizes of the items, whereas the weights on these edges would only depend on the costs of the items. Hence in the resulting LP formulation, the constraints depend only on the sizes, and the costs only appear in the objective function.

1. Alice samples a uniformly random number $i \in [n]$. If $i \notin A$, continue to the next step, otherwise Alice outputs 0 and terminates the communication.

2. If $i = i^*$, Alice outputs $n \cdot (U - \alpha U)$ and terminates the communication, otherwise continue.

3. Alice sends i to Bob using $\lceil \log n \rceil$ bits of communication, and Bob sends b_i back to Alice.

4. Alice outputs $n \cdot s'_i b_i$.

The above communication costs $O(\log n)$ bits, all outputs are non-negative and can be computed with the information available to each player, and by linearity of expectation, the expected output is exactly the slack (3.3). Together with the $O(\log(1/\varepsilon) + \log n + t)$ bits communicated previously, we conclude that in this case there is a protocol that computes the slack in expectation with $O(\log(1/\varepsilon) + \log n + t)$ bits of communication.

In the other case, we have $s(I_{\text{large}} \cap B) < D - \tilde{\Delta}$. Because $b \in \{0,1\}^n$ is feasible, we get

$$s(B) \geqslant D \iff \underbrace{s(I_{\text{large}} \cap B)}_{<D-\tilde{\Delta}} + s(I_{\text{small}} \cap B) \geqslant D\,,$$

therefore we can bound $s(I_{\text{small}} \cap B)$ as

$$s(I_{\text{small}} \cap B) > \tilde{\Delta} \geqslant\; D - s(I_{\text{large}} \cap A) - \delta\widetilde{U} =\; s(I_{\text{small}} \cap A) + U - \delta\widetilde{U} \geqslant\; \widetilde{\sigma} + (1-\delta)\widetilde{U}\,, \tag{3.4}$$

where $\widetilde{\sigma}$ is the unique integer multiple of $\delta\widetilde{U}$ such that

$$\widetilde{\sigma} \leqslant s(I_{\text{small}} \cap A) < \widetilde{\sigma} + \delta\widetilde{U}\,. \tag{3.5}$$

Since $\widetilde{\sigma} \leqslant s(I_{\text{small}} \cap A) \leqslant Un \leqslant (1+\delta)\widetilde{U}n$, Alice can communicate $\widetilde{\sigma}$ to Bob with $O(\log(1/\varepsilon) + \log n)$ bits.

This implies

$$s(I_{\text{small}} \cap (B \setminus A)) =\; s(I_{\text{small}} \cap B) - s(I_{\text{small}} \cap (A \cap B)) >\; \widetilde{\sigma} + (1-\delta)\widetilde{U} - s(I_{\text{small}} \cap (A \cap B))\,.$$

Recall that by definition of $\widetilde{U}$, we have $(1+\delta)\widetilde{U} > U$, therefore

$$(1-2\delta)\widetilde{U} - \alpha U > (1-2\delta)\widetilde{U} - \alpha(1+\delta)\widetilde{U} = 0\,. \tag{3.6}$$

We now rewrite the slack as

$$\begin{aligned} s'(B \setminus A) - \alpha U =\; & \underbrace{s'(I_{\text{large}} \cap (B \setminus A))}_{=\sum_{i \in I_{\text{large}} \setminus A} s'_i b_i} + \underbrace{s(I_{\text{small}} \cap B) - \widetilde{\sigma} - (1-\delta)\widetilde{U}}_{\text{non-negative by (3.4)}} + \underbrace{s(I_{\text{small}} \cap (A \setminus B))}_{\sum_{i \in I_{\text{small}} \cap A} s_i(1-b_i)} \\ & + \underbrace{\widetilde{\sigma} - s(I_{\text{small}} \cap A) + (1-\delta)\widetilde{U} - \alpha U}_{\text{non-negative by (3.5) and (3.6)}}\,. \end{aligned}$$

The slack can now be readily computed using the following protocol.

1. Alice samples a uniformly random number $i \in [n+2]$. If $i = n+2$, Alice outputs the normalized value of the last term, i. e., $(n+2)\cdot(\widetilde{\sigma} - s(I_{\text{small}} \cap A) + (1-\delta)\widetilde{U} - \alpha U)$, and terminates the communication. Otherwise, she sends i to Bob using $O(\log n)$ bits.

2. If $i = n+1$, Bob outputs $(n+2)\cdot(s(I_{\text{small}} \cap B) - \widetilde{\sigma} - (1-\delta)\widetilde{U})$, and ends the communication. Otherwise, he replies to Alice with b_i.

3. If $i \in I_{\text{large}} \setminus A$, Alice outputs $(n+2)\cdot s'_i b_i$; if $i \in I_{\text{small}} \cap A$, she outputs $(n+2)\cdot s_i(1-b_i)$; otherwise she outputs 0.

We can verify that the outputs of both players can be computed with information available to them, and that the outputs are non-negative due to Equation (3.4), (3.5) and (3.6), and the definition of the variables. □

4 Flow-cover inequalities

Adapting the combinatorial insight behind the knapsack cover inequalities to other settings has led to numerous strong relaxations. In particular, the *flow cover inequalities* have been used to strengthen LPs for many problems such as the Fixed Charge Network Flow problem [15] and the Single-Demand Facility Location problem [14]. In this section, we describe the application of flow cover inequalities to the Single-Demand Facility Location problem as used in [14], and then show that we can adapt our approach from the previous section to give an $O(\log^2 n)$-bit two-party communication protocol that computes a weakened version of these inequalities.

In the Single-Demand Facility Location problem, we are given a set F of n facilities, such that each facility $i \in F$ has a capacity s_i, an opening cost f_i, and a per-unit cost c_i to serve the demand. The goal is to serve the demand D by opening a subset $S \subseteq F$ of facilities such that the combined cost of opening these facilities and serving the demand is minimized. The authors of [14] cast this problem as an Integer Program, and showed that its natural LP relaxation has an unbounded integrality gap. To reduce this gap, they strengthened the relaxation by adding the so-called flow cover inequalities that we define shortly. (See Section 3 in [14] for a more elaborate discussion.)

A feasible solution (x,y) with $y \in \{0,1\}^n$ and $x \in [0,1]^n$ for the Single-Demand Facility Location LP can be thought of as follows: for each $i \in F$, $y_i \in \{0,1\}$ indicates if the i-th facility is open, and $x_i \in [0,1]$ indicates the fraction of the demand D being served by the i-th facility. A feasible solution (x,y) must then satisfy that

1. the demand is *met*, i. e., $\sum_i x_i = 1$;

2. no facility is supplying more than its capacity, i. e., $0 \leqslant x_i D \leqslant y_i s_i$ for all $i \in F$.

For a subset $J \subseteq F$ of facilities and a feasible solution (x,y), we denote by $B = \{i \in F : y_i = 1\} \subseteq [F]$ the set of *open* facilities according to y, and we define the quantity $x(J)$ to be the overall demand served by the facilities in J, i. e., $x(J) = \sum_{i \in J} x_i D$.[7] We also define the quantities $s(\cdot)$ and $s'(\cdot)$ as in Section 3.1.

[7] Note that since we are assuming that (x,y) is feasible, we get that $x(J) = x(J \cap B)$.

Carnes and Shmoys [14] showed that adding the flow cover inequalities (FCI) reduces the integrality gap of the natural LP relaxation down to 2. These inequalities are defined as follows: for any *infeasible* set $A \subseteq F$ (i. e., $A \subseteq F$ such that $s(A) < D$), and for all partitions of $F \setminus A = F_1 \sqcup F_2$, the following inequality holds for all feasible solutions (x, y):

$$s'(F_1 \cap B) + x(F_2 \cap B) \geqslant U\,, \tag{FCI}$$

where $U = D - s(A)$ is the residual demand and $s'_i = \min\{s_i, U\}$. For brevity, we refer to an infeasible set A along with some partition $F_1 \sqcup F_2 = F \setminus A$ as an *infeasible tuple* (A, F_1, F_2). Note that for $F_2 = \emptyset$, the flow-cover inequalities are the same as the knapsack cover inequalities. The main result of this section is formally stated in the following theorem:

Theorem 4.1. *For all $\varepsilon \in (0,1)$, capacities $s_1, \ldots, s_n \in \mathbb{R}_+$ and demand $D \in \mathbb{R}_+$, there exists a size-$(1/\varepsilon)^{O(1)} n^{O(\log n)}$ extended formulation defining an LP relaxation of the Single-Demand Facility Location problem with integrality gap at most $2+\varepsilon$.*

As in the knapsack cover inequalities, the goal is to compute the slack of a *relaxed* version of (FCI) in expectation for any feasible solution (x, y) and any infeasible tuple (A, F_1, F_2). Namely, let $\varepsilon \in (0,1)$ and $\alpha = 2/(2+\varepsilon)$. Our goal is to design an $O(\log^2 n + \log(1/\varepsilon))$-complexity two-party communication protocol with private randomness and nonnegative outputs whose expected output equals $s'(F_1 \cap B) + x(F_2 \cap B) - \alpha U$. That is, we want to compute the slack with respect to a given (weakened) flow-cover inequality $s'(F_1 \cap B) + x(F_2 \cap B) \geqslant \alpha U$, where the RHS of (FCI) is replaced by αU. This implies the existence of an LP of size $(1/\varepsilon)^{O(1)} n^{O(\log n)}$ with an integrality at most $2+\varepsilon$ for the Single-Demand Facility Location problem.

In Section 4.1, we set up the notation and define a class of feasible solutions with a certain special structure which we refer to as *canonical* feasible solutions. We design the promised communication protocol restricted to canonical solutions in Section 4.2, and extend it to arbitrary feasible solutions in Section 4.3.

4.1 Preliminaries

Let (x, y) be a feasible solution for the flow-cover problem with demand D, and let $B = \{i \in F : y_i = 1\}$ denote the support of y. In this terminology, B only indicates which facilities are open, but it does not capture the *relative* demand being served through each of them. However this distinction will be essential for designing the protocol, hence we partition B into three disjoint sets $B = \widetilde{F}_1 \sqcup \widetilde{F}_2 \sqcup \widetilde{F}_3$ as follows:

$$\widetilde{F}_1 = \{i \in B : x_i D = s_i y_i = s_i\}\,, \qquad \widetilde{F}_2 = \{i \in B : 0 < x_i D < s_i\}\,, \qquad \widetilde{F}_3 = \{i \in B : x_i D = 0\}\,.$$

We first focus on feasible solutions (x, y) that exhibit a certain structure, and then generalize to arbitrary solutions. Specifically, we restrict our attention here and in Section 4.2 to *canonical* feasible solutions defined as follows.

Definition 4.2. A feasible solution (x, y) with associated sets $\widetilde{F}_1, \widetilde{F}_2, \widetilde{F}_3$ is *canonical* if $\widetilde{F}_2$ contains at most one facility, i. e., $|\widetilde{F}_2| \leqslant 1$. In other words, in a canonical feasible solution, there is at most one facility j that supplies a non-zero demand $x_j D > 0$ which is *not* equal to its full capacity s_j.

Recall that we are interested in computing

$$s'(F_1 \cap B) + x(F_2 \cap B) - \alpha U \tag{4.1}$$

in expectation, which can be expanded as follows:

$$s'(F_1 \cap \widetilde{F}_1) + s'(F_1 \cap \widetilde{F}_2) + s'(F_1 \cap \widetilde{F}_3) + x(F_2 \cap \widetilde{F}_1) + x(F_2 \cap \widetilde{F}_2) + x(F_2 \cap \widetilde{F}_3) - \alpha U\,. \tag{4.2}$$

We get from the definition of the set $\widetilde{F}_3$ that the second to last term in the above equation is 0 when restricted to canonical feasible solutions. In fact, one can completely get rid of the overall contribution of $\widetilde{F}_3$ in the above equation, since intuitively, *closing* down the facilities in $\widetilde{F}_3$ should not alter the feasibility of the solution, and hence Equation (4.2) should still be positive even without accounting for the contribution of $s'(F_1 \cap \widetilde{F}_3)$. In the communication protocol setting, this intuition translates to designing a protocol that only deals with canonical feasible solutions restricted to $\widetilde{F}_3 = \emptyset$.

To see that this is without loss of generality, consider a canonical feasible solution (x,y) such that $\widetilde{F}_3$ is *not* empty, and let $(x,\bar{y})$ be the projection of (x,y) on $\widetilde{F}_1 \cup \widetilde{F}_2$—that is, for all $i \in B \setminus \widetilde{F}_3$, set $\bar{y}_i = y_i$, and for all $i \in \widetilde{F}_3$, set $\bar{y}_i = 0$. It follows that $(x,\bar{y})$ is also a canonical feasible solution, as the items whose support is $\widetilde{F}_3$ do not contribute to the feasibility of the solution, and the cardinality of $\widetilde{F}_2$ does not change. Thus, for any infeasible tuple (A,F_1,F_2), Equation (4.2) applied to $(x,\bar{y})$ can be written as

$$s'(F_1 \cap \widetilde{F}_1) + s'(F_1 \cap \widetilde{F}_2) + x(F_2 \cap \widetilde{F}_1) + x(F_2 \cap \widetilde{F}_2) - \alpha U\,, \tag{4.3}$$

which is also non-negative, as it is the slack of $(x,\bar{y})$ and (A,F_1,F_2). Therefore, for any feasible solution (x,y), the slack as given by Equation (4.2) can be viewed as the summation of Equation (4.3) and the non-negative term $s'(F_1 \cap \widetilde{F}_3)$. The latter is easy to compute with a small communication protocol,[8] thus if Alice and Bob can devise a communication protocol Π that computes (4.3) in expectation, they can then easily compute (4.2) in expectation. For example, Alice can generate a uniformly random bit $b \in \{0,1\}$, and

- if $b = 0$, then Alice and Bob run the protocol that computes $s'(F_1 \cap \widetilde{F}_3)$, and return *twice* its output;
- if $b = 1$, then Alice and Bob run the protocol Π that computes (4.3), and return *twice* its output.

Moreover, since $|\widetilde{F}_2| \leqslant 1$, and using the fact that $x_i D = s_i$ for $i \in \widetilde{F}_1$, we can further simplify Equation (4.3) as follows:

$$s'(F_1 \cap \widetilde{F}_1) + s(F_2 \cap \widetilde{F}_1) + \gamma(x,y,A,F_1,F_2) - \alpha U\,, \tag{4.4}$$

where the function $\gamma := \gamma(x,y,A,F_1,F_2)$ is defined as

$$\gamma = \begin{cases} s'_j & \text{if } \widetilde{F}_2 = \{j\} \subseteq F_1\,, \\ x_j D & \text{if } \widetilde{F}_2 = \{j\} \subseteq F_2\,, \\ 0 & \text{if } \widetilde{F}_2 = \{j\} \subseteq A, \text{ or } \widetilde{F}_2 = \emptyset\,. \end{cases} \tag{4.5}$$

For simplicity of notation, we drop the parameters from $\gamma(x,y,A,F_1,F_2)$ when they are clear from the context.

[8] To compute $s'(F_1 \cap \widetilde{F}_3)$, Bob samples an index $i \in [n]$. If $i \notin \widetilde{F}_3$, he outputs 0 and terminates the protocol, otherwise he sends i to Alice. If $i \in F_1$, Alice outputs $n \cdot s'(i)$, otherwise, she outputs 0.

4.2 Randomized protocol for canonical feasible solutions

In what follows, we define a randomized communication protocol where Alice gets an infeasible tuple (A, F_1, F_2), and Bob gets a *canonical* feasible solution (x, y) with $\widetilde{F}_3 = \emptyset$, and the goal is to compute the value of (4.4) in expectation.

For a fixed $\varepsilon > 0$, we define $\alpha := \alpha(\varepsilon) = 2/(2+\varepsilon)$, $\delta := \delta(\varepsilon) = \varepsilon/(6+2\varepsilon)$ as in the min-knapsack case. Like the protocol for the knapsack cover inequalities, Alice sends Bob $O(\log n)$ bits at the beginning so that Bob knows $I_{\text{large}}, I_{\text{small}}, \widetilde{U}, \widetilde{\sigma}$ and $\widetilde{\Delta}$. Recall that I_{large} is the set of large items (i. e., $i \in F$ such that $s_i \geqslant U$), I_{small} is the set of small items, $\widetilde{U}$ is an under-approximation of the residual demand U, $D - \widetilde{\Delta}$ is an over-approximation of $s(I_{\text{large}} \cap A)$ and $\widetilde{\sigma}$ is an under-approximation of $s(I_{\text{small}} \cap A)$. Moreover, knowing his input (x, y), Bob can construct the sets $\widetilde{F}_1$ and $\widetilde{F}_2$. Thus, by exchanging an additional $O(\log n)$ bits, Alice and Bob can both figure out which condition is satisfied for Equation (4.5).

To compute the value of (4.4) in expectation, we distinguish between the following cases:

Case 1: Either $\widetilde{F}_2 = \emptyset$, or $\widetilde{F}_2 = \{j\}$ and $j \in A \cup F_1$. In this case, we have that the value γ is either 0 or $s'_j y_j$. Bob now checks if

$$s(I_{\text{large}} \cap (\widetilde{F}_1 \cup \widetilde{F}_2)) \geqslant D - \widetilde{\Delta}. \tag{4.6}$$

Equation (4.6) holds: In the same way as in the min-knapsack protocol, Alice and Bob exchange $O(\log^2 n)$ bits to identify an index $i^* \in I_{\text{large}}$ such that $i^* \in ((\widetilde{F}_1 \cup \widetilde{F}_2) \setminus A)$. More precisely, this index i^* belongs to one of the following three sets: either $i^* \in F_1 \cap \widetilde{F}_1$, or $i^* \in F_2 \cap \widetilde{F}_1$, or $i^* = j$ and $\widetilde{F}_2 = \{j\}$. Alice and Bob can thus exchange $O(1)$ more bits to figure out the condition that i^* satisfies. In what follows, we design an $O(\log n)$-communication protocol to handle each of these cases.

If $i^* \in F_2 \cap \widetilde{F}_1$, then Equation (4.4) can be rewritten as

$$s'(F_1 \cap \widetilde{F}_1) + s((F_2 \cap \widetilde{F}_1) \setminus \{i^*\}) + \gamma + (s_{i^*} - \alpha U) \,. \tag{4.7}$$

One can see that each of the above four terms is non-negative, and similar to the min-knapsack protocol, Alice and Bob can exchange $O(\log n)$ bits and compute the value of (4.7) as follows:

1. Bob sends Alice the index j using $\lceil \log(n) \rceil$ bits if and only if $\widetilde{F}_2 = \{j\}$, and he sends $n+1$ if $\widetilde{F}_2 = \emptyset$.
2. Alice samples a uniformly random index $i \in [n+1]$. If $i = n+1$, Alice uses the knowledge of $\widetilde{F}_2$ (and thus γ) to compute the normalized value of the last terms, that is, she outputs $(n+1) \cdot (\gamma + s_{i^*} - \alpha U)$, and terminates the communication. Otherwise, she sends i to Bob using $\lceil \log(n) \rceil$ bits.
3. If $i \in \widetilde{F}_1$, Bob sends 1 to Alice; otherwise, Bob outputs 0 and terminates the communication.
4. If $i \in F_1$, Alice outputs $(n+1) \cdot s'_i$; if $i \in F_2 \setminus \{i^*\}$, she outputs $(n+1) \cdot s_i$; otherwise she outputs 0.

The above communication costs $O(\log n)$ bits, all outputs are non-negative and can be computed with the information available to each player, and by linearity of expectation, the expected output is exactly the slack (4.4) when $i^* \in F_2 \cap \widetilde{F}_1$.

The case where $i^* \in F_1 \cap \widetilde{F}_1$ is handled similarly.

In the remaining case, we have $\widetilde{F}_2 = \{j\}$ and $i^* = j \in F_1 \cap I_{\text{large}}$, and hence $\gamma = s'_j > \alpha U$. This can be handled by changing the second step of the protocol described earlier in such a way that Alice outputs $(n+1) \cdot (s'_j - \alpha U)$ if $i = n+1$.

Equation (4.6) does not hold: Recall that since (x,y) is a feasible solution (and $\widetilde{F}_3 = \emptyset$), we have

$$\begin{aligned} D &\leq x(\widetilde{F}_1) + x(\widetilde{F}_2) \\ &= x(I_{\text{small}} \cap \widetilde{F}_1) + x(I_{\text{small}} \cap \widetilde{F}_2) + x(I_{\text{large}} \cap \widetilde{F}_1) + x(I_{\text{large}} \cap \widetilde{F}_2) \\ &\leq x(I_{\text{small}} \cap \widetilde{F}_1) + x(I_{\text{small}} \cap \widetilde{F}_2) + s(I_{\text{large}} \cap \widetilde{F}_1) + s(I_{\text{large}} \cap \widetilde{F}_2) \\ &= x(I_{\text{small}} \cap \widetilde{F}_1) + x(I_{\text{small}} \cap \widetilde{F}_2) + s(I_{\text{large}} \cap (\widetilde{F}_1 \cup \widetilde{F}_2)). \end{aligned}$$

By the assumption that Equation (4.6) does not hold, together with the argument in Equation (3.4), we conclude that

$$x(I_{\text{small}} \cap \widetilde{F}_1) + x(I_{\text{small}} \cap \widetilde{F}_2) > \widetilde{\Delta} \geqslant \widetilde{\sigma} + (1-\delta)\widetilde{U}. \tag{4.8}$$

Note that since $|\widetilde{F}_2| \leqslant 1$, we get that

$$x(I_{\text{small}} \cap \widetilde{F}_2) = \begin{cases} 0 & \text{if } \widetilde{F}_2 = \emptyset, \\ 0 & \text{if } \widetilde{F}_2 = \{j\} \subseteq I_{\text{large}}, \\ x_j D & \text{if } \widetilde{F}_2 = \{j\} \subseteq I_{\text{small}}. \end{cases}$$

We also have that $x(I_{\text{small}} \cap \widetilde{F}_1) = s(I_{\text{small}} \cap \widetilde{F}_1)$ by the definition of $\widetilde{F}_1$. Together this gives that the summation $s(I_{\text{small}} \cap \widetilde{F}_1) + x(I_{\text{small}} \cap \widetilde{F}_2)$ is lower bounded by $\widetilde{\sigma} + (1-\delta)\widetilde{U}$. We rewrite (4.4) as

$$\begin{aligned} &s'(F_1 \cap \widetilde{F}_1) + s(F_2 \cap \widetilde{F}_1) + \gamma - \alpha U \qquad (4.9)\\ &= s'(I_{\text{large}} \cap F_1 \cap \widetilde{F}_1) + s(I_{\text{large}} \cap F_2 \cap \widetilde{F}_1) + s(I_{\text{small}} \cap (\widetilde{F}_1 \setminus A)) + \gamma - \alpha U \\ &= s'(I_{\text{large}} \cap F_1 \cap \widetilde{F}_1) + s(I_{\text{large}} \cap F_2 \cap \widetilde{F}_1) + s(I_{\text{small}} \cap (A \setminus B)) + s(I_{\text{small}} \cap A \cap \widetilde{F}_2) \\ &\quad + s(I_{\text{small}} \cap \widetilde{F}_1) - s(I_{\text{small}} \cap A) + \gamma - \alpha U. \end{aligned}$$

The non-negativity of the first three terms is straightforward, and Alice and Bob can compute them by exchanging $O(\log n)$ bits.[9] By adding and subtracting $(\widetilde{\sigma} + (1-\delta)\widetilde{U} - x(I_{\text{small}} \cap \widetilde{F}_2))$ to the remaining terms in (4.9), we can rearrange the terms and rewrite the rest as the sum of the following three non-negative terms that we can easily compute:

$$\begin{aligned} &\Big(s(I_{\text{small}} \cap \widetilde{F}_1) - \widetilde{\sigma} - (1-\delta)\widetilde{U} + x(I_{\text{small}} \cap \widetilde{F}_2)\Big) + \Big(\widetilde{\sigma} + (1-\delta)\widetilde{U} - \alpha U - s(I_{\text{small}} \cap A)\Big) \\ &+ \Big(s(I_{\text{small}} \cap A \cap \widetilde{F}_2) + \gamma - x(I_{\text{small}} \cap \widetilde{F}_2)\Big). \qquad (4.10) \end{aligned}$$

[9]For instance, to compute $s'(I_{\text{large}} \cap F_1 \cap \widetilde{F}_1)$, Alice samples uniformly $i \in [n]$ and sends it to Bob, Bob responds with $b = 1$ if $i \in \widetilde{F}_1$ and $b = 0$ otherwise. Alice then outputs $n \cdot s'_i$ if $i \in I_{\text{large}} \cap F_1$ and $b = 1$, and 0 otherwise. The protocols for the second and the third terms are very similar.

The non-negativity of the first part follows from (4.8), and Bob has all the information to compute it on his own. The non-negativity of the second part follows from our definition of $\widetilde{\sigma}$ and $\widetilde{U}$, and their relation to δ and α. Moreover, Alice has all the information to compute this part.

To see that the third part (i. e., $s(I_{\text{small}} \cap A \cap \widetilde{F}_2) + \gamma - x(I_{\text{small}} \cap \widetilde{F}_2))$ is also non-negative and can easily be computed by one of the players, note the following.

1. If $x(I_{\text{small}} \cap \widetilde{F}_2) = 0$, then clearly it is non-negative. In this case, Bob communicates the set $\widetilde{F}_2$ to Alice using $O(\log n)$ bits so that she knows whether $\widetilde{F}_2 = \emptyset$, or the item j if $\widetilde{F}_2 = \{j\}$ and $j \in I_{\text{large}}$. Once $\widetilde{F}_2$ is known to Alice, she can compute both $s(I_{\text{small}} \cap A \cap \widetilde{F}_2)$ and γ (recall that γ would be either 0 or $s'_j y_j = U$).
2. If $x(I_{\text{small}} \cap \widetilde{F}_2) = x_j D \neq 0$, then we have that $\widetilde{F}_2 = \{j\}$ and $j \in I_{\text{small}}$. From our assumption of Case 1, we also have that $j \in A \cup F_1$. Since A and F_1 are two disjoint sets, we get that:
   (a) If $j \in A$, then
   $$\underbrace{s(I_{\text{small}} \cap A \cap \widetilde{F}_2)}_{s_j y_j} + \underbrace{\gamma}_{0} - x_j D = s_j y_j - x_j D \geqslant 0\,.$$
   (b) If $j \in F_1$, then
   $$\underbrace{s(I_{\text{small}} \cap A \cap \widetilde{F}_2)}_{0} + \underbrace{\gamma}_{s_j y_j} - x_j D = s_j y_j - x_j D \geqslant 0\,.$$
   Thus it is also non-negative, and Bob can compute it on his own in this case.

This concludes the communication protocol in the case where either $\widetilde{F}_2 = \emptyset$, or $\widetilde{F}_2 = \{j\}$ where $j \in A \cup F_1$.

Case 2: $\widetilde{F}_2 = \{j\}$ and $j \in F_2$. In this case $\gamma = x_j D$. This case is quite similar to Case 1, with the difference being that Bob checks at the beginning if

$$s(I_{\text{large}} \cap \widetilde{F}_1) \geqslant D - \widetilde{\Delta},$$

i. e., without including $\widetilde{F}_2$ compared to (4.6).

If the condition was indeed satisfied, then the same reasoning as the first part of Case 1 resolves this case. Otherwise, we get

$$s(I_{\text{small}} \cap \widetilde{F}_1) + x_j D > \widetilde{\sigma} + (1-\delta)\widetilde{U}\,, \tag{4.11}$$

and using Equation (4.9) from the second part of Case 1 yields that that *first four* terms in this case are non-negative and easy to compute. Similarly, adding and subtracting $(\widetilde{\sigma} + (1-\delta)\widetilde{U})$ to the *last four* terms of (4.9), and rearranging the terms we get

$$\Big(s(I_{\text{small}} \cap \widetilde{F}_1) - \widetilde{\sigma} - (1-\delta)\widetilde{U} + x_j D\Big) + \Big(\widetilde{\sigma} + (1-\delta)\widetilde{U} - \alpha U - s(I_{\text{small}} \cap A)\Big)\,.$$

The first part of the summation is non-negative by Equation (4.11) and can be computed by Bob. The second part is the same as the second part in Equation (4.10). It is non-negative by definition and can be computed by Alice. This completes the proof.

This concludes the promised communication protocol in the case where Alice is given an infeasible tuple (A, F_1, F_2), and Bob is given a canonical feasible solution with $\widetilde{F}_3 = \emptyset$. As argued in Section 4.1, this generalizes to any canonical feasible solution *without* any restriction on $\widetilde{F}_3$.

4.3 Randomized protocol for arbitrary feasible solutions

We now extend the communication protocol of canonical feasible solutions to arbitrary feasible solutions. To that end, we denote by $\mathcal{R} = \{(x^1, y^1), (x^2, y^2), \dots, (x^r, y^r)\}$ the set of all canonical feasible solutions.

In this non-restricted setting, Alice still gets an infeasible tuple (A, F_1, F_2), but Bob gets a feasible solution (x, y) that is not necessarily canonical, and the goal remains to compute the slack of the corresponding flow-cover inequality (i. e., Equation (4.1)) in expectation. We show that the communication protocol that we developed in the previous section can be used as a black-box to handle this general case, by noting that any feasible solution (x, y) can be written as a convex combination of canonical feasible solutions $(x^1, y^1), (x^2, y^2), \dots, (x^r, y^r)$. This is formalized in Lemma 4.3.

Lemma 4.3. *Let $\mathcal{R} = \{(x^1, y^1), (x^2, y^2), \dots, (x^r, y^r)\}$ be the set of all the canonical feasible solutions for the flow cover problem, then any feasible solution (x, y) can be written as*

$$(x, y) = \sum_{k=1}^{r} \lambda_k (x^k, y^k)\,, \tag{4.12}$$

such that $\lambda_k \geqslant 0$ for all $1 \leqslant k \leqslant r$, and $\sum_k \lambda_k = 1$.

To see that this is enough, note that for any constraint the slack of (x, y), say $a^T(x, y) - b \geq 0$, equals the convex combination of the slacks of the canonical feasible solutions:

$$\sum_{k=1}^{r} \lambda_k \cdot \Big(a^T(x^k, y^k) - b\Big)\,.$$

Thus in order to compute the slack of (4.1) in expectation, Bob samples a canonical feasible solution $(x^k, y^k) \in \mathcal{R}$ with probability λ_k, then together with Alice, they compute the slack of

$$\sum_{i \in F_1} s'_i y_i^k + \sum_{i \in F_2} x_i^k D - \alpha U$$

as discussed in the previous section.

We conclude this section by proving Lemma 4.3.

Proof of Lemma 4.3. Given a feasible solution (x, y), define its support $\widetilde{F}^{x,y} = \{i : i \in F, \text{ and } y_i = 1\}$, and define the set $\mathcal{R}^{x,y}$ to be the set of all canonical feasible solutions whose support equals $\widetilde{F}^{x,y}$, i. e.,

$$\mathcal{R}^{x,y} = \{(x', y) : (x', y) \in \mathcal{R}\} \subseteq \mathcal{R}\,.$$

Without loss of generality, we assume that $\widetilde{F}^{x,y} = [n]$ to simplify the presentation.

We now consider the following polytope $P(y)$:

$$P(y) = \left\{ \begin{array}{ll} z \in [0,1]^n, & \text{such that:} \\ (*)\ \sum_{i=1}^n z_i = 1\,, & \\ (**)\ 0 \leqslant z_i \leqslant \frac{s_i y_i}{D} & \text{for all } 1 \leqslant i \leqslant n\,. \end{array} \right\}$$

Note that for any feasible solution (x,y) to the flow cover problem, we have that $x \in P(y)$. Moreover, we get from Definition 4.2 that for any canonical feasible solution $(x',y) \in \mathcal{R}^{x,y}$, all except at most one item $i \in [n]$, either has $x'_i = 0$ or $x'_i D = s_i y_i$. Thus x' satisfies at least $n-1$ linearly independent constraints of type $(**)$ with equality. Conversely, if a point $x \in P(y)$ satisfies at least $n-1$ constraints of type $(**)$ with equality, then $(x,y) \in \mathcal{R}^{x,y}$.

Recall that a point z is an *extreme point* solution of $P(y)$ iff there are n linearly independent constraints that are set to equality by z. Since constraint $(*)$ is an equality constraint and is linearly independent from any set of $n-1$ constraints from $(**)$, we conclude that $\{x' : (x',y) \in \mathcal{R}^{x,y}\}$ is the set of all extreme points of $P(y)$. This implies that for any $x \in P(y)$, there exists $\lambda_k \geqslant 0$ for each $1 \leqslant k \leqslant r$ such that $\sum_k \lambda_k = 1$ and

$$x = \sum_{k=1}^{r} \lambda_k x^k\,.$$

Since all these points have the same y-support, it follows that

$$(x,y) = \sum_{k=1}^{r} \lambda_k (x^k, y^k)\,. \qquad \square$$

5 Constructing the extended formulation

We proved the first part of Theorem 1.1 in Section 3. Here, we prove the second part of Theorem 1.1 and show how to *construct* an extended formulation in quasi-polynomial time. The construction is based for one part on our ideas, and for the other part on recent results by Fiorini, Huyhn and Weltge [24] that were established after this article was submitted.

Second part of Theorem 1.1. We use the same notation as in the proof of Lemma 3.1. Remember that $I_{\text{large}} \subseteq [n]$ is the set of large items, $I_{\text{small}} = [n] \setminus I_{\text{large}}$ is the set of small items, and $\tilde{\Delta}$ is an approximation of $D - s(I_{\text{large}} \cap A)$ with $\tilde{\Delta} < D - s(I_{\text{large}} \cap A)$. For every possible I_{large} and $\tilde{\Delta}$, let $f_{I_{\text{large}},\tilde{\Delta}} : \{0,1\}^n \to \{0,1\}$ be the weighted threshold function defined by

$$f_{I_{\text{large}},\tilde{\Delta}}(x) = 1 \iff \sum_{i \in I_{\text{large}}} s_i x_i \geqslant D - \tilde{\Delta}\,.$$

By Theorem 2.5, there exists an $O(\log^2 n)$-depth monotone circuit with fan-in 2 that computes $f_{I_{\text{large}},\tilde{\Delta}}$. Moreover, this circuit can be constructed in time $2^{O(\log^2 n)} = n^{O(\log n)}$. In order to use the results of [24], we convert the circuit computing $f_{I_{\text{large}},\tilde{\Delta}}$ into a size-$n^{O(\log n)}$ monotone Boolean formula $\phi_{I_{\text{large}},\tilde{\Delta}}$. This can also be done in time $n^{O(\log n)}$. Thus $\phi_{I_{\text{large}},\tilde{\Delta}}$ is a formula in the variables x_i ($i \in [n]$) and

$$\phi_{I_{\text{large}},\tilde{\Delta}}(x) = 1 \iff f_{I_{\text{large}},\tilde{\Delta}}(x) = 1$$

for all $x \in \{0,1\}^n$.

Now consider the following expression:

$$\psi := \bigwedge_{I_{\text{large}},\tilde{\Delta}} \left(\phi_{I_{\text{large}},\tilde{\Delta}} \vee \left(\sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta} \right) \right) .$$

We regard ψ as a formula in which each atom is either a variable x_i ($i \in [n]$) or some linear inequality, and each connector is either $\wedge$ (AND) or $\vee$ (OR). We can represent ψ as a tree each of whose leaves is labeled either by a variable x_i for some $i \in [n]$ (*variable-leaf*) or by an inequality $\sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta}$ for some I_{large} and $\tilde{\Delta}$ (*inequality-leaf*), and each of whose inner nodes is either an *AND node* or an *OR node*.

Let $Q \subseteq [0,1]^n$ be any polyhedral relaxation of the set of feasible solutions

$$S := \left\{ x \in \{0,1\}^n : \sum_{i=1}^{n} s_i x_i \geqslant D \right\} .$$

Here we take simply $Q := [0,1]^n$. Following [24], we recursively define a new relaxation $\psi(Q)$ from ψ.

Consider the tree of ψ. We define a polytope for each node of the tree, in a bottom-to-top fashion. First, consider a leaf of the tree. If the leaf is a variable-leaf labeled by x_i, we define the corresponding polytope to be $Q \cap \{x : x_i = 1\}$. Otherwise, the leaf is an inequality-leaf labeled by the inequality $\sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta}$ for some choice of I_{large} and $\tilde{\Delta}$. In this case we associate the polytope

$$Q \cap \left\{ x : \sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta} \right\}$$

to the leaf. Next, consider an inner node of the formula tree. For this inner node, we define a polytope that is either the intersection of the polytopes for its children, in case the inner node is an AND node, or the convex hull of the union of the polytopes for its children, in case the inner node is an OR node. We denote by $\psi(Q)$ the polytope for the root node of the tree.

First, we claim that $\psi(Q)$ is still a relaxation of S. Indeed, pick an arbitrary I_{large} and $\tilde{\Delta}$, and let $x \in S$ be any feasible solution. If x satisfies $\sum_{i \in I_{\text{large}}} s_i x_i \geqslant D - \tilde{\Delta}$, one can easily check that it belongs to the polytope for the root of the subtree that computes $\phi_{I_{\text{large}},\tilde{\Delta}}$, see [24] for details. It is convenient to denote this polytope by $\phi_{I_{\text{large}},\tilde{\Delta}}(Q)$. Otherwise, x satisfies $\sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta}$. In this case, it is in

$$Q \cap \left\{ x : \sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta} \right\} .$$

It follows that every $x \in S$ is in $\psi(Q)$.

Second, assuming that Q has a size-k extended formulation, $\psi(Q)$ has an extended formulation whose size is at most $(k+1)$ times the number of leaves in the tree of ψ. This follows from [24]. That is, $\psi(Q)$ has an extended formulation of size at most $(k+1)n(1/\varepsilon)^{O(1)}n^{O(1)}n^{O(\log n)} = (k+1)n^{O(\log n)}$. (Remember that there are at most n choices for I_{large} since we can assume the items to be sorted, and at most $(1/\varepsilon)^{O(1)}n^{O(1)} = n^{O(\log n)}$ choices for $\tilde{\Delta}$.) For $Q = [0,1]^n$ we have $k = 2n$, thus $\psi([0,1]^n)$ has a quasi-polynomial-size extended formulation. Moreover, the extended formulation can be constructed in quasi-polynomial time since all the $\phi_{I_{\text{large}},\tilde{\Delta}}$'s can be constructed in that time. Lastly, $\psi(Q)$ is built from the polytopes for the leaves of the tree of ψ by taking intersections and convex hulls of unions, for which we can construct extended formulations efficiently, see for instance [21].

Third, we claim that the integrality gap of $\psi([0,1]^n)$ is at most $(2+\varepsilon)$. This is due to the fact that for all $a \in \{0,1\}^n$ the weakened knapsack cover (KC) inequality

$$\sum_{i:a_i=0} s'_i x_i \geqslant \alpha U \tag{5.1}$$

is valid for $\psi([0,1]^n)$, where as before $\alpha = 2/(2+\varepsilon)$ is a number slightly below 1. As in the proof of Lemma 3.1, to every such inequality corresponds a unique set I_{large} and number $\tilde{\Delta}$. Below, we verify that (5.1) is valid for both polytopes

$$\phi_{I_{\text{large}},\tilde{\Delta}}([0,1]^n) \quad \text{and} \quad [0,1]^n \cap \left\{ x : \sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta} \right\}.$$

The claim follows directly.

- The weakened KC inequality (5.1) is valid for $\phi_{I_{\text{large}},\tilde{\Delta}}([0,1]^n)$. Remember that $D - \tilde{\Delta} > s(I_{\text{large}} \cap A)$. Thus every $x \in \{0,1\}^n$ such that
$$\sum_{i \in I_{\text{large}}} s_i x_i \geqslant D - \tilde{\Delta}$$
satisfies the inequality
$$\sum_{i:a_i=0, i \in I_{\text{large}}} x_i \geqslant 1\,.$$
Since this last inequality is a *pitch*-1 inequality,[10] it is valid for $\phi_{I_{\text{large}},\tilde{\Delta}}([0,1]^n)$, see [24]. Hence, for each $x \in \phi_{I_{\text{large}},\tilde{\Delta}}([0,1]^n)$ we get
$$\sum_{i:a_i=0} s'_i x_i \geqslant \sum_{i:a_i=0, i \in I_{\text{large}}} s'_i x_i = \sum_{i:a_i=0, i \in I_{\text{large}}} U x_i \geqslant U \geqslant \alpha U\,.$$

- The weakened KC inequality (5.1) is valid for
$$[0,1]^n \cap \left\{ x : \sum_{i \in I_{\text{small}}} s_i x_i \geqslant \tilde{\Delta} \right\}.$$

[10] The *pitch* of an inequality $\sum_{j \in J} c_j x_j \geq d$ is the smallest number p such that the inequality $\sum_{j \in J} c_j \geq d$ holds for every $J \subseteq \text{supp}(c) \subseteq [n]$ with $|J| \geq p$.

For each

$$x \in [0,1]^n \cap \left\{ x : \sum_{i \in I_{\mathrm{small}}} s_i x_i \geqslant \tilde{\Delta} \right\}$$

we have

$$\sum_{i:a_i=0} s'_i x_i \geqslant \sum_{i:a_i=0, i \in I_{\mathrm{small}}} s'_i x_i = \sum_{i:a_i=0, i \in I_{\mathrm{small}}} s_i x_i \geqslant \tilde{\Delta} - s(I_{\mathrm{small}} \cap A) \geqslant U - \delta \widetilde{U} \geqslant (1-\delta)U \geqslant \alpha U .$$

Above, the second inequality follows from $\sum_{i \in I_{\mathrm{small}}} s_i x_i \geqslant \tilde{\Delta}$ and $-x_i \geqslant -1$ for $i \in I_{\mathrm{small}} \cap A$. □

6 Conclusion

After the recent series of strong negative results on extended formulations, we have presented a positive result inspired by a connection to monotone circuits. Namely, we obtain the first quasi-polynomial-size LP relaxation of min-knapsack with constant integrality gap from polylog-depth circuits for weighted threshold functions.

This result sheds new light on the approximability of min-knapsack via small LPs by connecting it to the complexity of monotone circuits. For instance, it follows from our results that proving that no $n^{O(1)}$-size LP relaxation for min-knapsack can have integrality gap at most α for some $\alpha > 2$ would rule out the existence of $O(\log n)$-depth monotone circuits with bounded fan-in for weighted threshold functions on n inputs, which is an open problem.

Acknowledgements

We thank the anonymous SODA reviewers and ToC reviewers for their valuable comments on the conference version and the journal version of this paper, respectively.

References

[1] MIKLÓS AJTAI, JÁNOS KOMLÓS, AND ENDRE SZEMERÉDI: Sorting in $c \log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983. [doi:10.1007/BF02579338] 11

[2] HYUNG-CHAN AN, MOHIT SINGH, AND OLA SVENSSON: LP-based algorithms for Capacitated Facility Location. *SIAM J. Comput.*, 46(1):272–306, 2017. Conference version in FOCS'14. [doi:10.1137/151002320] 2

[3] EGON BALAS: Facets of the knapsack polytope. *Mathematical Programming*, 8(1):146–164, 1975. [doi:10.1007/BF01580440] 3

[4] NIKHIL BANSAL, NIV BUCHBINDER, AND JOSEPH (SEFFI) NAOR: Randomized competitive algorithms for generalized caching. *SIAM J. Comput.*, 41(2):391–414, 2012. Conference version in STOC'08. [doi:10.1137/090779000] 5

[5] NIKHIL BANSAL, ANUPAM GUPTA, AND RAVISHANKAR KRISHNASWAMY: A constant factor approximation algorithm for generalized min-sum set cover. In *Proc. 21st Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'10)*, pp. 1539–1545, 2010. [doi:10.1137/1.9781611973075.125] 5

[6] NIKHIL BANSAL AND KIRK PRUHS: The geometry of scheduling. *SIAM J. Comput.*, 43(5):1684–1698, 2014. Conference version in FOCS'10. [doi:10.1137/130911317, arXiv:1008.4889] 2, 5, 6

[7] ABBAS BAZZI, SAMUEL FIORINI, SANGXIA HUANG, AND OLA SVENSSON: Small extended formulation for knapsack cover inequalities from monotone circuits. In *Proc. 28th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'17)*, pp. 2326–2341, 2000. ACM DL. 1

[8] ABBAS BAZZI, SAMUEL FIORINI, SEBASTIAN POKUTTA, AND OLA SVENSSON: No small linear program approximates Vertex Cover within a factor of $2-\varepsilon$. *Math. Oper. Res.*, 2018. Conference version in FOCS'15. [doi:10.1287/moor.2017.0918, arXiv:1503.00753] 6

[9] AMOS BEIMEL AND ENAV WEINREB: Monotone circuits for weighted threshold functions. In *Proc. 20th IEEE Conf. on Computational Complexity (CCC'05)*, pp. 67–75. IEEE, 2005. [doi:10.1109/CCC.2005.12] 8

[10] AMOS BEIMEL AND ENAV WEINREB: Monotone circuits for monotone weighted threshold functions. *Inform. Process. Lett.*, 97(1):12–18, 2006. Conference version in CCC'05. [doi:10.1016/j.ipl.2005.09.008] 5, 8

[11] DANIEL BIENSTOCK: Approximate formulations for 0-1 knapsack sets. *Oper. Res. Lett.*, 36(3):317–320, 2008. [doi:10.1016/j.orl.2007.09.003] 3

[12] DANIEL BIENSTOCK AND BENJAMIN MCCLOSKY: Tightening simple mixed-integer sets with guaranteed bounds. *Math. Program.*, 133(1-2):337–363, 2012. [doi:10.1007/s10107-010-0435-x] 3

[13] GÁBOR BRAUN, SAMUEL FIORINI, SEBASTIAN POKUTTA, AND DAVID STEURER: Approximation limits of linear programs (beyond hierarchies). *Math. Oper. Res.*, 40(3):513–796, 2015. Conference version in FOCS'12. [doi:10.1287/moor.2014.0694, arXiv:1204.0957] 6, 8

[14] TIM CARNES AND DAVID B. SHMOYS: Primal-dual schema for capacitated covering problems. *Math. Program.*, 153(2):289–308, 2015. Conference version in IPCO'08. [doi:10.1007/s10107-014-0803-z] 5, 6, 14, 15

[15] ROBERT D. CARR, LISA K. FLEISCHER, VITUS J. LEUNG, AND CYNTHIA A. PHILLIPS: Strengthening integrality gaps for capacitated network design and covering problems. In *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'00)*, pp. 106–115, 2000. ACM DL. 3, 5, 6, 11, 14

[16] DEEPARNAB CHAKRABARTY, ELYOT GRANT, AND JOCHEN KÖNEMANN: On column restricted and priority integer covering programs. In *Proc. 14th Ann. Conf. on Integer Programming and Combinatorial Optimization (IPCO '10)*, 2010. [doi:10.1007/978-3-642-13036-6_27, arXiv:1003.1507] 5

[17] SIU ON CHAN, JAMES LEE, PRASAD RAGHAVENDRA, AND DAVID STEURER: Approximate constraint satisfaction requires large LP relaxations. *J. ACM*, 63(4):34:1–34:22, 2016. Conference version in FOCS'13. [doi:10.1145/2811255] 6

[18] XI CHEN, IGOR CARBONI OLIVEIRA, AND ROCCO A. SERVEDIO: Addition is exponentially harder than counting for shallow monotone circuits. In *Proc. 49th STOC*, pp. 1232–1245, 2017. [doi:10.1145/3055399.3055425, arXiv:1508.03061] 8

[19] MAURICE CHEUNG, JULIÁN MESTRE, DAVID B. SHMOYS, AND JOSÉ VERSCHAE: A primal-dual approximation algorithm for min-sum single-machine scheduling problems. *SIAM J. Discrete Math.*, 31(2):825–838, 2017. [doi:10.1137/16M1086819, arXiv:1612.03339] 5, 6

[20] MAURICE CHEUNG AND DAVID B. SHMOYS: A primal-dual approximation algorithm for min-sum single-machine scheduling problems. In *APPROX'11*, pp. 135–146, 2011. [doi:10.1007/978-3-642-22935-0_12] 5

[21] MICHELE CONFORTI, GÉRARD CORNUÉJOLS, AND GIACOMO ZAMBELLI: Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010. [doi:10.1007/s10288-010-0122-z] 3, 23

[22] HOSSEIN EFSANDIARI, MOHAMMAD TAGHI HAJIAGHAYI, JOCHEN KÖNEMANN, HAMID MAHINI, DAVID L. MALEC, AND LAURA SANITÀ: Approximate deadline-scheduling with precedence constraints. In *Proc. 23rd Ann. European Symp. on Algorithms (ESA'15)*, pp. 483–495. Springer, 2015. [doi:10.1007/978-3-662-48350-3_41, arXiv:1507.00748] 5, 6

[23] YURI FAENZA, SAMUEL FIORINI, ROLAND GRAPPE, AND HANS RAJ TIWARY: Extended formulations, non-negative factorizations and randomized communication protocols. *Math. Program.*, 153(1):75–94, 2015. Conference version in ISCO'12. [doi:10.1007/s10107-014-0755-3, arXiv:1105.4127] 8

[24] SAMUEL FIORINI, TONY HUYNH, AND STEFAN WELTGE: Strengthening convex relaxations of 0/1-sets using Boolean formulas. 2017. [arXiv:1711.01358] 7, 21, 22, 23

[25] SAMUEL FIORINI, SERGE MASSAR, SEBASTIAN POKUTTA, HANS RAJ TIWARY, AND RONALD DE WOLF: Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17:1–17:23, 2015. Conference version in STOC'12. [doi:10.1145/2716307] 6

[26] MIKA GÖÖS, RAHUL JAIN, AND THOMAS WATSON: Extension complexity of independent set polytopes. *SIAM J. Comput.*, 47(1):241–269, 2018. Conference version in FOCS'16. [doi:10.1137/16M109884X, arXiv:1604.07062] 4

[27] PETER L. HAMMER, ELLIS L. JOHNSON, AND URI N. PELED: Facet of regular 0-1 polytopes. *Math. Program.*, 8(1):179–206, 1975. [doi:10.1007/BF01580442] 3

[28] PAVEL HRUBEŠ: On the nonnegative rank of distance matrices. *Inform. Process. Lett.*, 112(11):457–461, 2012. [doi:10.1016/j.ipl.2012.02.009] 4

[29] MAURICIO KARCHMER AND AVI WIGDERSON: Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990. Conference version in STOC'88. [doi:10.1137/0403021] 5, 8, 9

[30] RICHARD M. KARP: Reducibility among combinatorial problems. In R. E. MILLER AND J. W. THATCHER, editors, *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York, 1972. PDF. [doi:10.1007/978-1-4684-2001-2_9] 2

[31] EUGENE L. LAWLER: Fast approximation algorithms for knapsack problems. *Math. Oper. Res.*, 4(2):339–356, 1979. Conference version in FOCS'77. [doi:10.1287/moor.4.4.339] 2

[32] JAMES R. LEE, PRASAD RAGHAVENDRA, AND DAVID STEURER: Lower bounds on the size of semidefinite programming relaxations. In *Proc. 47th STOC*, pp. 567–576, 2015. [doi:10.1145/2746539.2746599] 6

[33] RETSEF LEVI, ANDREA LODI, AND MAXIM SVIRIDENKO: Approximation algorithms for the capacitated multi-item lot-sizing problem via flow-cover inequalities. *Math. Oper. Res.*, 33(2):461–474, 2008. Conference version in IPCO'07. [doi:10.1287/moor.1070.0305] 5

[34] JULIÁN MESTRE AND JOSÉ VERSCHAE: A 4-approximation for scheduling on a single machine with general cost function. *CoRR*, 2014. [arXiv:1403.0298] 5

[35] SABURO MUROGA: *Threshold Logic and Its Applications*. Wiley-Interscience, 1971. 11

[36] SABURO MUROGA, IWAO TODA, AND SATORU TAKASU: Theory of majority decision elements. *J. Franklin Institute*, 271(5):376–418, 1961. [doi:10.1016/0016-0032(61)90702-5] 11

[37] MANFRED W. PADBERG, TONY J. VAN ROY, AND LAURENCE A. WOLSEY: Valid inequalities for fixed charge problems. *Operations Research*, 33(4):842–861, 1985. [doi:10.1287/opre.33.4.842] 2

[38] SEBASTIAN POKUTTA AND MATHIEU VAN VYVE: A note on the extension complexity of the knapsack polytope. *Oper. Res. Lett.*, 41(4):347–350, 2013. [doi:10.1016/j.orl.2013.03.010] 6

[39] THOMAS ROTHVOSS: The matching polytope has exponential extension complexity. *J. ACM*, 64(6):41:1–41:19, 2017. Conference version in STOC'14. [doi:10.1145/3127497, arXiv:1311.2369] 6

[40] LAURENCE A. WOLSEY: Faces for a linear inequality in 0-1 variables. *Math. Program.*, 8(1):165–178, 1975. [doi:10.1007/BF01580441] 3

[41] MIHALIS YANNAKAKIS: Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.*, 43(3):441–466, 1991. Conference version in STOC'88. [doi:10.1016/0022-0000(91)90024-Y] 4, 7, 8

AUTHORS

Abbas Bazzi
Software engineer
Google, Zürich, Switzerland
bazzi@google.com

Samuel Fiorini
Associate professor
Université libre de Bruxelles
Brussels, Belgium
sfiorini@ulb.ac.be
http://homepages.ulb.ac.be/~sfiorini/

Sangxia Huang
Research scientist
Sony Mobile Communications
Lund, Sweden
huang.sangxia@gmail.com
http://huang.sangxia.info/home

Ola Svensson
Assistant professor
EPFL, Lausanne, Switzerland
ola.svensson@epfl.ch
http://theory.epfl.ch/osven/

ABOUT THE AUTHORS

ABBAS BAZZI graduated from EPFL in 2017; his advisor was Ola Svensson. His thesis focused on the strength and limitations of Linear Programming relaxations. After finishing his Ph. D., he joined Google Zürich as Software Engineer.

SAMUEL FIORINI graduated from Université libre de Bruxelles (ULB, Brussels) in 2001 under the supervision of Jean-Paul Doignon. He has held research positions at CWI, MIT, and GERAD - HEC Montreal. In 2005, he joined the faculty of the Department of Mathematics and the Department of Computer Science at ULB. His research focuses on combinatorial optimization, more precisely polyhedral combinatorics and approximation algorithms. He is the principal investigator of the ERC grant FOREFRONT, focusing on extended formulations. In addition to spending time with his family, windsurfing and climbing bring balance to his life.

SANGXIA HUANG graduated from KTH Royal Institute of Technology in 2015 under the supervision of Johan Håstad. The topic of his thesis is hardness of approximation and probabilistically checkable proofs. He worked as a postdoc at EPFL in Ola Svensson's group for two years. He then returned to Sweden and joined Sony Mobile Communications as a research engineer in machine learning.

OLA SVENSSON (not to be confused with the singer Ola Svensson) graduated from IDSIA in 2009; his advisor was Monaldo Mastrolilli. The subject of his thesis was the approximability of graph and scheduling problems. After spending two years as a postdoc with Johan Håstad at KTH Royal Institute of Technology, Sweden, he is now back in Switzerland as an assistant professor in the theory group at EPFL. Apart from doing research, he enjoys the Alps that are in complete contrast to the flat but still beautiful landscape in South Sweden where he grew up.