# Approximating Convex Quadratic Programming is P-Complete

Maria Serna
Fatos Xhafa

Report LSI–95–58–R

# Approximating Convex Quadratic Programming is P-complete

Maria Serna[*] Fatos Xhafa[*]

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya, Pau Gargallo 5
08028-Barcelona, Spain

**Abstract**

In this paper we show that the problem of Approximating Convex Quadratic Programming is P-complete. We also consider two approximation problems related to it, *Solution Approximation* and *Value Approximation* and show both of these are P-complete. On the other hand, we show that we can approximate in NC those instances of Quadratic Programming (QP) that are "smooth" and "positive." Finally, we present a problem called Product Arrangement of a graph that is similar to Linear Arrangement. We formulate it as a Positive QP and prove that there is an NCAS for those instances that are "dense."

*Keywords*: Computational complexity, Parallel algorithms, Quadratic programming

## 1 Introduction

The problem of optimizing a convex function on a convex domain is called *convex programming* or *convex optimization*. Special cases of convex programming arise by letting the function and/or the domain be of a special type. Let $Q$ be an $n \times n$ positive semidefinite matrix, $A$ an $m \times n$ matrix, $a \in R$, $b \in R^m$ and $c \in R^n$. We consider here convex functions of the type $f(x) = 1/2\ x^TQx + c^Tx + a$ and regions $D = \{x \in R^n\ :\ Ax \leq b\}$. The convex programming imposed by these restrictions is called *Convex Quadratic Programming* (CQP).

Quadratic Programming (in the general case) is NP-complete [13, 15]. On the other hand, CQP can be solved sequentially in polynomial time, through the ellipsoid method [5] or the interior point algorithm [6]. Here we are interested in the parallel complexity of this problem. 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. Also, NCAS is the class of problems that have an NC approximation scheme (see for e.g. [4, page 111]). In this paper, we show that there is no NC algorithm for CQP. We cannot even approximate it in NC, unless P=NC. To prove this we make use of the fact that Linear Programming (LP) and the problem of approximating it are P-complete [2, 14].

We consider two approximation problems related to CQP [6, 14]. *Solution Approximation* consists of finding a feasible solution to CQP whose norm is close to that of an optimal one. *Value Approximation* is to find a vector $x$ such that the value of the objective function on it

[*]This research was supported by the ESPRIT BRA Program of the European Community under contract no. 7141, project ALCOM II.

is close to the optimal value, without the condition that $x$ be feasible. We show both of these problems are P-complete.

An instance of QP is called positive if all the entries of ($Q$,$c$,$A$,$b$) and $a$ are non-negative. Recall that Positive LP has an NCAS [8]. Here we address the same question for Positive QP. We consider a restricted version of QP, called Smooth QP (see Definition 3), and show that there is an NCAS for positive instances of Smooth QP. This also implies an NCAS for smooth and positive instances of CQP. Next, we present an application. We consider a problem similar to the Linear Arrangement of a graph, called *Product Arrangement.* As in the case of Linear Arrangement, we show that this problem is NP-complete. We observe that this problem can be formulated as a Positive QP. Then, we show that if the graph instance is "dense," we can approximate Product Arrangement in NC within an error of $\epsilon$.

## 2 Definition of the Problems

Let $Q$ be an $n \times n$ positive semidefinite matrix, $A$ an $m \times n$ matrix of range $m$, $a \in R$, $b \in R^m$ and $c \in R^n$. In all definitions, the entries of ($Q$,$c$,$A$,$b$) are assumed to be integers, unless otherwise stated. The following problems are used throughout the paper.
**Linear Programming** problem (LP) is

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \end{array}$$

**Convex Quadratic Programming** with linear constraints (CQP) is

$$\begin{array}{ll} \text{minimize} & f(x) = 1/2\, x^T Q x + c^T x + a \\ \text{subject to} & Ax \leq b \end{array}$$

**Definition 1** *Let $x^*$ be an optimal solution to CQP and $\epsilon \in (0,1]$. The* `Solution Approximation Problem` *to CQP, denoted $\epsilon$-CQPS, is to compute a feasible rational $n$-vector $x$ such that*

$$||x^*||_p \geq ||x||_p \geq \epsilon ||x^*||_p$$

*where $||x||_p = (\sum_{i=1}^n x_i^p)^{1/p}$.*

**Definition 2** *The* `Value Approximation Problem` *to CQP, denoted $\epsilon$-CQPV, is to compute a rational $n$-vector $x$ such that*

$$\frac{1}{\epsilon} f(x^*) \geq f(x) \geq f(x^*).$$

Let $\epsilon$-LPS ($\epsilon$-LPV) be the Solution Approximation (respectively, Value Approximation) problem corresponding to LP. It was shown in [14] that both of these are P-complete.

**Theorem 1** [14] *$\epsilon$-LPS and $\epsilon$-LPV are P-complete for any $\epsilon \in (0,1]$.*

## 3 Approximating Convex Quadratic Programming

The following theorem shows that CQP cannot be approximated in NC, unless P=NC.

**Theorem 2** *Approximating Convex Quadratic Programming is P-complete.*

*Proof.* Firstly, we show that there is an NC reduction from LP to CQP. The reduction goes as follows. Suppose we have an instance of LP:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \le b \end{array}$$

We construct an instance of CQP as follows:

$$\begin{array}{ll} \text{minimize} & f(x) = c^T x + q(x) \\ \text{subject to} & Ax \le b \end{array}$$

where $q(x) = 1/2\,||Ax - b||^2 = 1/2\,x^T A^T Ax - x^T A^T b + 1/2\,b^T b$. Note that $q(x) \ge 0$ for any $x$ and that $q(x) = 0$ for all $x$ with $Ax = b$. Therefore, $q(x)$ as a quadratic form is positive semidefinite (see for e.g. [11, pages 12–14]).

It holds that if $x^*$ is an optimal solution to LP, then $x^*$ is also an optimal solution for CQP and vice-versa. For that, first recall that an optimal solution to LP is a vertex of the convex polyhedra (the set of feasible solutions), such a point satisfies $Ax = b$. Further, we notice that the quadratic form $q(x)$ achieves its minimum in solutions $x$ with $Ax = b$ and that its minimum value is equally zero. Therefore, for solutions $x^*$ with $Ax^* = b$ we have

$$c^T x^* = f(x^*).$$

So any feasible solution $x^*$ to $Ax \le b$ that minimizes CQP also minimizes LP and vice-versa. Note that this reduction is in NC. Since LP and the problem of approximating it are P-complete, the theorem follows. □

Based on the non-approximability of $\epsilon$-LPS and $\epsilon$-LPV, we also show that even the problems of Solution Approximation and Value Approximation to CQP are P-complete.

**Theorem 3** *$\epsilon$-CQPS and $\epsilon$-CQPV are P-complete for any $\epsilon \in (0, 1]$.*

*Proof.* We define a restricted version of linear programming, denoted LPP, as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} x_i \\ \text{subject to} & Ax \le b, \quad \text{where } x_i = 0 \text{ or } 1 \end{array}$$

whose optimal solution is unique.

Let $\epsilon$-LPPS and $\epsilon$-LPPV be the corresponding approximation problems. Consider the CQP obtained by LPP as above (Theorem 2) and its corresponding approximation problems. We can easily see that if $x^*$ is the optimal solution (to both LPP and CQP) then we have

$$||x^*||_p = ||x^*||_1 = \sum_{i=1}^{n} x_i^* \qquad (1)$$

and

$$\sum_{i=1}^{n} x_i^* = f(x^*) \qquad (2)$$

Now, from (1), (2) and Theorem 1 our theorem follows. □

## 3.1 Smooth Quadratic Programming

Here we consider a restricted version of QP, called Smooth QP, defined as follows. For a more general definition see [1].

**Definition 3** *Let $x = (x_i)_{i=1}^n$. Smooth Quadratic Programming has the following form:*

$$\textit{minimize} \quad g(x) = \sum a_{ij}x_ix_j + \sum b_ix_i + c$$

$$\textit{subject to} \quad Wx \leq d \textit{ with } x_i \in \{0,1\}, \quad 1 \leq i \leq n$$

*where $a_{ij}$, $w_{ij}$ are $O(1)$, $b_i = \Theta(n)$, $c = \Theta(n^2)$ and $d$ is a constant.*

Given an instance of Smooth QP, we can approximate it in NC if it is, in addition, positive. The motivation for this is that Positive Linear Programming has a NCAS [8]. Note that this result will also hold for those instances of CQP that are smooth and positive. To prove this result, we will make use of the following two lemmas. The first one is a well known technique on how to estimate the sum of $n$ numbers by random sampling (see for e.g. [10]).

**Lemma 1 (Sampling Lemma)**
*Let $p = \sum_{i=1}^n a_i$, where each $a_i$ is $O(1)$. If we pick randomly a subset of $s = O(\log n/\epsilon^2)$ of them and compute their sum $q$, then with high probability $p - \epsilon n \leq qn/s \leq p + \epsilon n$.*

The second lemma by Raghavan and Thompson shows how to round a fractional solution of a linear program to an integer solution of the same linear program.

**Lemma 2** [12] **(Randomized Rounding)**
*Let $x = (x_i)_{i=1}^n$ be a vector with $0 \leq x_i \leq 1$ that satisfies linear constraints $A_ix = b_i$, where each entry of $A_i$ is $O(1)$. Construct $y = (y_i)_{i=1}^n$ randomly by setting $y_i = 1$ with probability $x_i$ and 0 otherwise. Then, with high probability, $A_iy = b_i + O(\sqrt{n}\log n)$.*

We prove the following theorem.

**Theorem 4** *Given a positive instance of Smooth QP and a fixed $\epsilon$, we can find in NC a 0,1 assignment to variables $x_i$ such that*

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

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

In fact, it suffices to prove the following theorem.

**Theorem 5** *Suppose there is a 0,1 solution to the following Positive Quadratic Program*

$$\begin{cases} x^TAx + bx \leq c \\ Wx \leq d \end{cases} \tag{3}$$

*where $a_{ij}$, $w_{ij} = O(1)$, $b_i = \Theta(n)$ and $c$, $d$ are constants. Then for any fixed $\epsilon$, we can find in NC an assignment of 0,1 values to $x$ such that*

$$x^TAx + bx \leq c + \epsilon n^2 \tag{4}$$

*Proof.* The proof closely follows a theorem from [1]. 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 is applied to find its fractional solution in NC. Then, we round the fractional solution to a 0,1 solution using the Randomized Rounding. The reduction is as follows. Suppose $x^{\#}$ is a solution of (3) and let $r^{\#} = x^{\#}A + b$. We can write (3) as

$$\begin{cases} x^T A + b = r^{\#} \\ r^{\#} x \leq c \\ Wx \leq d \\ 0 \leq x_i \leq 1 \end{cases} \qquad (5)$$

for which, $x^{\#}$ is also a feasible solution. Let $x$ be its fractional solution. By rounding this solution to a 0,1 solution $y$, we have that with high probability, i.e. probability $1 - O(1/n)$,

$$\begin{array}{l} y^T a_i + b_i = r_i^{\#} + O(\sqrt{n \log n}) \\ r^{\#} x \leq c + O(n) \cdot O(\sqrt{n \log n}) \\ W_i y \leq d + O(\sqrt{n \log n}) \end{array} \qquad (6)$$

Note that inequality (6) holds because $r_i^{\#} = \Theta(n)$. Consequently, we will have

$$\begin{aligned} y^T A y + by &= (y^T A + b) y \\ &= (r^{\#} + O(\sqrt{n \log n})) y \\ &\leq c + O(n^{3/2} \log n) \\ &\leq c + \epsilon n^2 \end{aligned}$$

We can write (5) if we knew the values $r_i^{\#}$. Instead, we use estimates $r_i$ for them such that

$$|r_i^{\#} - r_i| < \epsilon n \qquad (7)$$

for which the above arguments also hold (see [1] for details). 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 $r = (r_i)_{i=1}^n$ we use the Sampling Lemma to produce $n^{O(1/\epsilon^2)}$ estimates for $r^{\#}$. We can run in parallel $n^{O(1/\epsilon^2)}$ positive linear programs and take as estimate the best one. The values $r_i$ are found as follows.

1. Choose a set $S$ of $k = O(\log n/\epsilon^2)$ indices at random.

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

$$r_i = b_i + \frac{n}{k} \sum_{j \in S} a_{ij} s_j$$

   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 $b_i$ is a constant, see first equality in (5). Applying the Sampling Lemma on the set $\{a_{ij} x_j^*\}$, results in estimates that satisfy (7).

Now we show that both the Sampling Lemma and the Randomized Rounding can be done in NC. Regarding the Sampling Lemma, let $X_{i_1}, X_{i_2}, \ldots, X_{i_k}$ with $k = O(\log n/\epsilon^2)$ be random variables defined by

$$X_{i_j} = \begin{cases} 1 & \text{if index } i_j \text{ is chosen, and} \\ 0 & \text{otherwise.} \end{cases}$$

The only condition we need for these variables is to be pair-wise independent. Note that the sample space generated by random variables $X_{i_j}$, $1 \leq j \leq k$ is $O(2^k) = n^{O(1/\epsilon^2)}$ therefore, such sample space can be exhaustively searched in NC. On the other hand, the Randomized Rounding technique is parallelized (deterministically) in [9] by using the method of conditional probabilities. This method is applicable in our case, too. □

# 4 An Application

Smooth Quadratic 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 [1]. Unfortunately, it is difficult to find problems whose QP is, in addition, positive. Following we describe a problem similar to Linear Arrangement, called Product Arrangement. We first show that this problem is NP-complete and then we observe that it can be formulated as a positive QP. Furthermore, it turns out that when the graph instance of $n$ vertices is *dense* i.e. it has minimum degree $\Theta(n)$, then the corresponding QP is smooth. Thus for dense instances of this problem Theorem 4 is applicable.

## 4.1 Product Arrangement of a Graph

We define the problem of Product Arrangement of a graph analogously to that of Linear Arrangement.

**Definition 4 Maximum Product Arrangement (MaxPA).**
*Given an undirected graph $G = (V, E)$ and a natural number $c$, determine whether there exists a one-to-one mapping $f : V \rightarrow \{1, \ldots, |V|\}$ such that*

$$\sum_{\{u,v\} \in E} f(u) \cdot f(v) \geq c.$$

In a similar way **Minimum Product Arrangement (MinPA)** is defined.

Let MaxLA (MinLA) be the Maximum (respectively, Minimum) Linear Arrangement Problem and MaxC the Maximum Cut problem on graphs that are not bipartite. We show a logarithmic space reduction from MaxC to MaxPA and since MaxC is NP-complete [7], this shows MaxPA is also NP-complete.

**Proposition 1** *Maximum Product Arrangement Problem is NP-complete.*

*Proof.* Our proof is based on the reduction given in [3], where MaxC was polynomially reduced to MaxLA. Given an instance $[G' = (V', E'),\ c']$ for MaxC, the instance $[G = (V, E),\ c]$ for MaxPA is constructed as follows:

$$V = V' \cup \{u_1, u_2, \ldots, u_{n^3}\}, \quad E = E', \quad c = c' \cdot p$$

where $n = |V'|$ and $p = n(n + n^3)$. Now, suppose we have a cut $C(S, \overline{S})$ such that $|C(S, \overline{S})| \geq c'$. We consider the mapping $f$ that satisfies the following three conditions.

(1) If $v \in S$ then $1 \leq f(v) \leq |S|$.

(2) If $v \in \{u_1, u_2, \ldots, u_{n^3}\}$ then $|S| < f(v) \leq |S| + n^3$.

(3) If $v \in \overline{S}$ then $|S| + n^3 < f(v) \leq n + n^3$.

Now, let us prove that

$$\sum_{\{u,v\}\in E} f(u) \cdot f(v) \geq c' \cdot p.$$

For that, we write

$$\sum_{\{u,v\}\in E} f(u) \cdot f(v) = \sum_1 f(u) \cdot f(v) + \sum_2 f(u) \cdot f(v) + \sum_3 f(u) \cdot f(v)$$

where $\sum_1$ is taken over the edges of the cut, $\sum_2$ is taken over the edges with both endpoints in $S$ and $\sum_3$ is taken over the edges with both endpoints in $\overline{S}$. We notice that

$$\sum_1 f(u) \cdot f(v) \geq |C(S, \overline{S})| \cdot n^3 \quad \text{and} \quad \sum_3 f(u) \cdot f(v) \geq n^6.$$

The first inequality holds because any edge of the cut has its product greater then $n^3$ and the second holds because there is at least an edge with both endpoints in $\overline{S}$, otherwise we can take the cut $C(\overline{S}, S)$. But the cardinality of any cut is $|C(S, S')| \leq n^2$ so we have,

$$\begin{aligned} \textstyle\sum_{\{u,v\}\in E} f(u) \cdot f(v) & \geq |C(S, S')| \cdot n^3 + n^6 \\ & \geq n^2 \cdot |C(S, S')| + n^4 \cdot |C(S, S')| \\ & \geq c' \cdot n(n + n^3) \end{aligned}$$

as required. For the other direction, suppose we have a mapping $f$ for $G = (V, E)$ for which

$$\sum_{\{u,v\}\in E} f(u) \cdot f(v) \geq c.$$

Consider, for each $i$, $1 \leq i \leq |V|$, the set $S_i = \{v \mid v \in V \text{ and } f(v) \leq i\}$. This defines a cut $C(S_i, \overline{S_i})$. Let $C(S, \overline{S}) = C(S_j, \overline{S_j})$ be the maximum of these cuts. We rearrange the vertices of $V$ in the following way. The vertices of $V'$ do not change their position while all the isolated vertices $u_i$ are put in between $S \cap V'$ and $\overline{S} \cap V'$. Call this arrangement $f'$. In such an arrangement the following holds

$$\sum_{\{u,v\}\in E} f'(u) \cdot f'(v) \geq \sum_{\{u,v\}\in E} f(u) \cdot f(v) \geq c' \cdot p.$$

Indeed, the vertices $u_i$, $1 \leq i \leq n^3$, are isolated but if a vertex $u_i$ is interposed with $u$ and $v$ for any edge $\{u, v\}$ then $f(u) \cdot f(v)$ is increased by the $min\{f(u), f(v)\}$. Clearly, in the new arrangement each vertex $u_i$, $1 \leq i \leq n^3$, contributes the maximum value in the total sum of the products of the edges. Now, the sum of all products $f'(u) \cdot f'(v)$ can be divided into two parts: the sum over the edges of the cut and that of the edges not in the cut. For any edge $\{u, v\}$ of the cut, we have

$$f'(u) \cdot f'(v) \leq |S|(|S| + |\overline{S}| + n^3) \leq n(n + n^3).$$

So,

$$n(n + n^3) \cdot |C(S, \overline{S})| \geq \sum_{\{u,v\}\in C(S,\overline{S})} f'(u) \cdot f'(v).$$

For the second part, we may have edges of length 1, 2 and up to $n-1$. Let $L_k$ be the product of the edges of length $k$. Since any edge of this type is of the form $\{i, i+k\}$, we have that

$$L_k = \sum_{i=1}^{n-k} i(i+k) = \frac{(n-k)(n-k+1)(2n+k+1)}{6} \leq n^3.$$

Therefore

$$L = \sum_{i=1}^{n-1} L_k \leq n^4.$$

Thus, the second part is bounded above by $n^4$ because it is bounded above by $L$. Now, putting altogether we have that

$$n(n+n^3) \cdot |C(S, \overline{S})| + L \geq c' \cdot n(n+n^3)$$

and since $|C(S, \overline{S})|$ and $c'$ are integers we have that $|C(S, \overline{S})| \geq c'$. □

Now, we show that MinPA is also NP-complete by showing that MaxPA is logspace reduced to MinPA.

**Proposition 2** *Minimum Product Arrangement is NP-complete.*

*Proof.* Given the graph $G = (V, E)$ of $n$ vertices and the natural number $c$ for which there exists a labelling $f$ such that $\sum_{\{u,v\}\in E} f(u) \cdot f(v) \geq c$, we construct the complementary graph of $G$, $\overline{G} = (V, \overline{E})$ and let $c' = L_n - c$ where

$$L_n = (1 \cdot 2 + 1 \cdot 3 + \cdots + 1 \cdot n) + \cdots + (i \cdot (i+1) + i \cdot (i+2) + \cdots + i \cdot n) + \cdots + (n-1) \cdot n$$

is the sum of the products of the edges of the complete graph. The labelling we take is $f' = f$, therefore we have

$$\sum_{\{u,v\}\in E} f(u) \cdot f(v) + \sum_{\{u,v\}\in \overline{E}} f(u) \cdot f(v) = L_n.$$

Thus

$$\sum_{\{u,v\}\in \overline{E}} f(u) \cdot f(v) \leq L_n - c = c'.$$

This proves the proposition. □

## 4.2 Minimum Product Arrangement as a Positive Quadratic Program

We observe that MinPA can be written as a positive QP. Let us consider the following model:

$$(*) \qquad \begin{array}{lll} \text{minimize} & \sum_{\{i,j\}\in E} x_i x_j & \\ \text{subject to} & x_i \neq x_j, & i \neq j \\ & x_i \in \{1, \ldots, n\}, & 1 \leq i \leq n \end{array}$$

In order to express the constraints on variables $x_i$ linearly we introduce 0,1 variables $\delta_{ik}$,

$$\delta_{ik} = \begin{cases} 1 & \text{if } x_i = k, \text{ and} \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i,\ k \leq n$ and put on them the conditions:

$$\delta_{ik} + \delta_{jk} \leq 1, \quad \forall \{i,j\} \in E, \quad 1 \leq k \leq n.$$

That is, variables $x_i$ and $x_j$ cannot take the same value $k$ and

$$\delta_{i1} + \delta_{i2} + \cdots + \delta_{in} = 1, \quad 1 \leq i \leq n.$$

That is, $x_i$ takes a value in $\{1, \ldots, n\}$. Now, we write $(*)$ in terms of $\delta_{ik}$ by substituting

$$x_i = 1 \cdot \delta_{i1} + 2 \cdot \delta_{i2} + \cdots + n \cdot \delta_{in}, \quad 1 \leq i \leq n.$$

Therefore, we have

$$\begin{cases} \text{minimize} & n^2 \cdot \sum_{\{i,j\} \in E} (\frac{1}{n}\delta_{i1} + \frac{2}{n}\delta_{i2} + \cdots + 1 \cdot \delta_{in})(\frac{1}{n}\delta_{j1} + \frac{2}{n}\delta_{j2} + \cdots + 1 \cdot \delta_{jn}) \\ \text{subject to} & \\ & \delta_{ik} + \delta_{jk} \leq 1, \qquad \forall \{i,j\} \in E, \quad 1 \leq k \leq n \\ & \delta_{i1} + \delta_{i2} + \cdots + \delta_{in} = 1, \qquad 1 \leq i \leq n \\ & \delta_{ik} \in \{0,1\}, \qquad 1 \leq i\ , k \leq n. \end{cases}$$

Clearly, this is a positive quadratic program.

It is easy to see that when the graph instance is dense then the corresponding instance of MinPA can be written as a smooth QP. Under this condition we have the following:

**Theorem 6** *There is an NCAS for dense instances of Minimum (Maximum) Product Arrangement of a graph.*

## Open Questions

Find an NCAS for positive QP. We have also shown MaxPA is NP-complete for bipartite graphs, however, the reduction is somewhat complicated. It follows a similar proof in [3]. It would be interesting to come up with a simpler reduction.

## Acknowledgements

We thank Josep Díaz for many helpful discussions and suggestions. Many thanks to Raymond Greenlaw for his comments that significantly improved this paper. Ray also suggested us the case for bipartite graphs.

## References

1. S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings 27th Annual ACM Symposium on Theory of Computing*, pages 284–293, 1995.

2. D. Dobkin, J.R. Lipton, and S. Reiss. Linear Programming is logspace hard for P. *Information Processing Letters*, 8:96–97, 1979.

3. S. Even and Y. Shiloah. NP-completeness of several arrangements problems. Technical Report No. 43, Dept. of Computer Science, Technion, Haifa, Israel., 1975.

4. R. Greenlaw, H.J. Hoover, and W.L. Ruzzo. *Limits to parallel computation: P–completeness theory.* Oxford University Press, 1995.

5. M. Grotschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

6. A. Karmarkar. A new polynomial time algorithm in linear programming. *Proc. 16th Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.

7. R.M. Karp. *Reducibility Among Combinatorial Problems.* Complexity of Computer Computations, eds. R.E. Miller and J.W. Thatcher, Plenum Press, 1972.

8. M. Luby and N. Nisan. A parallel algorithm for positive linear pogramming. In *Proceedings 25th ACM Symposium on Theory of Computing*, pages 448–457, 1993.

9. 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.

10. R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

11. A.L. Peresini, F.E. Sullivan, and J.J. Uhl, Jr. *The Mathematics of Nonlinear Programming.* eds. Ewing, J.H. Springer Verlag, 1988.

12. P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

13. S. Sahni. Computationally Related Problems. *SIAM Journal of Computing*, 3:262–279, 1974.

14. M.J. Serna. Approximating linear programming is log-space complete for P. *Information Processing Letters*, 37:233–236, 1991.

15. S.A. Vavasis. Quadratic Programming is in NP. *Information Processing Letters*, 36:73–77, 1990.

**Departament de Llenguatges i Sistemes Informàtics**
Universitat Politècnica de Catalunya

**Research Reports – 1995**

LSI–95–1–R "Octree simplification of polyhedral solids", Dolors Ayala and Pere Brunet.

LSI–95–2–R "A note on learning decision lists", Jorge Castro.

LSI–95–3–R "The complexity of searching implicit graphs", José L. Balcázar.

LSI–95–4–R "Design quality metrics for object-oriented software development", Alonso Peralta, Joan Serras, and Olga Slavkova.

LSI–95–5–R "Extension orderings", Albert Rubio.

LSI–95–6–R "Triangles, ruler, and compass", R. Joan-Arinyo.

LSI–95–7–R "The modifiability factor in the LESD project: definition and practical results", Nuria Castell and Olga Slavkova.

LSI–95–8–R "Learnability of Kolmogorov-easy circuit expressions via queries", José L. Balcázar, Harry Buhrman, and Montserrat Hermo.

LSI–95–9–R "A case study on prototyping with specifications and multiple implementations", Xavier Franch.

LSI–95–10–R "Evidence of a noise induced transition in fluid neural networks", Jordi Delgado and Ricard V. Solé.

LSI–95–11–R "Supporting transaction design in conceptual modelling of information systems", Joan A. Pastor-Collado and Antoni Olivé.

LSI–95–12–R "Computer (PC) assisted drawing of diagrams for forecasting soaring weather", Lluís Pérez Vidal.

LSI–95–13–R "Animats adaptation to complex environments as learning guided by evolution", Mario Martin, Màrius Garcia, and Ulises Cortés.

LSI–95–14–R "Learning to solve complex tasks by reinforcement: A new algorithm", Mario Martin and Ulises Cortés.

LSI–95–15–R "Analysis of Hoare's FIND algorithm with median-of-three partition", Peter Kirschenhofer, Conrado Martínez, and Helmut Prodinger.

LSI–95–16–R "MDCO to B-Rep conversion algorithm", Dolors Ayala, Carlos Andújar, and Pere Brunet.

LSI-95-17-R "Augmented regular expressions: A formalism to describe, recognize, and learn a class of context-sensitive languages", René Alquézar and Alberto Sanfeliu.

LSI-95-18-R "On parallel versus sequential approximation", Maria Serna and Fatos Xhafa.

LSI-95-19-R "A set of rules for a constructive geometric constraint solver", Robert Joan-Arinyo and Antoni Soto.

LSI-95-20-R "From degenerate patches to triangular and trimmed patches", Marc Vigo, Núria Pla, and Pere Brunet.

LSI-95-21-R "Learning Ordered Binary Decision Diagrams", Ricard Gavaldà and David Guijarro.

LSI-95-22-R "The complexity of learning minor closed graph classes", Carlos Domingo and John Shawe-Taylor.

LSI-95-23-R "Approximating the permanent is in RNC", J. Díaz, M. Serna, and P. Spirakis.

LSI-95-24-R "Constraint satisfaction as global optimization", Pedro Meseguer and Javier Larrosa.

LSI-95-25-R "A rule-constructive geometric constraint solver", R. Joan-Arinyo and Antoni Soto.

LSI-95-26-R "External schemas in object oriented databases" (written in Spanish), José Samos.

LSI-95-27-R "An approach to belief in strong Kleene logic", Gustavo Núñez and Matías Alvarado.

LSI-95-28-R "Dynamic belief modeling", Antonio Moreno and Ton Sales.

LSI-95-29-R "Proceedings ICLP Workshop on Deductive Databases and Abduction", to appear. H. Decker, U. Geske, A. Kakas, C. Sakama, D. Seipel, and T. Urpí (editors).

LSI-95-30-R "Difference lists and difference bags for logic programming of categorial deduction", F. Xavier Lloré and Glyn Morrill.

LSI-95-31-R "An experiment on automatic semantic tagging of dictionary senses", German Rigau.

LSI-95-32-R "Slices of meaning: a study on nouns of portions", Salvador Climent and Maria Antònia Martí.

LSI-95-33-R "An HPSG grammar of Spanish implemented in ALE: A solution for dealing with subcategorisation alternances", Salvador Climent and Xavier Farreres.

LSI-95-34-R "Optimal and nearly optimal static weighted skip lists", Conrado Martínez and Salvador Roura.

LSI-95-35-R "Evaluation of expressions in a multiparadigm framework", Xavier Burgués and Xavier Franch.

LSI-95-36-R "Equivalence between executable OOZE and algebraic specification", Vicent-Ramon Palasí Lallana.

LSI-95-37-R "Automatic deduction of the behavioral equivalence between two algebraic specifications", Vicent-Ramon Palasí Lallana.

LSI-95-38-R "A method for explaining the behaviour of conceptual models", Antoni Olivé and Maria-Ribera Sancho.

LSI-95-39-R "Simple PAC learning of simple decision lists", Jorge Castro and José L. Balcázar.

LSI-95-40-R "Parallel approximation of MAX SNP and MAX NP problems", Maria Serna and Fatos Xhafa.

LSI-95-41-R "Automatically extracting translation links using a wide coverage semantic taxonomy", German Rigau, Horacio Rodriguez, and Jordi Turmo.

LSI-95-42-R "Belief increasing in SKL model frames", Matías Alvarado and Gustavo Núñez.

LSI-95-43-R "An incremental algorithm based on edge swapping for constructing restricted Delaunay triangulations", Marc Vigo Anglada.

LSI-95-44-R "Dealing with non-functional properties in an imperative programming language", Xavier Franch.

LSI-95-45-R "Universal computation in fluid neural networks", Ricard V. Solé and Jordi Delgado.

LSI-95-46-R "Phase transitions and complex systems", Ricard V. Solé, Susanna C. Manrubia, Bartolo Luque, Jordi Delgado, and Jordi Bascompte.

LSI-95-47-R "On the average complexity of some P-complete problems", Maria Serna and Fatos Xhafa.

LSI-95-48-R "An adaptive algorithm to compute the medial axis transform of 2-D polygonal domains", R. Joan-Arinyo, L. Pérez-Vidal, and E. Gargallo-Monllau.

LSI-95-49-R "Symbolic constraints in constructive geometric constraint solving", Christoph M. Hoffmann and Robert Joan-Arinyo.

LSI-95-50-R "Lambda extensions of rewrite orderings", Jean-Pierre Jouannaud and Albert Rubio.

LSI-95-51-R "Randomization of search trees by subtree size", Conrado Martínez and Salvador Roura.

LSI-95-52-R "Succinct circuit representations and leaf languages are basically the same concept", Bernd Borchert and Antoni Lozano.

LSI-95-53-R "Generalising discontinuity", Glyn Morrill and Josep-Maria Merenciano.

LSI-95-54-R "Concurrent AVL revisited: self-balancing distributed search trees", Luc Bougé, Joaquim Gabarró, and Xavier Messeguer.

LSI-95-55-R "A co-semidecision procedure for behavioral equivalence", Vicent-Ramon Palasí Lallana.

LSI-95-56-R "Reduction of behavioral equivalence to inductive theorems", Vicent-Ramon Palasí Lallana.

LSI-95-57-R "Where should concurrent rotations take place to rebalance a distributed arbitrary search tree?", Luc Bougé, Joaquim Gabarró and Xavier Messeguer.

LSI-95-58-R "Approximating convex quadratic programming is P-complete", Maria Serna and Fatos Xhafa.

---

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, http://www-lsi.upc.es/www/