Duality Gap, Computational Complexity and NP Completeness: A Survey

Prabhu Manyem
Department of Mathematics
Shanghai University
Shanghai 200444, China

December 30, 2010

Abstract

In this paper, we survey research that studies the connection between the computational complexity of optimization problems on the one hand, and the duality gap between the primal and dual optimization problems on the other. We further look at a similar phenomenon in finite model theory relating to complexity.

1 Introduction

In optimization problems, the **duality gap** is the difference between the optimal solution values of the primal problem and the dual problem. The relationship between the duality gap and the computational complexity of optimization problems has been implicitly studied for the last few decades. The connection between the two phenomenon has been subtly acknowledged. The *gap* has been exploited to design good approximation algorithms for NP-hard optimization problems [1, 9, 16]. However, we have been unable to locate a single piece of literature that addresses this issue explicitly.

This report is an attempt to bring a great deal of evidence together and specifically address this issue. Does the existence of polynomial time algorithms for the primal and the dual problems mean that the duality gap is zero? Conversely, does the existence of a duality gap imply that either the primal problem or the dual problem is (or both are) NP-hard? Is there an inherent connection between computational complexity and *strong duality* (that is, zero duality gap)?

1.1 Motivation

The apparent connection between the duality gap and computational complexity was considered more than thirty years ago. Linear Programming (LP) is a well known optimization problem. In the mid 1970s, before Khachiyan published his ellipsoid algorithm, LP was thought to be polynomially solvable precisely because, it obeys *strong duality*; that is, the duality gap is zero. For a good description of the ellipsoid algorithm, the reader is referred to the good book by Fang and Puthenpura [5]. Strong duality also places the decision version of Linear Programming in the class NP $\cap$ CoNP; see Lemma 7 below.

We now provide a definition for decision problems, also known as the decision versions of optimization problems:

Definition 1. *($\mathsf{D}_1(\mathsf{r})$: Decision problem corresponding to a given minimization problem)*

Given. *An objective function $f(\mathbf{x})$, as well as m_1 number of constraints $\mathbf{g}(\mathbf{x}) = \mathbf{b}$ and m_2 number of constraints $\mathbf{h}(\mathbf{x}) \geq \mathbf{c}$, where $\mathbf{x} \in \mathbb{R}^n$ is a vector of variables, $\mathbf{b} \in \mathbb{R}^{m_1}$ and $\mathbf{c} \in \mathbb{R}^{m_2}$ are constants. Also given is a parameter $r \in \mathbb{R}$.*

To Do. *Determine if there is a feasible solution $\boldsymbol{S}$ to the given set of constraints such that $f(\mathbf{x}) \leq r$.*

(Analogously, if the given problem is maximization, then the corresponding decision problem will be to determine if there is a feasible solution $\boldsymbol{S}$ such that $f(\mathbf{x}) \geq r$, when the constraints are $\mathbf{g}(\mathbf{x}) = \mathbf{b}$ and $\mathbf{h}(\mathbf{x}) \leq \mathbf{c}$.) ■

A word of caution: For decision problems, the term "feasibility" includes the constraint on the objective function; if the objective function constraint is violated, the problem becomes infeasible.

Let us start with the definition of the Lagrangian dual:

Definition 2. *[2] (Lagrangian Dual)*
Suppose we are given a minimization problem P_1 such as

$$\begin{aligned} \textit{Minimize} \quad & f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}, \\ \textit{subject to} \quad & \mathbf{g}(\mathbf{x}) = \mathbf{b}, \quad \mathbf{h}(\mathbf{x}) \geq \mathbf{c}, \\ \textit{where} \quad & \mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^{m_1} \textit{ and } \mathbf{c} \in \mathbb{R}^{m_2}. \end{aligned} \tag{1}$$

Let $\mathbf{u} \in \mathbb{R}^{m_1}$ and $\mathbf{v} \in \mathbb{R}^{m_2}$ be two vectors of variables with $\mathbf{v} \geq \mathbf{0}$.
The column vectors $\boldsymbol{b} = [b_1 \ b_2 \ \cdots \ b_{m_1}]^T$ and $\boldsymbol{c} = [c_1 \ c_2 \ \cdots \ c_{m_2}]^T$.

*For a given **primal** problem as in P_1, let us define the Lagrangian **dual** problem P_2:*

$$\begin{aligned} \textit{Maximize} \quad & \theta(\mathbf{u}, \mathbf{v}) \\ \textit{subject to} \quad & \mathbf{v} \geq \mathbf{0}, \\ \textit{where} \quad & \theta(\mathbf{u}, \mathbf{v}) = \inf_{\mathbf{x} \in \mathbb{R}^n} \ \{f(\mathbf{x}) + \textstyle\sum_{i=1}^{m_1} u_i(g_i(\mathbf{x}) - b_i) + \sum_{j=1}^{m_2} v_j(h_j(\mathbf{x}) - c_i)\}. \end{aligned} \tag{2}$$

■

Note that $g_i(\mathbf{x}) - b_i = 0$ $[h_j(\mathbf{x}) - c_j \geq 0]$ is the i^{th} equality $[j^{th}$ inequality$]$ constraint.

We now turn our attention to the relationship between the duality of an optimization problem, and membership in the complexity classes NP and CoNP of the corresponding set of decision problems. Decision problems are those with yes/no answers, as opposed to optimization problems that return an optimal solution (if a feasible solution exists).

2 Background: Duality and the classes NP and CoNP

Corresponding to P_1 defined above in (1), there is a set D_1 of decision problems, defined as $D_1 = \{D_1(r) \mid r \in \mathbb{R}\}$. The definition of $D_1(r)$ was provided in Definition 1.

Let us now define the computational classes NP, CoNP and P. For more details, the reader is referred to either [1] or [14].

Definition 3. *It is well known that **NP** (respectively **P**) is the class of decision problems for which there exist non-deterministic (respectively deterministic) Turing machines which provide Yes/No answers in time that is polynomial in the size of the input instance. In particular, for problems in P and NP, if the answer is **yes**, the Turing machine (TM) is able to provide an "evidence" (in technical terms called a* certificate*), such as a feasible solution to a given instance.*

*The class **CoNP** of decision problems is similar to NP, except for one key difference: the TM is able to provide a certificate only for **no** answers.*

From the above, it follows that for an instance of a problem in NP ∩ CoNP, the corresponding Turing machine can provide a certificate for both yes and no instances. ■

For example, if $D_1(r)$ in Definition 1 above is in NP, the certificate will be a feasible solution; that is, an $\mathbf{x} \in \mathbb{R}^n$ which obeys the constraints

$$\mathbf{g}(\mathbf{x}) = \mathbf{b},\ \mathbf{h}(\mathbf{x}) \leq \mathbf{c} \text{ and } f(\mathbf{x}) \geq r. \tag{3}$$

On the other hand, if $D_1(r) \in$ CoNP, the certificate will be an $\mathbf{x} \in \mathbb{R}^n$ that violates at least one of the $m_1 + m_2 + 1$ constraints in (3).

Remark 4. *For problems in NP, for Yes instances, extracting a solution from the certificate is not always an efficient (polynomial time) task. Similarly, in the case of CoNP, pinpointing a violation from a Turing machine certificate[1] is not guaranteed to be efficient either.*

Remark 5. *P ⊆ NP, because any computation that can be carried out by a deterministic TM can also be carried out by a non-deterministic TM. The problems in P are decidable deterministically in polynomial time.*

The class P is the same as its complement Co-P. That is, P is closed under complementation.

Furthermore, Co-P (≡ P) is a subset of CoNP. We know that P is a subset of NP. Hence P ⊆ NP ∩ CoNP. Thus for an instance of a problem in P, the corresponding Turing machine can provide a certificate for both yes and no instances. ■

We are now ready to define what is meant by a *tight dual*, and how it relates to the intersection class of problems, NP ∩ CoNP. Note that for two problems to be tight duals, it is sufficient if they are tight with respect to just one type of duality (such as Lagrangian duality, for example).

Definition 6. *We say that two optimization problems P_a and P_b are **dual to each other** if the dual of one problem is the other.*

[1]We thank WenXun Xing and PingKe Li for ponting out the above.

Suppose P_a and P_b are dual to each other, with zero duality gap; that is, P_a and P_b are tight duals. For any $r \in \mathbb{R}$, let $D_a(r)$ and $D_b(r)$ be the decision versions of P_a and P_b respectively.

*Let **TD** be the class of all decision problems whose optimization versions have tight duals. That is, **TD** is the set of all problems $D_a(r)$ and $D_b(r)$ for any $r \in \mathbb{R}$.* ■

One way in which duality gaps are related to the classes NP and CoNP is as follows:

Lemma 7. *[14] TD $\subseteq$ NP $\cap$ CoNP.*

From Remark 5 and Lemma 7, we know that both TD and P are subsets of NP $\cap$ CoNP. But is there a containment relationship between TD and P? That is, is either TD $\subseteq$ P or P $\subseteq$ TD? This is the subject of further study in this paper, with particular reference to Lagrangian duality.

Remark 8. *We should mention that in several cases, given a primal problem P, even if we are able to find a dual problem D such that the dual of P is D, it does not necessarily follow that the dual of D is P. That is,the dual of the dual need not be the primal. P and D are not necessarily duals of each other. We do not include such (P, D) pairs in **TD**. Among primal-dual pairs of problems, TD is a restricted class.*

3 Lack of Strong Duality results in NP hardness

In this section, we will review results from the literature, which show that the lack of strong duality imply that the optimization problem in question is NP-hard, assuming that the primal problem obeys the *constraint qualification* assumption as stated below in Definition 11. Here we work with Lagrangian duality. Results for other types of duality such as Fenchel, geometric and canonical dualities require further investigation.

Let us define what we mean by weak duality (as opposed to tight duality or strong duality):

Definition 9. *Given a primal problem P_1 and a dual problem P_2, as defined in Definition 2, the pair (P_1, P_2) is said to obey **weak duality** if the following condition is satisfied for any feasible solution $\boldsymbol{x}$ to the primal and any feasible solution ($\mathbf{u}$, $\mathbf{v}$) to the dual:*

$$\theta(\mathbf{u}, \mathbf{v}) \leq f(\mathbf{x}). \tag{4}$$

In Definition (9) above, if the inequality is replaced by an equality, then the primal and dual problems are said to be obey **strong duality**.

The following theorem from [2] guarantees that the feasible solutions to Lagrangian dual problems (1) and (2) indeed obey *weak duality*:

Theorem 10. *If $\boldsymbol{x}$ is a feasible solution to the primal problem in (1) and ($\mathbf{u}$, $\mathbf{v}$) is a feasible solution to the dual problem in (2), then $f(\mathbf{x}) \geq \theta(\mathbf{u}, \mathbf{v})$.* ■

We shall now define a special type of convex program, one with an assumption about the existence of a feasible solution in the interior of the domain:

Definition 11. ***Convex program with constraint qualification***.
Given. *A convex set* $X \subset \mathbb{R}^n$*, two convex functions* $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ *and* $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{m_1}$*, as well as an affine function* $\mathbf{h}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{m_2}$*.*

Constraint qualification assumption. *There is an* $\mathbf{x_0} \in X$ *such that* $\mathbf{g}(\mathbf{x_0}) < \mathbf{0}$*,* $\mathbf{h}(\mathbf{x_0}) = \mathbf{0}$*, and* $\mathbf{0} \in int\, \mathbf{h}(X)$*, where* $\mathbf{h}(X) = \bigcap_{\mathbf{x} \in X} h(\mathbf{x})$*.*

To do. *Minimize* $f(\mathbf{x})$*, subject to*
$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$*,* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ *and* $\mathbf{x} \in X$*.*

For the remainder of this section, we will assume primal constraint qualification; that is, we assume that constraint qualification is applied to the primal optimization problem. The following theorem provides sufficient conditions under which strong duality can occur:

Theorem 12. ***Strong Duality*** *[2]. If (i) the primal problem is given as in Definition 11, and (ii) the primal and dual problems have feasible solutions, then the primal and dual optimal solution values are equal (that is, the duality gap is zero):*

$$\inf\{f(\mathbf{x}) : \mathbf{x} \in X,\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0},\ \mathbf{h}(\mathbf{x}) = \mathbf{0}\} = \sup\{\theta(\mathbf{u}, \mathbf{v}) : \mathbf{v} \geq \mathbf{0}\},$$

where $\theta(\mathbf{u}, \mathbf{v})$*, the dual objective function, is equal to* $\inf_{\mathbf{x} \in X} \{f(\mathbf{x}) + \sum_{i=1}^{m_1} u_i g_i(\mathbf{x}) + \sum_{j=1}^{m_2} v_j h_j(\mathbf{x})\}$*.*

Using the contrapositive statement of Theorem 12, we get the following result:

Corollary 13. *(to Theorem 12) If there exists a duality gap using Lagrangian duals, then either the primal or the dual is not a convex optimization problem. (Remember, we are assuming constraint qualification.)*

The *Subset Sum* problem is defined as follows: Given a set S of positive integers $\{d_1, d_2, \cdots, d_k\}$ and another positive integer d_0, is there a subset P of S, such that the sum of the integers in P equals d_0?

Using a polynomial time reduction from the *Subset Sum* problem to a non-convex optimization problem, Murty and Kabadi (1987) showed the following:

Theorem 14. *[12] If an optimization problem is non-convex, it is NP-hard.*

(The converse is not true. A convex optimization problem in general is NP-hard, for example, Standard Quadratic Programming. See [11].)

From Corollary 13 and Theorem 14, it follows that

Theorem 15. *Assuming constraint qualification, if there exists a duality gap using Lagrangian duals, then either the primal or the dual is NP-hard.*

These results are true for Lagrangian duality. For other types of duality such as Fenchel, geometric and canonical dualities, this requires further investigation.

We should mention that there are quite a few papers in the literature on *convexification*; that is, in some cases, an original non-convex formulation has been shown to have an equivalent convex formulation, which sometimes makes the problem amenable to polynomial time solvability. Some authors call this “hidden convexity”; see [3] and [4].

4 Does Strong Duality Imply Polynomial Time Solvability?

At this time, such a proof (of whether a duality gap of zero implies polynomial time solvability of the primal and the dual problems) appears possible only for very simple problems, since estimating the duality gap appears extremely challenging for many problems.

Some of the problems where this is true include Convex Programming (and in particular, Linear Programming), mainly by using Interior Point algorithms. For example, see the book by Nesterov and Nemirovskii [13].

More investigation is needed to answer to this question in general.

However, there have been some results recently using Canonical duality for certain types of quadratic programs [6, 17]. Consider a standard quadratic programming (primal) problem:

$$\begin{array}{rl} \text{Maximize} & P(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}_0^T\mathbf{x} \\ \text{subject to} & \mathbf{b}_i^T\mathbf{x} \le c_i,\ 1 \le i \le m, \end{array} \tag{5}$$

where $A = A^T \in \mathbb{R}^{n\times n}$, $\mathbf{b}_i \in \mathbb{R}^n$ for $1 \le i \le m$, and $\mathbf{x} \in \mathbb{R}^n$. A is a symmetric matrix. We can write the Lagrangian function as

$$L(\mathbf{x},\lambda) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \left(\mathbf{b}_0 + \sum_{i=1}^{m}\lambda_i\mathbf{b}_i\right)^T \mathbf{x} - \sum_{i=1}^{m}\lambda_i c_i,\ \ \lambda_i \ge 0. \tag{6}$$

The first order necessary condition among the Karush-Kuhn-Tucker (KKT) conditions yields $A\mathbf{x} + \mathbf{b}_0 + \sum_{i=1}^{m}\lambda_i\mathbf{b}_i = 0$, from which we get the value of $\mathbf{x}$ as

$$\mathbf{x} = -A^{-1}\left(\mathbf{b}_0 + \sum_{i=1}^{m}\lambda_i\mathbf{b}_i\right), \tag{7}$$

assuming that A is an invertible matrix. Substituting this value of $\mathbf{x}$ back into the Lagrangian function yields

$$Q(\lambda) = -\frac{1}{2}\left(\mathbf{b}_0 + \sum_{i=1}^{m}\lambda_i\mathbf{b}_i\right)^T A^{-1}\left(\mathbf{b}_0 + \sum_{i=1}^{m}\lambda_i\mathbf{b}_i\right) - \sum_{i=1}^{m}\lambda_i c_i. \tag{8}$$

We then get a dual problem, the canonical dual P^d, as follows:

$$\begin{array}{rl} \text{Maximize} & Q(\lambda), \\ \text{subject to} & A \succ 0 \text{ and } \lambda \ge 0. \end{array} \tag{9}$$

(The relation $A \succ 0$ means the matrix A should be positive definite.)

Now, $Q(\lambda)$ is a concave function, to be maximized in the dual variable λ; hence P^d can be solved efficiently, since we are maximizing a concave function. Thus in general, we can get a lower bound for the primal problem quickly; and in some cases, we can get a strong dual with zero duality gap, which provides an optimal solution for the primal problem in polynomial time.

4.1 Preliminary Results: Canonical Duality and the Complexity Classes NP and CoNP

We now turn our attention to the relationship between the duality of an optimization problem, and membership in the complexity classes NP (and CoNP) of the corresponding set of decision problems. Recall that decision problems are those with Yes/No answers.

We consider quadratic programming[2] problems with a single quadratic constraint [17]. The primal problem P_0 is given by:

$$\begin{aligned} \text{Minimize} \quad & P(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{f}_0^T\mathbf{x} \\ \text{subject to} \quad & \tfrac{1}{2}\mathbf{x}^T B\mathbf{x} \le \mu, \end{aligned} \tag{10}$$

where A and B are non-zero $n \times n$ symmetric matrices, $\mathbf{f} \in \mathbf{R}^n$, and $\mu \in \mathbf{R}$. (A, B, $\mathbf{f}$ and μ are given.) Corresponding to the primal P_0, the canonical dual problem P^d is as follows:

$$\begin{aligned} \text{Maximize} \quad & P^d(\sigma) = -\tfrac{1}{2}\mathbf{f}^T(A+\sigma B)^{-1}\mathbf{f} - \mu\sigma \\ \text{subject to} \quad & \sigma \in \mathcal{F} = \{\sigma \ge 0 \mid A + \sigma B \succ 0\}, \end{aligned} \tag{11}$$

assuming that $\mathcal{F}$ is non-empty. The following theorem appeared in [17]:

Theorem 16. *(Strong duality theorem) When the maximum value $P^d(\sigma^*)$ in (11) is finite, strong duality between the primal problem P_0 and the dual problem P^d holds, and the optimal solution for P_0 is given by $x^* = (A + \sigma^* B)^{-1}\mathbf{f}$.*

That is, P_0 and P^d are tight duals. Combining this result with Lemma 7 above, we can conclude that

Remark 17. *Decision versions of Problems P_0 in (1) and P^d in (6) are members of the intersection class NP $\cap$ CoNP.*

Furthermore, from Theorem 3 in [17], it is easy to see that problems P_0 and P_d can be solved in polynomial time. The authors in [17] use what they call a "boundarification" technique which moves the analytical solution $\bar{x}$ to a global minimizer x^* on the boundary of the primal domain. This strengthens the conjecture that

Conjecture 18. *If two optimization problems P_a and P_b are such that one of them is the dual of the other, that is, they exhibit strong duality, then both are polynomially solvable.*

The observations above are for quadratic programming problems with a single quadratic constraint. It would be interesting to see what happens for quadratic programming problems with two constraints, whether strong duality still holds, and whether both the primal and the dual are still polynomially solvable.

Ramana [15] exhibited strong duality for the semidefinite programming (SDP) problem. However, the complexity of SDP is unknown; the author in [15] showed that the decision version of SDP is NP-complete if and only if NP = CoNP.

[2]Thanks to Shu-Cherng Fang for his input here.

5 Descriptive Complexity and Fixed Points

On a final note, we would like to briefly describe a similar phenomenon which occurs in the field of Descriptive Complexity, which is the application of Finite Model theory to computational complexity. In particular, we would like to mention least fixed point (LFP) computation. A full description would be beyond the scope of this paper. However, we would like to briefly mention a few related concepts and phenomenon.

For a good description of least fixed points (LFP) in existential second order (ESO) logic, the reader is referred to [7] (chapters 2 and 3) and [8]. If the input structures are ordered, then expressions in LFP logic can describe polynomial time (PTIME) computation [7].

The input instance to an LFP computation consists of a *structure* $\mathbf{A}$, which includes a domain set A and a set of (first order) relations R_i, each with arity r_i, $1 \leq i \leq J$. The LFP computation works by a stagewise addition of tuples from A, to a new relation P (of some arity k). If P_i represents the relation (set of tuples) after stage i, then $P_i \subseteq P_{i+1}$. The transition from P_i to P_{i+1} is through an operator Φ, such that $P_{i+1} = \Phi(P_i)$. At the beginning, P is empty, that is, $P_0 = \emptyset$. For some value of i, say when $i = f$, if $P_f = P_{f+1}$, a *fixed point* has been reached.

Without going into details, let us just say that such a fixed point, reached as above, is also a *least* fixed point (LFP) if the operator Φ can be chosen in a particular manner. The interested reader is referred to [7] (chapter 2) for details.

Note that the number of elements in P can be at most $|A|^k$ (where $|A|$ is the number of elements in A), which is polynomial in the size of the domain. Hence $f \leq |A|^k$, so an LFP is achieved within a polynomial number of stages.

Similar to LFP, we can also define a *greatest* fixed point (GFP). This is obtained by doing the reverse; we start with the entire set A^k of k-ary tuples from the universe A, and then removing tuples from P in stages. At the beginning, $P_0 = A^k$. In further stages, $P_i \supset P_{i+1}$. The GFP is reached at stage g if $P_g = P_{g+1}$.

The logic that includes LFP and GFP expressions is known as **LFP logic**. It expresses decision problems (those with a Yes/No answer), such as those in Definitions 1 and 3. To be feasible, a solution should also obey the objective function constraint ($f(\mathbf{x}) \geq K$ or $f(\mathbf{x}) \leq K$).

The LFP computation expresses decision problems based on maximization. Before the fixed point is reached, the solution is infeasible; that is, the number of tuples in the fixed point relation P is insufficient. However, once the fixed point is reached, the solution becomes feasible. Similarly, the GFP computation expresses decision problems based on minimization.

Problem. An interesting problem arising in LFP Logic is this: For what type of primal-dual optimization problem pairs will the LFP and GFP computation meet at the same fixed point? Does this mean that such a pair is polynomially solvable?

6 Conclusion and Further Study

In this paper, we have touched the tip of the iceberg on a very interesting problem, that of connecting the computational hardness of an optimization problem with its duality characteristics. A lot more study is required in this area.

Another issue is that of *saddle point* for Lagrangian duals. This is a decidable problem; we can do brute force and find the primal and dual optimal solutions; this will tell us if there is a duality gap. If the gap is zero, then there is a saddle point.

(Jeroslow [10] showed that the integer programming problem with quadratic constraints is undecidable if the number of variables is unbounded, which is an extreme condition. However, if each variable has a finite upper and lower bound, then the number of solutions is finite and thus it is possible to determine the best solution in finite time.)

However, this problem would be NP-complete, unless we can tell whether it has a saddle point by looking at the structure of the problem or by running a polynomial time algorithm.

We hope that this paper would motivate further research into this interesting topic.

7 Acknowledgments

This research was supported by (i) a visiting fellowship from the National Cheng Kung University (NCKU), Tainan, Taiwan; (ii) the Shanghai Leading Academic Discipline Project #S30104; (iii) the National Natural Science Foundation of China (# 11071158). Support from all sources is gratefully acknowledged.

In addition, I would like to thank Wen-Xun Xing and Ping-Ke Li (Tsinghua University, Beijing) and Shu-Cherng Fang (North Carolina State University, USA) for very helpful comments.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Combinatorial Optimization Problems and Their Approximability Properties.* Springer (Germany), 1999.

[2] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Non-Linear Programming: Theory and Algorithms (third edition).* Wiley Interscience, 2006.

[3] Amir Beck. Convexity Properties Associated with Nonconvex Quadratic Matrix Functions and Applications to Quadratic Programming. *Journal of Optimization Theory and Applications*, 142:1–29, 2009.

[4] Aharon Ben-Tal and Marc Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathemalical Programming*, 72:51–63, 1996.

[5] S-C. Fang and S. Puthenpura. *Linear Optimization and Extensions: Theory and Algorithms.* Prentice Hall (New Jersey), 1993.

[6] David Y. Gao. Canonical duality theory and solutions to constrained nonconvex quadratic programming. *Journal of Global Optimization*, 29(4):377–399, 2004.

[7] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications.* Texts in Theoretical Computer Science. Springer, 2007.

[8] Heinz D. Ebbinghaus and Jörg Flum. *Finite Model Theory.* Springer, 1999.

[9] Dorit Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, MA, 1997.

[10] R. G. Jeroslow. There Cannot be any Algorithm for Integer Programming with Quadratic Constraints. *Operations Research*, 21:221–224, 1973.

[11] Mirjam Dür. Copositive Programming — a Survey, November 2009. Available at `http://www.optimization-online.org/DB_HTML/2009/11/2464.html`.

[12] Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[13] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications.* Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[14] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley (Reading, Massachusetts), 1994.

[15] Motakuri V. Ramana. An Exact Duality Theory for Semidefinite Programming and Its Complexity Implications. *Mathematical Programming*, 77:129–162, 1997.

[16] Vijay V. Vazirani. *Approximation Algorithms.* Springer (Germany), 2001.

[17] W. Xing, S-C. Fang, D.Y. Gao, R-L. Sheu, and L. Zhang. Canonical Dual Solutions to the Quadratic Programming over a Quadratic Constraint. Technical report, Department of Mathematical Sciences, Tsinghua University, Beijing, China, 2008.