ON THE COMPLEXITY OF SOLVING THE GENERALIZED SET PACKING PROBLEM APPROXIMATELY

Nimrod Megiddo†

Abstract. The generalized set packing problem (GSP) is as follows. Given a family F of subsets of $M = \{1, \cdots, m\}$ and a vector $b \in R^m$, find a subfamily $F' \subset F$ of maximum cardinality such that for every $i \in M$, i does not belong to more than b_i members of F'. The subproblem (GSP_k) consists of those instances of (GSP) where each member of M belongs to at most k members of F. It is shown that for every k, if there is a polynomial-time approximation algorithm for (GSP_k) with a positive performance ratio then GSP_k has a polynomial approximation scheme. This generalizes a result of Garey and Johnson with regard to the maximum independent set problem in a graph.

† IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

The generalized set packing problem is the following integer linear programming problem:

$$(GSP) \qquad \begin{aligned} \text{Maximize} \quad & e^T x \\ \text{subject to} \quad & Ax \le b \\ & x \in \{0,1\}^n \end{aligned}$$

where $A \in \{0,1\}^{m \times n}$ is a zero-one matrix, $b \in Z^m$ is an integral vector, and e denotes a vetor of 1's. For an application of the problem, consider a scheduling problem where jobs require processing during certain time intervals. Thus, A_{ij} equals 1 if and only if job j requires processing during the i-th period. The value of b_i gives the number of jobs that can run in parallel during the i-th period. The problem is to select a set of jobs of maximum cardinality which can be processed subject to the constraints.

The definitions in the present paragraph are taken from [GJ]. Given an instance I of (GSP), denote by $opt(I)$ the value of its optimal solution. If F is an approximation algorithm for (GSP), let $F(I)$ denote the value of the solution delivered by A, and let $r(F) = \inf_I\{F(I)/opt(I)\}$. The ratio $r(F)$ is the *worst-case performance ratio* of the algorithm F. An *approximation scheme* is an algorithm $\overline{F}$ that receives together with the problem another parameter $\epsilon > 0$. Thus, for a fixed ϵ there is an implicit algorithm F_ϵ that is derived from $\overline{F}$. If for every $\epsilon > 0$ the derived algorithm F_ϵ runs in polynomial time and has a performance ratio $r(F_\epsilon) \ge 1 - \epsilon$ then $\overline{F}$ is said to be a *polynomial approximation scheme* [GJ].

The maximum independent set problem is a special case of (GSP) where A has exactly two 1's in each row (i.e., A is the node-arc incidence matrix of some graph), and $b = e$. Garey and Johnson [GJ] proved the following:

Proposition 1. *If the maximum independent set problem has a polynomial-time approximation algorithm with a positive performance ratio then it also has a polynomial approximation scheme.*

We show that Proposition 1 can be extended to a parameterized family of subproblems of the generalized set packing problem. For every integer $k \ge 2$, let (GSP_k) denote the set of instances of (GSP) where the matrix A has at most k 1's per row. A generalization of Proposition 1 is as follows.

Proposition 2. *For every $k \geq 2$, if there is a polynomial-time approximation algorithm for (GSP_k) with a positive performance ratio then there is a polynomial approximation scheme for (GSP_k).*

Proof: The proof generalizes Garey and Johnson's proof of Proposition 1. Consider an instance of (GSP_k), given by a pair (A, b) $(A \in \{0,1\}^{m\times n},\ b \in Z^m)$. We produce from (A, b) another instance $(\overline{A}, \overline{b})$, $\overline{A} \in \{0,1\}^{\overline{m}\times\overline{n}}$, $\overline{b} \in Z^{\overline{m}}$ (where $\overline{m} \leq mn(n^{k-1}+1)$ and $\overline{n} = n^2$) as follows (see an example below). The rows of $\overline{A}$ are of two types. The first mn rows are of the first type and are obtained by placing n copies of the matrix A in a diagonal configuration. Also, the first mn entries of $\overline{b}$ contain n copies of b. The other rows are of the second type and produced as follows. From each row i of A with k_i 1's $(k_i \leq k)$ we produce n^{k_i} rows of $\overline{A}$. Specifically, a typical row is produced from the i'th row of A in the following way. For each j such that $A_{ij} = 1$, pick a number ℓ_j, $n(j-1)+1 \leq \ell_j \leq nj$. Given the choices ℓ_j, create a row of $\overline{A}$ by placing 1's in the k_i positions of the form ℓ_j and 0's in all the other positions. Obviously, for each i the number of different ways to choose the ℓ_j's is n^{k_i}. The corresponding entry of $\overline{b}$ is set to b_i. Consider the generalized set packing problem with $(\overline{A}, \overline{b})$. The number of variables is n^2 and it is convenient to use here doubly indexed ones z_{jh}, defined by $z_{jh} \equiv x_{n(j-1)+h}$. Under this equivalence, the symbols $\overline{A}z$ and $e^T z$ are well-defined. Consider any zero-one solution $z = (z_{jh})$ of the system $\overline{A}z \leq \overline{b}$. Denote $x^{(j)} = (z_{j1}, \cdots, z_{jn})^T$ $(j = 1, \cdots, n)$. It is easy to see that $x^{(j)}$ is a solution of the system $Ax \leq b$. Now, denote $y_j = \prod_h z_{jh}$ $(j = 1, \cdots, n)$. We claim that $y = (y_1, \cdots, y_n)^T$ is a zero-one solution of $Ax \leq b$. For if $A_i y > b_i$ for some row i then by choosing ℓ_j (for each j such that $A_{ij} = 1$) so that $y_j = z_{j\ell_j}$, we identify a row i' of $\overline{A}$ for which $\overline{A}_{i'}z > \overline{b}_{i'}$. Moreover,

$$e^T z = \sum_j e^T x^j = \sum_{j:y_j=1} e^T x^j \leq (e^T y)\max_j\{e^T x^j\} \leq \max\left\{(e^T y)^2, \max_j\{(e^T x^j)^2\}\right\} .$$

On the other hand, if $y = x^{(j)} = x'$ $(j = 1, \cdots, n)$ then $e^T z = (e^T x')^2$. Thus, if the optimal value of the given problem is equal to V then the optimal value of the problem with $(\overline{A}, \overline{b})$ is equal to V^2. Now, suppose there exists a polynomial-time approximation algorithm F whose performance ratio is $r > 0$. Let z denote the solution given by F for the system with $(\overline{A}, \overline{b})$. Thus, $(e^T z)/V^2 > r$. Let x^* denote one of the vectors $y, x^{(1)}, \cdots, x^{(n)}$ so that $e^T x^*$ is maximal. It follows from the above that $(e^T x^*)/V \geq \sqrt{r}$. This means that there is also a polynomial-time approximation algorithm with a performance ratio of $\sqrt{r}$. Repeating this arguement sufficiently many times, we see that for every $\epsilon > 0$ a ratio of $1 - \epsilon$ can be guaranteed. ∎

Example 3. Consider the pair

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 1 \end{bmatrix} .$$

Here the pair $(\overline{A}, \overline{b})$ is the following:

$$
\overline{A} = \begin{bmatrix}
1 & 1 & 1 & & & & & & \\
1 & 1 & 0 & & & & & & \\
& & & 1 & 1 & 1 & & & \\
& & & 1 & 1 & 0 & & & \\
& & & & & & 1 & 1 & 1 \\
& & & & & & 1 & 1 & 0 \\
1 & & & 1 & & & 1 & & \\
1 & & & 1 & & & & 1 & \\
1 & & & 1 & & & & & 1 \\
1 & & & & 1 & & 1 & & \\
1 & & & & 1 & & & 1 & \\
1 & & & & 1 & & & & 1 \\
1 & & & & & 1 & 1 & & \\
1 & & & & & 1 & & 1 & \\
1 & & & & & 1 & & & 1 \\
& 1 & & 1 & & & 1 & & \\
& 1 & & 1 & & & & 1 & \\
& 1 & & 1 & & & & & 1 \\
& 1 & & & 1 & & 1 & & \\
& 1 & & & 1 & & & 1 & \\
& 1 & & & 1 & & & & 1 \\
& 1 & & & & 1 & 1 & & \\
& 1 & & & & 1 & & 1 & \\
& 1 & & & & 1 & & & 1 \\
& & 1 & 1 & & & 1 & & \\
& & 1 & 1 & & & & 1 & \\
& & 1 & 1 & & & & & 1 \\
& & 1 & & 1 & & 1 & & \\
& & 1 & & 1 & & & 1 & \\
& & 1 & & 1 & & & & 1 \\
& & 1 & & & 1 & 1 & & \\
& & 1 & & & 1 & & 1 & \\
& & 1 & & & 1 & & & 1 \\
1 & & & 1 & & & & & \\
1 & & & & 1 & & & & \\
1 & & & & & 1 & & & \\
& 1 & & 1 & & & & & \\
& 1 & & & 1 & & & & \\
& 1 & & & & 1 & & & \\
& & 1 & 1 & & & & & \\
& & 1 & & 1 & & & & \\
& & 1 & & & 1 & & &
\end{bmatrix}
\qquad
\overline{b} = \begin{bmatrix}
2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1
\end{bmatrix} .
$$

Reference

[GJ] M. R. Garey and D. S. Johnson, *Computers and intractibility: A Guide to the theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.