Approximating Integer Programs with Positive Right-hand Sides

Peter Jonsson[1*] and Johan Thapper[2**]

[1] Department of Computer and Information Science, Linköpings universitet, SE-581 83 Linköping, Sweden
[2] École polytechnique, Laboratoire d'informatique (LIX), 91128 Palaiseau Cedex, France

Abstract. We study minimisation of integer linear programs with positive right-hand sides. We show that such programs can be approximated within the maximum absolute row sum of the constraint matrix A whenever the variables are allowed to take values in $\mathbb{N}$. This result is optimal under the unique games conjecture. When the variables are restricted to bounded domains, we show that finding a feasible solution is **NP**-hard in almost all cases.

1 Introduction

We study the approximability of minimising integer linear programs with positive right-hand sides. Let n and m be positive integers, representing the number of variables and the number of inequalities, respectively. Let $\mathbf{x}^T = (x_1, \ldots, x_n)$ be a vector of n variables, A be an integer $m \times n$ matrix, $\mathbf{b} \in (\mathbb{Z}^+)^m$, and $\mathbf{c} \in (\mathbb{Q}^+ \cup \{0\})^n$. Finally, let X be some given subset of $\mathbb{N}^n$. We consider here various restrictions of the following integer linear program:

$$\begin{array}{ll} \text{Minimise} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \in X. \end{array} \tag{IP}$$

Typically, X is either $\mathbb{N}^n$ or $\{\mathbf{x} \in \mathbb{Z}^n \mid \mathbf{0} \leq \mathbf{x} \leq \mathbf{d}\}$ for some $\mathbf{d} \in (\mathbb{Z}^+)^n$, where the inequalities are to hold componentwise. A commonly occurring instance of the latter case is when $X = \{0,1\}^n$, so-called 0-1 programming. In all but very restricted cases, (IP) is

* Partially supported by the *Center for Industrial Information Technology* (CENIIT) under grant 04.01 and by the *Swedish Research Council* (VR) under grant 2006-4532.
** Partially supported by the contract DGA-LIX.

NP-hard to solve to optimality. Instead, the effort is directed towards finding approximation algorithms and improving the bound within which it is possible to find approximate solutions. Formally, a minimisation problem Π is said to be *approximable within* (a real constant) $c \geq 1$ if there exists a polynomial time algorithm A such that for all instances x of Π, $A(x)/\textsc{Opt}(x) \leq c$.

Let $\mathbf{a_j}^T = (a_{j1}, \ldots, a_{jn}) \in \mathbb{Z}^n$ be the jth row of A. We will use the norm $\|\mathbf{a_j}\|_1 = \sum_{i=1}^{n} |a_{ji}|$ as well as the *maximum absolute row sum norm* of A, defined as $\|A\|_\infty = \max_{1 \leq j \leq m} \|\mathbf{a_j}\|_1$. Let $(\text{IP})_k$ denote the subset of (IP) where $||A||_\infty \leq k$. We show that $(\text{IP})_k$ can unconditionally be approximated within k when $X = \mathbb{N}^n$, but cannot be approximated within $k - \varepsilon$, $\varepsilon > 0$, if Khot's *unique games conjecture* holds [9]. We also show that finding a feasible solution to (IP) is **NP**-hard in almost all cases when $X = \{0, \ldots, a-1\}^n$.

1.1 Previous work

The approximability of the program (IP) has been extensively studied in the case when A is restricted to non-negative entries. In this case, the problem is usually referred to as a (generalised, or capacitated) *covering problem*. Among the problems described by such programs one finds the Minimum Knapsack Problem, Minimum Vertex Cover (and its k-uniform hypergraph counterpart, described below) and various *network design problems* [2]. We will refer to (IP) with non-negative A as (CIP) (covering integer program). Here, X is often taken to be $\{\mathbf{x} \in \mathbb{Z}^n \mid \mathbf{0} \leq \mathbf{x} \leq \mathbf{d}\}$. Indeed, optimal solutions remain feasible after introduction of the bounds $x_i \leq \lceil \max_j b_j/a_{ji} \rceil$.

Hall and Hochbaum [7] restrict A in (CIP) to a 0/1-matrix and give an $\|A\|_\infty$-approximating algorithm for the case when $X = \{0,1\}^n$. Bertsimas and Vohra [1] study the general (CIP) with $X = \{0,1\}^n$ as well as $X = \mathbb{N}^n$. They use both a randomised rounding heuristic with a nonlinear rounding function and deterministic rounding using information about the dual program. For $X = \{0,1\}^n$, they show that (CIP) can be approximated within $\|A\|_\infty$ using both a deterministic rounding function and a dual heuristic. For $X = \mathbb{N}^n$, they obtain an $\|A\|_\infty + 1$ approximating algorithm. Carr, Fleischer, Leung and Phillips [2] lower the integrality gap of (CIP) with $X = \{0,1\}^n$

by introducing additional inequalities into the program to obtain an approximation ratio equal to the maximal number of non-zero entries in a row of A. Their claim that the proof immediately generalises to the case when the variables are bounded by any fixed $d > 1$ seems to be incorrect, but a complete proof for general d is given by Pritchard [11]. Koufogiannakis and Young [10] present an approximation algorithm for a general framework of *monotone covering* problems, with an approximation ratio equal to the maximal number of variables upon which a constraint depends. The constraints must be monotone (closed upwards), but can be non-convex. This framework in particular includes problems such as (CIP) and MINIMUM SET COVER.

2 Unbounded domain

We assume that $X = \mathbb{N}^n$ throughout this section. Lower bounds for $(\mathrm{IP})_k$ are discussed in Section 2.1. We aim to prove the following result:

Proposition 1. *$(IP)_k$ can be approximated within k.*

The problem $(\mathrm{IP})_1$ is solvable in polynomial time: initially, let $x_i = 0$ for all i, and for each inequality $x_i \geq b$, update x_i to $\max\{x_i, b\}$. Any inequality of the form $-x_i \geq b$ implies that there are no solutions. In order to prove Proposition 1 for $k \geq 2$, we give a deterministic 'rounding'-scheme, which produces an integer solution from a rational one, while increasing the value of the objective function by at most k. For an integer $k \geq 2$ and $x \in \mathbb{Q}^+ \cup \{0\}$, define the following operation:

$$\hat{x} = \begin{cases} 0 & \text{if } 0 \leq x < 1/k \\ 1 & \text{if } 1/k \leq x < 2/k \\ \lceil (k-1)x \rceil & \text{otherwise.} \end{cases}$$

For a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, let $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)^T$. Note that $\mathbf{c}^T\hat{\mathbf{x}} \leq k \cdot \mathbf{c}^T\mathbf{x}$. We will show that in addition, $\hat{\mathbf{x}}$ satisfies $A\hat{\mathbf{x}} \geq \mathbf{b}$ by showing that for any integer $b \geq 1$, we have $\mathbf{a} \cdot \hat{\mathbf{x}} \geq b$ whenever $\mathbf{a} \cdot \mathbf{x} \geq b$ for any vector $\mathbf{a} = (a_1, \ldots, a_n)^T$ with $\|\mathbf{a}\|_1 \leq k$. In order to do this, we first introduce a scaling of $\hat{x}$ which will be easier to work with. Let $x' = \hat{x}/(k-1)$ and extend to vectors, $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_k)^T$, as before.

Our first step is to bound the difference $\Delta = \mathbf{a} \cdot \mathbf{x} - \mathbf{a} \cdot \mathbf{x}'$ from above. Let $\delta_i = a_i(x_i - x'_i)$ so that $\Delta = \sum_{i=1}^{n} \delta_i$. Let $t_i = \mathrm{sgn}(a_i) \cdot x_i$ and $t'_i = \mathrm{sgn}(a_i) \cdot x'_i$. Then, $\delta_i = |a_i|(t_i - t'_i)$. Figure 1 illustrates how the t'_i are determined from the t_i in the cases which give positive contributions to Δ. Each arrow represents an interval, and for a t_i in a particular interval, t'_i can be found at the arrow head. Note that there are only two such intervals on the positive axis. To the left of L_5 follows an infinite sequence of left arrows, each of size equal to that of L_5.

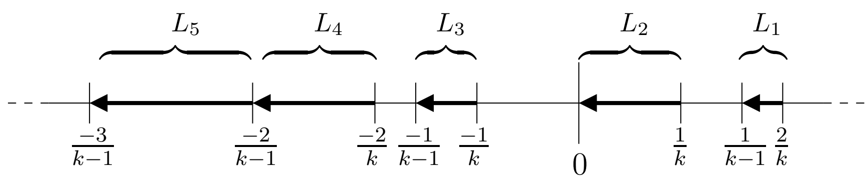

Fig. 1. The intervals $L_1, \ldots, L_5$ represented by arrows.

Formally, the intervals $L_i, i \geq 1$ are defined as follows:

$$
\begin{aligned}
L_1 &= \{x \in \mathbb{Q} \mid 1/(k-1) \leq x < 2/k\} \\
L_2 &= \{x \in \mathbb{Q} \mid 0 \leq x < 1/k\} \\
L_3 &= \{x \in \mathbb{Q} \mid -1/(k-1) < x \leq -1/k\} \\
L_4 &= \{x \in \mathbb{Q} \mid -2/(k-1) \leq x \leq -2/k\} \\
L_i &= \{x \in \mathbb{Q} \mid -(i-2)/(k-1) \leq x < -(i-3)/(k-1)\} (i \geq 5)
\end{aligned}
$$

When $k = 2$, the interval L_1 vanishes while L_3 and L_4 become adjacent. Let $L = \bigcup_{i \geq 1} L_i$. Now, δ_i can be bounded as follows, given the location of t_i:

$$
\begin{cases}
0 \leq \delta_i/|a_i| < (k-2)/k(k-1) & \text{if } t_i \in L_1 \\
0 \leq \delta_i/|a_i| < 1/k & \text{if } t_i \in L_2 \\
0 \leq \delta_i/|a_i| \leq 1/k(k-1) & \text{if } t_i \in L_3 \\
0 \leq \delta_i/|a_i| \leq 2/k(k-1) & \text{if } t_i \in L_4 \\
0 \leq \delta_i/|a_i| < 1/(k-1) & \text{if } t_i \in L_j, j \geq 5 \\
\delta_i \leq 0 & \text{if } t_i \notin L.
\end{cases}
$$

Note that when $k = 2$, the upper bound on $\delta_i/|a_i|$ for $t_i \in L_4$ is actually strict, since $-2/k$ is an integer. Thus, $\delta_i < |a_i|/(k-1)$, for all $i \geq 1$.

Lemma 1. *Let $b \geq 1$ and $k \geq 2$ be integers. If $\mathbf{a} \cdot \mathbf{x} \geq b$ and $\|\mathbf{a}\|_1 \leq k$, then $\Delta < 1$.*

Proof. Assume that there is an index l such that $t_l \notin L$. Then, $|a_l| > 0$ so $\sum_{i \neq l} |a_i| \leq k-1$. We then have

$$\Delta \leq \sum_{i \neq l} \delta_i < \sum_{i \neq l} \frac{|a_i|}{k-1} \leq \frac{k-1}{k-1} = 1. \tag{1}$$

Therefore, we can assume that for all $1 \leq i \leq n$ we have $t_i \in L$. Let $X_1 = \{i \mid t_i \in L_1\}$ and $X_2 = \{i \mid t_i \in L_2\}$. We will bound Δ by separately bounding the three parts of the sum with index sets X_1, X_2 and $\{1, \ldots, n\} \setminus (X_1 \cup X_2)$. Let $p = \sum_{i \in X_1} |a_i|$ and $q = \sum_{i \in X_2} |a_i|$. Since $t_i \geq 0$ if and only if $i \in X_1 \cup X_2$, we must have that $\sum_{i \in X_1 \cup X_2} |a_i| t_i \geq \mathbf{a} \cdot \mathbf{x} \geq b \geq 1$, hence $p \cdot 2/k + q \cdot 1/k > 1 \Leftrightarrow 2p + q > k$. Upper bounding the three parts yields:

$$\Delta < p \cdot \frac{k-2}{k(k-1)} + q \cdot \frac{1}{k} + (k-p-q) \cdot \frac{1}{k-1} = \frac{k^2 - 2p - q}{k(k-1)} < 1.$$

The lemma follows. $\square$

We can now use Lemma 1 to prove the following lemma, and complete the proof of Proposition 1.

Lemma 2. *Let $b \geq 1$ and $k \geq 2$ be integers. If $\mathbf{a} \cdot \mathbf{x} \geq b$ and $\|\mathbf{a}\|_1 \leq k$, then $\mathbf{a} \cdot \hat{\mathbf{x}} \geq b$.*

Proof. From Lemma 1, we have $\Delta = \mathbf{a} \cdot \mathbf{x} - \mathbf{a} \cdot \mathbf{x}' < 1$ which can be rearranged to $\mathbf{a} \cdot \mathbf{x}' > \mathbf{a} \cdot \mathbf{x} - 1 \geq b - 1$. Multiplication by $k-1$ now yields

$$\mathbf{a} \cdot \hat{\mathbf{x}} = (k-1)\mathbf{a} \cdot \mathbf{x}' > (k-1)(b-1).$$

When $k \geq 2$ and $b \geq 1$, we have $(k-1)(b-1) \geq b-1$, i.e., $\mathbf{a} \cdot \hat{\mathbf{x}} > b-1$. By integrality, it follows that $\mathbf{a} \cdot \hat{\mathbf{x}}$ must in fact be greater than, or equal to b. $\square$

Proof (Proposition 1). It remains to prove the statement for $k \geq 2$. For an instance of $(\mathrm{IP})_k$, we solve the LP-relaxation and obtain a solution $\mathbf{x}$ such that $\mathbf{Ax} \geq \mathbf{b}$, and $\mathbf{c}^T\mathbf{x}$ is less than or equal to the optimum of the corresponding integer program. In particular, $\mathbf{a} \cdot \mathbf{x} \geq b$ for every row vector $\mathbf{a}$ of $\mathbf{A}$. By Lemma 2, it follows that $\mathbf{a} \cdot \hat{\mathbf{x}} \geq b$, and therefore $\mathbf{A}\hat{\mathbf{x}} \geq \mathbf{b}$. The value of the objective function for the solution $\hat{\mathbf{x}}$ is $\mathbf{c}^T\hat{\mathbf{x}} \leq k \cdot \mathbf{c}^T\mathbf{x}$, hence we have approximated $(\mathrm{IP})_k$ within k. $\square$

2.1 Lower bounds

A k-uniform hypergraph H is a pair (V, E), where V is a set of vertices and each hyperedge $e \in E$ is a k-element subset of V. The Ek-Vertex-Cover problem is that of finding a minimum size vertex cover in a k-uniform hypergraph. Note that E2-Vertex-Cover is identical to the well-known Minimum Vertex Cover for ordinary graphs. Given H, there is an immediate reduction to a (CIP) with one variable x_i for each vertex $v_i \in V$ and one inequality of the form

$$x_{j_1} + x_{j_2} + \ldots + x_{j_k} \geq 1$$

for each hyperedge $e_j = \{v_{j_1}, v_{j_2}, \ldots, v_{j_k}\} \in E$. Here, the domain X may be any superset of $\{0,1\}^n$ since we can always obtain a feasible solution $\mathbf{x}' = \min\{\mathbf{1}, \mathbf{x}\}$ (componentwise) with at least as small measure. Consequently, if Ek-Vertex-Cover is not approximable within a constant α, then $(\mathrm{IP})_k$ is not approximable within α either.

The best lower bounds currently known under the assumption of $\mathbf{P} \neq \mathbf{NP}$ is 1.3606 for Minimum Vertex Cover by Dinur and Safra [4] and $k - 1 - \varepsilon$ for Ek-Vertex-Cover by Dinur et al. [3]. Stronger bounds are obtainable by exploiting stronger complexity theoretical assumptions such as Khot's unique games conjecture: Khot and Regev [9] show that modulo the truth of this conjecture, Ek-Vertex-Cover cannot be approximated within $k - \varepsilon$ for any $\varepsilon > 0$. Thus, we have good reasons to believe that it may in fact be **NP**-hard to approximate Ek-Vertex-Cover within $k - \varepsilon$ for any ε. Combined with Proposition 1, this bound yields a (conjectured) tight approximation constant of k of $(\mathrm{IP})_k$. This bound also matches the upper bound on (CIP) by Carr et al. [2].

3 Bounded domain

Let $X = \{0, 1, \ldots, a-1\}$, $a \geq 2$ and $k \geq 3$, with at least one of the inequalities strict. In this case, we show that it is **NP**-hard to find any feasible solution to $(\mathrm{IP})_k$. The exceptional case $a = 2$ and $k = 3$ turns out to be approximable within 3.

Proposition 2. *Let $a \geq 2$ and $X = \{0, 1, \ldots, a-1\}^n$. When $k \geq 4$, it is an* **NP***-hard problem to find feasible solutions to $(IP)_k$.*

Proof. We reduce from the problem One-in-three SAT, which has been shown to be **NP**-hard by Schaefer [12]. An instance of One-in-three SAT is given by a set of clauses $\{C_1, \ldots, C_m\}$ over variables $\mathcal{U} = \{u_1, \ldots, u_n\}$, where each clause is a disjunction $u_i \vee u_j \vee u_l$ of exactly three variables. The question is whether or not there exists a satisfying assignment such that precisely one variable in each clause is assigned the value 1. Note that we do not allow negations of the variables. This is in agreement with Schaefer's original formulation.

For each propositional variable u_i occurring in a One-in-three SAT instance, create a variable x_i, and for each clause $C = u_i \vee u_j \vee u_l$, add the following inequalities:

$$\begin{aligned} x_i,\, x_j,\, x_l &\geq a-2 \\ x_i + x_j + x_l &\geq 3(a-2)+1 \end{aligned} \tag{2}$$

The first equation restricts the variables x_i, x_j, x_l to the set $\{a-2, a-1\}$. The second equation ensures that at least one of the variables x_i, x_j and x_l is assigned the value $a-1$. Furthermore, we add a (unique) new variable y, and the following inequalities

$$\begin{aligned} y &\geq a-1 \\ x_i + x_j &\leq 2y - 1 \Leftrightarrow 2y - x_i - x_j \geq 1 \\ x_i + x_l &\leq 2y - 1 \\ x_j + x_l &\leq 2y - 1 \end{aligned} \tag{3}$$

Since y must be $a-1$, the last three inequalities, together with the fact that $x_i, x_j, x_l \in \{a-2, a-1\}$ implies that at most one variable from $\{x_i, x_j, x_l\}$ takes the value $a-1$. We can now solve the original One-in-three SAT-instance by assigning $u_i = 0$ if $x_i = a-2$ and $u_i = 1$ if $x_i = a-1$. It follows that finding a solution to $(\mathrm{IP})_k$ is **NP**-hard. □

Corollary 1. *If $X = \{0, 1, \ldots, a-1\}^n$ with $a > 2$, then the problem of finding a feasible solution to (IP)$_3$ is* **NP***-hard.*

Proof. The proof of Proposition 2 can be altered in the following way to produce the result in the corollary. First, replace the equations (2) with $x_i + x_j + x_l \geq 1$. Then, replace $2y - 1$ in equations (3) with $y - (a - 2)$. Finally, let $u_i = 0$ if and only if $x_i = 0$. □

We are left with one remaining case:

Proposition 3. *The problem (IP)$_3$ with domain $X = \{0, 1\}$ can be approximated within 3.*

Proof. Let $\mathbf{x}$ be an optimal solution to the LP-relaxation of IP$_3$. We round $\mathbf{x}$ to an integer solution $\hat{\mathbf{x}}$ as follows:

$$\hat{x}_i = \begin{cases} 0 & \text{if } x_i < 1/3, \\ 1 & \text{otherwise.} \end{cases}$$

This increases the value of the objective function by at most 3. The proof follows a similar strategy as that of Proposition 1. Define $t_i = \mathrm{sgn}(a_i) \cdot x_i$ and $\hat{t}_i = \mathrm{sgn}(a_i) \cdot \hat{x}_i$. We note again that it suffices to show that

$$\Delta = \mathbf{a} \cdot \mathbf{x} - \mathbf{a} \cdot \hat{\mathbf{x}} = \sum_{i=1}^{n} |a_i|(t_i - \hat{t}_i) < 1,$$

since we then have $\mathbf{a} \cdot \hat{\mathbf{x}} > \mathbf{a} \cdot \mathbf{x} - 1 \geq b - 1$, and the result follows from the integrality of $\mathbf{a} \cdot \hat{\mathbf{x}}$.

Let $X_1 = \{i \mid -1 \leq t_i \leq -1/3\}$ and $X_2 = \{i \mid -1/3 < t_i < 1/3\}$, and let $p = \sum_{i \in X_1} |a_i|$ and $q = \sum_{i \in X_2} |a_i|$. The values of p and q must satisfy

$$b \leq \mathbf{a} \cdot \mathbf{x} \leq p \cdot (-1/3) + q \cdot 1/3 + (3 - p - q) \cdot 1 \tag{4}$$

which implies $p \cdot 2/3 + q \cdot 1/3 \leq 1$, where we have used 1 as a lower bound for b. In fact, we have $p \cdot 2/3 + q \cdot 1/3 < 1$: if $q = 0$, this follows immediately from the non-strict inequality, and if $q > 0$, then the term $q \cdot 1/3$ in (4) is a strict upper bound on $\sum_{i \in X_2} t_i$.

To finish the proof, note that $\{i \mid t_i - \hat{t}_i > 0\} \subseteq X_1 \cup X_2$. We therefore have the bound

$$\Delta \leq \sum_{i \in X_1 \cup X_2} |a_i|(t_i - \hat{t}_i) \leq p \cdot 2/3 + q \cdot 1/3 < 1,$$

and the proposition follows. □

4 Discussion and Future Work

We have obtained a tight approximation of a general class of integer linear programs under the parameterisation $\|A\|_\infty \leq k$. It is however important to note, that the result in Section 2 is tight only with respect to this particular parameterisation. It is still imaginable that there exists an approximation algorithm which approximates (IP) within, for example, the maximum number of non-zero entries in any row of A as was the case for the (CIP) problems. The approach of Carr et al. [2] and Pritchard [11] is based on adding so-called Knapsack Cover (KC) inequalities to the program, which lowers the integrality gap. The exponentially many inequalities are then handled using a separation oracle. We note that for (IP), there seems to be no natural counterpart to the KC-inequalities. The main obstacle is that the validity of the inequalities of (IP) does not uniformly improve upon increasing individual variables, as is the case for ordinary covering problems.

A first step in this direction could be to look for a 2-approximation of (IP) with at most two variables per inequality (or, indeed, prove that such an algorithm is unlikely to exist.) An algorithm *is* known for arbitrary right-hand sides when the variables are bounded, see Hochbaum et al. [8]. The idea behind the proof is to reduce the problem to program with only *monotone* inequalities ($ax - by \geq c$, where $a, b > 0$.) This system can then be solved in pseudo-polynomial time, depending on the upper bounds of the variables. The value of the final solution can then easily be seen to be off by at most a factor of 2. To use a similar approach, one would like to prove that polynomial time solvability is retained for monotone inequalities, when arbitrary right-hand sides and bounded domain is substituted with positive right-hand sides and unbounded domain. We note that this can be seen as a *constraint satisfaction problem* over an infinite domain, and that the constraint language of monotone inequalities is invariant under the operations min and max.

Dobson [5] and Fisher and Wolsey [6] both analyse greedy algorithms for (CIP) and derive bounds of $\mathcal{O}(\log d)$, where d is the maximum *column sum* of A. As for the KC-inequalities, the correctness of these algorithms crucially uses the non-negativity of the A-matrix, and a direct generalisation to (IP) fails. Nevertheless, it seems reasonable to assume that some kind of column-sum bound for (IP) should exist.

References

1. D. Bertsimas and R. Vohra. Rounding algorithms for covering problems. *Mathematical Programming*, 80(1):63–89, 1998.
2. R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms (SODA-2000)*, pages 106–115, 2000.
3. I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM Journal on Computing*, 34(5):1129–1146, 2005.
4. I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.
5. G. Dobson. Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Mathematics of Operations Research*, 7(4):515–531, 1982.
6. M. L. Fisher and L. A. Wolsey. On the greedy heuristic for continuous covering and packing problems. *SIAM Journal on Algebraic Discrete Methods*, 3(4):584–591, 1982.
7. N. G. Hall and D. S. Hochbaum. A fast approximation algorithm for the multicovering problem. *Discrete Applied Mathematics*, 15:35–40, 1986.
8. D. S. Hochbaum, N. Megiddo, J. Naor, and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, 62:69–83, 1993.
9. S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
10. C. Koufogiannakis and N. E. Young. Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular cost. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP-2009)*, pages 634–652, 2009.
11. D. Pritchard. Approximability of sparse integer programs. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA-2009)*, pages 83–94, 2009.
12. T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing (STOC-1978)*, pages 216–226, 1978.