

Article



Doubly Nonnegative and Semidefinite Relaxations for the Densest *k*-Subgraph Problem

Chuan-Hao Guo * , Yuan Guo and Bei-Bei Liu

School of Economics and Management, Zhejiang Sci-Tech University, Hangzhou 310018, China; 201820803002@mails.zstu.edu.cn (Y.G.); 201720803003@mails.zstu.edu.cn (B.-B.L.)

* Correspondence: guo-ch@zstu.edu.cn

Received: 19 October 2018; Accepted: 19 January 2019; Published: 24 January 2019



Abstract: The densest *k*-subgraph (DkS) maximization problem is to find a set of *k* vertices with maximum total weight of edges in the subgraph induced by this set. This problem is in general NP-hard. In this paper, two relaxation methods for solving the DkS problem are presented. One is doubly nonnegative relaxation, and the other is semidefinite relaxation with tighter relaxation compare with the relaxation of standard semidefinite. The two relaxation problems are equivalent under the suitable conditions. Moreover, the corresponding approximation ratios' results are given for these relaxation problems. Finally, some numerical examples are tested to show the comparison of these relaxation problems, and the numerical results show that the doubly nonnegative relaxation is more promising than the semidefinite relaxation for solving some DkS problems.

Keywords: densest *k*-subgraph; doubly nonnegative relaxation; semidefinite relaxation; approximation ratio; NP-hard

MSC: 90C10; 90C20; 49M20; 65K05

1. Introduction

In this paper, the densest *k*-subgraph (DkS) problem [1,2] is considered. For a given graph *G* and a parameter *k*, the DkS problem consists in finding a maximal average degree in the subgraph induced by the set of *k* vertices. This problem was first introduced by Corneil and Perl as a natural generalization of the maximum clique problem [3]. It is NP-hard on restricted graph classes such as chordal graphs [3], bipartite graphs [3] and planar graphs [4]. The DkS problem is a classical problem of combinatorial optimization and arises in several applications, such as facility location [5], community detection in social networks, identifying protein families and molecular complexes in protein–protein interaction networks [6], etc. Since the DkS problem is in general NP-hard, there are a few approximation methods [7–9] for solving it. It is well-known that semidefinite relaxation is a powerful and computationally efficient approximation technique for solving a host of very difficult optimization problems, for instance, the max-cut problem [10] and the boolean quadratic programming problem [11]. It also has been at the center of some of the very exciting developments in the area of signal processing [12,13].

Optimization problems over the doubly nonnegative cone arise, for example, as a strengthening of the Lovasz- ϑ -number for approximating the largest clique in a graph [14]. The recent work by Burer [15] stimulated the interest in optimization problems over the completely positive cone. A tractable approximation to such problem being defined as an optimization problem over the doubly nonnegative cone. By using the technique of doubly nonnegative relaxation, Bai and Guo proposed an effective and promising method for solving multiple objective quadratic programming problems in [16]. For more details and developments of this technique, one may refer to [17–19] and the references

therein. It is worth pointing out that the cone of doubly nonnegative matrices is a subset of a positive semidefinite matrices cone. Thus, the doubly nonnegative relaxation is more promising than the basic semidefinite relaxation. Moreover, such relaxation problems can be efficiently solved by some popular package software.

In this paper, motivated by the idea of doubly nonnegative relaxation and semidefinite relaxation, the two relaxation methods for solving the DkS problem are presented. One is doubly nonnegative relaxation, and the other is semidefinite relaxation with tighter relaxation. Furthermore, we prove that the two relaxation problems are equivalent under the suitable conditions. Some approximation accuracy results about these relaxation problems are also given. Finally, we report some numerical examples to show the comparison of the two relaxation problems. The numerical results show that the doubly nonnegative relaxation is more promising than the semidefinite relaxation for solving some DkS problems.

The paper is organized as follows: we present doubly nonnegative relaxation and a new semidefinite relaxation with tighter relaxation for the DkS problem in Sections 2.1 and 2.2, respectively. In Section 3, we prove that the two new relaxations proposed in Section 2 are equivalent. In Section 4, some approximation accuracy results for the proposed relaxation problems are given. Some comparative numerical results are reported in Section 5 to show the efficiency of the proposed new relaxations. Moreover, some concluding remarks are given in Section 6.

2. Two Relaxations for the Densest k-Subgraph Problem

First of all, the definition of the densest *k*-subgraph (DkS) problem is given as follows.

Definition 1 (Densest k-subgraph). For a given graph G(V, E), where V is the vertex set and E is the edge set. The DkS problem on G(V, E) is the problem of finding a vertex subset of V of size k with the maximum induced average degree.

Given a symmetric $n \times n$ matrix $A = (a_{ij})$, the weighted graph with vertex set $\{1, 2, ..., n\}$ associates with A in such a way: the edge [i, j] with the weight a_{ij} is introduced in the graph. Then, A is interpreted as the weighted adjacency matrix of the graph with the vertex set $V = \{1, 2, ..., n\}$. Based on Definition 1, the DkS problem consists of determining a subset $V_1 \subseteq V$ consisting of k vertices such that the total weight of edges in the subgraph spanned by V_1 is maximized. To select subgraphs, assign a decision variable $y_i \in \{0, 1\}$ for each node $(y_i = 1 \text{ if the node is taken, and } y_i = 0 \text{ if the node is not})$. The weight of the subgraph given by y is $y^T Ay$. Thus, the DkS problem can be phrased as the 0 - 1 quadratic problem

(DkS)
$$\max \begin{array}{l} \frac{1}{2}y^T Ay, \\ \text{s.t.} \quad y^T e = k, \\ y \in \{0, 1\}^n \end{array}$$

It is known that the (DkS) problem is NP-hard [20], even though *A* is assumed to be positive semidefinite, since the feasible space of the (DkS) problem is nonconvex. For solving this problem efficiently, we present the two new relaxations for the (DkS) problem in the following subsections, based on the idea of approximation methods.

2.1. Doubly Nonnegative Relaxation

Note that the quadratic term $y^T A y$ in the (DkS) problem can also be expressed as $A \bullet y y^T$. By introducing a new variable $Y = y y^T$ and taking lifting techniques, we could reformulate the (DkS) problem into the following completely positive programming problem:

$$(CPP_{DkS}) \qquad \begin{array}{l} \max \quad \frac{1}{2}A \bullet Y, \\ \text{s.t.} \quad y^T e = k, \\ e^T Y e = k^2, \\ \text{diag}(Y) = y, \\ \left[\begin{array}{c} 1 \quad y^T \\ y \quad Y\end{array}\right] \in \mathbb{C}_{1+n}, \end{array}$$

where \mathbb{C}_{1+n} is defined as follows:

$$\mathbb{C}_{1+n} := \left\{ X \in \mathbb{S}_{1+n} : \ X = \sum_{h \in H} z^h (z^h)^T \right\} \cup \{0\},$$

and for some finite vectors $\{z^h\}_{h\in H} \subset \mathbb{R}^+_{1+n} \setminus \{0\}$.

The following theorem shows the relationship between the (DkS) problem and the (CPP_{DkS}) problem. Its proof is similar to the one of Theorem 2.6 in [15] and is omitted here.

Theorem 1. (*i*) The (DkS) problem and the (CPP_{DkS}) problem have the same optimal values of objective functions, i.e., $Opt(DkS) = Opt(CPP_{DkS})$; (*ii*) if (y^*, Y^*) is an optimal solution for the (CPP_{DkS}) problem; then, y^* is in the convex hull of optimal solutions for the (DkS) problem.

On one hand, according to Definition 2, it is obviously that the (CPP_{DkS}) problem is equivalent to the (DkS) problem. On the other hand, in view of the definition of convex cone, \mathbb{C}_{1+n} is a closed convex cone, and is called completely positive matrices cone. Thus, the (CPP_{DkS}) problem is convex. However, since checking whether or not a given matrix belongs to \mathbb{C}_{1+n} is NP-hard, which has been shown by Dickinson and Gijen in [21], the (CPP_{DkS}) problem is still NP-hard. Thus, \mathbb{C}_{1+n} has to be replaced or approximated by some computable cones. For example, \mathbb{R}_n^+ and \mathbb{S}_n^+ are both computable cones; furthermore, \mathbb{N}_n^+ is also a computable cone.

It is worth mentioning that Diananda's decomposition theorem [22] can be reformulated as follows, and its proof can be found in it.

Theorem 2. $\mathbb{C}_n \subseteq \mathbb{S}_n^+ \cap \mathbb{N}_n^+$ holds for all n. If $n \leq 4$, then $\mathbb{C}_n = \mathbb{S}_n^+ \cap \mathbb{N}_n^+$.

The matrices cone $\mathbb{S}_n^+ \cap \mathbb{N}_n^+$ is sometimes called "doubly nonnegative matrices cone". Of course, in dimension $n \ge 5$, there are matrices which are doubly nonnegative but not completely positive, the counterexample can be seen in [23].

By using Theorem 2, the (CPP_{DkS}) problem can be relaxed to the problem

$$(\text{DNNP}_{\text{DkS}}) \qquad \begin{array}{l} \max & \frac{1}{2}A \bullet Y, \\ \text{s.t.} & y^T e = k, \\ & e^T Y e = k^2, \\ & \text{diag}(Y) = y, \\ & \left[\begin{array}{c} 1 & y^T \\ y & Y \end{array}\right] \in \mathbb{S}^+_{1+n} \cap \mathbb{N}^+_{1+n'} \end{array}$$

which is called the doubly nonnegative relaxation for the (DkS) problem. Some explanations are given below for this relaxation problem.

Remark 1. Obviously, the (DNNP_{DkS}) problem has a linear objective function and the linear constraints as well as a convex conic constraint, so it is a linear conic programming problem. Meanwhile, it is notable that $\mathbb{S}_{1+n}^+ \cap \mathbb{N}_{1+n}^+ \subseteq \mathbb{S}_{1+n}^+$ and the types of variables in both the sets are the same, which further implies that the (DNNP_{DkS}) problem could be solved by some popular package softwares for solving semidefinite programs.

2.2. New Semidefinite Relaxation

It is well-known that semidefinite relaxation is a powerful approximation technique for solving a host of combinatorial optimization problems. In this subsection, we present a new semidefinite relaxation with tighter bound for the (DkS) problem.

The idea of the standard lifting is to introduce the symmetric matrix of rank one $Y = yy^T$. With the help of *Y*, we could express the integer constraints $y_i \in \{0, 1\}$ as $Y_{ii} = y_i$, and the quadratic objective function $y^T A y$ as $A \bullet Y$. Thus, we can get the following equivalent formulation of the (DkS) problem

max
$$\frac{1}{2}A \bullet Y$$
,
s.t. $y^T e = k$,
 $e^T Y e = k^2$,
 $Y = yy^T$,
diag $(Y) = y$,
rank $(Y) = 1$.

Notice then that the hard constraint in the above problem is the constraint rank(Y) = 1, which is moreover difficult to handle. Thus, we can relax the above problem to the following standard semidefinite relaxation problem by dropping the rank-one constraint

$$(I - SDR_{DkS}) \qquad \begin{array}{l} \max & \frac{1}{2}A \bullet Y, \\ \text{s.t.} & y^T e = k, \\ & e^T Y e = k^2, \\ & \text{diag}(Y) = y, \\ & \left[\begin{array}{c} 1 & y^T \\ y & Y \end{array}\right] \in \mathbb{S}^+_{1+n}. \end{array}$$

For the $(I - SDR_{DkS})$ problem, some remarks are given below.

Remark 2. (i) Obviously, the $(I - SDR_{DkS})$ problem is also a linear conic programming problem, it has the same objective function and the equality constraints with the $(DNNP_{DkS})$ problem. The only difference between the $(I - SDR_{DkS})$ problem and the $(DNNP_{DkS})$ problem is that the $(DNNP_{DkS})$ problem has $\frac{n(n+1)}{2} + n$ nonnegative constraints more than the $(I - SDR_{DkS})$ problem.

(ii) Since $\mathbb{S}_{1+n}^+ \cap \mathbb{N}_{1+n}^+ \subseteq \mathbb{S}_{1+n}^+$, it holds that $Opt(DNNP_{DkS}) \leq Opt(I - SDR_{DkS})$ from Remark 2 (i). Thus, the bound of the $(DNNP_{DkS})$ problem is not larger than the one of the $(I - SDR_{DkS})$ problem. In Section 5, we implement some numerical experiments to show the comparison between the $(I - SDR_{DkS})$ problem and the $(DNNP_{DkS})$ problem from the computational point of view.

Note that the (DkS) problem is inhomogeneous, but we can homogenize it as follows. First, let z = 2y - e in the (DkS) problem, it follows that $z \in \{-1, 1\}^n$. Thus, the change of variable $y \to z$ gives the following equivalent formulation of the (DkS) problem:

$$(\widehat{\text{DkS}}) \qquad \begin{array}{l} \max \quad \frac{1}{8}z^TAz + \frac{1}{4}z^TAe + \frac{1}{8}e^TAe,\\ \text{s.t.} \quad z^Te = 2k - n,\\ z \in \{-1,1\}^n. \end{array}$$

Entropy 2019, 21, 108

Then, with the introduction of the extra variable t, the (DkS) problem can be expressed as a homogeneous problem

$$\max \begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} \frac{1}{8}e^TAe & \frac{1}{8}e^TA \\ \frac{1}{8}Ae & \frac{1}{8}A \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix},$$

$$\overline{(DkS)} \qquad \text{s.t.} \begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{2}e^T \\ \frac{1}{2}e & \mathbf{0} \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix} = 2k - n,$$

$$\begin{bmatrix} t \\ z \end{bmatrix} \in \{-1, 1\}^{1+n},$$

where **0** is a zero matrix with appropriate dimension.

Remark 3. The (\overline{DkS}) problem is equivalent to the (\widehat{DkS}) problem in the following sense: if $\begin{bmatrix} t^* \\ z^* \end{bmatrix}$ is an optimal solution to the (\overline{DkS}) problem, then z^* (resp. $-z^*$) is an optimal solution to the (\widehat{DkS}) problem with $t^* = 1$ (resp. $t^* = -1$).

By using the standard semidefinite relaxation technique, and letting let $S = \begin{bmatrix} t \\ z \end{bmatrix} \begin{bmatrix} t & z^T \end{bmatrix}$, the $\overline{(DkS)}$ problem can be relaxed to the following problem:

$$\max \begin{bmatrix} \frac{1}{8}e^{T}Ae & \frac{1}{8}e^{T}A \\ \frac{1}{8}Ae & \frac{1}{8}A \end{bmatrix} \bullet S,$$

$$(SDR_{\overline{DkS}}) \qquad \text{s.t.} \begin{bmatrix} 0 & \frac{1}{2}e^{T} \\ \frac{1}{2}e & \mathbf{0} \\ \text{diag}(S) = e, \\ S \in \mathbb{S}_{1+n}^{+}.$$

Moreover, again by using the standard semidefinite relaxation technique directly to the (DkS) problem, we have from $Z = zz^T$,

$$(SDR_{\widehat{DkS}}) \qquad \begin{array}{l} \max \quad \frac{1}{8}A \bullet Z + \frac{1}{4}z^{T}Ae + \frac{1}{8}e^{T}Ae \\ \text{s.t.} \quad z^{T}e = 2k - n, \\ e^{T}Ze = (2k - n)^{2}, \\ \text{diag}(Z) = e, \\ \begin{bmatrix} 1 & z^{T} \\ z & Z \end{bmatrix} \in \mathbb{S}_{1+n}^{+}. \end{array}$$

The $(SDR_{\overline{DkS}})$ problem and the $(SDR_{\overline{DkS}})$ problem are both standard semidefinite relaxation problems for the (DkS) problem. The upshot of the formulations of these two relaxation problems is that they can be solved very conveniently and efficiently, to some arbitrary accuracy, by some readily available software packages, such as CVX. Note that there is only one difference between these two relaxation problems, i.e., the $(SDR_{\widehat{DkS}})$ problem has one equality constraint more than the $(SDR_{\overline{DkS}})$ problem. In Section 5, some comparative numerical results are reported to show the effectiveness of these two relaxations problems for solving some random (DkS) problems, respectively.

It is worth noting that the constraint $z \in \{-1, +1\}^n$ in the (DkS) problem further implies

$$(1+z_i)(1+z_j) \ge 0 \Rightarrow 1+z_i+z_j+Z_{ij} \ge 0, \ \forall 1 \le i \le j \le n$$
(1)

always holds. Thus, adding Formula (1) to the $(SDR_{\widehat{DkS}})$ problem, we come up with the following new semidefinite relaxation problem

$$\max \quad \frac{1}{8}A \bullet Z + \frac{1}{4}z^{T}Ae + \frac{1}{8}e^{T}Ae,$$

s.t. $z^{T}e = 2k - n,$
 $e^{T}Ze = (2k - n)^{2},$
(II - SDR_{DkS}) $\operatorname{diag}(Z) = e,$
 $1 + z_{i} + z_{j} + Z_{ij} \ge 0, \forall 1 \le i \le j \le n,$
 $\begin{bmatrix} 1 & z^{T} \\ z & Z \end{bmatrix} \in \mathbb{S}_{1+n}^{+}.$

Obviously, the relationship $Opt(II - SDR_{DkS}) \leq Opt(SDR_{\widehat{DkS}})$ holds since the feasible set of the $(II - SDR_{DkS})$ problem is the subset of the feasible set of the $(SDR_{\widehat{DkS}})$ problem and the two problems have the same objective function.

Up to now, three new semidefinite relaxation problems for the (DkS) problem are established, i.e., the $(SDR_{\overline{DkS}})$ problem, the $(SDR_{\overline{DkS}})$ problem and the $(II - SDR_{DkS})$ problem, in which the upper bound of the $(II - SDR_{DkS})$ problem is more promising than the one of the $(SDR_{\overline{DkS}})$ problem. In the following sections, we will further investigate the relationship between these three problems with the $(DNNP_{DkS})$ problem.

3. The Equivalence between the Relaxation Problems

The previous section establishes the doubly nonnegative relaxation (i.e., the (DNNP_{DkS}) problem) and the semidefinite relaxation with tighter bound (i.e., the (II – SDR_{DkS}) problem) for the (DkS) problem. Note that the (DNNP_{DkS}) problem has *n* inequality constraints more than the (II – SDR_{DkS}) problem. In this section, we will prove the equivalence between the two relaxations. First of all, the definition of the equivalence of two optimization problems is given as follows.

Definition 2. *Two problems P1 and P2 are called equivalent if they satisfy the following two conditions:*

(*i*) Opt(P1) = Opt(P2).
(*ii*) *If from a solution of P1, then a solution of P2 is readily found, and vice versa.*

In order to establish the equivalence for the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem, a crucial theorem is given below and the details of its proof can be seen in [24] (Appendix A.5.5).

Theorem 3 (Schur Complement). Let matrix $M \in \mathbb{S}^n$ is partitioned as

$$M = \left[\begin{array}{cc} D & B \\ B^T & C \end{array} \right].$$

If det $D \neq 0$, the matrix $H = C - B^T D^{-1}B$ is called the Schur complement of D in M. Then, the following relations hold:

(*i*) $M \succ \mathbf{0}$ *if and only if* $D \succ \mathbf{0}$ *and* $H \succ \mathbf{0}$. (*ii*) If $D \succ \mathbf{0}$, then $M \succeq \mathbf{0}$ if and only if $H \succeq \mathbf{0}$.

To the end, by using Definition 2 and Theorem 3, we have the following main equivalence theorem.

Theorem 4. Suppose that the feasible sets $\text{Fea}(\text{DNNP}_{\text{DkS}})$ and $\text{Fea}(\text{II} - \text{SDR}_{\text{DkS}})$ are both nonempty. Then, the $(\text{DNNP}_{\text{DkS}})$ problem and the $(\text{II} - \text{SDR}_{\text{DkS}})$ problem are equivalent.

Proof. First of all, we prove that $Opt(DNNP_{DkS}) \ge Opt((II - SDR_{DkS}))$.

Suppose that (z^*, Z^*) is an optimal solution of the $(II - SDR_{DkS})$ problem, and let

$$Y = \frac{1}{4}(ee^{T} + e(z^{*})^{T} + z^{*}e^{T} + Z^{*}), \ y = \frac{1}{2}(e + z^{*}).$$
⁽²⁾

Directly from $e^T z^* = 2k - n$ and Equation (2), we have

$$y^{T}e = \frac{1}{2}(e+z^{*})^{T}e = \frac{1}{2}(n+e^{T}z^{*}) = k.$$
(3)

By Equation (2) and $e^T Z^* e = (2k - n)^2$, it holds that

$$e^{T}Ye = \frac{1}{4}e^{T}(ee^{T} + e(z^{*})^{T} + (z^{*})e^{T} + Z^{*})e$$

= $\frac{1}{4}(n^{2} + 2n(z^{*})^{T}e + (2k - n)^{2})$
= k^{2} . (4)

Since diag(Z^*) = e, Equation (2) further implies that

diag(Y) =
$$\frac{1}{4}(e + 2z^* + \text{diag}(Z^*)) = \frac{1}{2}(e + z^*) = y.$$
 (5)

From $1 + z_i^* + z_j^* + Z_{ij}^* \ge 0$ for all $1 \le i \le j \le n$, it holds that

$$Y_{ij} \ge 0, \forall 1 \le i \le j \le n.$$
(6)

Combining with Equation (5), it is true that from Formula (6)

$$y_i \ge 0, \forall 1 \le i \le n. \tag{7}$$

By Theorem 3 (ii) and Equation (2), it follows that

$$\begin{array}{ll} Y - yy^T &= \frac{1}{4}(ee^T + e(z^*)^T + z^*e^T + Z^*) - \frac{1}{4}(e + z^*)(e + z^*)^T \\ &= \frac{1}{4}(Z^* - z^*(z^*)^T) \succeq \mathbf{0}, \end{array}$$

i.e.,

$$\begin{bmatrix} 1 & y^T \\ y & Y \end{bmatrix} \in \mathbb{S}^+_{1+n}.$$
(8)

From Equations (3), (4) and (6)–(8), we have (y, Y) defined by Equation (2) is a feasible solution of the (DNNP_{DkS}) problem. Moreover, again from Equation (2), we obtain

$$\begin{split} \frac{1}{2}A \bullet Y &= \frac{1}{8}A \bullet (ee^{T} + e(z^{*})^{T} + z^{*}e^{T} + Z^{*}) \\ &= \frac{1}{8}A \bullet Z^{*} + \frac{1}{4}(z^{*})^{T}Ae + \frac{1}{8}e^{T}Ae \\ &= \text{Opt}((\text{II} - \text{SDR}_{\text{DkS}})), \end{split}$$

i.e., $Opt(DNNP_{DkS}) \ge Opt((II - SDR_{DkS}))$.

Conversely, given an optimal solution (y^*, Y^*) to the (DNNP_{DkS}) problem, and let

$$Z_{ij} = 1 - 2y_i^* - 2y_j^* + 4Y_{ij}^*, \ z_i = 2y_i^* - 1, \ \forall 1 \le i \le j \le n.$$
(9)

Since $diag(Y^*) = y^*$, Equation (9) further implies that

$$diag(Z) = e - 4y^* + 4diag(Y^*) = e.$$
 (10)

Moreover,

$$1 + z_i + z_j + Z_{ij} = 1 + (2y_i^* - 1) + (2y_j^* - 1) + 1 - 2y_i^* - 2y_j^* + 4Y_{ij}^*$$

= $4Y_{ii}^* \ge 0, \ \forall 1 \le i \le j \le n.$ (11)

From Equation (9) and $e^T y^* = k$, it follows that

$$z^{T}e = (2y^{*} - e)^{T}e = 2(y^{*})^{T}e - n = 2k - n.$$
(12)

Again from Equation (9) and $e^T Y^* e = k^2$, it is true that

$$e^{T}Ze = e^{T}(ee^{T} - 2e(y^{*})^{T} - 2y^{*}e^{T} + 4Y^{*})e$$

= $n^{2} - 4kn + 4k^{2} = (2k - n)^{2}.$ (13)

From Equation (9) and Theorem 3 (ii), it holds that

$$Z - zz^{T} = ee^{T} - 2e(y^{*})^{T} - 2y^{*}e^{T} + 4Y^{*} - (2y^{*} - e)(2y^{*} - e)^{T}$$

= 4(Y^{*} - y^{*}(y^{*})^{T}) \succeq 0. (14)

By Equations (11)–(14), we can conclude that (z, Z) defined by Equation (9) is a feasible solution of the (II – SDR_{DkS}) problem. Furthermore, we have

$$\begin{array}{ll} \frac{1}{8}A \bullet Z &+ \frac{1}{4}z^{T}Ae + \frac{1}{8}e^{T}Ae \\ &= \frac{1}{8}A \bullet (ee^{T} - 2e(y^{*})^{T} - 2y^{*}e^{T} + 4Y^{*}) + \frac{1}{4}(2y^{*} - e)^{T}Ae + \frac{1}{8}e^{T}Ae \\ &= \frac{1}{2}A \bullet Y^{*} = \operatorname{Opt}(\mathrm{DNNP}_{\mathrm{DkS}}), \end{array}$$

i.e., $Opt(DNNP_{DkS}) \leq Opt(II - SDR_{DkS})$.

Summarizing the analysis above, we obtain $Opt(DNNP_{DkS}) = Opt(II - SDR_{DkS})$. From Equations (2) and (9), we observe that (y, Y) defined by Equation (2) is an optimal solution for the $(DNNP_{DkS})$ problem and (z, Z) defined by Equation (9) is also an optimal solution for the $(II - SDR_{DkS})$ problem, respectively. According to Definition 2, we conclude that the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem are equivalent. \Box

The above Theorem 4 shows that $Opt(DNNP_{DkS}) = Opt(II - SDR_{DkS})$. Note that the $(DNNP_{DkS})$ problem has *n* inequality constraints more than the $(II - SDR_{DkS})$ problem, thus the computational cost of solving $(DNNP_{DkS})$ problem may be greater than that of the $(II - SDR_{DkS})$ problem.

4. The Approximation Accuracy

The above section shows that the $(DNNP_{DkS})$ problem is equivalent to the $(II - SDR_{DkS})$ problem which has the tighter upper bound compared to the $(SDR_{\widehat{DkS}})$ problem (see Theorem 4). In this section, we further investigate the approximation accuracy of the $(DNNP_{DkS})$ problem for solving the (DkS) problem, comparing with the standard semidefinite relaxation problems which was proposed in the above sections, under some conditions.

To simplify the expression, we denote $\begin{bmatrix} \frac{1}{8}e^T A e & \frac{1}{8}e^T A \\ \frac{1}{8}A e & \frac{1}{8}A \end{bmatrix}$ by *N*. First of all, note that, if $k = \frac{n}{2}$, then the (SDR_{DkS}) problem is simplified to the following problem:

$$(\widetilde{\text{SDR}_{DkS}}) \qquad \max \quad N \bullet \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix},$$
$$(\widetilde{\text{SDR}_{DkS}}) \qquad \text{s.t.} \qquad \operatorname{diag}(Z) = e,$$
$$\begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} \in \mathbb{S}^+_{1+n},$$

and the $(II - SDR_{DkS})$ problem can be simplified as follows:

$$(\text{II} - \widetilde{\text{SDR}}_{\text{DkS}}) \qquad \begin{array}{l} \max \quad N \bullet \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix}, \\ \text{s.t.} \quad \text{diag}(Z) = e, \\ 1 + z_i + z_j + Z_{ij} \ge 0, \ \forall 1 \le i \le j \le n, \\ \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} \in \mathbb{S}^+_{1+n}. \end{array}$$

Combining Theorem 3 in [25] with the corresponding known approximation accuracy of semidefinite relaxation for some quadratic programming problems [26], we immediately have that the following theorem holds.

Theorem 5. If $k = \frac{n}{2}$, then we have

$$\frac{2}{\pi} Opt(SDR_{\widehat{DkS}}) = \frac{2}{\pi} Opt(SD\widetilde{R_{DkS}}) \le Opt(DkS) = Opt(\widehat{DkS}) \le Opt(SD\widetilde{R_{DkS}}) = Opt(SDR_{\widehat{DkS}})$$

and

$$\frac{2}{\pi} Opt(SDR_{\widehat{DkS}}) \le Opt(DNNP_{DkS}) = Opt(II - SDR_{DkS}) = Opt(II - \widetilde{SDR}_{DkS}) \le Opt(SDR_{\widehat{DkS}})$$

for $N \succeq 0$. Moreover, if $N \succeq 0$ and $N_{ij} \le 0$, $\forall i \neq j$, we further have

$$0.87856Opt(SDR_{\widehat{DkS}}) \le Opt(DkS) = Opt(\widehat{DkS}) \le Opt(SDR_{\widehat{DkS}})$$

and

$$0.87856Opt(SDR_{\widehat{DkS}}) \le Opt(DNNP_{DkS}) = Opt(II - SDR_{DkS}) \le Opt(SDR_{\widehat{DkS}})$$

hold.

In the following analysis, we assume that $k \neq \frac{n}{2}$. We first observe that

 $Opt(DkS) = Opt(\overline{DkS}),$

since the $(\overline{\text{DkS}})$ problem is the homogeneous problem for the (DkS) problem. In addition, note that the constraint $\begin{bmatrix} t \\ z \end{bmatrix} \in \{-1, 1\}^{1+n}$ in the $(\overline{\text{DkS}})$ problem can be relaxed to the quadratic form $\begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix} = n + 1$. Thus, the $(\overline{\text{DkS}})$ problem can be relaxed to

$$\max \begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} \frac{1}{8}e^T A e & \frac{1}{8}e^T A \\ \frac{1}{8}A e & \frac{1}{8}A \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix},$$

(DKS) s.t.
$$\begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{2}e^T \\ \frac{1}{2}e & \mathbf{0} \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix} = 2k - n,$$
$$\begin{bmatrix} t & z^T \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix} = n + 1,$$

which can further be relaxed to the standard semidefinite programming problem

$$\max \begin{bmatrix} \frac{1}{8}e^{T}Ae & \frac{1}{8}e^{T}A \\ \frac{1}{8}Ae & \frac{1}{8}A \end{bmatrix} \bullet S,$$

$$(SDR_{\widetilde{DkS}}) \qquad \text{s.t.} \begin{bmatrix} 0 & \frac{1}{2}e^{T} \\ \frac{1}{2}e & \mathbf{0} \end{bmatrix} \bullet S = 2k - n,$$

$$I \bullet S = n + 1,$$

$$S \in \mathbb{S}^{+}_{1+n}.$$

Obviously, diag(*S*) = *e* implies that $\sum_{i} S_{ii} = n + 1$, i.e., $I \bullet S = n + 1$, but we could not obtain diag(*S*) = *e* from $I \bullet S = n + 1$. These results further imply that

$$Opt(SDR_{\overline{DkS}}) \le Opt(SDR_{\widetilde{DkS}}).$$

Similar to the Theorem 4.2 in [27], we have that the following approximation accuracy theorem holds.

Theorem 6. $Opt(DkS) \le Opt(DkS) \le Opt(SDR_{\widetilde{DkS}}) \le 2log(67)Opt(DkS).$

Up to now, we not only establish the equivalence between the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem, but also some approximation accuracy results about the $(DNNP_{DkS})$ problem and some standard semidefinite relaxation problems are given. In the following Section 5, we will implement some numerical experiments to give a flavour of the actual behaviour of the $(DNNP_{DkS})$ problem and some semidefinite relaxation problems.

5. Numerical Experiments

In this section, some random (DkS) examples are tested to show the efficiency of the proposed relaxation problems. These relaxation problems are all solved by CVX [28], which is implemented by using MATLAB R2010a on the Windows XP platform, and on a PC with 2.53 GHz CPU. The corresponding comparative numerical results are reported in the following parts.

To give a flavour of the behaviour of the above relaxation problems, we consider results for the following test examples. The data of the test examples are given in Table 1.

P25	n = 25;	k = 5;	<pre>seed = 1, 2,, 50; rand('seed', seed);</pre>
			$A = round(10 \times rand(n)); A = tril(A, -1) + triu(A', 0);$
P30	n = 30;	k = 10;	<pre>seed = 2012;randn('seed',seed);A = randn(n,n);</pre>
			A = tril(A, -1) + triu(A', 0);
P40	n = 40;	k = 15;	seed = 2017; rand('seed', seed); $A = rand(n, n)$;
			A = tril(A, -1) + triu(A', 0);
P50	n = 50;	k = 10;	<pre>seed = 2001, 2002,, 2050; rand('seed', seed);</pre>
			$G = (rand(n) < 0.5); A = round(20 \times rand(n) - 10); A = triu(A, 1);$
			$A = A \cdot *G; A100 = A + A';$
P60	n = 60;	k = 20;	<pre>seed = 2020; rand('seed', seed); A = round(rand(n, n));</pre>
			A = tril(A, -1) + triu(A', 0);
P80	n = 80;	k = 30;	<pre>seed = 2023; rand('seed', seed); A = round(rand(n, n));</pre>
			A = tril(A, -1) + triu(A', 0);

Table 1. Data of the test examples.

The first column of Table 1 denotes the name of the test examples, n and k stand for the number of vertices of the given graph and the finding subgraph, respectively. The last column denotes the procedures for generating the coefficient matrices A in the (DkS) problem. The more detailed explanations for the procedures are given as follows:

• P25. 50 random examples are generated from the 'seed = 1,2,...,50'. The corresponding coefficient matrices *A* of order n = 25 with integer weights are drawn from $\{0, 1, ..., 10\}$.

• P30. This example is generated by the MATLAB function randn from the 'seed = 2012'. The elements of *A* satisfy the standard normal distribution.

• P40. This example is generated by MATLAB function rand from the 'seed = 2017'. The elements of *A* satisfy the standard uniform distribution on the interval (0, 1).

• P50. We generate 50 examples of order n = 50 from the 'seed = 2001,2002,...,2050'. For each coefficient matrix of these examples, half of the entries are randomly set to zero, and the remaining ones are chosen from $\{-10, -9, ..., 9, 10\}$.

• P60. This example is generated by MATLAB function rand from the 'seed = 2020'. The elements of A are drawn from $\{0, 1\}$.

• P80. This example is generated by MATLAB function rand from the 'seed = 2023'. The coefficient matrix A of order n = 80 with integer weights drawn from $\{0, 1, ..., 10\}$.

First of all, the performances of the (DNNP_{DkS}) problem and the (II – SDR_{DkS}) problem as well as the (I – SDR_{DkS}) problem, for solving P25 and P50, are compared. We use the performance profiles described in Dolan and Moré's paper [29]. Our profiles are based on optimal values (i.e., average degree) and the number of iterations of these relaxation problems. The Cumulative Probability denotes the cumulative distribution function for the performance ratio within a factor $\tau \in R$, i.e., is the probability that the solver will win over the rest of the solvers. The corresponding comparative results of performance are shown in Figures 1 and 2.

The comparative results for P25 are shown in Figure 1. It is obvious that the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem have the same performance, which is a bit better than that of the $(I - SDR_{DkS})$ problem from the viewpoint of optimal values. In view of the number of iterations, the performance of the $(I - SDR_{DkS})$ problem is the best, and the performance of the $(II - SDR_{DkS})$ problem is better than that of the $(DNNP_{DkS})$ problem.

The performance of the three relaxation problems for solving P50 is shown in Figure 2. The results show that the performance of the $(DNNP_{DkS})$ problem is the same as that of the $(II - SDR_{DkS})$ problem; they are both much better than that of the $(I - SDR_{DkS})$ problem in view of optimal values—although the performance of the $(I - SDR_{DkS})$ problem is better than the one of the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem from the viewpoint of the number of iterations.



Figure 1. The performance of the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem as well as the $(I - SDR_{DkS})$ problem for solving P25. The performance profile of (**left**) figure is based on optimal values, and (**right**) figure is based on the number of iterations.



Figure 2. The performance of the (DNNP_{DkS}) problem and the (II – SDR_{DkS}) problem as well as $(I – SDR_{DkS})$ problem for solving P50. The performance profile in the (**left**) figure is according to optimal values, and the (**right**) figure is based on the number of iterations.

All of results show in Figures 1 and 2 further imply that the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem can generate more promising bounds for solving P25 and P50, compared with the $(I - SDR_{DkS})$ problem, while the number of iterations is a bit more. Moreover, the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem have the same performance based on optimal values, although the performance of the $(II - SDR_{DkS})$ problem is better than that of the $(DNNP_{DkS})$ problem from the viewpoint of the number of iterations, for solving P25 and P50.

In order to further show the computational efficiency of the $(DNNP_{DkS})$ problem, which is compared with the $(II - SDR_{DkS})$ problem and some other types of semidefinite relaxation problems proposed in [30], for solving some (DkS) problems. The test examples A50 and A100 are chosen from [30]. (R-20), (R-24) and (R-MET) denote the three semidefinite relaxation problems proposed in [30], respectively. The corresponding numerical results are shown in Table 2, where "-" means that the corresponding information about the number of iterations is not given in [30]. The results show that the computational efficiency of the $(DNNP_{DkS})$ problem is better than the one of the $(II - SDR_{DkS})$ problem from the viewpoints of optimal values and number of iterations, respectively. Note that the performance of the $(DNNP_{DkS})$ problem and the $(II - SDR_{DkS})$ problem are both much better than that of (R - 20) and (R - 24). Moreover, the performance of the $(DNNP_{DkS})$ problem is more competitive with (R - MET) for solving these two problems.

Example	Solving Model	Optimal Value	Number of Iterations
	(DNNP _{DkS})	337.4	51
	$(II - SDR_{DkS})$	337.416	57
A50	(R - 20)	365.508	_
	(R - 24)	365.831	_
	(R - MET)	333.65	_
	(DNNP _{DkS})	505.989	54
	$(II - SDR_{DkS})$	506.126	68
A100	(R - 20)	607.691	_
	(R - 24)	611.070	_
	(R - MET)	490.50	_

Table 2. Numerical results for some DkS examples in [30].

Finally, we further compare the efficiency of the $(DNNP_{DkS})$, $(II - SDR_{DkS})$, $(I - SDR_{DkS})$ and $(SDR_{\widehat{DkS}})$ problems, for solving examples P30, P40, P60 and P80. The comparative results are shown in Table 3. The results signify that the efficiency of the $(DNNP_{DkS})$ problem is always better than that of the $(II - SDR_{DkS})$ problem from the viewpoint of optimal values and the number of iterations as well

as CPU time, respectively, for solving these examples. The performance of the $(I - SDR_{DkS})$ problem and the $(SDR_{\widehat{DkS}})$ problem are almost the same for solving these examples. Moreover, note that the optimal value of the $(DNNP_{DkS})$ problem for solving P80 is larger than that of the $(II - SDR_{DkS})$ problem. Thus, we can conclude that it may be more promising to use the $(DNNP_{DkS})$ problem than to use the $(II - SDR_{DkS})$ problem for solving some specific (DkS) problems in practice.

Example	Solving Model	Optimal Value	Number of Iterations	CPU Time (seconds)
	(DNNP _{DkS})	30.5894	52	3.4
D2 0	$(II - SDR_{DkS})$	30.5903	70	5.0
P 50	$(I - SDR_{DkS})$	33.3382	33	0.7
	$(\text{SDR}_{\widehat{\text{DkS}}})$	33.3383	34	0.8
	(DNNP _{DkS})	74.4657	50	14.7
P40	$(II - SDR_{DkS})$	74.4658	66	20.6
140	$(I - SDR_{DkS})$	75.5198	33	1.5
	$(\text{SDR}_{\widehat{\text{DkS}}})$	75.5197	34	1.6
	(DNNP _{DkS})	150.385	52	105.5
P 60	$(II - SDR_{DkS})$	150.387	60	125.3
100	$(I - SDR_{DkS})$	154.272	40	3.2
	$(\text{SDR}_{\widehat{\text{DkS}}})$	154.273	35	2.2
	$(DNNP_{DkS})$	230.912	84	739.9
D8 0	$(II - SDR_{DkS})$	312.644	85	770.4
1 00	$(I - SDR_{DkS})$	318.945	41	5.7
	$(SDR_{\widehat{DkS}})$	318.945	34	3.2

Table 3. Numerical results for some (DkS) examples.

6. Conclusions

In this paper, the DkS problem is studied, whose goal is to find a *k*-vertex subgraph such that the total weight of edges in this subgraph is maximized. This problem is NP-hard on bipartite graphs, chordal graphs, and planar graphs. By using the advantages of the structure of the DkS problem, the doubly nonnegative relaxation and the new semidefinite relaxation with tighter relaxation for solving the DkS problem are established, respectively. Moreover, we prove that the two relaxation problems are equivalent under the suitable conditions, and give some approximation accuracy results for these relaxation problems. Finally, the comparative numerical results show that the efficiency of the doubly nonnegative relaxation is better than the one of semidefinite relaxation for solving some DkS problems.

Author Contributions: Methodology, C.-H.G.; software, C.-H.G., Y.G. and B.-B.L.; validation, C.-H.G.; formal analysis, C.-H.G. and B.-B.L.; investigation, C.-H.G. and Y.G.; data curation, C.-H.G. and Y.G.; writing—original draft preparation, C.-H.G.; writing—review and editing, C.-H.G., Y.G. and B.-B.L.; visualization, C.-H.G. and Y.G.; supervision, C.-H.G.; project administration, C.-H.G.; funding acquisition, C.-H.G.

Funding: This work is supported by National Natural Science Foundation of China under Grant No. 11501350 and Science Foundation of Zhejiang Sci-Tech University (ZSTU) under Grant No. 16092098-Y.

Acknowledgments: The authors thank the reviewers for their very helpful suggestions, which led to substantial improvements of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

\mathbb{R}_n^+	the cone of nonnegative vectors with dimension <i>n</i>
\mathbb{S}_n	the cone of $n \times n$ symmetric matrices
\mathbb{S}_n^+	the cone of $n \times n$ positive semidefinite matrices
\mathbb{S}_n^{++}	the cone of $n \times n$ positive definite matrices
\mathbb{N}_n^+	the cone of $n \times n$ nonnegative symmetric matrices
е	the vector of ones
Ι	the identity matrix with appropriate dimension
$X \succeq (\succ) 0$	stands for X is positive semidefinite (positive definite), i.e, $X \in \mathbb{S}_n^+(\mathbb{S}_n^{++})$
$A \bullet B$	trace($A^T B$) for two conformal matrices A and B
diag(M)	the column vector whose elements are the diagonal elements of the given matrix M
Opt(*)/Fea(*)	the optimal objective value/the set of feasible solutions of the given problem $(*)$

References

- 1. Feige, U.; Kortsarz, G.; Peleg, D. The Dense k-Subgraph Problem1. Algorithmica 2001, 29, 410–421. [CrossRef]
- 2. Roupin, F.; Billionnet, A. A deterministic approximation algorithm for the densest *k*-subgraph problem. *Int. J. Oper. Res.* **2008**, *3*, 301–314. [CrossRef]
- 3. Corneil, D.; Perl, Y. Clustering and domination in perfect graphs. *Discret. Appl. Math.* **1984**, *9*, 27–39. [CrossRef]
- 4. Keil, J.; Brecht, T. The complexity of clustering in planar graphs. *J. Combin. Math. Combin. Comput.* **1991**, *9*, 155–159.
- 5. Pisinger, P. Upper bounds and exact algorithms for dispersion problems. *Comput. Oper. Res.* 2006, 33, 1380–1398. [CrossRef]
- 6. Goldstein, D.; Langberg, M. The dense *k*-subgraph problem. *arXiv* **2009**, arXiv:0912.5327.
- 7. Backer, J.; Keil, J. Constant factor approximation algorithms for the densest *k*-subgraph problem on proper interval graphs and bipartite permutation graphs. *Inf. Process. Lett.* **2010**, *110*, 635–638. [CrossRef]
- Bhaskara, A.; Charikar, M.; Chlamtac, E.; Feige, U.; Vijayaraghavan, A. Detecting high log-densities: An O (*n* 1/4) approximation for densest *k*-subgraph. In Proceedings of the 42nd ACM Symposium on Theory of Computing, Cambridge, MA, USA, 6–8 June 2010; pp. 201–210.
- 9. Liazi, M.; Milis, I.; Zissimopoulos, V. A constant approximation algorithm for the densest *k*-subgraph problem on chordal graphs. *Inf. Process. Lett.* **2008**, *108*, 29–32. [CrossRef]
- 10. Goemans, M.; Williamson, D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* **1995**, *42*, 1115–1145. [CrossRef]
- 11. Poljak, S.; Rendl, F.; Wolkowicz, H. A recipe for semidefinite relaxation for (0, 1)-quadratic programming. *J. Glob. Optim.* **1995**, *7*, 51–73. [CrossRef]
- 12. Ma, W.; Davidson, T.; Wong, K.; Luo, Z.; Ching, P. Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA. *IEEE Trans. Signal Process.* **2002**, *50*, 912–922.
- 13. Ma, W.; Su, C.; Jalden, J.; Chang, T.; Chi, C. The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 1038–1052. [CrossRef]
- 14. Lovasz, L. Stable sets and polynomials. Discret. Math. 1994, 124, 137–153. [CrossRef]
- 15. Burer, S. On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program* **2009**, *120*, *479–495*. [CrossRef]
- 16. Bai, Y.; Guo, C. Doubly nonnegative relaxation method for solving multiple objective quadratic programming problems. *J. Ind. Manag. Optim.* **2014**, *10*, 543–556. [CrossRef]
- 17. Burer, S. Optimizing a polyhedral-semidefinite relaxation of completely positive programs. *Math. Program Comput.* **2010**, *2*, 1–19. [CrossRef]
- 18. Arima, N.; Kim, S.; Kojima, M.T.K. Lagrangian-Conic relaxations, part I: A unified framework and its applications to quadratic optimization problems. *Pac. J. Optim.* **2018**, *14*, 161–192.
- 19. Bomze, I. Copositive optimization-Recent developments and applications. *Eur. J. Oper. Res.* 2012, 216, 509–520. [CrossRef]

- 20. Pardalos, P.; Vavasis, S. Quadratic programming with one negative eigenvalue is NP-hard. *J. Glob. Optim.* **1991**, *1*, 15–22. [CrossRef]
- 21. Dickinson, P.; Gijben, L. On the Computational Complexity of Membership Problems for the Completely Positive Cone and its Dual; University of Groningen: Groningen, The Netherlands, 2011.
- 22. Diananda, P. On non-negative forms in real variables some or all of which are non-negative. In *Mathematical Proceedings of the Cambridge Philosophical Society;* Cambridge University Press: Cambridge, UK, 1962.
- 23. Berman, A.; Shaked-Monderer, N. Completely Positive Matrices; World Scientific: Singapore, 2003.
- 24. Boyd, S.; Vandenberghe, L. Convex Optimization; Cambridge University Press: Cambridge, UK, 2004.
- 25. Ge, D.; Ye, Y. On Doubly Positive Semidefinite Programming Relaxations. 2010. Available online: http://www.optimization-online.org/DB_HTML/2010/08/2709.pdf (accessed on 17 August 2010).
- 26. Luo, Z.; Ma, W.; So, A.C.; Ye, Y.; Zhang, S. Semidefinite relaxation of quadratic optimization problems. *Signal Proc. Mag. IEEE* **2010**, *27*, 20–34. [CrossRef]
- 27. He, S.; Luo, Z.; Nie, J.; Zhang, S. Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization. *SIAM J. Optim.* **2008**, *19*, 503–523. [CrossRef]
- 28. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming, Version 1.21. Available online: http://cvxr.com/cvx (accessed on 15 April 2010).
- 29. Dolan, E.; Moré, J. Benchmarking optimization software with performance profiles. *Math. Program.* 2002, *91*, 201–213. [CrossRef]
- 30. Rendl, F. Semidefinite relaxations for partitioning, assignment and ordering problems. *Q. J. Oper. Res.* **2012**, 10, 321–346. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).