

The Parallel Approximability of a Subclass of Quadratic Programming*

Maria Serna

Departament de LSI, UPC
Univ. Polytec. de Catalunya
Campus Nord, C6
Jordi Girona Salgado, 1-3
08034-Barcelona, Spain
E-mail: mjserna@lsi.upc.es

Fatos Xhafa

Departament de LSI, UPC
Univ. Polytec. de Catalunya
Campus Nord, C6
Jordi Girona Salgado, 1-3
08034-Barcelona, Spain
E-mail: fatos@lsi.upc.es

Abstract

In this paper we deal with the parallel approximability of a special class of Quadratic Programming (QP), called Smooth Positive Quadratic Programming. This subclass of QP is obtained by imposing restrictions on the coefficients of the QP instance. The Smoothness condition restricts the magnitudes of the coefficients while the positiveness requires that all the coefficients be non-negative. Interestingly, even with these restrictions several combinatorial problems can be modeled by Smooth QP. We show NC Approximation Schemes for the instances of Smooth Positive QP. This is done by reducing the instance of QP to an instance of Positive Linear Programming, finding in NC an approximate fractional solution to the obtained program, and then rounding the fractional solution to an integer approximate solution for the original problem. Then we show how to extend the result for positive instances of bounded degree to Smooth Integer Programming problems. Finally, we formulate several important combinatorial problems as Positive Quadratic Programs (or Positive Integer Programs) in packing/covering form and show that the techniques presented can be used to obtain NC Approximation Schemes for "dense" instances of such problems.

1 Introduction

Quadratic Programming represents a special class of Non-Linear Programming in which the objective function is quadratic and the constraints are linear. Quadratic Programs arise in a wide variety of applications including scheduling and allocation problems,

regression analysis in statistics, economic models of optimal sales revenues, etc. The sequential complexity of QP is well understood. QP, in its general form, is NP-complete. Sahni showed that QP is NP-hard [18] and later on Vavasis proved that QP is in the class NP [21].

In this paper we study the parallel approximability of a special class of QP. Recall, the class NC consists of problems that can be solved by a PRAM algorithm whose running time is polylogarithmic in instance size while using a polynomial number of processors. Thus a problem that is shown to be P-complete unlikely will belong to NC [5]. For those hard problems it is interesting to analyze whether we can get an approximate algorithm in NC. Problems easy to approximate belong to NCAS, the class of problems that have an NC Approximation Scheme, i.e., a family of NC algorithms $\{A_\epsilon\}_{\epsilon \geq 0}$, such that A_ϵ has approximation ratio $1 + \epsilon$, and whose running time depends arbitrarily on ϵ . For a discussion of non-approximable problems see, e.g., [4]. Usually the last behavior is proved by showing that adequate approximation problems are P-complete. For example, *Convex Quadratic Programming (CQP)* that is, Quadratic Programming where the objective function is convex is non-approximable in NC. Indeed, CQP is in P (it is solved in polynomial time through the ellipsoid method [6] or the interior point algorithm [8]), and we can easily notice that CQP is non-approximable in NC by making use of the fact that Approximating Linear Programming is P-complete [19]. In fact, we can state non-approximability results also for a couple of problems related to CQP. *Solution Approximation* consists in finding a feasible solution x to CQP whose norm is close to that of an optimal solution x^* . *Value Approx-*

*This research was partially supported by the ESPRIT Long Term Research Project No. 20244, ALCOM IT and CICT TIC97-1475-CE.

imation is to find a vector x such that the value of the objective function on it is close to the optimal value, without the condition that x be feasible. Considering all possible values for the error parameter ε , we get two families of problems, one for each type of approximation. Both these families can be easily shown P-complete, for any value of ε , thus proving that CQP is hard to approximate.

Our main attention here is on another special class of QP, called *Smooth Quadratic Programming*, in which we impose restrictions on the magnitudes of all the coefficients appearing in the instance. We show that, given an instance of Smooth QP, if it is, in addition, *positive*, i.e., all the coefficients are non-negative, then we can find an NC Approximation Scheme for the problem restricted to such instances. Recall that Luby and Nisan [11] gave a parallel approximation scheme for Positive Linear Programming (Positive LP) – the special case of LP in packing/covering form with all the coefficients of the instance being non-negative. We obtain the result by reducing the instance of QP to an instance of Positive LP, finding in NC an approximate fractional solution to the resulting program, and then rounding the fractional solution to an integer approximate solution for the original problem. The reduction from QP to LP uses the Random Sampling technique, and we show how to perform it deterministically in NC by using random walks in constant degree expanders. The rounding of the fractional solution involves the Randomized Rounding method of [17], and we show that the integer solution can be found also deterministically in NC. Furthermore, we prove that the NC Approximation Scheme for Smooth QP can be used to obtain NC Approximation Schemes for bounded degree *Smooth Integer Programs*.

We should point out, however, that the “positiveness” and “smoothness” conditions are restrictive, in the sense that it is difficult to find natural problems whose QP is at the same time positive and smooth. Fortunately, even with these conditions we are able to cast in our model several important combinatorial problems. These are MAX CUT, MAX DICUT, MAX 2SAT and MAX k SAT. First, we formulate all of these problems as Positive QP or Positive Integer Programs in the packing/covering form, and then we derive NC Approximation Schemes for their “dense” instances. Notice that these programs are different from those given in [3], in that the last ones are neither positive nor in packing/covering form. Interestingly, for the problems on graphs, imposing in their QP programs the smoothness condition is equivalent with that of the graph instance being dense. Our results for the prob-

lems mentioned above match those of the sequential case given in [3].

2 Definitions

In this section we give formal definitions of the problems used in the paper. Let Q be an $n \times n$ matrix, A an $m \times n$ matrix, $a \in \mathbb{R}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

Linear Programming problem (LP) is

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax \geq b, \quad x \geq 0 \end{aligned}$$

Positive LP in packing form is $\max_x \{c^T x : Ax \leq b; x \geq 0\}$, and in covering form $\min_y \{by : A^T y \geq c; y \geq 0\}$, where A , b and c have non-negative entries.

Quadratic Programming problem is

$$\begin{aligned} &\text{minimize} && 1/2 x^T Q x + c^T x + a \\ &\text{subject to} && Ax \geq b \end{aligned}$$

Smooth Integer Programming [3]. Let $x = (x_i)_{i=1}^n$. A Smooth degree- l Integer Program has the form

$$\begin{aligned} &\text{minimize} && p(x_1, \dots, x_n) \\ &\text{subject to} && Wx \geq d \\ &&& x_i \in \{0, 1\}, \quad 1 \leq i \leq n, \end{aligned} \tag{1}$$

where $p(x_1, \dots, x_n)$ is a degree- l polynomial in which the coefficient of the degree- i term is either 0 or $\Theta(n^{l-i})$.

3 Approximating Smooth Quadratic Programming

In this section we present our main result. We consider a restricted version of QP that we call Smooth Quadratic Programming (Smooth QP). Our definition for Smooth QP comes from the more general definition of [3] for Smooth Integer Programming.

Smooth Quadratic Programming (Smooth QP) is

$$\begin{aligned} &\text{minimize} && g(x) = \sum a_{ij} x_i x_j + \sum b_i x_i + a \\ &\text{subject to} && Wx \geq d \\ &&& x_i \in \{0, 1\}, \quad 1 \leq i \leq n, \end{aligned} \tag{2}$$

where a_{ij} , are $O(1)$, b_i are 0 or $\Theta(n)$, and a is 0 or $\Theta(n^2)$ and, finally, w_{ij} are $O(1)$ and such that for any $i = 1, \dots, m$, $\sum_{j=1}^n w_{ij} - d_i = \Omega(\log n)$.

Here is some intuition behind this subclass of QP. The condition on the magnitudes of the coefficients of the objective function assures that the optimal value is $O(n^2)$ and therefore while obtaining an $(1 + \varepsilon)$ -approximation of the optimum value, a large error (i.e. of order εn^2) is allowable. On the other hand, the restrictions on w_{ij} and d_i 's is to assure that after

rounding the fractional values of variables x_i to 0/1 values y_i , the integer solution y will satisfy $Wy \geq d$ with positive probability. This explains, intuitively, why the approximation of Smooth QP is (expected to be) more easy.

We show that, given an instance of Smooth QP, we can approximate it in NC if it is, in addition, positive. An instance of Smooth QP is called positive if all the coefficients of the instance are non-negative. We will show that for Positive Smooth QP we can achieve in NC an $(1 + \varepsilon)$ -approximation for any (constant) value of ε , thus the problem has an NC Approximation Scheme. As we will see this result can be easily extended to the general case, that is for positive instances of Smooth degree- l Integer Programming. To prove the result, we will make use of the following known results. The first one is a well known technique on how to estimate the sum of n numbers by random sampling (see, e.g., [15]).

Lemma 1 (Sampling Lemma)

Let $\{a_i\}_{i=1}^n$ be a set of n numbers, where each a_i is $O(1)$. Let $p = \sum_{i=1}^n a_i$ be their sum. If we pick uniformly at random a subset of $s = O(\log n/\varepsilon^2)$ of a_i 's and compute their sum q , then with high probability, i.e., with probability at least $1 - O(1/n)$, we have that $p - \varepsilon n \leq qn/s \leq p + \varepsilon n$.

The second result by Raghavan and Thompson [17] shows how to round a fractional solution of a linear program $\max\{\sum_{i=1}^n x_i : Ax \leq \mathbf{r}\}$ to an integer solution of the same linear program, known as **Randomized Rounding with Scaling**. Given such a linear program, let \tilde{v}^* be its optimal fractional value and $v' = \tilde{v}^*(1 - \delta)$, for a small $\delta > 0$.

Theorem 1 (Theorem 4.3 in [17])

Let $\delta_1 > 0$, $\delta_2 > 0$ be such that $\delta_1 + \delta_2 < 1$. If there is a constant $\delta \in (0, 1/2]$ such that

$$\sum_{i=1}^n \exp\left[-\frac{\delta^2 r_i}{(3(1-\delta))}\right] < \delta_2 \tag{3}$$

then there exists an integer solution to $Ax \leq \mathbf{r}$ with objective function value at least $v' - \sqrt{2v' \ln(1/\delta_1)}$.

Notice that the above theorem does not apply to general linear programs but only when restricted to those which satisfy condition (3). We will discuss on this point later but let us just mention that such a condition is to assure that after rounding the fractional values of variables x_i to 0/1 values y_i , y will satisfy $Ax \leq \mathbf{r}$ with high probability. Notice that, in particular, when $r_i = \Omega(\log n)$ the condition (3) holds.

We prove the following theorem.

Theorem 2 Given a positive instance of Smooth QP and a fixed ε , we can find in NC a 0,1 assignment to variables x_i such that x satisfies $Wx \geq d$, and

$$g(x_1, x_2, \dots, x_n) \leq g(x^*) + \varepsilon n^2,$$

where $g(x)$ is the objective function and $g(x^*)$ is its optimal value.

To prove the theorem, we first observe the following. Let us write (2) equivalently as¹:

$$\begin{aligned} &\text{minimize} && c \\ &\text{subject to} && x^T Ax + bx \leq c \\ &&& Wx \geq d \\ &&& x_i \in \{0, 1\}, \quad 1 \leq i \leq n. \end{aligned} \tag{4}$$

Therefore, for Theorem 2, it suffices to prove the following theorem.

Theorem 3 Suppose there is a 0,1 solution to the following Positive Quadratic System

$$\begin{aligned} &x^T Ax + bx \leq c \\ &Wx \geq d, \end{aligned} \tag{5}$$

where a_{ij} , w_{ij} are $O(1)$, $b_i = \Theta(n)$ and c is a constant. Then, for any fixed ε , we can find in NC an assignment of 0,1 values to x_i such that x satisfies $Wx \geq d$, and

$$x^T Ax + bx \leq c + \varepsilon n^2 \tag{6}$$

The proof follows the ideas of a theorem from [3]. Here we observe that under our conditions, the 0,1 solution can be found also in NC. The idea is to reduce the QP instance to an instance of LP. Note that the resulting instance will be positive and thus an algorithm for Positive Linear Programming [11] is applied to find a fractional solution in NC. Then, we round the fractional solution to a 0,1 solution using the Randomized Rounding. In contrast to the reduction of the sequential case, our reduction involves further modifications to the instance of LP in order to enable the Luby and Nisan algorithm, which works only for Positive LP in packing/covering form.

Two sources of randomness appear in this scheme. First, the reduction we will describe in the next section uses randomization, so we have to show how to achieve it in NC. And, secondly, randomness appears when we apply the Randomized Rounding.

¹The constant a in the objective function is negligible.

3.1 Reducing Positive Smooth QP to LP

The reduction is as follows. Let $x^\#$ be a feasible solution of (5), as supposed, and let us write $r^\# = x^\#A + b$. Since $x^T Ax + bx = (x^T A + b)x$, we can express (5) as

$$\begin{aligned} x^T A + b &\leq r^\# \\ r^\# x &\leq c \\ Wx &\geq d \\ 0 \leq x_i &\leq 1 \end{aligned} \tag{LP1}$$

for which $x^\#$ is also a feasible solution. The following observation is immediate.

Proposition 1 *If x is a feasible solution to (LP1) then x is also feasible for the system (5).*

Clearly, any coefficient in (LP1) is non-negative. However, (LP1) is not in the packing/covering form because in it we have both types of restrictions. To overcome this, we modify (LP1) appropriately by taking $z = \mathbf{1} - x$ (i.e., $z_i = 1 - x_i$), where $\mathbf{1}$ is the n -vector of all ones. Thus, (LP1) is written as

$$\begin{aligned} x^T A + b &\leq r^\# \\ r^\# x &\leq c \\ Wz &\leq W \cdot \mathbf{1} - d \\ x_i + z_i &= 1 \\ 0 \leq x_i, z_i &\leq 1, \quad 1 \leq i \leq n. \end{aligned} \tag{LP2}$$

We can suppose, without loss of generality, that $W \cdot \mathbf{1} - d \geq 0$ because otherwise the system $Wx \geq d$ would not have a solution. So, the above program (LP2) is still positive. The relation between (LP1) and (LP2) is given as follows.

Proposition 2 *The following hold*

- a) *if x is a feasible solution to (LP1) then $(x, \mathbf{1} - x)$ is also feasible for (LP2).*
- b) *if (x, z) is a feasible solution to (LP2) then x is a feasible solution to (LP1).*

Finally, to transform the conditions $x_i + z_i = 1$ into $x_i + z_i \leq 1$, we add to (LP2) an adequate objective function (see, e.g., [20]) resulting in the following program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^n (x_i + z_i) \\ \text{subject to} \quad & x^T A + b \leq r^\# \end{aligned} \tag{7}$$

$$r^\# x \leq c \tag{8}$$

$$\begin{aligned} Wz &\leq W \cdot \mathbf{1} - d \\ x_i + z_i &\leq 1 \\ 0 \leq x_i, z_i &\leq 1. \end{aligned} \tag{LP3}$$

We have transformed the restrictions “=” into “ \leq ,” yet this does not change the optimal solution because $x_i + z_i$ appears positively in the objective function, and since we maximize this “forces” those restrictions to hold with equality. Moreover, the feasible solutions of (LP2) and (LP3) are related as follows:

Proposition 3 (a) *if (x, z) is a feasible solution to (LP2) then it is also feasible for (LP3); (b) if (x, z) is an $(1 - \varepsilon)$ -optimal solution to (LP3) then we can construct (x', z') in NC such that z' is feasible to $Wz \leq W \cdot \mathbf{1} - d$ and*

$$\begin{aligned} x'^T A + b &\leq r^\# + K_1 \varepsilon n \\ r^\# x' &\leq c + K_2 \varepsilon n^2 \end{aligned}$$

where K_1 and K_2 are two constants computable in NC from the instance.

Proof. b) A near-optimal solution (x, z) to (LP3) can be found via the Luby & Nisan’s algorithm. Such solution will have cost at least $(1 - \varepsilon)n$ (supposing that (LP2) is feasible). Now, we define $z' = z$ and $x' = \mathbf{1} - z$. Since

$$\sum_{i=1}^n (x_i + z_i) \geq (1 - \varepsilon)n$$

we will have that $x_i + z_i \approx 1 - \varepsilon$ therefore, intuitively, the values of x'_i will not increase too much. More precisely, let, for any i , $x'_i = x_i + \varepsilon_i$, and also $e = (\varepsilon_1, \dots, \varepsilon_n)$. Notice that

$$\begin{aligned} \sum_{i=1}^n \varepsilon_i &= \sum_{i=1}^n (1 - x_i - z_i) = n - \sum_{i=1}^n (x_i + z_i) \\ &\leq n - (1 - \varepsilon)n = \varepsilon n. \end{aligned}$$

Clearly, the conditions $Wz \leq W \cdot \mathbf{1} - d$ will be satisfied by z' and for the rest of constraints they might be violated by x' . However, because of the magnitudes of $a_{ij} = O(1)$ and $r_i^\# = \Theta(n)$, for x' we have that

$$x'^T A + b = (x + e)^T A + b \leq r^\# + K_1 \varepsilon n \tag{9}$$

$$r^\# x' = r^\# (x + e) \leq c + K_2 \varepsilon n^2 \tag{10}$$

where K_1 and K_2 are two constants defined as follows: $K_1 = \max_{i,j} a_{ij}$ and $K_2 = \max_i \alpha_i$ where α_i are the upper bounds on $r_i^\#$, i.e., $r_i^\# \leq \alpha_i \cdot n$. \square

Next, having (x', z') , we apply the Randomized Rounding with Scaling (as explained previously, Theorem 4.3 in [17]) to $Wz \leq W \cdot \mathbf{1} - d$. We can apply this theorem because we know, from the definition of Smooth QP, that the i th component of $W \cdot \mathbf{1} - d$ is $\Omega(\log n)$ and therefore the condition of Raghavan & Thompson's theorem holds. Rounding the feasible fractional solution z gives a 0,1 solution u , that with high probability, satisfies $Wu \leq W \cdot \mathbf{1} - d$. Letting $y = \mathbf{1} - u$, we have $Wy \geq d$. Furthermore, for y we will have:

$$\begin{aligned} y^T A_i + b_i &\leq (r_i^\# + K_1 \varepsilon n) + O(\sqrt{n \log n}) \\ r^\# y &\leq (c + K_2 \varepsilon n^2) + O(n) \cdot O(\sqrt{n \log n}). \end{aligned} \quad (11)$$

Consequently,

$$\begin{aligned} y^T A y + b y &= (y^T A + b) y \\ &\leq \left((r^\# + K_1 \varepsilon n) + O(\sqrt{n \log n}) \right) y \\ &\leq r^\# y + K_1 \varepsilon n \mathbf{1} \cdot y + O(\sqrt{n \log n}) \mathbf{1} \cdot y \\ &\leq (c + K_2 \varepsilon n^2) + O(n) \cdot O(\sqrt{n \log n}) + \\ &\quad + K_1 \varepsilon n \mathbf{1} \cdot y + O(\sqrt{n \log n}) \mathbf{1} \cdot y \\ &\leq c + (K_1 + K_2 + 2) \varepsilon n^2. \end{aligned}$$

Thus, if we find the fractional solution with $\varepsilon' = \varepsilon / (K_1 + K_2 + 2)$ we will have, from above, $y^T A y + b y \leq c + \varepsilon n^2$, as desired. But, we can write (7) only if we knew the values $r_i^\#$. Instead, it is shown in [3] that using estimates r_i for them such that

$$|r_i^\# - r_i| < \varepsilon n \quad (12)$$

then (11) and (12) still hold. To show that such estimates can be found in NC, we first prove that they can be found in RNC and later we prove that they can be found also deterministically in NC. To find $r = (r_i)_{i=1}^n$ in RNC we use the Sampling Lemma to produce $n^{O(1/\varepsilon^2)}$ estimates for $r^\#$. We can run in parallel $n^{O(1/\varepsilon^2)}$ positive linear programs and take as estimate the one whose linear program has the best outcome. The values r_i are found as follows.

1. Choose a set S of $k = O(\log n / \varepsilon^2)$ indices at random.
2. In parallel, for each of $2^k = n^{O(1/\varepsilon^2)}$ possible assignments to variables with indices in S produce an estimate r of $r^\#$ by taking

$$r_i = b_i + \frac{n}{k} \sum_{j \in S} a_{ij} s_j, \quad 1 \leq i \leq n,$$

where s_j is the value assigned to the j -th variable.

Since the assignments are found exhaustively, in one of the assignments generated above we have $s_j = x_j^*$, that is, s_j is the j -th component of the optimal solution. In order to estimate $r_i^\#$ we only need to estimate $a_{ij} x_j^*$, since $r_i^\# = \sum a_{ij} x_j^* + b_i$ and b_i is a constant. Applying the Sampling Lemma on the set $\{a_{ij} x_j^*\}$, results in estimates that satisfy (12). \square

Comment When there are no linear restrictions $Wx \geq d$, the Randomized Rounding with Scaling is applied directly to the system $\{x^t A + b \leq r^\#, r^\# x \leq c\}$. Arora, Karger and Karpinski [3] have presented their scheme precisely for this case.

3.2 Derandomization

Now we show that both the Sampling Lemma and the Randomized Rounding can be done in NC. For the Sampling Lemma, we can use similar arguments to those for the sequential case [3]. Instead of choosing $k = O(\log n / \varepsilon^2)$ indices independently, it suffices to choose the vertices encountered in a random walk of length k on a constant degree expander. The number of such walks is $n^{O(1/\varepsilon^2)}$, i.e., polynomial in n and therefore can be handled in NC. So, it remains to explain the following two points: first, how to construct in NC a constant degree expander, and second, how to simulate the random walk in NC. Both these points have been extensively treated in [10, 13, 9, 1], and we give a simple explanation as indicated in [12]. A constant degree d expander on n nodes is an n -node d -regular graph in which the number of neighbors of any set of vertices S is larger than some positive constant multiple of the cardinality of S (see, e.g., [15, pages 108–112]). It is well known that the adjacency matrix of such an expander is a symmetric matrix whose greatest eigenvalue is d and whose second largest one is less than d . Such constant degree expanders can be explicitly constructed in NC for any n and d , even with the additional condition that the second greatest eigenvalue λ be less than $d^{9/10}$ [9]. Let $N = 2^r$, where $N = n^{O(1/\varepsilon^2)}$, and let us identify the set $\{0, 1\}^r$ with the nodes of a d -regular N -node graph expander G , where d is set to some constant value such that $\lambda \leq d/4$. Then, we choose the nodes y_1, \dots, y_N as the nodes visited in a random walk of length k , with $k = O(\log n / \varepsilon^2)$, starting from a random node z . The random walk is determined by the starting node z and integers $i_j \in \{1, \dots, d\}$, for $j \in \{1, \dots, k\}$ describing which edge to use in the j -th step of the walk. Thus, y_1 is the i_1 -th neighbor of z , and for $j \geq 2$, y_j is the i_j -th neighbor of y_{j-1} . The number of random bits used in this process is $r + O(k)$, and therefore the whole process can be simulated deterministically in NC [1, 16].

Regarding the Randomized Rounding, this technique, in its general setting, seems to be inherently sequential [14]. However, it is possible to derandomize it for several cases, and in particular, for the linear systems of all coefficients positive numbers. Application of this sort can be found in [14, 7] (known as Lattice Approximation Technique), or in [2] (applied to PIP's—Packing Integer Programs). In both cases the condition for the positive coefficients to be positive is crucial. The linear program of our case matches the required conditions of these techniques. \square

3.3 Extensions

From the arguments in the previous subsections we have already proved Theorem 2. This result can be extended easily to positive instances of Smooth degree- l Integer Programming, for fixed l . Indeed, in this case we reduce (as in the case of Smooth QP) the degree- l program to a degree- $(l - 1)$ program, and successively until a Smooth QP is obtained, whose approximate solution is also an approximate solution for the original problem. Each step introduces a small error but we do this a constant number of times overall. Note that the reduction is done in NC in l steps.

Theorem 4 *Given a positive instance of Smooth degree- l IP and a fixed ε , we can find in NC a 0,1 assignment to variables x_i such that*

$$p(x_1, x_2, \dots, x_n) \leq p(x^*) + \varepsilon n^2, \quad (13)$$

where p is the objective function and $p(x^*)$ is its optimal value.

It is clear from the definition of Smooth QP that for its optimal value $p(x^*)$ we have $p(x^*) \geq \delta n^2$, for some positive constant δ , and therefore (13) is written as

$$p(x_1, x_2, \dots, x_n) \leq (1 + \varepsilon/\delta)p(x^*)$$

which gives us an approximation scheme. Furthermore, the overall parallel time is bounded by a polylogarithm in the number of variables n . Thus we get

Theorem 5 *There are NC Approximation Schemes for Positive Smooth QP and for bounded degree Positive Smooth IP.*

4 Applications

Smooth QP's (and more generally, Smooth Integer Programs) are strong enough to represent combinatorial problems. These programs were recently used to obtain Polynomial Time Approximation Schemes for dense instances of several NP-hard problems [3], however in their formulation, the instances of QP or

degree- l Integer Programs are neither positive nor in packing/covering form. Thus, we have to first formulate them as positive instances in packing/covering form and then apply our method. Notice that in all our applications the linear restrictions $Wx \geq d$ do not appear, actually there will appear simply complementary ones, i.e., restrictions of type $x_i + z_i = 1$. In this section we will derive NCAS for the dense instances of the problems MAX CUT, MAX DICUT and MAX 2SAT. We also show how to extend the result of MAX 2SAT for the general case of MAX k SAT on n variables. It turns out that for MAX CUT, MAX DICUT the condition for the instance to be smooth is equivalent with that of the graph instance being “dense.” A graph instance of n vertices is considered dense if any vertex in it has degree $\Theta(n)$ and an instance of MAX- k -SAT with m clauses over n variables is considered to be dense if $m = \Omega(n^k)$.

4.1 MAX CUT and MAX DICUT

Recall that in a MAX CUT instance we are given an undirected graph $G = (V, E)$ and we want to find a partition of its vertices $V = L \cup R$ that maximizes the number of crossing edges from L to R . We associate to any vertex $i \in V$ two 0/1 variables l_i and r_i defined as follows: $l_i = 1$ (resp. $r_i = 1$) if $i \in L$ (resp. $i \in R$). Now, let us consider the following program for MAX CUT:

$$\begin{aligned} & \text{maximize} && \frac{1}{2} \sum_{\{i,j\} \in E} (l_i r_j + l_j r_i) \\ & \text{subject to} && l_i + r_i = 1 \\ & && l_i, r_i \in \{0, 1\}, \quad \forall i \in V. \end{aligned} \quad (14)$$

Clearly, any coefficient in (14) is non-negative. In order to apply Theorem 2, we still need to transform (14) to match its requirements. More precisely, we have to (a) transform the restrictions “=” into restrictions “ \leq ,” i.e., to write the program in the packing form, and (b) the coefficients of the instance should satisfy the *smoothness* condition. As in Subsection 3.1 we modify the objective function appropriately and obtain:

$$\begin{aligned} & \text{max} && \frac{1}{2} \sum_{\{i,j\} \in E} (l_i r_j + l_j r_i + l_i + l_j + r_i + r_j) \\ & \text{s.t.} && l_i + r_i \leq 1 \\ & && l_i, r_i \in \{0, 1\}, \quad \forall i \in V. \end{aligned} \quad (15)$$

As for (a), observe that in (15), the restrictions “=” are relaxed to “ \leq ,” yet this does not change the optimal solution since $l_i + r_i$ is multiplied by a positive constant in the objective function, and since our maximization “forces” that restriction to hold with equality. Thus, the optimal solution of both (15) and (14) is the same. Finally, for condition (b), first note that

the coefficients of the second degree terms are 1, and those of the first degree are d_i , the degree² of vertex i . But notice from the definition of Smooth QP that, in the objective function, the sum of first degree terms is $\Theta(n^2)$, therefore in order for (15) to approximate the value of the optimal cut, the sum of the second degree terms in the objective function of (15) has also to be of order $\Theta(n^2)$. This is achieved when the number of second degree terms is $\Theta(n^2)$, that means if the graph instance is dense. At this point we have the connection between the density of the graph and smoothness of QP instance. Therefore, applying Theorem 2 we obtain:

Theorem 6 *There is an NCAS for dense instances of MAX CUT.*

The MAX DICUT problem, i.e., MAX CUT for directed graphs, can be modeled as a Positive QP similar to (14). In this case, supposing that the edges of the cut are oriented from the left to the right, the modified objective function has only second degree terms $l_i r_j$. Therefore, Theorem 6 also holds for MAX DICUT.

4.2 MAX 2SAT

In any instance of MAX 2SAT we are given a set $\{C_j\}_{j=1}^m$ of m clauses over n variables x_1, \dots, x_n , where each clause has 2 literals, and we want to find a truth assignment to the variables that maximizes the number of satisfied clauses. For any variable x_i , we consider two other 0/1 variables t_i, f_i such that $t_i = 1$ (resp. $f_i = 1$) if the variable $x_i = \text{true}$ (resp. $x_i = \text{false}$) and 0 otherwise. Now, we will associate a degree-2 term to any clause such that for any assignment of t_i, f_i that evaluates the term to 1, the corresponding assignment of x_i evaluates the clause to true. The term $q_j(t_1, \dots, t_n, f_1, \dots, f_n)$ corresponding to clause C_j is constructed as follows. If x_i appears positively in C_j , then we replace it by f_i and by t_i otherwise, and finally subtract the result from 1. For example, from the clause $\bar{x}_1 \vee x_3$ we construct the term $1 - t_1 f_3$. Therefore, we obtain the following program:

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^m q_j(t_1, \dots, t_n, f_1, \dots, f_n) \\ & \text{subject to} && t_i + f_i = 1 \\ & && t_i, f_i \in \{0, 1\}, \quad 1 \leq i \leq n \end{aligned} \quad (16)$$

This program is not positive since the degree-2 terms appear negatively (as in $1 - t_1 f_3$). By letting $q_j(t_1, \dots, t_n, f_1, \dots, f_n) = 1 - T_j(t_1, \dots, t_n, f_1, \dots, f_n)$, we can consider instead of (16) the program with the objective function, the

²In fact, instead of d_i we can take any positive coefficient that is $\Theta(n)$.

sum of the degree-2 terms T_j and take minimization instead of maximization, and thus (16) is transformed into:

$$\begin{aligned} & \text{minimize} && \sum_{j=1}^m T_j(t_1, \dots, t_n, f_1, \dots, f_n) + \\ & && + \sum_{i=1}^n d_i(t_i + f_i) \\ & \text{subject to} && t_i + f_i \geq 1 \\ & && t_i, f_i \in \{0, 1\}, \quad 1 \leq i \leq n \end{aligned} \quad (17)$$

As in the previous cases we have relaxed conditions “=” to conditions “ \geq ” by adding to the objective function $t_i + f_i$ multiplied by a positive constant d_i . In order for (17) to be smooth we must choose $d_i = \Theta(n)$. Then, any approximation for (17) would yield an approximation also for (16) if we let the number of degree-2 terms be of order $\Omega(n^2)$, i.e., if we let the MAX 2SAT instance be “dense.” Therefore, similar to the above, we have

Theorem 7 *There is an NCAS for dense instances of MAX 2SAT.*

MAX k SAT. This problem can be modeled by a program similar to that of MAX 2SAT (17), but in this case $T_j(t_1, \dots, t_n, f_1, \dots, f_n)$ is a degree- k term. In order for the resulting program to be smooth, we have to choose the coefficients d_i to be of order $d_i = \Theta(n^{k-1})$, and therefore $\sum_{i=1}^n d_i(t_i + f_i)$ will be of order $\Theta(n^k)$. Consequently, the objective function of the resulting program would approximate the maximum number of satisfied clauses if the sum $\sum_{j=1}^m T_j(t_1, \dots, t_n, f_1, \dots, f_n)$ has $\Theta(n^k)$ terms, i.e., if the instance of MAX k SAT is dense. Next, from this observation, we conclude:

Theorem 8 *There is an NCAS for dense instances of MAX k SAT.*

In fact the above result holds also for dense instances of a restricted version of the MAX SAT problem on n variables whose clauses have up to $l = \text{polylog}(n)$ literals. Indeed, in this case we will have a degree- l Integer Program for which the arguments of Subsection 3.3 also hold, i.e., the reduction to a Smooth QP is done also in NC.

Corollary 1 *There is an NCAS for dense instances of MAX SAT of clauses on n variables and up to $\text{polylog}(n)$ literals per clause.*

Open Questions

Find an NCAS for Positive QP or Positive CQP. We believe that the class of Positive Quadratic Programs can be used to model and approximate also other combinatorial optimization problems. Some

candidates in this line are Multiplicity Scheduling problems, Minimum Bisection, Minimum Separator and possibly others. Finally, it would be interesting to find problems that are modeled by Smooth QP with linear restrictions. Our applications do not have such restrictions.

Acknowledgments

We thank Josep Díaz for helpful discussions and suggestions. Many thanks to Raymond Greenlaw for his continuous support and comments that significantly improved this paper. We are grateful to the anonymous referees for several useful comments.

References

- [1] M. Ajtai, J. Komlos, and E. Szemerédi. Deterministic Simulation in LOGSPACE. In *Proceedings of 19th ACM Symposium on Theory of Computing*, page 132, 1987.
- [2] N. Alon and A. Srinivasan. Improved Parallel Approximation of a Class of Integer Programming Problems. *Algorithmica*, 17:449–462, 1997.
- [3] S. Arora, D. Karger, and M. Karpinski. Polynomial Time Approximation Schemes for Dense Instances of NP-hard Problems. In *Proceedings of 27th Annual ACM Symposium on Theory of Computing*, pages 284–293, 1995.
- [4] J. Díaz, M. Serna, P. Spirakis, and J. Torán. *Paradigms for Fast Parallel Approximability*. Cambridge University Press, 1997.
- [5] R. Greenlaw, H.J. Hoover, and W.L. Ruzzo. *Limits to Parallel Computation: P-completeness Theory*. Oxford University Press, 1995.
- [6] M. Grötschel, L. Lovász, and A. Schrijver. The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica*, 1:169–197, 1981.
- [7] D.R. Karger and D. Koller. (De)randomized Construction of Small Sample Spaces in \mathcal{NC} . In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, 1994.
- [8] A. Karmarkar. A New Polynomial Time Algorithm in Linear Programming. *Proc. 16th Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.
- [9] R. Karp, N. Pippenger, and M. Sipser. Expanders, Randomness, or Time versus Space. In *First Annual Conference on Structure in Complexity Theory*, pages 325–329, 1986.
- [10] A. Lubotzky, R. Phillips, and P. Sarnak. Explicit Expanders and the Ramanujan Graphs. In *Proceedings of 20th ACM Symposium on Theory of Computing*, pages 240–246, 1988.
- [11] M. Luby and N. Nisan. A Parallel Approximation Algorithm for Positive Linear Programming. In *Proceedings of 25th ACM Symposium on Theory of Computing*, pages 448–457, 1993.
- [12] M. Luby and A. Wigderson. Pairwise Independence and Derandomization. Technical Report TR-95-035, International Computer Science Institute, 1995.
- [13] G. Margulis. Explicit Group-Theoretical Constructions of Combinatorial Schemes and their Application to the Design of Expanders and Superconcentrators. *Problems of Information Transmission*, 24:39–46, 1988.
- [14] R. Motwani, J.S. Naor, and M. Naor. The Probabilistic Method Yields Deterministic Parallel Algorithms. *Journal of Computer and System Sciences*, 49:478–516, 1994.
- [15] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [16] N. Nisan and D. Zuckerman. More Deterministic Simulation in Logspace. In *Proceedings of 25th ACM Symposium on Theory of Computing*, pages 235–244, 1993.
- [17] P. Raghavan and C. Thompson. Randomized Rounding: a Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica*, 7:365–374, 1987.
- [18] S. Sahni. Computationally Related Problems. *SIAM Journal of Computing*, 3:262–279, 1974.
- [19] M. Serna. Approximating Linear Programming is Logspace Complete for P. *Information Processing Letters*, 37:233–236, 1991.
- [20] L. Trevisan. Positive Linear Programming, Parallel Approximation and PCP's. In J. Díaz and M. Serna, editors, *Fourth European Symposium on Algorithms*, volume 1136 of *Lecture Notes in Computer Science*, pages 62–75. Springer-Verlag, 1996.
- [21] S.A. Vavasis. Quadratic Programming is in NP. *Information Processing Letters*, 36:73–77, 1990.