SIAM J. COMPUT.
Vol. 44, No. 4, pp. 1154–1172

ON THE NUMBER OF ITERATIONS FOR DANTZIG–WOLFE OPTIMIZATION AND PACKING-COVERING APPROXIMATION ALGORITHMS*

PHILIP KLEIN† AND NEAL E. YOUNG‡

Abstract. We give a lower bound on the iteration complexity of a natural class of Lagrangian-relaxation algorithms for approximately solving packing/covering linear programs. We show that, given an input with m random 0/1-constraints on n variables, with high probability, *any* such algorithm requires $\Omega(\rho \log(m)/\epsilon^2)$ iterations to compute a $(1+\epsilon)$-approximate solution, where ρ is the *width* of the input. The bound is tight for a range of the parameters (m, n, ρ, ϵ). The algorithms in the class include Dantzig–Wolfe decomposition, Benders' decomposition, Lagrangian relaxation as developed by Held and Karp for lower-bounding TSP, and many others (e.g., those by Plotkin, Shmoys, and Tardos and Grigoriadis and Khachiyan). To prove the bound, we use a discrepancy argument to show an analogous lower bound on the support size of $(1+\epsilon)$-approximate mixed strategies for random two-player zero-sum 0/1-matrix games.

Key words. Lagrangian relaxation, lower bound, packing, covering, linear programming

AMS subject classifications. 68Q17, 91A05, 68W25, 90C05

DOI. 10.1137/12087222X

1. Background.

We consider a class of algorithms that we call *Dantzig–Wolfe-type* algorithms. The class encompasses algorithms from three lines of research. One line began in 1958 with a method proposed by Ford and Fulkerson [9] for multicommodity flow. Dantzig and Wolfe [7] generalized it as follows. They suggested decomposing an arbitrary linear program into two sets of constraints as

$$\min\{c^{\mathsf{T}}x : Ax \geq b,\, x \in P\},$$

where P is a polyhedron, and using an algorithm that solves the program iteratively. In each iteration, the algorithm performs a single linear optimization over the polyhedron P—that is, in each iteration, the algorithm chooses a cost vector q and computes

$$\arg\min\{q^{\mathsf{T}}x : x \in P\}.$$

This approach, now called *Dantzig–Wolfe decomposition*, is especially useful when P is a Cartesian product $P_1 \times \cdots \times P_K$ and linear optimization over P decomposes into independent optimizations over each P_i.

Lagrangian relaxation. In 1971, Held and Karp [17, 18] proposed a now well-known lower bound for the traveling salesman tour, which they formulated (for some (A, b, c)) as the mathematical program

$$\max_u \Big[u^{\mathsf{T}}b + \min_{x\in P}(c - u^{\mathsf{T}}A)x\Big].$$

*Received by the editors April 3, 2012; accepted for publication (in revised form) June 18, 2015; published electronically August 26, 2015. A conference version of this paper appeared in Proceedings of the Seventh Conference on Integer Programming and Combinatorial Optimization (IPCO' 99), Lecture Notes in Comput. Sci. 1610, Springer, Berlin, 1999, pp. 320–327.
http://www.siam.org/journals/sicomp/44-4/87222.html
†Brown University, Providence, RI 02912 (klein@brown.edu). This author's research was partially supported by NSF grants CCR-9700146 and CCF-0635089.
‡University of California, Riverside, CA 92521 (neal.young@ucr.edu). This author's research was partially supported by NSF grants CCR-9720664, CNS-0626912, CCF-0729071, and CCF-1117954.

Here P is the polyhedron whose vertices are 1-trees (spanning trees plus one edge; a relaxation of traveling salesman tours). To compute an approximate solution, they suggested starting with an arbitrary assignment to u and then iterating as follows: find a minimum-cost 1-tree $T \in P$ with respect to the edge costs $q = c - u^{\intercal} A$; increase u_v for each node v of degree 3 or more in T, and then repeat.

As in Dantzig–Wolfe decomposition, their algorithm interacts with the polyhedron P only by repeatedly choosing a cost vector q and solving for $T = \arg\min\{q^{\intercal} x : x \in P\}$. The method has been applied to a variety of other problems and has come to be known as Lagrangian relaxation. It turns out to be the *subgradient method*, which dates back to the early 1960s.

Fractional packing and covering. In 1979, Shapiro [33] referred to "the correct combination of artistic expertise and luck" needed to make progress in subgradient optimization—although Dantzig–Wolfe decomposition and Lagrangian relaxation could sometimes be proved to converge in the limit, in practice, finding a way to compute and use queries that gave a reasonable convergence rate was an art.

In contrast, the third line of research provided guaranteed convergence rates. In 1990, Shahrokhi and Matula [32] gave an approximation algorithm for a special case of multicommodity flow, which was improved by Klein et al. [21], Leighton et al. [24], and others. Plotkin, Shmoys, and Tardos [31] generalized it to approximate fractional packing (defined below); Grigoriadis and Khachiyan obtained similar results independently [13]. Many subsequent algorithms (too many to list here) build on these results, extending them to fractional covering and to mixed packing/covering and improving the convergence bounds in various ways. Generally, these algorithms are also of Dantzig–Wolfe type: in each iteration, they do a single linear optimization over the polyhedron P.

This research direction is still active. Bienstock gives an implementation-oriented, operations-research perspective [3]. Arora, Hazan, and Kale give a computer-science perspective, highlighting connections to other fields such as learning theory [2]. An overview by Todd places them in the context of general linear programming [34].

In many applications, the total time for the algorithm is the number of iterations times the time per iteration. In most applications, the time per iteration (to solve the subproblem) is large (e.g., linear or more). Hence, a main research goal is to find algorithms that take as few iterations as possible. This paper concerns the following question: *How many iterations (i.e., linear optimizations over the underlying polyhedron P) do Dantzig–Wolfe-type algorithms require in order to compute approximate solutions to packing and covering problems?* We give lower bounds (worst-case and average-case) that match known worst-case upper bounds for a range of the relevant parameters.

Definition of Dantzig–Wolfe-type algorithms for packing/covering. We start with a formal definition of packing and covering.

Definition 1 (fractional packing and covering [31]). *An instance of* fractional packing (or fractional covering) *is a triple (A, b, P), where A is in $\mathbb{R}^{m\times n}$, b is in $\mathbb{R}^m_+$, and P is a polyhedron in $\mathbb{R}^n$ such that $Ax \geq 0$ for all $x \in P$. A* feasible solution *is any member of the set $\{x \in \mathbb{R}^n : Ax \leq b, x \in P\}$. (For covering, the constraint $Ax \leq b$ is replaced by $Ax \geq b$.)*

If such an x exists, the instance (A, b, P) is called feasible. *A* $(1+\epsilon)$-approximate solution *is an $x \in P$ such that $Ax \leq (1+\epsilon)b$ (for covering, such that $Ax \geq b/(1+\epsilon)$).*

Informally, a Dantzig–Wolfe-type algorithm, given a packing instance (A, b, P), computes a $(1+\epsilon)$-approximate solution, interacting with P only via linear optimizations of the following form:

$$\text{(1)} \qquad \textit{Given some } q \in \mathbb{R}^n_+\textit{, find an } x \in P \textit{ minimizing } q^\intercal x.$$

In our formal model, instead of P, the algorithm is given an *optimization oracle* for P, defined as follows.

Definition 2 (Dantzig–Wolfe-type algorithm for packing). *For any polyhedron $P \subseteq \mathbb{R}^n_+$, an* optimization oracle X_P for P *is a function $X_P : \mathbb{R}^n_+ \to P$ such that, for every input $q \in \mathbb{R}^n_+$, the output $x^* = X_P(q)$ satisfies $x^* \in P$ and $q^\intercal x^* = \min\{q^\intercal x : x \in P\}$.*

An algorithm is of Dantzig–Wolfe type *if, for each triple (A, b, X_P) where (A, b, P) is a packing instance and X_P is an optimization oracle for P, the algorithm (given input (A, b, X_P)) either decides correctly that the input (A, b, P) is infeasible or outputs a $(1 + \epsilon)$-approximate solution. The algorithm accesses P only by linear optimization via X_P: in each iteration, the algorithm computes one* oracle input $q \in \mathbb{R}^n_+$ *and then receives the* oracle output $X_P(q)$.

For covering, the definition is the same, with "max" replacing "min".

The oracle X_P above models how most Dantzig–Wolfe-type algorithms in the literature work and how they are analyzed: their analyses show that they finish within the desired time bound given any optimization oracle X_P for the polyhedron P. This paper studies the limits of such algorithms, or, more precisely, such analyses. For our lower bounds, all parts of the input (A, b, X_P), including X_P, are chosen by an adversary to the algorithm. Although the oracle X_P is not completely determined by the polyhedron P, the distinction between X_P and P is a minor technical issue.[1]

In the Held–Karp computation (for bounding the optimal traveling salesman tour) each oracle call $X_P(q)$ reduces to a minimum-spanning-tree computation with edge weights given by q. For multicommodity-flow problems, each oracle call typically reduces (depending on the underlying polyhedron) to either a shortest-path computation with edge weights given by q, a minimum-cost single-commodity-flow computation with edge costs given by q, or several such computations (one per commodity).

2. Main result: Lower bound on iteration complexity.

Recall our main question: *How many iterations (i.e., oracle calls) does a Dantzig–Wolfe-type algorithm require in order to compute a $(1 + \epsilon)$-approximate solution to a packing and covering problem*? Each call reveals some information about P. The algorithm must force the oracle to eventually reveal enough information to determine an $x \in P$ such that $Ax \le (1 + \epsilon)b$. In the worst case (for an adversarial oracle), how many calls does an optimal algorithm require? For fractional packing, the algorithm of [31] gives an upper bound of

$$O(\rho\epsilon^{-2} \log m),$$

where ρ, the *width* of the input, is $\rho(A, b, P) = \max_{x \in P} \max_i A_i x/b_i$ (where A_i denotes the ith row of A). Our main result (Theorem 11) is a lower bound that matches this upper bound for a range of parameters. Here is a simplified form of that lower bound.

Corollary 1 (iteration bound, simple form). *For every $\delta \in (0, 1/2)$, there exist positive $k_\delta, c_\delta > 0$ such that the following holds. For every two integers $m, n \ge k_\delta$ and*

[1] The value of $X_P(q)$ is determined by the polyhedron P for all oracle inputs $q \in \mathbb{R}^n_+$ except those that happen to be orthogonal to an edge of P, for which $\min\{q^\intercal x : x \in P\}$ has multiple minima, where $X_P(q)$ can break the tie arbitrarily.

every $\rho \geq 2$, there exists an input (A, b, X_P) (packing or covering, as desired) having m constraints, n variables, and width $O(\rho)$ with the following property:

For every $\epsilon \in (0, 1/10)$, every deterministic Dantzig–Wolfe-type algorithm and every Las Vegas–style[2] randomized Dantzig–Wolfe-type algorithm requires at least

$$c_\delta \cdot \min(\rho\,\epsilon^{-2} \log m,\, m^{1/2-\delta},\, n)$$

iterations to compute a $(1+\epsilon)$-approximate solution, given input (A, b, X_P).

That is, for every $\delta \in (0, 1/2)$, the worst-case iteration complexity of every Dantzig–Wolfe-type algorithm is at least $\Omega_\delta\big(\min(\rho\,\epsilon^{-2}\log m,\, m^{1/2-\delta},\, n)\big)$. Here we use the notation Ω_δ to signify that the constant factor hidden by the Ω notation is allowed to depend on δ (but no other parameters).

Section 4 sketches the proof idea. Section 6 gives a more detailed version (Theorem 11) with a full proof. Theorem 11 shows that in fact the bound holds with probability $1 - O(1/m^2)$ for random inputs drawn from a natural class: the polyhedron P is the regular n-simplex, $P = \{x \in \mathbb{R}^n_+ : \sum_i x_i = 1\}$, and the constraint matrix A is a random 0/1 matrix with independent and identically distributed (i.i.d.) entries. The resulting problem instance (A, b, P) is equivalent to finding an optimal mixed strategy for the column player of the two-player zero-sum game with payoff matrix A. (As a packing problem, the instance models the column player being the min player; as a covering problem, it models the column player being the max player.) The basic idea of the proof is to prove a corresponding lower bound on the minimum support size of any $(1+\epsilon)$-approximate solution $\hat{x}$ and then to argue that (for the inputs in question) each iteration increases the support size of $\hat{x}$ by at most 1.

Extending to products of polyhedra. Following one of the original models for Dantzig–Wolfe decomposition, many algorithms in the literature specialize when the polyhedron P is a Cartesian product $P = P_1 \times \cdots \times P_K$ of K polyhedra and optimization over P decomposes into independent optimizations over the individual polyhedra P_i. It is straightforward to extend our lower bound to this model by making A block-diagonal, thus forcing each subproblem to be solved independently. Extended in this way, the lower bound shows that the number of iterations (each optimizing over some individual polyhedron P_i) must be $\Omega(\sum_i \min(\epsilon^{-2}\rho_i \log m_i, m_i^{1/2-\delta}, n_i))$, where polyhedron P_i has n_i variables and width ρ_i, and A has m_i constraints on P_i's variables. This lower bound matches known upper bounds (e.g., $O(\sum_i \epsilon^{-2}\rho_i \log m_i)$) for a range of the parameters.

2.1. Comparison with previous and related works. Recall the known upper bound of $O(\rho\,\epsilon^{-2}\log m)$ iterations in the worst case (see, e.g., [31]). It follows that the lower bound here is tight for a certain range of the parameters, roughly in the regime $\rho\epsilon^{-2} \ll \min(\sqrt{m}, n)$. This suggests two directions for proving stronger upper bounds. The first direction is to look for better upper bounds outside of the regime $\rho\epsilon^{-2} \ll \min(\sqrt{m}, n)$. A few such bounds are known (e.g., $O(\min(\rho, m)\,\epsilon^{-2}\log m)$ iterations [11, 36] and $O(m(\epsilon^{-2} + \log m))$ iterations [14]), but these leave a large gap with respect to any known lower bound. The second direction is to consider non-Dantzig–Wolfe-type algorithms, as discussed later.

Dantzig–Wolfe-type algorithms that allow approximate *oracles.* Many Dantzig–Wolfe-type algorithms in the literature are known to work even if run with an *approximate* optimization oracle. Define a $(1+\epsilon)$-*approximate oracle* to be a function $X'_P : \mathbb{R}^n_+ \to P$ such that, for all $q \in \mathbb{R}^n$,

[2] An algorithm having zero probability of error.

the output $x' = X'_P(q)$ *satisfies* $x' \in P$ *and* $q^{\intercal} x' \le (1+\epsilon)\min\{q^{\intercal} x : x \in P\}$.

A typical analysis proves a worst-case performance guarantee such as the following: *for every input* (A, b, X'_P) *such that* X'_P *is a* $(1+\epsilon/10)$*-approximate oracle, the algorithm computes a correct output using* $O(\rho \log(m)/\epsilon^2)$ *oracle calls.* A common motivation is that approximate oracles can require less time per iteration, leading to faster total run times.

Such an algorithm is, formally, of Dantzig–Wolfe-type per Definition 2. (The reason is trivial: every exact optimization oracle X_P per Definition 2 is also a valid approximate oracle as defined above, so such an algorithm necessarily works with every exact oracle as well.) Hence, the lower bounds in Corollary 1 and Theorem 11 apply to every such algorithm.

As we discuss next, our lower bounds imply that to obtain a better upper bound requires not only (i) an algorithm that uses an optimization oracle that does something other than pure linear optimization over P but also (ii) an analysis that makes use of that additional requirement.

Non-Dantzig–Wolfe-type algorithms. To obtain better general upper bounds for the parameter regime where the lower bound is tight, one has to consider non-Dantzig–Wolfe-type algorithms. Indeed, since the appearance of the conference version of this paper [22], researchers [6, 4, 19, 30] have built on the methods of Nesterov [29] (see also Nemirovski [28]) to obtain polynomial-time approximation schemes whose running times have better dependence on ϵ. These algorithms bypass the lower bound by optimizing nonlinear convex functions instead of linear functions (or by linear optimization over P but with side constraints).

Bienstock and Iyengar [4] give an algorithm that for a given $\epsilon > 0$ and packing input

$$\{x \in \mathbb{R}^n : Ax \le b,\, x \in P\}$$

finds a $(1+\epsilon)$-approximate solution by using calls to a convex quadratic program over a set of the form

$$\{x \in P : \forall j.\ 0 \le x_j \le \lambda\},$$

where the value of λ can be adjusted by the algorithm in each iteration. Such an algorithm violates the assumption of our lower bound in two ways: the objective function is nonlinear, and the optimization takes place not over P but over the intersection of P with a hypercube of specified side-lengths. Bienstock and Iyengar also give an algorithm for covering; it similarly violates the assumptions of our lower bound. For their algorithms, the number of iterations is bounded by $O(\epsilon^{-1}\sqrt{Kn\log m})$, where K is the maximum number of nonzero elements in any row of A. Each iteration calls the quadratic-programming oracle.

How difficult is convex quadratic programming? Using the ellipsoid algorithm (see [27, 15]), quadratic programming over an n-dimensional convex set can be reduced to a polynomial number of calls to a linear-optimization oracle for that set. However, the polynomial is quite large. Bienstock and Iyengar also show that it suffices to approximate the convex quadratic objective function by a piecewise linear objective function. In either case, the required oracle is generally more expensive computationally than linear optimization over the original convex set.

Bienstock and Iyengar illustrate their method with an application to variants of multicommodity flow. Nesterov [30] also gives an approximation algorithm for a variant of multicommodity flow. In both cases, the number of iterations is proportional to ϵ^{-1} instead of ϵ^{-2}. However, the dependence of the overall running time on the size of the problem is worse, by a factor of at least the number of commodities.

Chudak and Eleutério build on the techniques of Nesterov to give an approximation scheme for a linear-programming relaxation of facility location [6]. The running time of their algorithm is $\tilde{O}((nm)^{3/2}/\epsilon)$, where nm is the number of facilities times the number of clients. In contrast, a Dantzig–Wolfe-type algorithm can be implemented to run in time $\tilde{O}(N/\epsilon^2)$, where $N \leq nm$ is the input size—the number of (facility, client) pairs with finite distance [37].

Iyengar, Phillips, and Stein [19] use the method of Nesterov to obtain approximation schemes for certain semidefinite programs. For problems previously addressed using the method of Plotkin, Shmoys, and Tardos [31], their running times, while proportional to ϵ^{-1}, have worse dependence on problem size.

For the important special case when the polyhedron P is the positive orthant (e.g., problems of the form $\max\{c^{\mathsf{T}}x : Ax \leq b, x \geq 0\}$), a recent breakthrough by Allen-Zhu and Orecchia runs in $\tilde{O}(N/\epsilon)$ time for packing, or $\tilde{O}(N/\epsilon^{1.5})$ time for covering, where N is the number of nonzeros in the constraint matrix [1]. The algorithms are not Dantzig–Wolfe-type algorithms.

Does the regime in which the bound is tight contain interesting problems? Recall that the bound is tight in (roughly) the regime $\rho\epsilon^{-2} \ll \min(\sqrt{m}, n)$. For some interesting classes of problems, the width ρ is either constant (for example, zero-sum games with payoffs in $[0, 1]$ and value bounded away from 0 and 1) or a function of m and/or n that grows slowly (a celebrated recent example is for maximum flow in undirected graphs [5], in which, for n-node graphs, the width is $\widetilde{O}(n^{1/3})$). "Small width" problems such as these (with, say, constant ϵ) lie in the regime.

Related lower bounds. Khachiyan [20] proves an $\Omega(\epsilon^{-1})$ lower bound on the number of iterations to achieve an error of ϵ. Grigoriadis and Khachiyan [13, Section 2.8] observe that for the packing problem "*find* $x \in \Delta^m$ *such that* $Ix \leq \mathbf{1}/m$" (where Δ^m is the m-simplex, I is the identity matrix, and $\mathbf{1}$ is the all-ones vector in $\mathbb{R}^m$) any 0.5-approximate solution x has to have support of size at least $m/2$, and that this gives an $m/2$ lower bound on the number of oracle calls for any Dantzig–Wolfe-type algorithm to return a 0.5-approximate solution. (Consider also that the *covering* problem "*find* $x \in \Delta^m$ *such that* $Ix \geq \mathbf{1}/m$" requires at least m iterations to return *any* approximate solution.) These inputs have large width, $\Theta(m)$, complementing our lower bound.

Grigoriadis and Khachiyan [13, Section 3.3] generalize their observation above to give a lower bound on the number of calls required by any algorithm in a class they call *restricted price-directed-decomposition* (PDD). Their model, different from the one studied here, focuses on product-of-polyhedra packing inputs of the form $x \in P = P_1 \times P_2 \times \cdots \times P_K$ and $Ax \leq b$. In each iteration, the algorithm computes a single vector y and the oracle returns an $x \in \widehat{P}$ minimizing $(y^{\mathsf{T}}A)x$, where $\widehat{P} = \{x \in P : \forall j.\ x_j \leq \mu_j\}$, for some vector μ (subject, crucially, to restrictions on μ). They show that any such algorithm must use at least $\min(m, k)/\text{polylog}\, m$ iterations to compute a 0.5-approximate solution.

Freund and Schapire [10], in independent work in the context of learning theory, prove a lower bound on the net "regret" of any adaptive strategy that plays repeated zero-sum games against an adversary. Their proof is based on repeated random games. They study a wider class of problems (giving the adversary more power), so their lower bound does not apply to Dantzig–Wolfe-type algorithms as defined here.

Sublinear-time randomized algorithms for explicit packing and covering. In the special case of two-player zero-sum games with payoff matrix A where each payoff A_{ij} is in $[0,1]$, randomized algorithms can compute solutions with additive error ϵ in sublinear time [12] (see also [23]). Deterministic algorithms cannot [12].

3. Small-support mixed strategies for zero-sum games.

To prove the lower bound on iteration complexity, we prove an analogous lower bound (Theorem 8) on the minimum support size[3] of any $(1+\epsilon)$-approximate mixed strategy x for two-player zero-sum games.[4] Here is a simplified form of the support-size lower bound.

COROLLARY 2 (support bound, simple form). *For every $\delta \in (0,1/2)$, there exist $k_\delta > 0$, $c_\delta > 0$ such that, for every two integers $m, n \geq k_\delta$ and every $p \in (0,1/2)$, there exists a two-player zero-sum matrix game A with m rows, n columns, and value $\Omega(p)$ having the following property:*

For every $\epsilon \in (0,1/10)$, every $(1+\epsilon)$-approximate mixed strategy for the column player of A (as either the max player or the min player) has support size at least

$$c_\delta \cdot \min(p^{-1}\,\epsilon^{-2}\log m,\, m^{1/2-\delta},\, n).$$

Section 4 sketches the proof idea. Section 5 fully proves a more detailed version (Theorem 8), showing that in fact the bound holds with probability $1 - O(1/m^2)$ when the payoff matrix A is a random 0/1 matrix with i.i.d. entries.

Matching upper bound. The lower bound in Theorem 8 matches (up to constant factors) a previous small-support upper bound by Lipton and Young [26]: *For every two-player zero-sum game with payoffs in* $[0,1]$ *and value p, each player has a* $(1+\epsilon)$*-approximate mixed strategy with support of size* at most $O(p^{-1}\epsilon^{-2}\log m)$, *where m is the number of pure strategies available to the opponent.* The proof is simple.[5] Derandomizing the proof via the method of conditional probabilities gives a Dantzig–Wolfe-type algorithm to compute the $(1+\epsilon)$-approximate strategy using $O(p^{-1}\,\epsilon^{-2}\log m)$ oracle calls [35].

In the context of Nash equilibria, similar small-support upper bounds have subsequently been shown and used for algorithmic upper bounds (see, e.g., [25, 8, 16]).

4. Proof ideas.

This section sketches how a support-size bound (Corollary 2) implies an iteration-complexity bound (Corollary 1) and how we prove a support-size bound such as Corollary 2. See sections 5 and 6 for the more detailed theorems that imply these corollaries, with detailed proofs based on the ideas sketched here.

How a support-size bound implies an iteration bound. We sketch the idea for packing. The idea also works for covering. Fix the parameters m, n, ρ as in Corollary 1. Let probability $p = 1/\rho$. Let A be the $m \times n$ payoff matrix for any zero-sum game with the properties described in Corollary 2.

[3] The support of x is the set $\{j : x_j \neq 0\}$.

[4] A *mixed strategy* for the column player of A is an $x \in \Delta^n$, where $\Delta^n = \{x \in \mathbb{R}^n_+ : \sum_j x_j = 1\}$ is the regular n-simplex. The *expected payoff (or value)* of x (for MAX as the column player) is $\min_i A_i x$. The value of the *game A* (with MAX as the column player) is $\max_{x\in\Delta^n} \min_i A_i x$, i.e., the maximum expected payoff of any mixed strategy. With MIN as the column player, the value of the game is $\min_{x\in\Delta^n} \max_i A_i x$. A $(1+\epsilon)$-*approximate* mixed strategy x is one whose expected payoff is within a factor of $1+\epsilon$ of the value of the game.

[5] Consider a mixed strategy that plays a pure strategy chosen uniformly from a multiset S of s pure strategies, where S is formed by sampling s times i.i.d. from the optimal mixed strategy. Use a standard Chernoff bound and the union bound to show that this mixed strategy has the desired properties with positive probability.

Let $V_{\min}(A)$ denote the value of the game with MIN (the min player) as the column player. Let $\mathsf{packing}(A)$ denote the packing problem (A, b, Δ^n), where each $b_i = V_{\min}(A)$ and $\Delta^n = \{x \in \mathbb{R}^n_+ : \sum_j x_j = 1\}$ is the simplex. This is equivalent to the zero-sum game with payoff matrix A and MIN as the column player. Via this equivalence, any $(1+\epsilon)$-approximate solution $\hat{x}$ for $\mathsf{packing}(A)$ is also a $(1+\epsilon)$-approximate mixed strategy for MIN as the column player of the game. Assuming Corollary 2, any such solution $\hat{x}$ must have support of size $\Omega_\delta(\min(\rho\,\epsilon^{-2}\log m,\, m^{1/2-\delta},\, n))$, where $\rho = 1/p$.

Whenever the Dantzig–Wolfe-type algorithm queries the oracle for Δ^n, the oracle can respond to the query q with a vertex of Δ^n. Each such vertex has just one nonzero coordinate. For the algorithm to be correct, the final solution $\hat{x}$ must be a convex combination of these vertices, so the number of queries must be at least the size of the support of $\hat{x}$. To finish, note that the width of $\mathsf{packing}(A)$ is $O(\rho)$ because the width is $1/V_{\min}(A) = 1/\Omega(p)$.

Proving the support-size bounds (e.g., Corollary 2). We sketch a proof of Corollary 2 when the column player is MIN. (The other case is similar.) Fix the parameters m, n, p as in Corollary 2. Let $\ell = p^{-1}\epsilon^{-2}\log m$ be the desired lower bound.

Take A to be a random 0/1 matrix with i.i.d. entries, where each entry A_{ij} is 1 with probability p. W.h.p., the value of A is at least $(1-\epsilon)p$. (This is easily proven by considering MAX's uniform mixed strategy.) Now consider any subgame B of A induced by just ℓ columns. The subgame B is highly skewed—there are many more rows for MAX than columns for MIN—so, by a discrepancy argument, w.h.p., the value of B is high: *at least* $(1+3\epsilon)p$. (Here is a sketch of the discrepancy argument. B is a random 0/1 matrix where each entry is 1 with probability p. Since the number of rows m is much higher than the number of columns ℓ, w.h.p. B has a substantial number of rows that have a relatively large number—at least $(1+5\epsilon)p\ell$—of ones, and, w.h.p., if MAX (the row player) plays uniformly on just these rows, MAX guarantees a payoff of at least $(1+3\epsilon)p$ for the subgame B.)

Then subgame B has value at least $(1+3\epsilon)p$, while A has value at most $(1+\epsilon)p$. Since $(1+\epsilon)^2 p < (1+3\epsilon)p$, no $(1+\epsilon)$-approximate mixed strategy $\hat{x}$ can be supported by just the columns of B. By a union bound over the $\binom{n}{\ell}$ submatrices B with ℓ columns, w.h.p., there is no such B that can support any $(1+\epsilon)$-approximate mixed strategy $\hat{x}$, in which case there is no $(1+\epsilon)$-approximate strategy $\hat{x}$ with support of size ℓ.

This yields the corollary for any single $\epsilon \in (0,1)$. To complete the argument, we extend the bound to all $\epsilon \in (0,1)$ simultaneously (for the given A) by applying the single-ϵ case for ϵ in a geometrically increasing sequence $\{\epsilon_0, 2\epsilon_0, 4\epsilon_0, \ldots, 1/10\}$ and then appealing to monotonicity for the remaining ϵ.

5. Theorem 8 (support bound).

In the rest of this section, we state and prove Theorem 8. Theorem 8 implies Corollary 2. We first give a few self-contained utility lemmas. The first is a standard Chernoff bound, which we give without proof.

LEMMA 3 (Chernoff bound). *Let X be the average of t independent, 0/1 random variables, each with expectation $p \in (0,1]$. For every $\epsilon \in (0,1]$,*

(i) $\Pr[X \le (1-\epsilon)p] \;\le\; \exp\left(-\epsilon^2 pt/3\right)$;

(ii) $\Pr[X \ge (1+\epsilon)p] \;\le\; \exp\left(-\epsilon^2 pt/3\right)$.

The next utility lemma states that the Chernoff bound above is tight up to constant factors in the exponent, as long as the bound is below $1/e$. That the Chernoff bound is tight (in most cases) is standard folklore.

LEMMA 4 (tightness of Chernoff bound). *Let X be the average of s independent, 0/1 random variables (r.v.). For every $\epsilon \in (0, 1/2]$ and $p \in (0, 1/2]$, if $\epsilon^2 ps \geq 3$, the following hold:*

(i) *If each r.v. is 1 with probability at most p, then* $\Pr[X \leq (1-\epsilon)p] \geq \exp\left(-9\epsilon^2 ps\right)$.

(ii) *If each r.v. is 1 with probability at least p, then* $\Pr[X \geq (1+\epsilon)p] \geq \exp\left(-9\epsilon^2 ps\right)$.

A detailed proof is in the appendix.

The third utility lemma leverages the Chernoff bound to give straightforward bounds on the likely value of random matrix games. Note that independence of the entries of the matrix is assumed only within each individual row.

LEMMA 5 (naive bounds on $V_{\max}()$ and $V_{\min}()$). *Let M be a random 0/1 $r \times c$ payoff matrix such that within each row of M the entries are independent. Let $\epsilon, p \in (0, 1]$.*

(i) *If each entry of M is 1 with probability at least p, then*

$$\Pr_M[V_{\min}(M^\intercal) \leq (1-\epsilon)p] \ = \ \Pr_M[V_{\max}(M) \leq (1-\epsilon)p] \ \leq \ r\ \exp(-c\,\epsilon^2 p/3).$$

(ii) *If each entry of M is 1 with probability at most p, then*

$$\Pr_M[V_{\max}(M^\intercal) \geq (1+\epsilon)p] \ = \ \Pr_M[V_{\min}(M) \geq (1+\epsilon)p] \ \leq \ r\ \exp(-c\,\epsilon^2 p/3).$$

Proof. (i) The equality in (i) holds because, by von Neumann's min-max theorem (strong LP duality), $V_{\min}(M^\intercal) = V_{\max}(M)$. We prove the inequality. MAX can play a uniform mixed strategy on the c columns. By the Chernoff bound, the probability that any given row then gives MIN expected payoff less than $(1-\epsilon)p$ is at most $\exp(-\epsilon^2 pc/3)$. By the union bound, the probability that *any* of the r rows gives MIN expected payoff less than $(1-\epsilon)p$ is at most $r\exp(-\epsilon^2 pc/3)$.

(ii) The proof is similar (MIN can play a uniform mixed strategy on the c columns). ☐

The next lemma uses the discrepancy argument outlined in the proof sketch in section 4 to quantify the disadvantage to the column player for playing a random game with many fewer columns than rows. The reader may wish to review Figure 2 for the notation.

LEMMA 6 (skewed game 1). *Let B be a random 0/1 $m \times s$ payoff matrix whose entries are i.i.d., each being 1 with probability $p \in (0, 1/2]$. Let $\epsilon \in (0, 1/10]$. Assume that $\epsilon^2 ps \geq 1$. Then, for $t = m\ \exp(-250\epsilon^2 ps)$, and $\beta = s\exp(-\epsilon^2 tp/15)$,*

(i) $\Pr_B[V_{\max}(B) \geq (1-3\epsilon)p] \leq\ 2\beta$;

(ii) $\Pr_B[V_{\min}(B) \leq (1+3\epsilon)p]\ \leq\ 2\beta$.

(When we apply the bound, s will be chosen so that t is large and β is small.)

Proof. (i) Let D be the submatrix formed by the $\lceil t/2 \rceil$ rows of B that have the fewest ones, as shown in Figure 1. Say that a row of B is *deviant* if the average of its entries is at most $p' = (1-5\epsilon)p$. We claim that *the probability that D has a nondeviant row is at most β.*

To prove the claim, let r.v. d be the number of deviant rows in B. By Lemma 4 (tightness of the Chernoff bound, with $\epsilon' = 5\epsilon$, using here the assumption that $\epsilon^2 ps \geq 1$ and that $\epsilon \leq 1/10$), the probability that a given row of B is deviant is at least $\exp(-9(5\epsilon)^2 ps) \geq t/m$ (by the choice of t). Thus, by the choice of t, the expected number of deviant rows is at least t. Since the rows of B are independent, by the Chernoff bound (with $\epsilon = 1/2$), the probability that $d \leq t/2$ is at most $\exp(-(1/2)^2 t/3) = \exp(-t/12)$, which (using $\epsilon \leq 1/10$, $p \leq 1/2$, and $s \geq 1$) is less

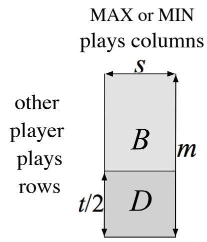

FIG. 1. *Given skewed matrix B, submatrix D contains the $t/2$ rows with the most (or least) 1's. By playing uniformly on the rows of D,* MAX *(or* MIN*) forces value at most $(1-3\epsilon)p$ (or at least $(1+3\epsilon)p$) w.h.p..*

than $\beta = s\exp(-\epsilon^2 tp/15)$. This proves the claim because if $d > t/2$, then all rows in D are deviant.

Conditioned on all $\lceil t/2 \rceil$ rows in D being deviant, within each row of D, by symmetry, the probability that any given entry equals 1 is at most $p' = (1-5\epsilon)p$. Also, within any column of D the entries are independent. Thus, part (ii) of Lemma 5 (the naive bounds) applied to the value of the *transpose*, $V_{\min}(D^\intercal)$, implies that $\Pr_B[V_{\min}(D^\intercal) \geq (1+\epsilon)p' \mid \textit{all rows of } D \textit{ are deviant}]$ is at most $s\exp(-(t/2)\epsilon^2 p'/3)$, which (using $\epsilon \leq 1/10$ and the choice of p') is at most $\beta = s\exp(-\epsilon^2 tp/15)$.

The latter bound and previous claim imply that, unconditionally, $\Pr_B[V_{\min}(D^\intercal) \geq (1+\epsilon)p']$ is at most $\beta + (1-\beta)\beta$, which is less than 2β.

By von Neumann's min-max theorem, $V_{\max}(D) = V_{\min}(D^\intercal)$. Since D consists of a subset of B's rows and MIN is the row player, $V_{\max}(B) \leq V_{\max}(D)$. Transitively, $V_{\max}(B) \leq V_{\min}(D^\intercal)$. With the preceding paragraph, this implies that $\Pr_B[V_{\max}(B) \geq (1+\epsilon)p']$ is at most 2β. To finish, note that $(1+\epsilon)p' = (1+\epsilon)(1-5\epsilon)p \leq (1-3\epsilon)p$, as $\epsilon \leq 1/5$.

(ii) Say that a row of B is deviant if the average of its entries is at *least* $p' = (1+5\epsilon)p$. Let D be the $\lceil t/2 \rceil$ rows of B with the *most* ones. Now proceed exactly as in part (i). To finish, note that $(1-\epsilon)p' = (1-\epsilon)(1+5\epsilon)p \geq (1+3\epsilon)p$, as $\epsilon \leq 1/5$. ∎

We use Lemma 6 only to prove the next lemma, which just specializes it to a convenient choice of s (the number of columns). Namely, we take $s = \lfloor 4\ell \rfloor$, where $\ell = \delta\, p^{-1}\epsilon^{-2}\ln(m)/\,1000$ is the lower bound we will seek later.

LEMMA 7 (skewed game 2). *Let B be a random $m \times s$ 0/1 payoff matrix whose entries are i.i.d., each entry being 1 with the same probability $p \in (0, 1/2]$. Let $\epsilon \in (0, 1/10]$ and $\delta \in (0, 1/2)$. Let $s = \lfloor \delta \ln(m)/\,250\epsilon^2 p \rfloor$. Assume that $s \leq m^{1/2-\delta}$, that $n \leq m^{1/\delta}$, and that m is sufficiently large (exceeding some constant that depends only on δ). Then*

(i) $\Pr_B[V_{\max}(B) \geq (1-3\epsilon)p] \;\leq\; 1/n^{2s}$;

(ii) $\Pr_B[V_{\min}(B) \leq (1+3\epsilon)p] \;\leq\; 1/n^{2s}$.

Proof. We check the technical assumptions necessary to apply Lemma 6 and check that the upper bound from that lemma implies the upper bound claimed in this lemma. By inspection of s, the condition $\epsilon^2 ps \geq 1$ of Lemma 6 is satisfied for $m = \exp(\Omega(1/\delta))$. To finish, we show, for this s and $t = m\exp(-250\epsilon^2 ps)$ from Lemma 6, that the upper bound $2s\exp(-\epsilon^2 tp/15)$ from that lemma is at most $1/n^{2s}$ (for large enough m).

If $s = 0$, the corollary is trivial, so assume without loss of generality that $s \geq 1$.

given parameters	*determined from given*
m = number of constraints (rows)	$A = m \times n$, 0/1 matrix w/ i.i.d. entries
n = number of variables (columns)	$\mathsf{packing}(A)$ = *"find $x \in \Delta^n$ minimizing $\max_i Ax$"*
δ = constant in (0,1/2)	$\mathsf{covering}(A)$ = *"find $x \in \Delta^n$ maximizing $\min_i Ax$"*
ρ = approximate width, $\rho \geq 2$	$V_{\min}(A)$ = value of game A if MIN plays cols
$p = 1/\rho = \Pr[A_{ij} = 1]$, $p \in (0, 1/2)$	$V_{\max}(A)$ = value of game A if MAX plays cols

FIG. 2. *Notation for Theorems* 8 *and* 11.

Then

(2)	$\delta \ln(m)/\,500\epsilon^2 p \leq s \leq \delta \ln(m)/\,250\epsilon^2 p$	By the choice of s and $s \geq 1$.
(3)	$t \geq m^{1-\delta}$	By substituting the right-hand side (RHS) of (2) for s in the definition of t and simplifying.
(4)	$m^{1-2\delta} \geq s^2$	Squaring both sides of assumption $s \leq m^{1/2-\delta}$.
(5)	$m^{1-\delta} \geq 50 \times 500\, m^{1-2\delta}/\delta^2$	For sufficiently large m (depending only on δ).
(6)	$m^{1-\delta} \geq 50 \times 500\, s^2/\delta^2$	Substituting RHS of (4) for $m^{1-2\delta}$ in (5).
(7)	$t \geq 50 \times 500\, s^2/\delta^2$	By transitivity on (3) and (6).
(8)	$t \geq 50s \ln(m)/\delta\epsilon^2 p$	By substituting the left-hand side (LHS) of (2) for one s in (7) and simplifying.
(9)	$t \geq 15[2s\ln(m)/\delta + \ln 2s]/\epsilon^2 p$	By (8) and calculation for large enough m.
(10)	$t \geq 15[2s \ln n + \ln 2s]/\epsilon^2 p = 15 \ln(2sn^{2s})/\epsilon^2 p$	By (9) and assumption $n \leq m^{1/\delta}$, that is, $\ln n \leq \ln(m)/\delta$.
(11)	$\epsilon^2 tp/15 \geq \ln(2sn^{2s})$	Rearranging (10).
	$2s \exp(-\epsilon^2 tp/15) \leq 1/n^{2s}$	By (11), taking exponentials and rearranging. $\square$

This concludes the utility lemmas. Next we state and prove Theorem 8.

THEOREM 8 (support-size bound). *For every constant $\delta \in (0, 1/2)$, there exists constant $k_\delta > 0$ such that the following holds. Fix arbitrary integers $m, n > k_\delta$ and arbitrary $p \in (0, 1/2)$. Let ϵ_0 be such that $p^{-1}\,\epsilon_0^{-2} \ln(m) = \min(m^{1/2-\delta}, n/9)$. Assume $n \leq m^{1/\delta}$ and $\epsilon_0 \leq 1/10$. Let A be a random $m \times n$ 0/1 matrix with i.i.d. entries, where each entry A_{ij} is 1 with probability p. Then, with probability $1 - O(1/m^2)$,*

1. *both $V_{\max}(A)$ and $V_{\min}(A)$ lie in the interval $[(1-\epsilon_0)p, (1+\epsilon_0)p]$, and*
2. *for all $\epsilon \in [\epsilon_0, 1/10]$, every $(1+\epsilon)$-approximate mixed strategy for the column player (as* MIN *or as* MAX*) has support of size at least $\delta\, p^{-1}\, \epsilon^{-2} \ln(m)\,/\,1000$.*

Proof. All probabilities in the proof are with respect to the random choice of A.

Part 1, *bounds on $V_{\min}(A)$ and $V_{\max}(A)$:* By the naive bound (Lemma 5), the probability that either $V_{\min}(A)$ or $V_{\max}(A)$ falls either before or after the interval is at most

$$2\,m\,\exp(-n\,\epsilon_0^2\,p/3) + 2\,n\,\exp(-m\,\epsilon_0^2\,p/3).$$

The first of the two terms is at most $2/m^2$ because the definition of ϵ_0 implies $\epsilon_0^2\,p \geq 9\ln(m)/n$.

Likewise, the second of the first two terms is at most $2/m^2$ because, using the definition of ϵ_0 again, $\epsilon_0^2\,p \ge m^{1/2-\delta}$, so

$$m\,\epsilon_0^2\,p \;\ge\; m^{1/2+\delta} \;\ge\; m^{1/2} \;\ge\; 3\,\ln n \;+\; 6\ln m$$

(using that m is large enough so that $0.9\sqrt{m} \ge 3\delta^{-1}\ln m \ge 3\ln n$ and $0.1\sqrt{m} \ge 6\ln m$).

Part 2. Define r.v. $S^*_{\min}(\epsilon)$ to be the minimum support size of any mixed strategy that achieves value $(1+\epsilon)p$ or less for MIN as the column player of A. Analogously let $S^*_{\max}(\epsilon)$ be the minimum support size that achieves value at least $(1-\epsilon)p$ for MAX as the column player.

Next we use the skewed-game lemma (Lemma 7) and the union bound to bound the probability that either player (playing the columns of A) has a good strategy with small support. Let $\ell(\epsilon) = \delta\,p^{-1}\epsilon^{-2}\ln(m)/1000$ denote the desired lower bound on the support size for a given ϵ.

OBSERVATION 8.1. *Let $\epsilon \in [\epsilon_0, 1/10]$. Let $s = \lfloor 4\,\ell(\epsilon)\rfloor = \lfloor \delta\ln(m)/\,250\epsilon^2 p\rfloor$. If $s \le m^{1/2-\delta}$ and $n \le m^{1/\delta}$, then*

(i) $\Pr_A[S^*_{\max}(3\epsilon) \le s] \;\le\; 1/n^s$, and

(ii) $\Pr_A[S^*_{\min}(3\epsilon) \le s] \;\le\; 1/n^s$.

(Note that $S^*_{\max}(3\epsilon) \le s$ iff MA]X can get value $(1-3\epsilon)p$ or more using at most s columns.)

Proof. (i) If MAX has a mixed strategy with support of size s that has value at least $(1-3\epsilon)p$, then A has an $m \times s$ submatrix B with $V_{\max}(B) \ge (1-3\epsilon)p$.

Consider all $\binom{n}{s}$ possible such submatrices B. By Lemma 7, given any one of these submatrices B, the probability of $V_{\max}(B) \ge (1-3\epsilon)p$ is at most $1/n^{2s}$. Thus, by the union bound, the probability that *any* such submatrix B of A has $V_{\max}(B) \ge (1-3\epsilon)p$ is at most $\binom{n}{s}/n^{2s} \le 1/n^s$.

The proof for (ii) is essentially the same. ◻

Observation 8.1 bounds the probability of failure for a *single* given ϵ. We want to show that w.h.p. the bound holds for all $\epsilon \in (0, 1/10]$ simultaneously. We start by considering a sequence Q of geometrically increasing ϵ values: $Q = \{2^i\epsilon_0 : i = 0, 1, 2, \ldots, \lfloor\log_2(0.1/\epsilon_0)\rfloor\}$.

The maximum ϵ in Q is just less than 0.1.

OBSERVATION 8.2. *With probability $1 - O(1/m^2)$, for all $\epsilon \in Q$, support of size $4\ell(\epsilon)$ is necessary for* MAX *to achieve value $(1-3\epsilon)p$ or for* MIN *to achieve value $(1+3\epsilon)p$. Specifically, for n and m large enough (as a function of δ), with probability $1 - O(1/m^2)$, for all $\epsilon \in Q$, $S^*_{\max}(3\epsilon) > 4\,\ell(\epsilon)$ and $S^*_{\min}(3\epsilon) > 4\,\ell(\epsilon)$.*

Proof. By Observation 8.1, for every ϵ in the set Q, the probability of the event $S^*_{\max}(3\epsilon) \le 4\,\ell(\epsilon)$ is at most $1/n^{\lfloor 4\ell(\epsilon)\rfloor}$. By the union bound, the probability that there exists an $\epsilon \in Q$ with $S^*_{\max}(3\epsilon) \le 4\,\ell(\epsilon)$ is at most $\sum_{\epsilon\in Q} 1/n^{\lfloor 4\ell(\epsilon)\rfloor}$. Using that $\ell(2^i\epsilon_0) = \ell(\epsilon_0)/4^i$ for $i \ge 0$ and the definition of Q, this sum is at most $\sum_{i=0}^{\infty} 1/n^{\lfloor 4^i\ell(0.1)\rfloor}$. The terms in this sum decrease supergeometrically, so the sum is proportional to its first term, which is at most $1/n^{\delta\ln(m)/10p\,-1}$, which is $O(1/m^2)$ as long as n and m are large enough (as a function of δ).

The proof for MIN is similar. ◻

To complete the proof of Theorem 8, we extend the previous observation to all $\epsilon \in [\epsilon_0, 1/10]$.

OBSERVATION 8.3. *With probability $1-O(1/m^2)$, for all $\epsilon \in [\epsilon_0, 1/10]$, support of size $\ell(\epsilon)$ is necessary for* MAX *to achieve value $(1-3\epsilon)p$ and for* MIN *to achieve value $(1+3\epsilon)p$. Specifically, for n and m large enough (as a function of δ), with probability $1 - O(1/m^2)$, for all $\epsilon \in [\epsilon_0, 1/10]$, $S^*_{\max}(3\epsilon) > \ell(\epsilon)$ and $S^*_{\min}(3\epsilon) > \ell(\epsilon)$.*

Proof. We show that if the event in Observation 8.2 happens, then the event desired above happens. Assume the former event happens, i.e., for all $\epsilon' \in Q$, $S^*_{\max}(3\epsilon') > 4\,\ell(\epsilon')$.

Now consider any $\epsilon \in [\epsilon_0, 1/10]$. By the choice of Q, there is some $\epsilon' \in Q$ such that $\epsilon \in (\epsilon'/2, \epsilon']$. Then we have

(i) $S^*_{\max}(3\epsilon) \geq S^*_{\max}(3\epsilon')$ (since $S^*_{\max}(\cdot)$ is monotone decreasing and $\epsilon \leq \epsilon'$),

(ii) $S^*_{\max}(3\epsilon') \geq 4\,\ell(\epsilon')$ (since $\epsilon' \in Q$), and

(iii) $\ell(\epsilon') > \ell(\epsilon)/4$ (by the definition of $\ell(\cdot)$ and $\epsilon > \epsilon'/2$).

By transitivity, we conclude that $S^*_{\max}(3\epsilon) > \ell(\epsilon)$ for all $\epsilon \in [\epsilon_0, 1/10]$.

The proof for $S^*_{\min}(\epsilon)$ is similar. ∎

We now finish the proof of Theorem 8, part (2). From part 1 of the theorem, with probability $1 - O(1/m^2)$, for all $\epsilon \in [\epsilon_0, 1/10]$, to achieve $(1+\epsilon)$-approximation, MAX must achieve absolute value at least $(1-\epsilon_0)p/(1+\epsilon) \geq (1-3\epsilon)p$. By Observation 8.3, with probability $1 - O(1/m^2)$, for all $\epsilon \in [\epsilon_0, 1/10]$, support size at least $\ell(\epsilon)$ is needed for MAX to achieve this absolute value. By the union bound, with probability $1 - O(2/m^2)$, every $(1+\epsilon)$-approximate strategy for MAX has support size at least $\ell(\epsilon)$. By a similar argument (using $(1+\epsilon_0)(1+\epsilon)p \leq (1+3\epsilon)p)$, with probability $1 - O(1/m^2)$, every $(1+\epsilon)$-approximate strategy for MIN also has support size at least $\ell(\epsilon)$. This completes the proof of Theorem 8. ∎

Before we prove Theorem 11, we observe that Corollary 2 is indeed just a simplified (and somewhat weaker) statement of Theorem 8.

Proof of Corollary 2. Fix any δ, m, n, and p as in the corollary. (Take k_δ in the corollary to be the same as in the theorem.) Assume without loss of generality that $n \leq m^{1/\delta}$. (Otherwise, decrease n to $n' = \lfloor m^{1/\delta} \rfloor \geq m^2$, apply the corollary to get a game with $m \times n'$ payoff matrix A', and then duplicate any of the columns $n - n'$ times to get an equivalent $m \times n$ game with the desired properties.)

Redefine $\ell(\epsilon) = p^{-1}\epsilon^{-2}\log m$. Fix ϵ_0 as in the theorem. The choice of ϵ_0 implies $\ell(\epsilon_0) = \Theta(\min(m^{1/2-\delta}, n))$, so the support bound desired for the corollary is equivalent to the following:

$\forall\, \epsilon \in (0, 1/10]$, *any* $(1+\epsilon)$*-approximate mixed strategy has support size* $\Omega_\delta\big(\ell\big(\max(\epsilon_0, \epsilon)\big)\big)$.

Assume without loss of generality that $\epsilon_0 \leq 1/10$. (If $\epsilon_0 > 1/10$, raise p until the corresponding ϵ_0 decreases to $1/10$; the corollary for the smaller p follows from the corollary for the larger p because in both cases the lower bound in question is the same: $\Omega_\delta(\min(m^{1/2-\delta}, n))$ for all $\epsilon \in (0, 1/10)$.)

Now we have $\epsilon_0 \leq 1/10$ and $n \leq m^{1/\delta}$. Applying Theorem 8, there are (many) $m \times n$ zero-sum matrix games with value $\Omega(p)$ such that, for all $\epsilon \in [\epsilon_0, 1/10]$, any $(1+\epsilon)$-approximate strategy for the column player requires support of size at least $\Omega_\delta(\ell(\epsilon))$. To finish, note that, for the remaining $\epsilon \in (0, \epsilon_0]$, any $(1+\epsilon)$-approximate strategy is also a $(1+\epsilon_0)$-approximate strategy, so it must have support size at least $\Omega_\delta(\ell(\epsilon_0)) = \Omega_\delta(\min(m^{1/2-\delta}, n))$, proving the corollary. ∎

6. Theorem 11 (iteration bound). Before we state and prove Theorem 11, we prove two utility lemmas. The first says that the output $\hat{x}$ of any Dantzig–Wolfe-type algorithm has to be a convex combination of the vectors output by the oracle.

LEMMA 9. *Suppose that a deterministic Dantzig–Wolfe-type algorithm, given some input* (A, b, X_P)*, returns a solution* $\hat{x} \in P$*. Then* $\hat{x}$ *must be a convex combination of the outputs returned by the oracle* X_P *during the computation. The same holds if the algorithm is randomized (and has zero probability of error).*

Proof. First we consider the deterministic case. Let Q denote the set of oracle inputs generated by the algorithm on input (A, b, X_P). Define polyhedron $P' \subseteq P$ to be the convex hull of the vectors output by X_P during the algorithm. That is, P' is the polyhedron whose vertices are $\{X_P(q) : q \in Q\}$. Suppose for contradiction that $\hat{x} \notin P'$, and consider the modified input (A, b, P'), with polyhedron P' instead of P. Define the oracle $X'_{P'}$ for the polyhedron P' such that $X'_{P'}(q)$ outputs a minimizer of $q^{\intercal}x$ among $x \in P'$. For $q \in Q$, break any ties among the minimizers for q by choosing $X'_{P'}(q) = X_P(q)$. This $X'_{P'}$ optimizes correctly over P'. Observe that it also has the following key property: *Let $q \in Q$ be any input that the algorithm gave to oracle X_P on input (A, b, X_P). Then, on input q, oracle $X'_{P'}$ gives the same output, $X_P(q)$, that X_P did.*

Consider rerunning the Dantzig–Wolfe-type algorithm, this time on the input $(A, b, X'_{P'})$. The Dantzig–Wolfe-type algorithm is deterministic, and, as observed above, $X_P(q) = X'_{P'}(q)$ for all inputs $q \in Q$ that the algorithm gave to the oracle when the algorithm ran on input (A, b, X_P). Recall that the algorithm interacts with the polyhedron only via the oracle (X_P or $X'_{P'}$). By induction on the number of queries, when run on $(A, b, X'_{P'})$, the algorithm behaves the same—that is, it makes the same sequence of queries and computes the same final answer $\hat{x}$—as it did when run on $(A, b, X'_{P'})$. But this is an incorrect output, as $\hat{x}$ is not in the polyhedron P' for the second input. This proves the lemma for the deterministic case.

Now consider running any (error-free) randomized Dantzig–Wolfe-type algorithm on (A, b, X_P). Suppose for contradiction that the algorithm has nonzero probability of producing an output $\hat{x}$ that is not a convex combination of the oracle outputs made during the run. Fix any such outcome that has positive probability, say $p' > 0$. Let Q, P', and $X'_{P'}$ be as in the proof above, and consider running the algorithm on input $(A, b, X'_{P'})$. With probability at least p', the algorithm will make the same random choices that it made in the first run. When this happens, then (as in the proof for the deterministic case) it returns the same vector $\hat{x}$, which is (just as for the deterministic case) an error, because $\hat{x} \notin P'$. Hence, the algorithm has positive probability of error on input $(A, b, X'_{P'})$. ∎

The next lemma is a convenient restatement of Theorem 8 in terms of $\mathsf{packing}(A)$ and $\mathsf{covering}(A)$.

Lemma 10. *For every constant $\delta \in (0, 1/2)$, there exists constant $k_\delta > 0$ such that the following holds. Fix any integers $m, n > k_\delta$ and any desired width $\rho \geq 2$. Let ϵ_0 be such that $\rho\,\epsilon_0^{-2}\ln(m) = \min(m^{1/2-\delta}, n/9)$. Assume $n \leq m^{1/\delta}$ and $\epsilon_0 \leq 1/10$. Let A be a random $m \times n$ 0/1 matrix with i.i.d. entries, where each entry A_{ij} is 1 with probability $p = 1/\rho$. Then, with probability $1 - O(1/m^2)$, for both* $\mathsf{packing}(A)$ *and* $\mathsf{covering}(A)$,

1. *the instance has width at most 2ρ;*
2. *for all $\epsilon \in [\epsilon_0, 1/10]$, every $(1+\epsilon)$-approximate solution has support size at least*

$$\delta\,\rho\,\epsilon^{-2}\ln(m)\,/\,1000.$$

Proof. Note that we take $p = 1/\rho$.

The $(1+\epsilon)$-approximate solutions to $\mathsf{packing}(A)$ and $\mathsf{covering}(A)$ are, respectively, the $(1+\epsilon)$-approximate mixed strategies for MIN and MAX (as the column player of the game with payoff matrix A). Thus, part 2 of Theorem 8 implies part 2 of the lemma.

Regarding part 1 of the lemma, suppose that part 1 of the theorem holds, so $V_{\min}(A)$ and $V_{\max}(A)$ are both at least $(1-\epsilon_0)p$. By definition of $\mathsf{packing}(A)$, each b_i is $V_{\min}(A)$, so the width is $\max_{x,i} A_i x / V_{\min}(A)$, where x ranges over the simplex. Since A is a 0/1 matrix and $\sum_j x_j = 1$ (and A is not all zeros, as $V_{\min}(A) \geq (1-\epsilon_0)p > 0$), we have $\max_{x,i} A_i x = \max_{ij} A_{ij} = 1$, so the width is $1/V_{\min}(A) \leq 2/p = 2\rho$. The same argument shows that $\mathsf{covering}(A)$ has width $1/V_{\max}(A) \leq 2\rho$. ∎

Next we state and prove Theorem 11.

THEOREM 11 (iteration bound). *For every constant $\delta \in (0, 1/2)$, there exists constant $k_\delta > 0$ such that the following holds. Fix arbitrary integers $m, n > k_\delta$ and an arbitrary desired width $\rho \geq 2$. Let ϵ_0 be such that $\rho\,\epsilon_0^{-2}\ln(m) = \min(m^{1/2-\delta}, n/9)$. Assume $n \leq m^{1/\delta}$ and $\epsilon_0 \leq 1/10$. Let A be a random $m \times n$ 0/1 matrix with i.i.d. entries, where each entry A_{ij} is 1 with probability $p = 1/\rho$. Then, with probability $1 - O(1/m^2)$, for both* $\mathsf{packing}(A)$ *and* $\mathsf{covering}(A)$,

1. *the instance has width at most 2ρ;*
2. *for all $\epsilon \in [\epsilon_0, 1/10]$, all deterministic Dantzig–Wolfe-type algorithms and all Las Vegas–style randomized Dantzig–Wolfe-type algorithms must make at least $\delta\,\rho\,\epsilon^{-2}\ln(m)\,/\,1000$ iterations to find a $(1+\epsilon)$-approximate solution.*

Proof. Fix any values of the parameters $\delta, m, n, \rho, p = 1/\rho$. Let A be as described. Assume the events 1 and 2 in Lemma 10 happen for A (as they do with probability $1 - O(1/m^2)$).

Part 1. Part 1 of the theorem is immediate from event 1 of Lemma 10.

Part 2. Let e_j denote the jth standard basis vector for $\mathbb{R}^n$, that is, the vector that is 1 in the jth coordinate and zero elsewhere, so that the set of vertices of Δ^n is $\{e_j : j \in [n]\}$.

Fix any oracle X_n whose output $X_n(q)$ for each input q is some vertex e_j of Δ^n (one minimizing $q^{\mathsf{T}} e_j$; breaking ties consistently). For any $\epsilon \in [\epsilon_0, 1/10]$, run the (deterministic or randomized) Dantzig–Wolfe-type algorithm on input (A, b, X_n). Let $\hat{x}$ be the $(1+\epsilon)$-approximate solution it returns.

By Lemma 9, $\hat{x}$ is a convex combination of the vectors returned by the oracle. Each such vector has just one nonzero coordinate. Thus, the number of nonzero coordinates in $\hat{x}$ is at most the number of iterations made by the algorithm. On the other hand, $\hat{x}$ is a $(1+\epsilon)$-approximate solution, so by event 2 of Lemma 10, the number of nonzero coordinates is at least the desired lower bound $\delta\,\rho\,\epsilon^{-2}\ln(m)\,/\,1000$. □

Finally, we observe that Corollary 1 is indeed just a simplified and somewhat weaker statement of Theorem 11. The proof is identical to the proof that Corollary 2 follows from Theorem 8.

Proof of Corollary 1. Fix any δ, m, n, and p as in the corollary. (Take k_δ in the corollary to be the same as in the theorem.) Assume without loss of generality that $n \leq m^{1/\delta}$. (Otherwise, decrease n to $n' = \lfloor m^{1/\delta} \rfloor \geq m^2$, apply the corollary to get $m \times n'$ packing or covering instances, and then duplicate any of the columns $n - n'$ times to get equivalent $m \times n$ instances with the desired properties.)

Let $\ell(\epsilon) = p^{-1}\epsilon^{-2}\log m$. Fix ϵ_0 as in the theorem. As $\ell(\epsilon_0) = \Theta(\min(m^{1/2-\delta}, n))$, the support bound desired for the corollary is equivalent to

$\forall \epsilon \in (0, 1/10]$, *any $(1+\epsilon)$-approximate solution has support size* $\Omega_\delta\big(\ell\big(\max(\epsilon_0, \epsilon)\big)\big)$.

Assume without loss of generality that $\epsilon_0 \leq 1/10$. (If $\epsilon_0 > 1/10$, lower ρ until the corresponding ϵ_0 decreases to $1/10$; the corollary for the larger ρ follows from the corollary for the smaller ρ because in both cases the lower bound in question is the same: $\Omega_\delta(\min(m^{1/2-\delta}, n))$ for all $\epsilon \in (0, 1/10)$.)

Now we have $\epsilon_0 \leq 1/10$ and $n \leq m^{1/\delta}$. Applying Theorem 11, there are (many) $m \times n$ packing/covering instances with width $O(\rho)$ such that, for all $\epsilon \in [\epsilon_0, 1/10]$, every $(1+\epsilon)$-approximate solution has support of size at least $\Omega_\delta(\ell(\epsilon))$. To finish, note that, for the remaining $\epsilon \in (0, \epsilon_0]$, any $(1+\epsilon)$-approximate solution is also a $(1+\epsilon_0)$-approximate solution, so it must have support size at least $\Omega_\delta(\ell(\epsilon_0)) = \Omega_\delta(\min(m^{1/2-\delta}, n))$, proving the corollary. □

Appendix. Lemma 4 (tightness of Chernoff bound). Here is the proof of Lemma 4—that a standard Chernoff bound is tight up to constant factors in the exponent for a particular range of the parameters (in particular, whenever the variables are 0 or 1, and 1 with probability 1/2 or less, and $\epsilon \in (0, 1/2)$, and the Chernoff upper bound is less than a particular constant). First we prove the following useful inequality.

LEMMA 12. *If* $1 \le \ell \le k-1$, *then*

$$\binom{k}{\ell} \;\ge\; \frac{1}{e\sqrt{2\pi\ell}}\Big(\frac{k}{\ell}\Big)^{\ell}\Big(\frac{k}{k-\ell}\Big)^{k-\ell}.$$

Proof. By Stirling's approximation, $i! = \sqrt{2\pi i}(i/e)^i e^{\lambda}$, where $\lambda \in [\frac{1}{12i+1}, \frac{1}{12i}]$. Thus, $\binom{k}{\ell}$ is

$$\frac{k!}{\ell!(k-\ell)!} \ge \frac{\sqrt{2\pi k}\,(\frac{k}{e})^k}{\sqrt{2\pi\ell}\,(\frac{\ell}{e})^{\ell}\;\sqrt{2\pi(k-\ell)}\,(\frac{k-\ell}{e})^{k-\ell}}\exp\Big(\frac{1}{12k+1}-\frac{1}{12\ell}-\frac{1}{12(k-\ell)}\Big)$$
$$\ge \frac{1}{\sqrt{2\pi\ell}}\Big(\frac{k}{\ell}\Big)^{\ell}\Big(\frac{k}{k-\ell}\Big)^{k-\ell}e^{-1}. \qquad \square$$

LEMMA 4 (tightness of Chernoff bound). *Let X be the average of s independent, 0/1 random variables (r.v.). For every $\epsilon \in (0, 1/2]$ and $p \in (0, 1/2]$, if $\epsilon^2 ps \ge 3$, the following hold:*

(i) *If each r.v. is 1 with probability at most p, then* $\Pr[X \le (1-\epsilon)p] \ge \exp\big(-9\epsilon^2 ps\big)$.

(ii) *If each r.v. is 1 with probability at least p, then* $\Pr[X \ge (1+\epsilon)p] \ge \exp\big(-9\epsilon^2 ps\big)$.

Proof. Part (i). Without loss of generality assume each 0/1 random variable in the sum X is 1 with probability *exactly* p. Note that $\Pr[X \le (1-\epsilon)p]$ equals the sum $\sum_{i=0}^{\lfloor(1-\epsilon)pk\rfloor}\Pr[X = i/k]$, and $\Pr[X = i/k] = \binom{k}{i}p^i(1-p)^{k-i}$.

Fix $\ell = \lfloor(1-2\epsilon)pk\rfloor + 1$. The terms in the sum are increasing, so the terms with index $i \ge \ell$ each have value at least $\Pr[X = \ell/k]$, so their sum has total value at least $(\epsilon pk - 2)\Pr[X = \ell/k]$. To complete the proof, we show that

$$(\epsilon pk - 2)\Pr[X = \ell/k] \;\ge\; \exp(-9\epsilon^2 pk).$$

The assumptions $\epsilon^2 pk \ge 3$ and $\epsilon \le 1/2$ give $\epsilon pk \ge 6$, so the left-hand side above is at least $\frac{2}{3}\epsilon pk\binom{k}{\ell}p^{\ell}(1-p)^{k-\ell}$. Using Lemma 12 to bound $\binom{k}{\ell}$, this is in turn at least $A\,B$, where $A = \frac{2}{3e}\epsilon pk/\sqrt{2\pi\ell}$ and $B = \big(\frac{k}{\ell}\big)^{\ell}\big(\frac{k}{k-\ell}\big)^{k-\ell}p^{\ell}(1-p)^{k-\ell}$. To finish we show that $A \ge \exp(-\epsilon^2 pk)$ and $B \ge \exp(-8\epsilon^2 pk)$.

OBSERVATION 4.1. $A \ge \exp(-\epsilon^2 pk)$.

Proof. The assumption $\epsilon^2 pk \ge 3$ implies $\exp(-\epsilon^2 pk) \le \exp(-3) \le 0.04$. To finish we show $A \ge 0.1$:

(12)	$12 \le pk$	By assumptions $\epsilon^2 pk \ge 3$ and $\epsilon \le 1/2$.
(13)	$\ell \le 1.1pk$	From (12) and $\ell \le pk + 1$ (from ℓ's definition).
(14)	$A \ge \dfrac{2}{3e}\epsilon\sqrt{pk/2.2\pi}$	Using (13) to substitute for ℓ in definition of A.
	$A \ge \dfrac{2}{3e}\sqrt{3/2.2\pi} \ge 0.1$	From (14) and $\epsilon\sqrt{pk} \ge \sqrt{3}$ (from $\epsilon^2 pk \ge 3$). $\square$

OBSERVATION 4.2. $B \geq \exp(-8\epsilon^2 pk)$.

Proof. Fix δ such that $\ell = (1-\delta)pk$. The choice of ℓ implies $\delta \leq 2\epsilon$, so the observation will hold as long as $B \geq \exp(-2\delta^2 pk)$. Taking each side of this latter inequality to the power $-1/\ell$ and simplifying, it is equivalent to

$$\frac{\ell}{pk}\Big(\frac{k-\ell}{(1-p)k}\Big)^{k/\ell-1} \;\leq\; \exp\Big(\frac{2\delta^2 pk}{\ell}\Big).$$

Substituting $\ell = (1-\delta)pk$ and simplifying, it is equivalent to

$$(1-\delta)\Big(1+\frac{\delta p}{1-p}\Big)^{\dfrac{1}{(1-\delta)p}-1} \;\leq\; \exp\Big(\frac{2\delta^2}{1-\delta}\Big).$$

Taking the logarithm of both sides and using $\ln(1+z) \leq z$ twice, it will hold as long as

$$-\delta + \frac{\delta p}{1-p}\Big(\frac{1}{(1-\delta)p}-1\Big) \;\leq\; \frac{2\delta^2}{1-\delta}.$$

The left-hand side above simplifies to $\delta^2/\,(1-p)(1-\delta)$, which is less than $2\delta^2/(1-\delta)$ because $p \leq 1/2$. ∎

Observations 4.1 and 4.2 imply $AB \geq \exp(-\epsilon^2 pk)\exp(-8\epsilon^2 pk)$. This implies part (i) of Lemma 4.

Part (ii). Without loss of generality assume each random variable is 1 with probability exactly p.

Note $\Pr[X \geq (1+\epsilon)p] = \sum_{i=\lceil(1-\epsilon)pk\rceil}^{n} \Pr[X = i/k]$. Fix $\hat{\ell} = \lceil(1+2\epsilon)pk\rceil - 1$.

The last ϵpk terms in the sum total at least $(\epsilon pk - 2)\Pr[X = \hat{\ell}/k]$, which is at least $\exp(-9\epsilon^2 pk)$. (The proof of that is the same as for (i), except with ℓ replaced by $\hat{\ell}$ and δ replaced by $-\hat{\delta}$ such that $\hat{\ell} = (1+\hat{\delta})pk$.) ∎

Acknowledgments. The authors thank editor Robert Krauthgamer and several anonymous referees for feedback that helped improve the presentation substantially, as well as Xinhua Zhang (Machine Learning Research Group, National ICT Australia) for pointing out that the argument in the conference version of this paper does not require $n \approx \Theta(m^{1/2})$ [22].

REFERENCES

[1] Z. Allen-Zhu and L. Orecchia, *Nearly-linear time positive LP solver with faster convergence rate* in Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC'15), pp. 229–236.

[2] S. Arora, E. Hazan, and S. Kale, *The multiplicative weights update method: A meta-algorithm and applications*, Theory Comput., 8 (2012), pp. 121–164.

[3] D. Bienstock, *Potential Function Methods for Approximately Solving Linear Programming Problems: Theory and Practice*, Kluwer Academic Publishers, Boston, MA, 2002.

[4] D. Bienstock and G. Iyengar, *Approximating fractional packings and coverings in $O(1/\varepsilon)$ iterations*, SIAM J. Comput., 35 (2006), pp. 825–854.

[5] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S. Teng, *Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, 2011, pp. 273–282.

[6] F. Chudak and V. Eleutério, *Improved approximation schemes for linear programming relaxations of combinatorial optimization problems*, in Integer Programming and Combinatorial Optimization, Springer, Berlin, 2005, pp. 81–96.

[7] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Oper. Res., 8 (1960), pp. 101–111.

[8] T. Feder, H. Nazerzadeh, and A. Saberi, *Approximating Nash equilibria using small-support strategies*, in Proceedings of the 8th ACM conference on Electronic commerce, 2007, pp. 352–354.

[9] L. R. Ford, Jr., and D. R. Fulkerson, *A suggested computation for maximal multicommodity network flow*, Management Sci., 5 (1958), pp. 97–101.

[10] Y. Freund and R. E. Schapire, *Adaptive game playing using multiplicative weights*, Games Econom. Behav., 29 (1999), pp. 79–103.

[11] N. Garg and J. Könemann, *Faster and simpler algorithms for multicommodity flow and other fractional packing problems*, SIAM J. Comput., 37 (2007), pp. 630–652.

[12] M. D. Grigoriadis and L. G. Khachiyan, *A sublinear-time randomized approximation algorithm for matrix games*, Oper. Res. Lett., 18 (1995), pp. 53–58.

[13] M. D. Grigoriadis and L. G. Khachiyan, *Coordination complexity of parallel price-directive decomposition*, Math. Oper. Res., (1996), pp. 321–340.

[14] M. D. Grigoriadis, L. G. Khachiyan, L. Porkolab, and J. Villavicencio, *Approximate max-min resource sharing for structured concave optimization*, SIAM J. Optim., 11 (2001), pp. 1081–1091.

[15] M. Grötschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms and Combinatorics, Springer, London, 2011.

[16] E. Hazan and R. Krauthgamer, *How Hard is It to Approximate the Best Nash Equilibrium?*, SIAM J. Comput., 40 (2011), pp. 79–91.

[17] M. Held and R. M. Karp, *The traveling salesman problem and minimum spanning trees*, Oper. Res., 18 (1971), pp. 1138–1162.

[18] M. Held and R. M. Karp, *The traveling salesman problem and minimum spanning trees: Part* II, Math. Programming, 1 (1971), pp. 6–25.

[19] G. Iyengar, D. J. Phillips, and C. Stein, *Approximating semidefinite packing programs*, SIAM J. Optim., 21 (2011), pp. 231–268.

[20] L. G. Khachiyan, *Convergence rate of the game processes for solving matrix games*, Zh. Vychisl. Mat. i Mat. Fiz., 17 (1977), pp. 1421–1431 (in Russian); U.S.S.R. Comput. Math and Math. Phys., 17 (1978), pp. 78–88 (in English).

[21] P. Klein, S. Plotkin, C. Stein, and É. Tardos, *Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts*, SIAM J. Comput., 23 (1994), pp. 466–487.

[22] P. Klein and N. E. Young, *On the number of iterations for Dantzig-Wolfe optimization and packing-covering approximation algorithms*, in Proceedings of the Seventh Conference on Integer Programming and Combinatorial Optimization (IPCO '99), Lecture Notes in Comput. Sci. 1610, Springer, Berlin, 1999, pp. 320–327.

[23] C. Koufogiannakis and N. E. Young, *A nearly linear-time PTAs for explicit fractional packing and covering linear programs*, Algorithmica, 70 (2014), pp. 648–674.

[24] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas, *Fast approximation algorithms for multicommodity flow problems*, J. Comput. Systems Sci., 50 (1995), pp. 228–243.

[25] R. J. Lipton, E. Markakis, and A. Mehta, *Playing large games using simple strategies*, in Proceedings of the 4th ACM conference on Electronic commerce, 2003, pp. 36–41.

[26] R. J. Lipton and N. E. Young, *Simple strategies for large zero-sum games with applications to complexity theory*, in Proceedings of the 26th Annual ACM Symposium on Theory of Computing, 1994, pp. 734–740.

[27] L. Lovasz, *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 50, SIAM, Philadelphia, 1986.

[28] A. Nemirovski, *Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim., 15 (2004), pp. 229–251.

[29] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[30] Y. Nesterov, *Fast gradient methods for network flow problems*, in Proceeding of the 20th International Symposium of Mathematical Programming, 2009.

[31] S. A. Plotkin, D. B. Shmoys, and É. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Math. Oper. Res., 20 (1995), pp. 257–301.

[32] F. Shahrokhi and D. W. Matula, *The maximum concurrent flow problem*, J. Assoc. Comput. Mach., 37 (1990), pp. 318–334.

[33] J. F. Shapiro, *A survey of Lagrangean techniques for discrete optimization*, Ann. Discrete Math., 5 (1979), pp. 113–138.

[34] M. J. Todd, *The many facets of linear programming*, Math. Program., 91 (2002), pp. 417–436.
[35] N. E. Young, *Randomized rounding without solving the linear program*, in Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1995, pp. 170–178.
[36] N. E. Young, *Sequential and parallel algorithms for mixed packing and covering*, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science 2001, pp. 538–546.
[37] N. E. Young, *Nearly Linear-Work Algorithms for Mixed Packing/Covering and Facility-Location Linear Programs*, preprint, arXiv:1407.3015, 2014.