ISSN 1342-2804

Research Reports on Mathematical and Computing Sciences

Conic relaxation approaches for equal deployment problems

Sena Safarina, Satoko Moriguchi, Tim J. Mullin, Makoto Yamashita

March 2017, B–485

Department of Mathematical and Computing Sciences
Tokyo Institute of Technology

SERIES B: Operations Research

Conic relaxation approaches for equal deployment problems

Sena Safarina [1], Satoko Moriguchi [2], Tim J. Mullin [3], and Makoto Yamashita [1]
Submitted: February 27, 2017.

Abstract: An important problem in the breeding of livestock, crops, and forest trees is the optimum of selection of genotypes that maximizes genetic gain. The key constraint in the optimal selection is a convex quadratic constraint that ensures genetic diversity, therefore, the optimal selection can be cast as a second-order cone programming (SOCP) problem. Yamashita et al. (2015) exploits the structural sparsity of the quadratic constraints and reduces the computation time drastically while attaining the same optimal solution.

This paper is concerned with the special case of equal deployment (ED), in which we solve the optimal selection problem with the constraint that contribution of genotypes must either be a fixed size or zero. This involves a nature of combinatorial optimization, and the ED problem can be described as a mixed-integer SOCP problem.

In this paper, we discuss conic relaxation approaches for the ED problem based on LP (linear programming), SOCP, and SDP (semidefinite programming). We analyze theoretical bounds derived from the SDP relaxation approaches using the work of Tseng (2003) and show that the theoretical bounds are not quite sharp for tree breeding problems. We propose a steepest-ascent method that combines the solution obtained from the conic relaxation problems with a concept from discrete convex optimization in order to acquire an approximate solution for the ED problem in a practical time. From numerical tests, we observed that among the LP, SOCP, and SDP relaxation problems, SOCP gave a suitable solution from the viewpoints of the optimal values and the computation time. The steepest-ascent method starting from the SOCP solution provides high-quality solutions much faster than an existing method that has been widely used for the optimal selection problems and a branch-and-bound method.

Keywords: Semidefinite programming, Second-order cone programming, Mixed-integer conic programming, Conic relaxation, Tree breeding, Equal deployment problem.
MCS2010 classification: 90C05 Linear programming, 90C11 Mixed integer programming, 90C22 Semidefinite programming, 90C25 Convex programming, 90C59 Approximation methods and heuristics, 90C90 Applications of mathematical programming, 92-08 Biology and other natural sciences (Computational methods).

[1] Department of Mathematical and Computing Science, Tokyo Institute of Technology, 2-12-1-W8-29 Ookayama, Meguro-ku, Tokyo 152-8552, Japan. (The work of M. Yamashita was partially supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research (C), 15K00032).)

[2] School of Business Administration, Faculty of Urban Liberal Arts, Tokyo Metropolitan University, 1-1 Minami-Osawa, Hachioji-shi, Tokyo 192-0397, Japan.

[3] The Swedish Forestry Research Institute (Skogforsk), Box 3, Sävar 918 21, Sweden; and 224 rue du Grand-Royal Est, QC, J2M 1R5, Canada. (The work of T. J. Mullin was partially supported by Föreningen Skogsträdsfördling (The Swedish Tree Breeding Foundation)

1 Introduction

Computational methods based on mathematical optimization have started gaining attention from breeding researchers, since the optimization methods provide efficient approaches and give theoretical aspects for the optimality of the obtained solutions. For example, optimal selection problems that determine the contributions of genotypes are studied for clonal seed orchards and dairy cattle [5, 13, 15, 19, 22, 24, etc].

A main objective in optimal selection problems is to attain the highest response from a genotype selection. Lindgren et al. [19] proposed a linear deployment in which the genotype contributions are basically proportional to their breeding values. This deployment was derived from a concept that the genotypes with higher breeding values should appear more frequently than those with lower values. An advantage of the linear deployment was the extremely low computation cost, since it could be computed by a greedy algorithm. However, the linear deployment worked well only when the pedigree situation was simple, that is, the candidate genotypes were unrelated. If the selected genotypes do not embrace enough diversity, the response will critically diminish through inbreeding depression [6, 40] due to accumulated kinship.

Meuwissen [22] introduced a quadratic constraint to control a group coancestry under an appropriate level. He developed the Lagrangian multiplier method to maximize the genetic response with the quadratic constraints. This method was implemented in a software package GENCONT [22], and it has been widely accepted among breeding researchers. A serious drawback of the Lagrangian multiplier method is that this method does not always generate optimal solutions. In contrast, Pong-Wong et al. [31] employed an SDP approach. This approach is based on mathematical optimization, and they demonstrated that this approach gave the optimal contributions exactly. This approach was extended in [1], but their SDP approach required long computation time even when they used parallel computing with the help of SDPA (a high-performance solver for SDPs) [45, 46]. Recently, Yamashita et al. [47] proposed an SOCP (second-order cone programming) approach and successfully reduced the computation time of the SDP approach attaining the same optimal solution.

The problems solved by the SDP approach [31] and the SOCP approach [47] are unequal deployment (UD) problems of form

$$\begin{array}{lll} \max & : & \boldsymbol{g}^T\boldsymbol{x} \\ \text{subject to} & : & \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta, \\ & & \boldsymbol{e}^T\boldsymbol{x} = 1, \\ & & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}. \end{array} \tag{1}$$

Throughout this paper, we use Z to denote the number of candidate genotypes. In the UD problem, the variable is the vector $\boldsymbol{x} \in \mathbb{R}^Z$, and x_i indicates the contribution of the ith genotype. We use a superscript T to denote the transpose of a vector or a matrix. The cost vector $\boldsymbol{g} \in \mathbb{R}^Z$ in the objective function is the estimated breeding value (EBV) [21]. Since this vector is computed separately, we regard $\boldsymbol{g}$ as a constant vector. The matrix $\boldsymbol{A} \in \mathbb{R}^{Z\times Z}$ is the Wright numerator matrix [43]. The elements of this matrix are given from the information of heredity diagram. We should emphasize that the matrix $\boldsymbol{A}$ is always symmetric and positive definite. Hence, with a given constant $\theta > 0$, the constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ is a convex constraint, and this quadratic constraint ensures that the group coancestry $\frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{2}$ in the selected group is kept under a permissible range θ. We use $\boldsymbol{e} \in \mathbb{R}^n$ to denote the vector of all ones, therefore, the constraint $\boldsymbol{e}^T\boldsymbol{x} = 1$ indicates that the total contribution of all the candidates is unity. In addition, the vectors $\boldsymbol{l} \in \mathbb{R}^Z$ and $\boldsymbol{u} \in \mathbb{R}^Z$ are the lower and upper bounds of the variable $\boldsymbol{x}$, respectively.

The name an *unequal* deployment indicates that the contributions need not to be equal. Since the variable $\boldsymbol{x}$ is a continuous variable and the constraints are linear or convex-quadratic, the UD problem can be cast a type of SOCP problems, as pointed in [47]. Therefore, the UD problem can be solved in a polynomial time algorithm, for example, interior-point methods for SOCP [2, 7, 38].

This paper is concerned with the special-case problem of *equal* deployment (ED) form

$$OPT_{ED} := \begin{array}{lll} \max & : & \boldsymbol{g}^T\boldsymbol{x} \\ \text{subject to} & : & \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta, \\ & & \boldsymbol{e}^T\boldsymbol{x} = 1, \\ & & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, \\ & & x_1, \ldots, x_n \in \left\{0, \frac{1}{N}\right\}. \end{array} \tag{2}$$

We use OPT_{ED} to denote the optimal value of this problem. The crucial difference from the UD problem is that the ED problem has the binary constraints $x_1, \ldots, x_n \in \left\{0, \frac{1}{N}\right\}$. We choose exactly N genotypes from Z candidates, and the selected N genotypes must contribute their genes equally. The ED problems fit breeding populations, where we consider the selected genotypes should contribute with the same amount and therefore we require a fixed-size population.

Weng et al. [39] solved the ED problem only with the linear constraints and the binary constraints using the "Solver" tool in Microsoft Excel. Meuwisen extended GENCONT to the ED problems incorporating some heuristic methods so that GENCONT generated approximate solutions that satisfy the binary constraints. The heuristic methods implemented in GENCONT are partially discussed in [42].

From the viewpoint of mathematical optimization, the most difficult constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ is a quadratic convex constraint. An ED problem (2) can thus be viewed as a mixed-integer second-order cone programing (MI-SOCP) problem. Many approaches have been explored to solve MI-SOCP efficiently. Ben-tal and Nemirovski [3] proposed a polyhedral relaxation that approximates a second-order cone with a polyhedron so that the resulting problem can be handled with software packages for mixed-integer linear programming. Drewes applied an outer approximation method and a branch-and-cut method [8]. For other approaches, a survey paper due to Benson and Saglam [4] is a good reference.

Theoretically speaking, MI-SOCP is an SOCP problem with integer constraints, hence, we can obtain an exact optimal solution if we rely on the branch-and-bound framework. However, we suffer from a long computation time if we pursue the exact solution. For example, CPLEX can directly handle MI-SOCP problems, but fails to complete the computation of a small case $Z = 1050$ and $N = 50$ (it tried to choose $N = 50$ genotypes from $Z = 1050$ candidates) in one week. Mullin and Belotti [23] combined the outer approximation method and the branch-and-bound method and reduced the computation time. However, it also requires half a day for the small case $Z = 200$ to attain the gap 0.5%, so it is still hard to say that this approach is practical for larger instances $Z \geq 5000$. To manage ED problems in a practical time, it is desirable that we find a high-quality approximate solution instead of the exact solution.

In this paper, we propose an integration of conic relaxation approaches and a steep-ascent method originally developed for discrete convex functions to derive a suitable solution for practical usage in a reasonable computation time.

An epoch-making paper on conic relaxation approach was the application of SDP problem to the max-cut problems by Goemans and Williamson [11]. They converted a feasible set of the max-cut problems into the space of positive semidefinite matrices with the rank-one constraint on the matrix variable, and they derived an SDP problem by ignoring this rank-one constraint. They showed that a solution generated with a randomized algorithm from an optimal solution of the resulting SDP problem gave very good approximation to the original max-cut problem. Following this achievement, the SDP relaxation approach has widely been applied to combinatorial optimization problems, see [41] and the references therein. Theoretical evaluation of the quality of the approximate solution were discussed in [14, 16, 29, 36, 49, etc]. Conic relaxation approaches are the relaxation approaches that employs linear programming (LP), SOCP or SDP problems. A remarkable points of the three conic programming problems (LP, SOCP, and SDP) is that they can be analyzed in the framework of Euclidean Jordan algebras [9, 32]. Hence, the resulting relaxation problems can be solved in polynomial time by interior-point methods [30] and many software packages are available [33, 35, 45]. Kim and Kojima [17] reported a numerical evaluation on the relaxation approaches using LP, SOCP, and SDP for some quadratic optimization problems.

On the other hand, discrete convex optimization has another abundant research direction. We might consider that a convex function in continuous space is a discrete convex function if we restrict the variable space to the integer points, although this naive intuition is not appropriate because such a function does not always have useful properties of convex functions, and some deep combinatorial or discrete-mathematical considerations are needed for discrete convexity. In the theory of discrete convex analysis [26], two convexity concepts, called L-convexity and M-convexity, play primary roles. L-convex functions and M-convex functions are convex functions with additional combinatorial properties distinguished by "L" and "M", which are conjugate to each other through a discrete version of the Legendre-Fenchel transformation. If a function is an M-convex function, a step-descent method proposed in [27] can find its global minimum.

In this paper, we first introduce conic relaxation problems for the ED problems, and discuss the relations between the relaxation problems. We analyze the theoretical bounds of the randomized algorithm starting from the solution of the SDP relaxation problem. However, when we numerically evaluate these bounds using tree-breeding datasets, we learn that these bounds are not so sharp. Instead of pursuing an exact solution by branch-and-bound frameworks that impose heavy computation costs, our focus is to acquire a favorable solution that is available in a practical computation time. To obtain such a solution, we develop a steep-ascent method that employs the solution obtained from the conic relaxation problems as a starting point. The usual steep-descent method [27] minimizes an objective function on a particular feasible set. Since the ED problem is a maximization problem, we consider a steep-ascent method instead of a steep-descent method. We embed the quadratic constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ into the objective function as a penalty term with a weight computed from the Lagrange multiplier. This new objective function is not an M-concave function, therefore, we cannot guarantee that the solution obtained by the steep-ascent method is a global solution of the ED problem. However, through numerical experiments, we observe that the steep-ascent method generates qualified solutions for the ED problem. In particular, the steep-ascent method starting with the SOCP relaxation problem attains the best performance among the LP, SOCP, and SDP relaxation problems. Actually, we verify from numerical experiments that this approach performs better than existing methods like GENCONT in the viewpoints of both solution quality and computation time.

The rest of this paper is organized as follows. In Section 2, we introduce LP, SOCP, and SDP relaxation problems for the ED problems, and we discuss the strength of these conic relaxations. In Section 3, we analyze the approximation rate of the SDP relaxation based on the work of Tseng [36]. Section 4 gives the details of the steep-ascent method specialized for the ED problems. In Section 5, we present numerical results to compare the conic relaxations and to evaluate the solution acquired by the steep-ascent method. We also compare this result with existing methods. In Section 6, we will give a conclusion and discuss future directions.

1.1 Notation

We use $|S|$ to denote the cardinality of a set S. The vector $\boldsymbol{e}_S$ is the vector of all ones of the lengths $|S|$. In contrast, we denote by $\boldsymbol{e}_i$ the vector of all zeros except one in the ith position. The symbol $\mathbb{S}^n$ is used to denote the space of $n \times n$ symmetric matrices, and $\boldsymbol{X} \succeq \boldsymbol{O}$ indicates that a symmetric matrix $\boldsymbol{X}$ is positive semidefinite. The inner-product between $\boldsymbol{A} \in \mathbb{S}^n$ and $\boldsymbol{X} \in \mathbb{S}^n$ is defined by $\boldsymbol{A} \bullet \boldsymbol{X} := \sum_{i=1}^n \sum_{j=1}^n A_{ij} X_{ij}$. The trace of a matrix $\boldsymbol{A} \in \mathbb{S}^n$ is given by $\text{Trace}(\boldsymbol{A}) := \sum_{i=1}^n A_{ii}$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \boldsymbol{0}$ indicates the element-wise non-negativity of $\boldsymbol{x}$, that is, $x_1, \ldots, x_n \geq 0$.

2 Conic relaxations for equally deployment problems

In this section, we first derive an SDP relaxation problem of an ED problem. Then, by a further relaxation of the positive semidefinite condition using a relaxation technique proposed in [17], we obtain an LP relaxation problem. Finally, we apply a continuous relaxation technique to the ED problem to obtain an SOCP relaxation problem. The reason we employ a different relaxation approach for only the SOCP relaxation is that we can exploit a structural sparsity in the Wright numerator matrix $\boldsymbol{A}$.

A standard form of SDP problems can be given as follow:

$$\begin{array}{lll} \min & : & \boldsymbol{C} \bullet \boldsymbol{X} \\ \text{subject to} & : & \boldsymbol{F}_i \bullet \boldsymbol{X} = b_i \ (i = 1, \ldots, m), \\ & & \boldsymbol{X} \succeq \boldsymbol{O}. \end{array} \tag{3}$$

In this standard form, the variable matrix is $\boldsymbol{X} \in S^n$. The input matrices in (3) are $\boldsymbol{C}$, $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_m \in \mathbb{S}^n$, while the vector $\boldsymbol{b} \in \mathbb{R}^n$ is an input vector. Shortly speaking, a standard SDP form minimizes a linear objective function over linear constraints and a positive semidefinite condition on $\boldsymbol{X}$.

As a first step to derive an SDP relaxation from the ED problem (2), we remove the variables that can be fixed from the box constraints. More precisely, if $l_i > 0$, we fix $x_i = \frac{1}{N}$. Similarly, we fix $x_i = 0$ if $u_i < \frac{1}{N}$. We ignore the cases $l_i > \frac{1}{N}$, $u_i < 0$ or $l_i > u_i$, since we can immediately detect the infeasibility of the ED problem. Then, we define two sets F and V so that the two sets separate the set $\{1, \ldots, Z\}$ disjointly and x_i is fixed to $c_i \in \left\{0, \frac{1}{N}\right\}$ for $i \in F$ while x_i remains as a decision variable for $i \in V$.

Without loss of generality, we assume that $V = \{1, 2, \ldots, |V|\}$, $F = \{|V|+1, |V|+2, \ldots, Z\}$, and $g_1 \geq g_2 \geq \ldots \geq g_{|V|}$. Along with these V and F, we introduce the vectors $\boldsymbol{x}_V$ and $\boldsymbol{c}_F$ that divide $\boldsymbol{x} \in \mathbb{R}^Z$ into the two parts $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_V \\ \boldsymbol{c}_F \end{pmatrix}$. We also divide the Wright numerator matrix $\boldsymbol{A}$ into the four parts; $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{VV} & \boldsymbol{A}_{VF} \\ \boldsymbol{A}_{FV} & \boldsymbol{A}_{FF} \end{pmatrix}$. The sizes of $\boldsymbol{A}_{VV}$, $\boldsymbol{A}_{FV} (= \boldsymbol{A}_{VF}^T)$, and $\boldsymbol{A}_{FF}$ are

$|V| \times |V|$, $|F| \times |V|$, and $|F| \times |F|$, respectively. We further partition the vectors and the matrices that appear in the ED problem into the corresponding parts;

$$
\begin{array}{rclcl}
OPT_{ED} & = & \max & : & \boldsymbol{g}_V^T \boldsymbol{x}_V + \boldsymbol{g}_F^T \boldsymbol{c}_F \\
 & & \text{subject to} & : & \boldsymbol{x}_V^T \boldsymbol{A}_{VV} \boldsymbol{x}_V + 2\boldsymbol{c}_F^T \boldsymbol{A}_{FV} \boldsymbol{x}_V + \boldsymbol{c}_F^T \boldsymbol{A}_{FF} \boldsymbol{c}_F \leq 2\theta, \\
 & & & & \boldsymbol{e}_V^T \boldsymbol{x}_V + \boldsymbol{e}_F^T \boldsymbol{c}_F = 1, \\
 & & & & x_i \in \left\{0, \frac{1}{N}\right\} \text{ for } i \in V.
\end{array} \tag{4}
$$

Note that we also removed the box constraints $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$ from the ED problem by fixing the variables in $\boldsymbol{x}_F$ to $\boldsymbol{c}_F$. We count the number of x_i that is fixed to c_i by $p := \left|\left\{i \in F : x_i = \frac{1}{N}\right\}\right|$. Therefore, we will choose $N-p$ genotypes from $|V|$ candidates in (4), while we choose N genotypes from Z candidates in the original ED problem (2).

Remark 2.1. *We can assume $p \leq N$ and $|V| \geq 2$ without loss of generality. In the case $p > N$, we can detect the infeasibility of the problem (2). If $|V| = 1$, we have $F = \{2, \ldots, Z\}$. Therefore, x_1 is also fixed with $x_1 = 1 - \sum_{i=2}^{Z} c_i$, and all the variables can be fixed without solving (4).*

We change the decision variables by $\boldsymbol{y}_V := 2N\boldsymbol{x}_V - \boldsymbol{e}_V \in \mathbb{R}^{|V|}$ and we use y_i to denote the ith element of $\boldsymbol{y}_V$. Then, the binary constraints $x_1, \ldots, x_{|V|} \in \left\{0, \frac{1}{N}\right\}$ are mapped to $y_1, \ldots, y_{|V|} \in \{-1, 1\}$. Even without employing this variable change, we can also directly apply the SDP relaxation method in a similar way to [12]. The reason we employed this variable change is for the later discussion in Section 4 so that most of the matrices $\boldsymbol{B}^k$ there will be diagonal matrices.

We will denote the ith element of $\boldsymbol{y}_V$ by y_i. We define $g_{\min} := \min\{g_i : i = 1, \ldots, Z\}$, $\bar{\boldsymbol{g}}_V := \frac{1}{4N}(\boldsymbol{g}_V - g_{\min}\boldsymbol{e}_V)$, $\bar{g} := \frac{1}{2N}(\boldsymbol{g}_V - g_{\min}\boldsymbol{e}_V)^T\boldsymbol{e}_V + (\boldsymbol{g}_F - g_{\min}\boldsymbol{e}_F)^T\boldsymbol{c}_F + g_{\min}$, $\bar{\boldsymbol{c}}_F := \boldsymbol{A}_{VV}\boldsymbol{e}_V + 2N\boldsymbol{A}_{VF}\boldsymbol{c}_F$, $\bar{\theta} := 2N^2(2\theta - \boldsymbol{c}_F^T\boldsymbol{A}_{FF}\boldsymbol{c}_F) - \frac{1}{2}\boldsymbol{e}_V^T\boldsymbol{A}_{VV}\boldsymbol{e}_V - 2N\boldsymbol{c}_F^T\boldsymbol{A}_{FV}\boldsymbol{e}_V$, and $\bar{N} := 2N(1 - \boldsymbol{e}_F^T\boldsymbol{c}_F) - |V| = 2(N-p) - |V|$. From these definitions, it is easy to check $\bar{\boldsymbol{g}}_V \geq \boldsymbol{0}$ and $\boldsymbol{g}^T\boldsymbol{x} = 2\bar{\boldsymbol{g}}_V^T\boldsymbol{y}_V + \bar{g}$ using $\boldsymbol{e}^T\boldsymbol{x} = 1$. We now have another expression of the ED problem;

$$
\begin{array}{rclcl}
OPT_{ED} & = & \max & : & 2\bar{\boldsymbol{g}}_V^T \boldsymbol{y}_V + \bar{g} \\
 & & \text{subject to} & : & \boldsymbol{y}_V^T \boldsymbol{A}_{VV} \boldsymbol{y}_V + 2\bar{\boldsymbol{c}}_F^T \boldsymbol{y}_V \leq 2\bar{\theta}, \\
 & & & & \boldsymbol{e}_V^T \boldsymbol{y}_V = \bar{N}, \\
 & & & & y_i \in \{-1, 1\} \text{ for } i \in V.
\end{array} \tag{5}
$$

By introducing a variable matrix $\boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}$, we apply the lift-and-project method of Lovász and Schrijver [20]. As a result, we obtain one more equivalent form;

$$
\begin{array}{rclcl}
OPT_{ED} & = & \max & : & \begin{pmatrix} 0 & \bar{\boldsymbol{g}}_V^T \\ \bar{\boldsymbol{g}}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} + \bar{g} \\
 & & \text{subject to} & : & \begin{pmatrix} -2\bar{\theta} & \bar{\boldsymbol{c}}_F^T \\ \bar{\boldsymbol{c}}_F & \boldsymbol{A}_{VV} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \leq 0, \\
 & & & & \begin{pmatrix} -2\bar{N} & \boldsymbol{e}_V^T \\ \boldsymbol{e}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\
 & & & & \begin{pmatrix} -\bar{N}^2 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_V\boldsymbol{e}_V^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\
 & & & & \begin{pmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_i\boldsymbol{e}_i^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0 \text{ for } i \in V, \\
 & & & & \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \succeq \boldsymbol{O}, \quad \text{rank}\left(\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix}\right) = 1.
\end{array} \tag{6}
$$

The key property for the equivalence between (5) and 6 is $\boldsymbol{Y}_{VV} = \boldsymbol{y}_V\boldsymbol{y}_V^T$ from the rank-1 constraint on the matrix $\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix}$. We will denote the (i,j)th element of $\boldsymbol{Y}_{VV}$ by Y_{ij}. The equality $Y_{ii} = y_i^2$ for $i = 1, \dots, |V|$ should holds for feasible solution of (6), hence $\begin{pmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_i\boldsymbol{e}_i^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0$ leads to the binary constraint $y_i \in \{-1, 1\}$. In (6), we introduced a redundant constraint $(\boldsymbol{e}_V\boldsymbol{e}_V^T) \bullet \boldsymbol{Y}_{VV} = \bar{N}^2$ that was derived from $(\boldsymbol{e}_V^T\boldsymbol{y}_V)^2 = \bar{N}^2$ and $\boldsymbol{Y}_{VV} = \boldsymbol{y}_V\boldsymbol{y}_V^T$. It is known that redundant constraints of this type make the SDP relaxation tighter, and we can often obtain better approximate solution. The hardest constraint in (6) is the rank-1 constraint. This constraint embraces a nature of combinatorial optimization. By removing this hardest constraint, we build an SDP relaxation problem and we denote its optimal value by OPT_{SDP}.

$$\begin{array}{llll} OPT_{SDP} & := & \max & : \begin{pmatrix} 0 & \bar{\boldsymbol{g}}_V^T \\ \bar{\boldsymbol{g}}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} + \bar{g} \\ & & \text{subject to} & : \begin{pmatrix} -2\bar{\theta} & \bar{\boldsymbol{c}}_F^T \\ \bar{\boldsymbol{c}}_F & \boldsymbol{A}_{VV} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \leq 0, \\ & & & \begin{pmatrix} -2\bar{N} & \boldsymbol{e}_V^T \\ \boldsymbol{e}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\ & & & \begin{pmatrix} -\bar{N}^2 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_V\boldsymbol{e}_V^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\ & & & \begin{pmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_i\boldsymbol{e}_i^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0 \text{ for } i \in V, \\ & & & \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \succeq \boldsymbol{O}. \end{array} \tag{7}$$

When we further relax the positive semidefinite constraint $\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \succeq \boldsymbol{O}$, we can obtain an LP relaxation problem. In general, a matrix $\boldsymbol{X} \in \mathbb{S}^n$ is positive semidefinite if and only if $\boldsymbol{u}^T\boldsymbol{X}\boldsymbol{u} \geq 0$ for $\forall \boldsymbol{u} \in \mathbb{R}^n$. For the positive semidefinite constraint of (7), we choose a set of vectors $\boldsymbol{u}_{ij} = \boldsymbol{e}_i - \boldsymbol{e}_j \in \mathbb{R}^{1+|V|}$ for $i = 1, \dots, |V|$ and $j = i+1, \dots, |V|+1$ as a subset of $\mathbb{R}^{1+|V|}$. We use $\hat{W}$ to denote the non-diagonal upper-triangular position of $\boldsymbol{Y}_{VV}$, that is $\hat{W} := \{(i,j) \in V \times V : i < j\}$. The key step to derive an LP relaxation problem is the following step:

$$\begin{array}{cl} & \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \succeq \boldsymbol{O} \\ \Leftrightarrow & \boldsymbol{u}^T \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \boldsymbol{u} \geq 0 \text{ for } \forall \boldsymbol{u} \in \mathbb{R}^{1+|V|} \\ \overset{\text{(relaxation)}}{\Rightarrow} & \boldsymbol{u}_{ij}^T \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \boldsymbol{u}_{ij} \geq 0 \text{ for } i = 1, \dots, |V| \text{ and } j = i+1, \dots, |V|+1 \\ \Leftrightarrow & \begin{cases} Y_{ii} \geq y_i^2 & \text{for } \quad i \in V, \\ Y_{ii}Y_{jj} \geq Y_{ij}^2 & \text{for } \quad (i,j) \in \hat{W}. \end{cases} \end{array}$$

From the constraints $Y_{ii} = 1$ for $i \in V$ in (7), the constraints $Y_{ii} \geq y_i^2$ and $Y_{ii}Y_{jj} \geq Y_{ij}^2$ are linear constraints in nature. Consequently, we reach an LP relaxation problem, whose optimal value is denoted as OPT_{LP}.

$$
\begin{array}{rcll}
OPT_{LP} & = & \max & : \begin{pmatrix} 0 & \bar{\boldsymbol{g}}_V^T \\ \bar{\boldsymbol{g}}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} + \bar{g} \\
 & & \text{subject to} & : \begin{pmatrix} -2\bar{\theta} & \bar{\boldsymbol{c}}_F^T \\ \bar{\boldsymbol{c}}_F & \boldsymbol{A}_{VV} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \leq 0, \\
 & & & \quad \begin{pmatrix} -2\bar{N} & \boldsymbol{e}_V^T \\ \boldsymbol{e}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\
 & & & \quad \begin{pmatrix} -\bar{N}^2 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_V \boldsymbol{e}_V^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0, \\
 & & & \quad \begin{pmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_i \boldsymbol{e}_i^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0 \text{ for } i \in V, \\
 & & & \quad -1 \leq y_i \leq 1 \quad \text{for} \quad i \in V, \\
 & & & \quad -1 \leq Y_{ij} \leq 1 \quad \text{for} \quad (i,j) \in \hat{W}, \\
 & & & \quad \boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}.
\end{array}
\tag{8}
$$

We now move our focus to an SOCP relaxation problem. In a similar way to the above step that derives (8) from (7), it may be possible to apply an SOCP relaxation technique developed in [17] to (7). In contrast, we utilize a continuous relaxation technique that converts the binary constraint $x_i \in \left\{0, \frac{1}{N}\right\}$ into a continuous constraint $0 \leq x_i \leq \frac{1}{N}$. The main reason of this continuous relaxation is that we can keep the efficient SOCP formula of [47] that extensively exploits a structural sparsity of the Wright numerator matrix $\boldsymbol{A}$.

A second-order cone of dimension q is defined by $\mathcal{K}^q := \left\{ \boldsymbol{x} \in \mathbb{R}^q : x_1 \geq \sqrt{\sum_{i=2}^n x_i^2} \right\}$. A standard form of second-order cone programming (SOCP) problem in this paper is given as follows:

$$
\begin{array}{ll}
\max & : \boldsymbol{c}^T \boldsymbol{x} \\
\text{subject to} & : \boldsymbol{F}\boldsymbol{x} = \boldsymbol{b}, \\
 & \quad \boldsymbol{h} - \boldsymbol{H}\boldsymbol{x} \in \mathcal{K}^q.
\end{array}
\tag{9}
$$

The decision variable here is $\boldsymbol{x} \in \mathbb{R}^n$ and the objective function is a linear function with a constant vector $\boldsymbol{c} \in \mathbb{R}^n$. The linear constraints are encoded with a matrix $\boldsymbol{F} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$. The second-oder cone constraint is given with a vector $\boldsymbol{h} \in \mathbb{R}^q$ and a matrix $\boldsymbol{H} \in \mathbb{R}^{q \times n}$. A more general SOCP formulation often includes a Cartesian product of second-order cones. However, only one second-order cone is enough for the discussions in this paper.

Yamashita et al. [47] introduced a new vector $\boldsymbol{z} := \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^Z$, and converted the quadratic constraint $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq 2\theta$ into $||\boldsymbol{B}\boldsymbol{z}|| \leq \sqrt{2\theta}$ with a matrix $\boldsymbol{B} \in \mathbb{R}^{Z \times Z}$ that satisfies $\boldsymbol{B}^T \boldsymbol{B} = \boldsymbol{A}^{-1}$. Though the Wright numerator matrix $\boldsymbol{A}$ itself is not a sparse matrix, the matrices $\boldsymbol{A}^{-1}$ and $\boldsymbol{B}$ possess favorable sparsity. The computation time reduction reported in [47] was mainly derived from these sparsity. Using these new vector $\boldsymbol{z}$ and matrix $\boldsymbol{B}$, we transformed the ED problem (2) into the following SOCP problem with integer constraints;

$$\begin{array}{lll} \max & : & (\boldsymbol{A}^{-1}\boldsymbol{g})^T\boldsymbol{z} \\ \text{subject to} & : & (\boldsymbol{A}^{-1}\boldsymbol{e})^T\boldsymbol{z} = 1, \\ & & \begin{pmatrix} \sqrt{2\theta} \\ \boldsymbol{B}\boldsymbol{z} \end{pmatrix} \in \mathcal{K}^{1+Z}, \\ & & [\boldsymbol{A}^{-1}\boldsymbol{z}]_i \in \left\{0, \frac{1}{N}\right\} \quad \text{for} \quad i \in V, \\ & & [\boldsymbol{A}^{-1}\boldsymbol{z}]_i = c_i \qquad \text{for} \quad i \in F. \end{array}$$

Here, we use the notation $[\boldsymbol{A}^{-1}\boldsymbol{z}]_i$ to denote the ith element of $\boldsymbol{A}^{-1}\boldsymbol{z}$. It may seem that we would remove $\boldsymbol{x}_F$ from this formulation by fixing $\boldsymbol{x}_F = \boldsymbol{c}_F$ and reduce the sizes the problem. However, such elimination would strongly diminish the efficiency of the SOCP problem, since it completely destroys the favorable sparsity that appear in $\boldsymbol{A}^{-1}$ and $\boldsymbol{B}$.

By applying the continuous relaxation to the binary constraints, we obtain an SOCP relaxation problem of the ED problem;

$$\begin{array}{llll} OPT_{SOCP} & := & \max & : & (\boldsymbol{A}^{-1}\boldsymbol{g})^T\boldsymbol{z} \\ & & \text{subject to} & : & (\boldsymbol{A}^{-1}\boldsymbol{e})^T\boldsymbol{z} = 1, \\ & & & & \begin{pmatrix} \sqrt{2\theta} \\ \boldsymbol{B}\boldsymbol{z} \end{pmatrix} \in \mathcal{K}^{1+Z}, \\ & & & & 0 \leq [\boldsymbol{A}^{-1}\boldsymbol{z}]_i \leq \frac{1}{N} \quad \text{for} \quad i \in V, \\ & & & & [\boldsymbol{A}^{-1}\boldsymbol{z}]_i = c_i \qquad \text{for} \quad i \in F. \end{array} \tag{10}$$

OPT_{ED}, OPT_{SDP}, OPT_{LP} and OPT_{SOCP}, respectively. From the derivation of the LP relaxation problem (8), it is natural that the SDP relaxation problem (7) gives closer an optimal value than the LP relaxation problem, that is, we know $OPT_{SDP} \leq OPT_{LP}$. In contrast, the relation of the SOCP relaxation (10) is not so explicit, since the SOCP relaxation was derived by a continuous relaxation independently from the SDP or LP relaxation.

The strength of these relaxation problems can be summarized in Lemma 2.3. For the discussion there, we prepare some notation and introduce an assumption. We use $\mathcal{S}_m(\boldsymbol{v})$ to denote the sum of the m smallest elements of $\boldsymbol{v} \in \mathbb{R}^n$. More precisely, when $\hat{v}_1 \leq \hat{v}_2 \leq \ldots \leq \hat{v}_n$ is the sorted vector of $\boldsymbol{v}$ in the ascending order, the definition of $\mathcal{S}_m(\boldsymbol{v})$ is given by $\mathcal{S}_m(\boldsymbol{v}) := \sum_{i=1}^m \hat{v}_i$. The symbol $\hat{A}_{\hat{W}}$ indicates the set of the collection of $\boldsymbol{A}_{VV}$ with respect to $\hat{W}$, that is, $\hat{A}_{\hat{W}} := \left\{A_{ij} : (i,j) \in \hat{W}\right\}$. We define a vector $\hat{\boldsymbol{y}}_V \in \mathbb{R}^{|V|}$ by $[\hat{y}_V]_i := 1$ for $i = 1, \ldots, N-p$ and $[\hat{y}_V]_i := -1$ for $i = N-p+1, \ldots, |V|$. This vector satisfies $\boldsymbol{e}_V^T\hat{\boldsymbol{y}}_V = \bar{N}$. In the following this discussion, we make the following assumption on the input data of the ED problem (2). From preliminary numerical tests, we verified that this assumption holds for practical datasets of pine orchards and datasets generated by simulations. The details of these dataset will be described in Section 5.

Assumption 2.2. *The input data of (2) satisfies*

$$\mathcal{S}_{\hat{N}}(\hat{A}_{\hat{W}}) \leq \frac{2\bar{\theta} - 2\,\mathit{Trace}(\boldsymbol{A}_{VV}) + \boldsymbol{e}_V^T\boldsymbol{A}_{VV}\boldsymbol{e}_V - 2\bar{\boldsymbol{c}}_F^T\hat{\boldsymbol{y}}}{4},$$

where $\hat{N} := \frac{\bar{N}^2 + |V|^2 - 2|V|}{4}$.

We should ensure that $\hat{N}$ is a positive integer, otherwise we need to manage a fractional number in the definition of $\mathcal{S}$. The positiveness is derived from $\bar{N}^2 + |V|^2 - 2|V| \geq \bar{N}^2 + 1 \geq 1$ by $|V| \geq 2$ of Remark 2.1, and $\hat{N}$ is integer by

$$\begin{aligned}\bar{N}^2 + |V|^2 - 2|V| &= \left\{2N(1 - \boldsymbol{e}_F^T\boldsymbol{c}_F) - |V|\right\}^2 + |V|^2 - 2|V| \\ &= \left\{2N(1 - \frac{p}{N}) - |V|\right\}^2 + |V|^2 - 2|V| \\ &= 4\left\{(N-p)^2 - |V|(N-p) + \frac{|V|(|V|-1)}{2}\right\}.\end{aligned}$$

We are now prepared to examine the relation between the relaxation problems.

Lemma 2.3. *It holds for the optimal values of the relaxation problems that*

$$OPT_{ED} \leq OPT_{SDP} \leq OPT_{SOCP}.$$

Furthermore, if Assumption 2.2 holds, then

$$OPT_{ED} \leq OPT_{SDP} \leq OPT_{SOCP} \leq OPT_{LP}.$$

Proof: [$OPT_{ED} \leq OPT_{SDP}$] When we derived (7), we ignored the rank-1 constraint in (6). From this derivation, for any feasible solution $\boldsymbol{x} \in \mathbb{R}^Z$ of (2), the corresponding vector $\boldsymbol{y}_V \in \mathbb{R}^{|V|}$ through the connections $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_V \\ \boldsymbol{c}_F \end{pmatrix}$, then $\boldsymbol{y}_V = 2N\boldsymbol{x}_V - \boldsymbol{e}_V$ is also a feasible solution of (7). Furthermore, from these connections hold, it holds that $\boldsymbol{g}^T\boldsymbol{x} = \bar{\boldsymbol{g}}^T\boldsymbol{y}_V + \bar{g}$. The objective functions of (6) and (7) are same and the feasible region of (7) is wider than that of (6) substantially, hence, we have $OPT_{ED} \leq OPT_{SDP}$.

[$OPT_{SDP} \leq OPT_{SOCP}$] We take any feasible solution $\boldsymbol{y}_V \in \mathbb{R}^{|V|}$ and $\boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}$ of (7). It is enough to check that $\boldsymbol{z} = \boldsymbol{A}\begin{pmatrix} \frac{\boldsymbol{y}_V + \boldsymbol{e}_V}{2N} \\ \boldsymbol{c}_F \end{pmatrix}$ is a feasible solution of (10).

From $\begin{pmatrix} -2\bar{N} & \boldsymbol{e}_V^T \\ \boldsymbol{e}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0$, we obtain $\boldsymbol{e}_V^T\boldsymbol{y}_V = \bar{N} = 2N(1 - \boldsymbol{e}_F^T\boldsymbol{c}_F) - |V|$, hence,

$$(\boldsymbol{A}^{-1}\boldsymbol{e})^T\boldsymbol{z} = \begin{pmatrix} \boldsymbol{e}_V \\ \boldsymbol{e}_F \end{pmatrix}^T \begin{pmatrix} \frac{\boldsymbol{y}_V + \boldsymbol{e}_V}{2N} \\ \boldsymbol{c}_F \end{pmatrix} = \frac{\boldsymbol{e}_V^T\boldsymbol{y}_V + |V|}{2N} + \boldsymbol{e}_F^T\boldsymbol{c}_F = 1.$$

By applying the Schur complement to the positive semidefinite condition $\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \succeq \boldsymbol{O}$, it holds $\boldsymbol{Y}_{VV} - \boldsymbol{y}_V\boldsymbol{y}_V^T \succeq \boldsymbol{O}$. Since $\boldsymbol{A} \bullet \boldsymbol{X} \geq 0$ holds for any two positive semidefinite matrices of the same dimension $\boldsymbol{A}$ and $\boldsymbol{X}$ [34] and the Wright numerator matrix is always positive definite, it holds $\boldsymbol{A}_{VV} \bullet (\boldsymbol{Y}_{VV} - \boldsymbol{y}_V\boldsymbol{y}_V^T) \geq 0$, therefore, $\boldsymbol{A}_{VV} \bullet \boldsymbol{Y}_{VV} \geq \boldsymbol{y}_V^T\boldsymbol{A}_{VV}\boldsymbol{y}_V$. Using the relation $\begin{pmatrix} -2\bar{\theta} & \bar{\boldsymbol{c}}_F^T \\ \bar{\boldsymbol{c}}_F & \boldsymbol{A}_{VV} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \leq 0$, we obtain $\boldsymbol{y}_V^T\boldsymbol{A}_{VV}\boldsymbol{y}_V + 2\bar{\boldsymbol{c}}_F^T\boldsymbol{y}_V \leq 2\bar{\theta}$. From the definitions of $\boldsymbol{y}_V, \bar{\boldsymbol{c}}_F, \bar{\theta}, \boldsymbol{B}$ and $\boldsymbol{z}$, we can derive $\boldsymbol{z}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{z} \leq 2\theta$, therefore, $\begin{pmatrix} \sqrt{2\theta} \\ \boldsymbol{B}\boldsymbol{z} \end{pmatrix} \in \mathcal{K}^{1+Z}$. From

$\boldsymbol{Y}_{VV} - \boldsymbol{y}_V \boldsymbol{y}_V^T \succeq \boldsymbol{O}$, we also have $Y_{ii} \geq y_i^2$ for $i = 1, \ldots, |V|$. Furthermore, due to the constraint $\begin{pmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{e}_i \boldsymbol{e}_i^T \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} = 0$, it holds $Y_{ii} = 1$ for $i = 1, \ldots, |V|$, consequently $-\boldsymbol{e}_V \leq \boldsymbol{y}_V \leq \boldsymbol{e}_V$. From $\boldsymbol{A}^{-1}\boldsymbol{z} = \begin{pmatrix} \frac{\boldsymbol{y}_V + \boldsymbol{e}_V}{2N} \\ \boldsymbol{c}_F \end{pmatrix}$, it is now clear that $0 \leq [\boldsymbol{A}^{-1}\boldsymbol{z}]_i \leq \frac{1}{N}$ for $i \in V$ and that $[\boldsymbol{A}^{-1}\boldsymbol{z}]_i = c_i$ for $i \in F$. Furthermore, the objective value of (7) at $\boldsymbol{y}_V$ is same as that of (10) at $\boldsymbol{z}$ if $\boldsymbol{z} = \boldsymbol{A}\begin{pmatrix} \frac{\boldsymbol{y}_V + \boldsymbol{e}_V}{2N} \\ \boldsymbol{c}_F \end{pmatrix}$. Hence, we obtain $OPT_{SDP} \leq OPT_{SOCP}$.

$[OPT_{SOCP} \leq OPT_{LP}]$ We first consider an LP problem

$$\begin{array}{lll} \min & : & \boldsymbol{c}^T \boldsymbol{\eta} \\ \text{subject to} & : & \sum_{i=1}^n \eta_i = K, \\ & & 0 \leq \eta_i \leq 1 \text{ for } i = 1, \ldots, n, \end{array} \tag{11}$$

where the decision variable is $\boldsymbol{\eta} \in \mathbb{R}^n$ and the input vector is $\boldsymbol{c} \in \mathbb{R}^n$ and K is a positive integer. The optimal value of this LP problem is $\mathcal{S}_K(\boldsymbol{c})$ and this value can be attained at $\hat{\boldsymbol{\eta}} \in \mathbb{R}^n$ such that $\hat{\eta}_i = 1$ for $i = 1, \ldots, K$ and $\hat{\eta}_i = 0$ for $i = K+1, \ldots, n$.

If we ignore the quadratic constraint of (10) and we reverse the variable into $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{z}$, we obtain an optimization problem of form

$$\begin{array}{lll} \max & : & \boldsymbol{g}_V^T \boldsymbol{x}_V + \boldsymbol{g}_F^T \boldsymbol{c}_F, \\ \text{subject to} & : & \boldsymbol{e}_V^T \boldsymbol{x}_V = 1 - \frac{p}{N}, \\ & & 0 \leq x_i \leq \frac{1}{N} \quad \text{for} \quad i \in V. \end{array} \tag{12}$$

Since $g_1 \geq g_2 \geq \ldots g_{|V|}$, the optimal value of (12) is given by $-\frac{\mathcal{S}_{N-p}(-\boldsymbol{g}_V)}{N} + \boldsymbol{g}_F^T \boldsymbol{c}_F$ in a similar way to (11) and an optimal solution is $\hat{\boldsymbol{x}}_V := \frac{\hat{\boldsymbol{y}}_V + \boldsymbol{e}_V}{2N}$. Therefore, it holds that $OPT_{SOCP} \leq \boldsymbol{g}_V^T \hat{\boldsymbol{x}}_V + \boldsymbol{g}_F^T \boldsymbol{c}_F$.

Next, we define ρ_{LP} to denote the optimal value of the following LP problem;

$$\begin{array}{lllll} \rho_{LP} & := & \min & : & \boldsymbol{A}_{VV} \bullet \boldsymbol{Y}_{VV} \\ & & \text{subject to} & : & (\boldsymbol{e}_V \boldsymbol{e}_V^T) \bullet \boldsymbol{Y}_{VV} = \bar{N}^2, \\ & & & & Y_{ii} = 1 \text{ for } i \in V, \\ & & & & -1 \leq Y_{ij} \leq 1 \text{ for } (i,j) \in \hat{W}, \\ & & & & \boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}. \end{array} \tag{13}$$

We convert this problem introducing $\bar{X}_{ij} := \frac{Y_{ij}+1}{2}$ for $(i,j) \in \hat{W}$. The following LP problem is equivalent to (13), therefore, its optimal value must be ρ_{LP}.

$$\begin{array}{lllll} \rho_{LP} & = & \min & : & 4\sum_{(i,j)\in\hat{W}} A_{ij}\bar{X}_{ij} - 2\sum_{(i,j)\in\hat{W}} A_{ij} + \text{Trace}(\boldsymbol{A}_{VV}) \\ & & \text{subject to} & : & \sum_{(i,j)\in\hat{W}} \bar{X}_{ij} = \hat{N}, \\ & & & & 0 \leq \bar{X}_{ij} \leq 1 \text{ for } (i,j) \in \hat{W}. \end{array} \tag{14}$$

The structure of this problem is same as (11), hence, it holds that $\rho_{LP} = 4\mathcal{S}_{\hat{N}}(\hat{A}_{\hat{N}}) - \boldsymbol{e}_V^T \boldsymbol{A}_{VV} \boldsymbol{e}_V + 2\text{Trace}(\boldsymbol{A}_{VV})$.

Let $\hat{\boldsymbol{Y}}_{VV}$ be a part of an optimal solution of (13). From Assumption 2.2, it holds that

$$\boldsymbol{A}_{VV} \bullet \hat{\boldsymbol{Y}}_{VV} + 2\bar{\boldsymbol{c}}_F^T \hat{\boldsymbol{y}}_V = \rho_{LP} + 2\bar{\boldsymbol{c}}_F^T \hat{\boldsymbol{y}}_V = 4\mathcal{S}_{\hat{N}}(\hat{A}_{\hat{N}}) - \boldsymbol{e}_V^T \boldsymbol{A}_{VV} \boldsymbol{e}_V + 2\text{Trace}(\boldsymbol{A}) + 2\bar{\boldsymbol{c}}_F^T \hat{\boldsymbol{y}}_V \leq 2\bar{\theta}.$$

Furthermore, $\hat{\boldsymbol{y}}_V$ satisfies $-1 \leq \hat{y}_i \leq 1$ for $i \in V$ and $\boldsymbol{e}_V^T \hat{\boldsymbol{y}}_V = \bar{N}$ by its definition and $\hat{\boldsymbol{Y}}_{VV}$ satisfies all the constraints of (13). Consequently, the pair $\hat{\boldsymbol{y}}_V$ and $\hat{\boldsymbol{Y}}_{VV}$ is a feasible solution of (8) and this leads to the inequality we wanted to obtain.

$$OPT_{LP} \geq \begin{pmatrix} 0 & \bar{\boldsymbol{g}}_V^T \\ \bar{\boldsymbol{g}}_V & \boldsymbol{O} \end{pmatrix} \bullet \begin{pmatrix} 1 & \hat{\boldsymbol{y}}_V^T \\ \hat{\boldsymbol{y}}_V & \hat{\boldsymbol{Y}}_{VV} \end{pmatrix} + \bar{g} = 2\bar{\boldsymbol{g}}_V^T \hat{\boldsymbol{y}}_V + \bar{g} = \boldsymbol{g}_V^T \hat{\boldsymbol{x}}_V + \boldsymbol{g}_F^T \boldsymbol{c}_F \geq OPT_{SOCP}.$$

□

Remark 2.4. *Since an optimal solution of a further relaxation problem of (8)*

$$\begin{array}{lll} \max & : & 2\bar{\boldsymbol{g}}_V^T \boldsymbol{y}_V + \bar{g} \\ \textit{subject to} & : & \boldsymbol{e}_V^T \boldsymbol{y}_V = \bar{N}, \\ & & -1 \leq y_i \leq 1 \textit{ for } i \in V \end{array}$$

is $\hat{\boldsymbol{y}}_V$, its optimal value $2\bar{\boldsymbol{g}}_V^T \hat{\boldsymbol{y}}_V + \bar{\boldsymbol{g}}$ must be an upper bound of OPT_{LP}. On the other hand, from the proof of Lemma 2.3, when Assumption 2.2 holds, there exists some $\hat{\boldsymbol{Y}}_{VV} \in \mathbb{S}^{|V|}$ such that the pair of $\hat{\boldsymbol{y}}_V$ and $\hat{\boldsymbol{Y}}_{VV}$ is a feasible solution of (8) with the objective value $2\bar{\boldsymbol{g}}_V^T \hat{\boldsymbol{y}}_V + \bar{\boldsymbol{g}}$. Therefore, $\hat{\boldsymbol{y}}_V$ is also an optimal solution of (8). This indicates that we can obtain the solution of (8) at the computation cost for sorting $\boldsymbol{g}_V$ instead of solving (8) as an LP problem, when Assumption 2.2 holds.

This remark implies that the LP relaxation (8) is not so tight against the original ED problem (2). In contrast, we observed through preliminary numerical tests that the vector $\begin{pmatrix} \hat{\boldsymbol{x}}_V \\ \boldsymbol{c}_F \end{pmatrix}$ defined with an optimal solution $\hat{\boldsymbol{x}}_V$ of (12), is not always a feasible solution of (10) even if Assumption 2.2 holds. Therefore, the feasible region of the SOCP relaxation problem is strictly narrower than that of the LP relaxation problem, and the SOCP relaxation gives a tighter approximation than the LP relaxation in general, even though the relaxation were derived independently.

Remark 2.5. *The SDP relaxation problem (7) has no interior-feasible point.*

If a pair $\boldsymbol{y}_V \in \mathbb{R}^{|V|}$ and $\boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}$ satisfies all the constraint of (7), the pair is a feasible point. When the matrix $\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix}$ is a positive definite matrix for some feasible point $\boldsymbol{y}_V \in \mathbb{R}^{|V|}$ and $\boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}$, we say that (7) has an interior-feasible point. We can show that $\begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix}$ is not positive definite for any feasible point of (7). To show this, we take a feasible point $\boldsymbol{y}_V \in \mathbb{R}^{|V|}$ and $\boldsymbol{Y}_{VV} \in \mathbb{S}^{|V|}$. Then, we have $\boldsymbol{e}_V^T \boldsymbol{y}_V = \bar{N}$ and $(\boldsymbol{e}_V \boldsymbol{e}_V^T) \bullet \boldsymbol{Y}_{VV} = \bar{N}^2$. If $\bar{N} \neq 0$, it holds

$$\begin{pmatrix} 1 \\ -\boldsymbol{e}_V/\bar{N} \end{pmatrix}^T \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{e}_V/\bar{N} \end{pmatrix} = 1 - 2\boldsymbol{e}_V^T \boldsymbol{y}_V/\bar{N} + \boldsymbol{e}_V^T \boldsymbol{Y}_{VV} \boldsymbol{e}_V/\bar{N}^2 = 0.$$

In addition, for the case $\bar{N} = 0$, it holds

$$\begin{pmatrix} 0 \\ \boldsymbol{e}_V \end{pmatrix}^T \begin{pmatrix} 1 & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix} \begin{pmatrix} 0 \\ \bar{N} \end{pmatrix} = \boldsymbol{e}_V^T \boldsymbol{Y}_{VV} \boldsymbol{e}_V = \bar{N}^2 = 0.$$

In either case, there exists a nonzero vector that makes the quadratic from zero, the matrix is not positive definite, therefore, (7) has no interior-feasible point.

3 Theoretical evaluation of the SDP relaxation problems with a randomized algorithm

The solutions obtained from the conic relaxation problem (7), (8) and (10) are not always a feasible solution of the ED problem (2), since we ignored some constraints of the NP-hard problem to derive the conic relaxation problems that are solvable in polynomial time. When SDP relaxation approaches are used, examine randomized algorithms often follow to generate feasible solutions. A randomized algorithm using the solutions obtained through SDP relaxation problems was first introduced for max-cut problems in [11]. They showed that the expectation objective value obtained by their randomized algorithm on average was at least 0.878 of that of an SDP relaxation problem. Since the optimal value of a max cut problem exists between an objective value of any feasible solution and the value obtained from the SDP relaxation problem, their algorithm have an expected approximation factor of 0.878. Many researches followed [11] to extend its results to more general quadratic-constraint problems using the framework of SDP relaxation methods. Among them, Tseng [36] discussed one of the most general cases and gave its probabilistic analysis. Wu et al. [44] also analyzed the expectation values using a different randomized algorithm.

In this section, we employ the result of [36] to give theoretical bounds on the expected objective value of a randomized algorithm. Tseng [36] applied the SDP relaxation methods to a quadratically-constrained quadratic programming (QCQP) problem:

$$\begin{array}{rcll} \overline{OPT}_{QCQP} & := & \max & : \ \boldsymbol{y}^T\boldsymbol{A}^0\boldsymbol{y} + (\boldsymbol{b}^0)^T\boldsymbol{y} + c^0 \\ & & \text{subject to} & : \ \boldsymbol{y}^T\boldsymbol{A}^k\boldsymbol{y} + (\boldsymbol{b}^k)^T\boldsymbol{y} + c^k \leq 0 \text{ for } k = 1,\ldots,m. \end{array} \quad (15)$$

Here, the variable is $\boldsymbol{y} \in \mathbb{R}^n$, while the input data are $\boldsymbol{A}^0,\ldots,\boldsymbol{A}^m \in \mathbb{S}^n$, $\boldsymbol{b}^0,\ldots,\boldsymbol{b}^m \in \mathbb{R}^n$ and $c^0, c^1 \ldots, c^m \in \mathbb{R}$. For simplicity, the constant in the objective function is fixed to $c^0 = 0$. The QCQP originally discussed in [36] is a minimization problem, but we consider a maximization problem since the ED problem (2) is a maximization problem.

When we apply the lift-and-project method of Lovász and Schrijver [20] to (15), the resultant SDP relaxation problem is given as follows:

$$\begin{array}{rcll} \overline{OPT}_{SDP} & := & \max & : \ \boldsymbol{B}^0 \bullet \boldsymbol{Y} \\ & & \text{subject to} & : \ \boldsymbol{B}^k \bullet \boldsymbol{Y} \leq 0 \text{ for } k = 1,\ldots,m, \\ & & & \ \ \ \boldsymbol{B}^{m+1} \bullet \boldsymbol{Y} = 1, \quad \boldsymbol{Y} \succeq \boldsymbol{O} \end{array} \quad (16)$$

where $\boldsymbol{B}^k := \begin{pmatrix} c^k & (\boldsymbol{b}^k)^T \\ \boldsymbol{b}^k & \boldsymbol{A}^k \end{pmatrix}$ for $k = 0,\ldots,m$ and $\boldsymbol{B}^{m+1} := \begin{pmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{O} \end{pmatrix}$, and the decision variable is $\boldsymbol{Y} \in \mathbb{S}^{1+n}$. In the following discussions, we start the row or column index of $\boldsymbol{Y}$ from zero, therefore, the elements of $\boldsymbol{Y}$ are denoted by $Y_{00}, Y_{01}, \ldots, Y_{nn}$. It is known that if we add the rank-1 constraint rank$(\boldsymbol{Y}) = 1$ to (16), the two problems (15) and (16) are equivalent. In other words, we ignored the rank-1 constraint from (15) to derive (16), hence, $\overline{OPT}_{QCQP} \leq \overline{OPT}_{SDP}$.

The randomized algorithm of [36] can be summarized as follow. We assume that (15) and (16) are feasible and that (16) has an optimal solution, denoted as $\boldsymbol{Y}^*$. This solution $\boldsymbol{Y}^*$ is factorzied with a matrix $\boldsymbol{V} \in \mathbb{R}^{(1+n)\times(1+n)}$ such that $\boldsymbol{Y}^* = \boldsymbol{V}^T\boldsymbol{V}$. Such $\boldsymbol{V}$ is available, for example, by the Cholesky factorization or the eigenvalue decomposition. We use $\boldsymbol{v}^0, \boldsymbol{v}^1, \ldots, \boldsymbol{v}^n \in \mathbb{R}^{1+n}$ to denote the columns of $\boldsymbol{V}$. Then, a vector $\boldsymbol{v} \in \mathbb{R}^{1+n}$ is chosen randomly from the unit sphere in $\mathbb{R}^{1+n}$

based on uniform distribution. Finally, the randomized algorithm outputs a solution $\tilde{\boldsymbol{y}} \in \mathbb{R}^{1+n}$ defined by

$$\tilde{y}_i := \sqrt{Y^*_{ii}}\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0)\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^i) \text{ for } i = 0, \dots, n$$

where $\text{sign}(a) = 1$ if $a \geq 0$ and $\text{sign}(a) = -1$ if $a < 0$. We remark that from the definition of $\boldsymbol{B}^{m+1}$, it always holds that $Y^*_{00} = 1$, hence, $\tilde{y}_0 = \sqrt{1}(\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0))^2 = 1$.

The set $\mathcal{I}$ is introduced to indicate diagonal-matrix constraints of (15);

$$\mathcal{I} := \left\{k \in \{1, 2, \dots, m\} : \boldsymbol{A}^k \text{ is a diagonal matrix and } \boldsymbol{b}^k = \boldsymbol{0}\right\}.$$

To measure a shift in the objective function, ρ^0_{SDP} is defined as the optimal value of the following SDP problem

$$\begin{array}{rlll} \rho^0_{SDP} & := & \min & : \boldsymbol{B}^0 \bullet \boldsymbol{Y} \\ & & \text{subject to} & : \boldsymbol{B}^k \bullet \boldsymbol{Y} = \boldsymbol{B}^k \bullet \boldsymbol{Y}^* \text{ for } k \in \mathcal{I}, \\ & & & \boldsymbol{B}^{m+1} \bullet \boldsymbol{Y} = 1, \quad \boldsymbol{Y} \succeq \boldsymbol{O}. \end{array} \tag{17}$$

Tseng [36] showed a relation between the expected objective value of the generated solution $\tilde{\boldsymbol{y}}$ and the optimal values of the SDP problems.

Theorem 3.1. [36, Theorem 2] *If the SDP relaxation problem (16) has an optimal solution $\boldsymbol{Y}^*$ and a set $\left\{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y}^T\boldsymbol{A}^k\boldsymbol{y} + (\boldsymbol{b}^k)^T\boldsymbol{y} + c^k \leq 0, k \in \mathcal{I}\right\}$ is bounded, then*

$$E[\tilde{\boldsymbol{y}}^T\boldsymbol{A}^0\tilde{\boldsymbol{y}} + (\boldsymbol{b}^0)^T\boldsymbol{y}] \geq \frac{2}{\pi}\overline{OPT}_{SDP} + \left(1 - \frac{2}{\pi}\right)\rho^0_{SDP}.$$

Let us return to the ED problem (2). We analyze the performance of the output solution $\tilde{\boldsymbol{y}}_V$ that is generated by the above randomized algorithm using the optimal solution $\boldsymbol{Y}^*$ of the SDP relaxation problem (7). From the form of (7), the objective value at $\tilde{\boldsymbol{y}}_V$ is $2\boldsymbol{g}_V^T\tilde{\boldsymbol{y}} + \bar{g}$. The following lemma provides a theoretical aspects on the expected value of this objective function.

Lemma 3.2. *For the ED problem (2), the expected objective value obtained through the randomized algorithm is bounded by*

$$\frac{2}{\pi}OPT_{SDP} + \left(1 - \frac{2}{\pi}\right)(-2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g}) \leq E[2\bar{\boldsymbol{g}}_V^T\tilde{\boldsymbol{y}}_V + \bar{g}] \leq \alpha OPT_{SDP} + (1 - \alpha)(2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g}),$$

where $\alpha := \min\left\{\frac{2}{\pi}\frac{\theta}{1-\cos\theta} : 0 \leq \theta \leq \pi\right\} \approx 0.878$.

Proof:

First, we derive the lower bound of the objective function by use of Theorem 3.1. To embed the SDP relaxation problem (7) arising from the ED problem into the framework developed in [36], we embed the variable vector $\boldsymbol{y}_V$ and matrix $\boldsymbol{Y}_{VV}$ into the matrix $\boldsymbol{Y} \in \mathbb{S}^{1+|V|}$ as $\boldsymbol{Y} = \begin{pmatrix} Y_{00} & \boldsymbol{y}_V^T \\ \boldsymbol{y}_V & \boldsymbol{Y}_{VV} \end{pmatrix}$. In particular, we identify $y_i = Y_{0i} = Y_{i0}$ for $i = 1, \dots, |V|$. For the input matrices $\boldsymbol{B}^0, \ldots, \boldsymbol{B}^{2|V|+5}$, we prepare

$$\begin{cases} c^0 = 0, & \boldsymbol{b}^0 = \bar{\boldsymbol{g}}_V, & \boldsymbol{A}^0 = \boldsymbol{O}, \\ c^1 = -2\bar{\theta}, & \boldsymbol{b}^1 = \bar{\boldsymbol{c}}_F, & \boldsymbol{A}^1 = \boldsymbol{A}_{VV}, \\ c^2 = -2\bar{N}, & \boldsymbol{b}^2 = \boldsymbol{e}_V, & \boldsymbol{A}^2 = \boldsymbol{O}, \\ c^3 = 2\bar{N}, & \boldsymbol{b}^3 = -\boldsymbol{e}_V, & \boldsymbol{A}^3 = \boldsymbol{O}, \\ c^4 = -\bar{N}^2, & \boldsymbol{b}^4 = \boldsymbol{0}, & \boldsymbol{A}^4 = \boldsymbol{e}_V\boldsymbol{e}_V^T, \\ c^5 = \bar{N}^2, & \boldsymbol{b}^5 = \boldsymbol{0}, & \boldsymbol{A}^5 = -\boldsymbol{e}_V\boldsymbol{e}_V^T, \\ c^{5+i} = -1, & \boldsymbol{b}^{5+i} = \boldsymbol{0}, & \boldsymbol{A}^{5+i} = \boldsymbol{e}_i\boldsymbol{e}_i^T \text{ for } i \in V, \\ c^{5+|V|+i} = 1, & \boldsymbol{b}^{5+|V|+i} = \boldsymbol{0}, & \boldsymbol{A}^{5+|V|+i} = -\boldsymbol{e}_i\boldsymbol{e}_i^T \text{ for } i \in V. \end{cases}$$

The number of input matrices in the form of (16) is $m = 2|V| + 5$. For example, $\boldsymbol{B}^{5+i} \bullet \boldsymbol{Y} \leq 0$ and $\boldsymbol{B}^{5+|V|+i} \bullet \boldsymbol{Y} \leq 0$ lead to $Y_{ii} = 1$ for $i \in V$. In addition, $Y_{00} = 1$ is guaranteed by $\boldsymbol{B}^{m+1} \bullet \boldsymbol{Y} = 1$.

The set of diagonal constraints is $\mathcal{I} = \{5 + i : i \in V\} \cup \{5 + |V| + i : i \in V\}$. From this $\mathcal{I}$, the feasible set of (17) is given by $\mathcal{F} := \left\{\boldsymbol{Y} \in \mathbb{S}^{1+|V|} : Y_{ii} = 1 \text{ for } i = 0, 1, \ldots, |V| \text{ and } \boldsymbol{Y} \succeq \boldsymbol{O}\right\}$. Hence, we obtain

$$\rho^0_{SDP} = \min\left\{\boldsymbol{B}^0 \bullet \boldsymbol{Y} : \boldsymbol{Y} \in \mathcal{F}\right\} = \min\left\{2\sum_{i\in V}\bar{g}_i Y_{0,i} : \boldsymbol{Y} \in \mathcal{F}\right\} = -2\bar{g}_V^T\boldsymbol{e}_V.$$

Here, a combination of the matrix-completion method [10, 28, 48] with a property $\bar{\boldsymbol{g}}_V \geq \boldsymbol{0}$ ensures that an optimal solution of this minimization problem is given as $\boldsymbol{Y} = \begin{pmatrix} 1 & -\boldsymbol{e}_V^T \\ -\boldsymbol{e}_V & \boldsymbol{e}_V\boldsymbol{e}_V^T \end{pmatrix}$.

We should note that the SDP relaxation problem (7) has a constant term $\bar{g}$ in the objective function, but we have to set $c^0 = 0$ to employ Theorem 3.1. By taking the shift of $\bar{g}$ into account, Theorem 3.1 gives a lower bound;

$$\begin{aligned} E[2\bar{\boldsymbol{g}}_V^T\tilde{\boldsymbol{y}}_V + \bar{g}] &= E[2\bar{\boldsymbol{g}}_V^T\tilde{\boldsymbol{y}}_V] + \bar{g} \\ &\geq \frac{2}{\pi}(\overline{OPT}_{SDP}) - \left(1 - \frac{2}{\pi}\right)2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g} \\ &= \frac{2}{\pi}(OPT_{SDP} - \bar{g}) - \left(1 - \frac{2}{\pi}\right)2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g} \\ &= \frac{2}{\pi}OPT_{SDP} + \left(1 - \frac{2}{\pi}\right)(-2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g}). \end{aligned}$$

To consider an upper bound, we first evaluate $E[\tilde{y}_i]$ for $i \in V$. From $Y^*_{ii} = 1$ for $i \in \{0\} \cup V$ in (7) and the definition of $\tilde{\boldsymbol{y}}_V$, and it holds that $\tilde{y}_i = 1$ if $\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0) = \text{sign}(\boldsymbol{v}^T\boldsymbol{v}^i)$, and $\tilde{y}_i = -1$ if $\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0) = -\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^i)$. The discussion in [11] indicates that the probability of the event $\text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0) = \text{sign}(\boldsymbol{v}^T\boldsymbol{v}^0)$ is given as $1 - \frac{1}{\pi}\text{arccos}(Y^*_{0i})$. Therefore, we have

$$\begin{aligned} E[\tilde{y}_i] &= 1 \cdot \left(1 - \frac{1}{\pi}\text{arccos}(Y^*_{0i})\right) + (-1) \cdot \left\{1 - \left(1 - \frac{1}{\pi}\text{arccos}(Y^*_{0i})\right)\right\} \\ &= 1 - \frac{2}{\pi}\text{arccos}(Y^*_{0i}) \leq \alpha(Y^*_{0i} - 1) + 1. \end{aligned}$$

The last inequality was derived from the inequality $\frac{\text{arccos}(y)}{\pi} \geq \alpha\frac{1-y}{2}$ for $-1 \leq y \leq 1$ (Lemma 3.4 of [11]) and $-1 \leq Y^*_{0i} \leq 1$ due to $\boldsymbol{Y}^* \succeq \boldsymbol{O}$ and $Y^*_{00} = Y^*_{ii} = 1$.

Table 1: Theoretical bounds on the expected values by the randomized algorithm

Z	2θ	lower bound	expected value	upper bound	OPT_{SDP}
200	0.0334	16.161	25.812	30.340	25.386
1050	0.0627	5.075	32.305	112.600	24.938
2045	0.0711	279.259	446.089	2007.212	438.659
5050	0.1081	5.775	284.965	806.205	42.786

As a result, we obtain an inequality

$$\begin{aligned} E[2\bar{\boldsymbol{g}}_V^T\tilde{\boldsymbol{y}}_V] + \bar{g} &= 2\sum_{i\in V}\bar{g}_i E[\tilde{y}_i] + \bar{g} \\ &\leq 2\alpha\sum_{i\in V}\bar{g}_i Y_{0i}^* + 2(1-\alpha)\boldsymbol{g}_V^T\boldsymbol{e}_V + \bar{g} \\ &= 2\alpha\bar{\boldsymbol{g}}_V\boldsymbol{y}_V^* + 2(1-\alpha)\boldsymbol{g}_V^T\boldsymbol{e}_V + \bar{g} \\ &= \alpha(OPT_{SDP} - \bar{g}) + 2(1-\alpha)\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g} \\ &= \alpha OPT_{SDP} + (1-\alpha)(2\bar{\boldsymbol{g}}_V^T\boldsymbol{e}_V + \bar{g}). \end{aligned}$$

□

From a theoretical viewpoint, Lemma 3.2 gives the bounds on the expected objective value $E[2\bar{\boldsymbol{g}}_V^T\tilde{\boldsymbol{y}}_V + \bar{g}]$ of the randomized algorithm. When we executed preliminary experiments, we observed that the interval between the lower and upper bounds are not so sharp. Table 1 presents the lower and upper bounds and the expected objective value. The dataset we used here is a subset of datasets in Section 5. The first column shows Z, the number of genotype candidates. We fix the number of chosen candidates to $N = 50$. The third and fourth columns are the lower and the upper bounds in Lemma 3.2, respectively. The expected objective value is shown in the third column, and it is obtained by generating the random vector $\boldsymbol{v}$ thousand times and taking the average of the thousand trials. The fifth column is the optimal value of the SDP relaxation problem 7.

For the smallest size $Z = 200$, the gap between the lower and the upper bound was not so large. However, when we tried the larger problems, the gap was getting worse. In particular, the ratio of the upper bound to the expected objective value for the case $Z = 5255$ goes beyond 5.34.

Another aspect in the randomized algorithm is that the expected objective value is always larger than OPT_{SDP}. A reason of this unfavorable aspect is that the generated solution $\tilde{\boldsymbol{y}}_V$ is not guaranteed to satisfy the constraint $\boldsymbol{y}_V^T\boldsymbol{A}_{VV}\boldsymbol{y}_V + 2\bar{\boldsymbol{c}}_F^T\boldsymbol{y}_V \leq 2\bar{\theta}$ that corresponds to $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ of (2). Though Theorem 4 of [36] estimates the number of randomly generated solutions required for approximate feasible solutions with high probability, this cannot be applied to the discussion in this paper, since the current discussion does not fully satisfy the assumption of the theorem.

Due to this weaker bounds reported in Table 1, we are determined to seek an optimization method that can obtain a reasonable solution for practical use. This motivated us to develop a local search method based on the steep-descent method for discrete convex functions.

4 Steepest-ascent method

In contrast to mixed-integer linear programming problems for which many solvers have been developed, a principal difficulty in the ED problem (2) arises from the nonlinear constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$. To obtain a sensible solution in a short time, we embed the violation against this constraint into the objective function as a penalty term using a penalty weight $\lambda \geq 0$ and focus the following optimization problem

$$\begin{array}{lll} \max & : & f_\lambda(\boldsymbol{x}) := \boldsymbol{g}^T\boldsymbol{x} - \lambda \max\{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - 2\theta, 0\} \\ \text{subject to} & : & \boldsymbol{x} \in \hat{\mathcal{F}} \end{array} \tag{18}$$

where $\hat{\mathcal{F}} := \left\{\boldsymbol{x} \in \mathbb{R}^Z : \boldsymbol{e}^T\boldsymbol{x} = 1, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, x_1, \ldots, x_Z \in \left\{0, \frac{1}{N}\right\}\right\}$.

We give a validity of (18) by the next lemma which shows that if we take a large λ, this optimization problem with a penalty term (18) is equivalent to the original problem (2) .

Lemma 4.1. *Let $\boldsymbol{x}(\lambda) \in \mathbb{R}^Z$ be an optimal solution of (18). There exists a $\hat{\lambda} > 0$ such that $\boldsymbol{x}(\lambda)$ is an optimal solution of (2) for $\forall \lambda \geq \hat{\lambda}$.*

Proof: Let $\hat{\phi}$ be the optimal value of the following optimization problem;

$$\begin{array}{llll} \hat{\phi} & := & \min & : \max\{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - 2\theta, 0\} \\ & & \text{subject to} & : \boldsymbol{x} \in \hat{\mathcal{F}}. \end{array} \tag{19}$$

From this definition, ϕ can take either zero or a positive number.

If $\hat{\phi} = 0$, the quadratic constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ holds for $\forall \boldsymbol{x} \in \hat{\mathcal{F}}$. Therefore, this constraint vanishes from (2) and the penalty term in (18) has no effect. Hence, the two problems (2) and (18) are equivalent for any $\lambda \geq 0$.

For the case $\hat{\phi} > 0$, since $\hat{\mathcal{F}}$ is composed of a finite number of points, the reciprocal number of $\hat{\phi}$ is a finite number. Therefore, we can take $\hat{\lambda} = \frac{\max\{g_i : i=1,\ldots,Z\} - \min\{g_i : i=1,\ldots,Z\} + 1}{\hat{\phi}}$. To show this by a contradiction, we assume that $\boldsymbol{x}(\lambda)^T\boldsymbol{A}\boldsymbol{x}(\lambda) \leq 2\theta$ does not hold for $\lambda \geq \hat{\lambda}$. Then, we have

$$\boldsymbol{g}^T\boldsymbol{x}(\lambda) - \lambda\left(\boldsymbol{x}(\lambda)^T\boldsymbol{A}\boldsymbol{x}(\lambda) - 2\theta\right) \leq \max\{g_i : i = 1, \ldots, Z\} - \hat{\lambda}\hat{\phi} < \min\{g_i : i = 1, \ldots, Z\}.$$

Here, we used $\boldsymbol{e}^T\boldsymbol{x}(\lambda) = 1$ and $\boldsymbol{x}(\lambda) \geq \boldsymbol{0}$ since $\boldsymbol{x}(\lambda) \in \hat{\mathcal{F}}$. On the contrary, from the assumption that (2) has a feasible point, the objective value of (18) at this feasible point is at least $\min\{g_i : i = 1, \ldots, Z\}$. This indicates that $\boldsymbol{x}(\lambda)$ can not be an optimal solution of (18) if $\boldsymbol{x}(\lambda)^T\boldsymbol{A}\boldsymbol{x}(\lambda) > 2\theta$. Therefore, we can restrict the feasible region of (18) to the set $\{\boldsymbol{x} \in \mathbb{R}^Z : \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta\} \cap \hat{\mathcal{F}}$, and the objective function of (18) is reduced to $\boldsymbol{g}^T\boldsymbol{x}$. Consequently, the optimal solution of (18) is also optimal for (2). □

Since the computation for $\hat{\phi}$ is almost as hard as the original ED problem, it is not practical to compute $\hat{\phi}$. In addition, when we maximize $f_\lambda(\boldsymbol{x})$, extremely large λ makes the computation numerically unstable. As an appropriate value for the penalty weight λ, we make the use of the Lagrangian multiplier λ_0 developed in Meuwissen [22];

$$\lambda_0 := \sqrt{\frac{(\boldsymbol{g}^T\boldsymbol{A}^{-1}\boldsymbol{g})(\boldsymbol{e}^T\boldsymbol{A}^{-1}\boldsymbol{e}) - (\boldsymbol{g}^T\boldsymbol{A}^{-1}\boldsymbol{e})^2}{8\theta(\boldsymbol{e}^T\boldsymbol{A}^{-1}\boldsymbol{e}) - 4}}.$$

This λ_0 corresponds to the Lagrangian multiplier of the constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 2\theta$ in the following optimization problem.

$$\begin{array}{lll} \text{max} & : & \boldsymbol{g}^T\boldsymbol{x} \\ \text{subject to} & : & \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 2\theta, \\ & & \boldsymbol{e}^T\boldsymbol{x} = 1. \end{array} \tag{20}$$

The approach of [22] first solves (20), then applies some heuristic method to obtain a solution of (2). Therefore, a maximization of $\boldsymbol{g}^T\boldsymbol{x} - \lambda_0(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - 2\theta)$ over $\boldsymbol{e}^T\boldsymbol{x} = 1$ is a natural derivation when we consider (20). For $f_\lambda(\boldsymbol{x})$, we often employ λ such that $\lambda \geq \lambda_0$, since we put a strong emphasis on the violation with respect to $\max\{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - 2\theta, 0\}$.

We now discuss (18) from the viewpoint of convex functions. The function $-f_\lambda(\boldsymbol{x})$ is a convex function in the continuous space $\mathbb{R}^Z$, since $\boldsymbol{A} \succeq \boldsymbol{O}$ and $\lambda \geq 0$. Hence, the problem (18) can be cast as a minimization of a convex function over a discrete feasible set.

An M-convex function [26] is a discrete convex function defined on a set in which the sum of the elements of a feasible point is constant. A steepest-descent method for M-convex functions was developed in [27]. When an M-convex function $f^M(\boldsymbol{x})$ with a feasible set $\mathcal{F}^M$ is given, the steepest-descent method starts from an initial point $\boldsymbol{x}^0 \in \mathcal{F}^M$, and finds the next point $\boldsymbol{x}^1$ from a neighborhood $\mathcal{N}(\boldsymbol{x}^0) \subset \mathcal{F}^M$ that decreases the objective function $f^M(\boldsymbol{x})$ with the largest margin. Here, $\mathcal{N}(\boldsymbol{x}^0) := \{\boldsymbol{x} + \boldsymbol{e}_i - \boldsymbol{e}_j \in \mathcal{F}^M : i, j = 1, \ldots, n\}$. In other words, $\boldsymbol{x}^1$ is chosen so that $f^M(\boldsymbol{x}_1) \leq f^M(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{x}^0)$. The steepest-descent method continues the search in neighbors, and it eventually can find a global minimizer since any local minimizer is a global minimizer when the objective function f^M is an M-convex function.

Though $-f_\lambda(\boldsymbol{x})$ is not an M-convex function since it can encompass multiple local minimizers that are not always global minimizers, the optimization problem with the penalty term (18) has resemblances to a minimization of an M-convex function. In particular, the feasible set $\hat{\mathcal{F}}$ satisfies $\sum_{i=1}^Z x_i = 1$ and the function $-f_\lambda(\boldsymbol{x})$ is a convex function in the continuous space $\mathbb{R}^Z$. Therefore, we can expect that the steepest-descent method for M-convex functions will give good direction to solve (18). Furthermore, we can exploit the solution obtained by the conic relaxation problems in Section 3 to generate a starting point $\boldsymbol{x}^0$.

When we adjust the steepest-descent method implemented in the software package ODICON [1] [37] to solve (18), we obtain Algorithm 4.2. Since (18) is a maximization problem, Algorithm 4.2 is a steepest-ascent method.

Algorithm 4.2. *A steep-ascent method with a conic relaxation problem for the optimization problem with the penalty term arising from the ED problem*

Step 1: *Solve a conic relaxation problem (7), (8) or (10). If (10) is solved, let* $\boldsymbol{x}^*$ *be its optimal solution. For (8) and (7), let* $\boldsymbol{y}_V^*$ *be its optimal solution and set* $\boldsymbol{x}^*$ *by* $\boldsymbol{x}_V^* := \boldsymbol{y}_V^*$ *and* $\boldsymbol{x}_F^* := \boldsymbol{c}_F$*.*

Step 2: *By sorting* $\boldsymbol{x}^*$*, separate* V *into the two disjoint set* $V_{\frac{1}{N}}$ *and* V_0 *such that* $x_i^* \geq x_j^*$ *for* $i \in V_{\frac{1}{N}}$*,* $j \in V_0$ *and that* $|V_{\frac{1}{N}}| = N - p$ *(ties are broken arbitrary). Set the initial point* $\boldsymbol{x}^0 \in \mathbb{R}^Z$ *by* $x_i^0 := \frac{1}{N}$ *for* $i \in V_{\frac{1}{N}}$*,* $x_j^0 := 0$ *for* $j \in V_0$*, and* $\boldsymbol{x}_F^0 := \boldsymbol{c}_F$*. Set the iteration counter* $h := 0$*.*

[1] `http://ist.ksc.kwansei.ac.jp/~tutimura/odicon/index.en.html`

Step 3: *Select the steepest swap* $i^h \in V_{\frac{1}{N}}$ *and* $j^h \in V_0$ *such that*

$$f_\lambda(\boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_{i^h} + \frac{1}{N}\boldsymbol{e}_{j^h}) \geq f_\lambda(\boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_i + \frac{1}{N}\boldsymbol{e}_j) \text{ for } i \in V_{\frac{1}{N}}, j \in V_0.$$

Step 4: *If there is no improvement, that is* $f_\lambda(\boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_{i^h} + \frac{1}{N}\boldsymbol{e}_{j^h}) \leq f_\lambda(\boldsymbol{x}^h)$*, output* $\boldsymbol{x}^h$ *as a solution and stop.*

Step 5: *Set* $\boldsymbol{x}^{h+1} := \boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_{i^h} + \frac{1}{N}\boldsymbol{e}_{j^h}$*. Swap* i^h *and* j^h *by* $V_{\frac{1}{N}} := V_{\frac{1}{N}} \cup \{j^h\} \backslash \{i^h\}$ *and* $V_0 := V_0 \cup \{i^h\} \backslash \{j^h\}$*. Set* $h := h + 1$ *and return to Step 3.*

In Step 2, the number of $\frac{1}{N}$ in $\boldsymbol{x}^0$ is exactly N. Due to Step 5, this property is kept through the iterations in the algorithm, hence, the number of $\frac{1}{N}$ in $\boldsymbol{x}^h$ is also exactly N for any $h \geq 1$. When no improvement can be found, the algorithm stops by Step 4.

Most computation cost of each iteration in Algorithm 4.2 is consumed at the evaluations of f_λ in Step 3. The number of the evaluations is determined by the size of neighbor around $\boldsymbol{x}^h$, that is, $|V_{\frac{1}{N}}| \times |V_0| = (N - p) \times (|V| - (N - p))$. Therefore, the case $N - p = \frac{|V|}{2}$ requires the heaviest computation cost. Furthermore, to reduce the computation cost, we focus the evaluation of $(\boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_{i^h} + \frac{1}{N}\boldsymbol{e}_{j^h})^T \boldsymbol{A}(\boldsymbol{x}^h - \frac{1}{N}\boldsymbol{e}_{i^h} + \frac{1}{N}\boldsymbol{e}_{j^h})$. Since ODICON was designed to handle general functions, it accessed all the elements of $\boldsymbol{A}$ for each i^h and j^h. By expanding the part as $(\boldsymbol{x}^h)^T \boldsymbol{A}(\boldsymbol{x}^h) + \frac{2}{N}\left(-\boldsymbol{e}_{i^h} + \boldsymbol{e}_{j^h}\right)^T (\boldsymbol{A}\boldsymbol{x}^h) + \frac{1}{N^2}\left(A_{i^h i^h} + A_{j^h j^h} - 2A_{i^h j^h}\right)$, we evaluate $(\boldsymbol{x}^h)^T \boldsymbol{A}(\boldsymbol{x}^h)$ and $(\boldsymbol{A}\boldsymbol{x}^h)$ only once for each iteration of Algorithm 4.2. This saves 95% of the computation time for Step 3 compared to ODICON.

5 Numerical results

In this section, we report numerical results to verify the performance of the proposed algorithm, Algorithm 4.2. We implemented Algorithm 4.2 with Matlab R2014b. We compared the proposed algorithm with GENCONT [22], the branch-and-bound method implemented in OPSEL 2.0 [24, 23], and IBM CPLEX 12.62. We used an Windows PC with Core i7 3770K (3.5 GHz) and 32 GB memory space for cases. Only when the 32 GB memory space was not enough, we used a Linux server with Opteron 4386 (3.10 GHz) and 128 GB memory space. To solve the LP problem (8), the SOCP problem (10), and the SDP problem (7), we employed CPLEX, ECOS [7], and SDPT-3 [35], respectively. For the steepest-ascent method, we set $\lambda = 2\lambda_0$ as the penalty weight in the function $f_\lambda(\boldsymbol{x})$ of (18).

The data tested in the numerical experiments of this paper are practical datasets of pine orchards available at the Dryad Digital Repository[2] and datasets generated by a simulation software package [25].

Tables 2 and 3 present the comparison of the three conic relaxation approaches. The number N of chosen genotype is set 50 and 150 in Table 2 and Table 3, respectively. The first column in the tables shows the name of algorithms; for example, CR (LP) is the result of conic relaxation problem (in this case, an LP problem) and SA (LP) is the result after the application of Algorithm 4.2 starting from the solution of CR (LP). The names for SOCP and SDP are indicated with the same rule. For CR (LP)-s and SA (LP)-s, we applied Remark 2.4 to (8) and obtain its solution by sorting

[2] `http://dx.doi.org/10.5061/dryad.9pn5m`

$\boldsymbol{g}_V$. The second column is Z, the number of genotype candidates, while the third column is 2θ. The fourth and fifth columns are the objective value $\boldsymbol{g}^T\boldsymbol{x}$ and the value $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ of each algorithm. For the CR rows, these two values were evaluated at $\boldsymbol{x}^*$, the solution at Step 1 of Algorithm 4.2, and for the SA rows, they were computed with the output solution $\boldsymbol{x}^h$ at Step 4. The sixth column is the iteration number of Algorithm 4.2. The seventh column is the value of $f_\lambda(\boldsymbol{x}^h)$, where h is the iteration number indicated in the sixth column. For the CR row, note that $\boldsymbol{x}^*$ must satisfy the quadratic constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$, but $\boldsymbol{x}^0$ does not always satisfy it. In contrast, for the SA rows, $f_\lambda(\boldsymbol{x}^h)$ is given at the final solution of Step 4. The last column is the computation time in seconds. Since SA (LP) uses the result of CR (LP), the computation time of SA (LP) is the sum of the computation time of CR (LP) and the steep-ascent method. In a similar way, SA (SOCP) and SA (SDP) are also the sums.

For the smallest case $Z = 200$ and $N = 50$, the SDP relaxation problem attains a remarkable result. Since the solution in SA (SDP) satisfies the constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$, this solution is a feasible solution of the original ED problem (2). In addition, it holds that $OPT_{ED} \leq OPT_{SDP}$ from Lemma 2.3. Therefore, we know $25.207 \leq OPT_{ED} \leq 25.386$ and we obtain the optimal value of the ED problem up to an error 0.710%. Since this error is much better than the theoretical bounds discussed in Lemma 3.2, the combination of the SDP relaxation and the steepest-ascent method performs very well in this case.

In addition, we can make sure that the objective values of CR (LP) and CR (LP)-s are same, and this indicates that Assumption 2.2 holds in the numerical tests. As a result, we can obtain the solution of the LP relaxation problem (8) without solving it as an LP problem, as noted in Remark 2.4.

From Tables 2 and 3, we observe in the cases $Z \leq 5050$ that $OPT_{SDP} \leq OPT_{SOCP} \leq OPT_{LP}$ from the values of $\boldsymbol{g}^T\boldsymbol{x}$ in the CR (LP), CR (SOCP), and CR (SDP) rows. This result supports the validity of Lemma 2.3. However, we also observe for large instances $Z \geq 10100$ that $\boldsymbol{g}^T\boldsymbol{x}$ of CR (SDP) is lower than that of CR (SOCP). A principal reason of this inconsistent phenomenon is a premature termination of SDPT-3. These inaccurate values of CR (SDP) were mainly caused by the lack of interior-feasible points in (7); see Remark 2.5. The SOCP relaxation problem (10) provides highly numerical stability compared to the SDP relaxation problem (7). This can be regarded as an advantage of the SOCP relaxation approach.

As a next viewpoints, the violations of the solution generated by the steep-ascent method against $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ are remarkably small. This is mainly because we set λ large enough based on the Lagrangian multiplier λ_0. Therefore, the maximization of the function with the penalty term (18) can provide a suitable solution for the ED problem (2).

A comparison of the results of the steepest-ascent methods that starts from the three conic relaxation indicates that if SDPT-3 obtained sensible solutions ($Z \leq 5050$), the output objective values $\boldsymbol{g}^T\boldsymbol{x}$ of SA (SDP) and SA (SOCP) were close to each other, but much higher than that of SA (LP). This implies that the SOCP relaxation problem and the SDP relaxation problem provided good starting points for the steepest-ascent method. This point is also indicated in the iteration number of the steepest-ascent methods. The iteration numbers of SA (SDP) and SA (SOCP) are much less than that of SA (LP). For example, when $Z = 2045$ and $N = 100$, SA (SDP) and SA (SOCP) required only two and three iterations, respectively, while SA (LP) required 65 iterations. Therefore, we can infer that the solutions of the SDP relaxation problem and the SOCP relaxation problem are close to local maximizer of $f_\lambda(\boldsymbol{x})$.

When we move our focus from the solution quality to the computation time, the computation of

Table 2: The comparison of the convex relaxation approaches ($N = 50$)

Algorithm	Z	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	iter	$f_\lambda(\boldsymbol{x})$	time (s)
CR (LP)	200	0.0334	28.068	0.0574	0	-133.895	0.07
SA (LP)			25.029	0.0334	21	25.029	0.10
CR (LP)-s			28.068	0.0574	0	-133.895	0.01
SA (LP)-s			25.029	0.0334	21	25.029	0.04
CR (SOCP)			26.156	0.0334	0	-41.484	0.02
SA (SOCP)			25.090	0.0334	13	25.090	0.06
CR (SDP)			25.386	0.0321	0	18.978	1.29
SA (SDP)			25.207	0.0334	4	25.207	1.30
CR (LP)	1050	0.0627	30.754	0.1362	0	-198.235	0.37
SA (LP)			22.707	0.0627	23	22.707	0.51
CR (LP)-s			30.754	0.1362	0	-198.235	0.01
SA (LP)-s			22.707	0.0627	23	22.707	0.15
CR (SOCP)			25.284	0.0627	0	19.621	0.08
SA (SOCP)			24.831	0.0627	2	24.831	0.09
CR (SDP)			24.938	0.0617	0	24.721	27.94
SA (SDP)			24.846	0.0627	2	24.846	27.96
CR (LP)	2045	0.0711	504.217	0.4566	0	-26197.137	1.16
SA (LP)			414.591	0.0710	32	414.591	1.47
CR (LP)-s			504.217	0.4566	0	-26197.137	0.01
SA (LP)-s			414.591	0.0710	32	414.591	0.32
CR (SOCP)			439.353	0.0711	0	293.122	0.06
SA (SOCP)			438.386	0.0710	2	438.386	0.09
CR (SDP)			438.659	0.0706	0	438.457	145.57
SA (SDP)			438.457	0.0710	1	438.457	145.59
CR (LP)	5050	0.1081	57.630	0.3672	0	-1185.866	10.17
SA (LP)			38.696	0.1080	23	38.696	11.17
CR (LP)-s			57.630	0.3672	0	-1185.866	0.01
SA (LP)-s			38.696	0.1080	23	38.696	0.98
CR (SOCP)			43.036	0.1081	0	42.456	0.21
SA (SOCP)			42.691	0.1080	3	42.691	0.37
CR (SDP)			42.786	0.0980	0	41.327	2221.22
SA (SDP)			42.431	0.1080	3	42.431	2221.40
CR (LP)	10100	0.0701	62.377	0.2368	0	-1305.4682	46.84
SA (LP)			41.284	0.0701	32	41.284	49.87
CR (LP)-s			62.377	0.2368	0	-1305.468	0.01
SA (LP)-s			41.284	0.0701	32	41.284	3.29
CR (SOCP)			47.445	0.0701	0	21.094	0.54
SA (SOCP)			46.568	0.0701	2	46.568	0.87
CR (SDP)			21.265	0.0545	0	13.369	5577.80†
SA (SDP)			44.662	0.0701	45	44.662	5582.46†
CR (LP)	15222	0.0388	603.783	0.4568	0	-67047.589	129.55
SA (LP)			438.791	0.0388	42	438.791	139.03
CR (LP)-s			603.783	0.4568	0	-67047.589	0.01
SA (LP)-s			438.791	0.0388	42	438.791	6.45
CR (SOCP)			468.367	0.0388	0	-1042.485	0.99
SA (SOCP)			460.769	0.0388	9	460.769	2.56
CR (SDP)			288.739	0.0195	0	314.493	17433.38†
SA (SDP)			460.409	0.0388	43	460.409	17441.93†

† indicates numerical instability

Table 3: The comparison of the convex relaxation approaches ($N = 100$)

Algorithm	Z	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	iter	$f_\lambda(\boldsymbol{x})$	time (s)
CR (LP)	200	0.0258	24.654	0.0304	0	-22.392	0.01
SA (LP)			23.355	0.0258	18	23.355	0.07
CR (LP)-s			24.654	0.0304	0	-22.392	0.01
SA (LP)-s			23.355	0.0258	18	23.355	0.06
CR (SOCP)			24.015	0.0258	0	0.950	0.01
SA (SOCP)			23.412	0.0258	13	23.412	0.08
CR (SDP)			23.640	0.0255	0	21.783	0.82
SA (SDP)			23.521	0.0258	5	23.521	0.84
CR (LP)	1050	0.0539	27.637	0.1214	0	-208.680	0.39
SA (LP)			19.805	0.0539	42	19.805	0.98
CR (LP)-s			27.637	0.1214	0	-208.680	0.01
SA (LP)-s			19.805	0.0539	42	19.805	0.590
CR (SOCP)			22.432	0.0539	0	18.537	0.08
SA (SOCP)			22.321	0.0539	5	22.321	0.15
CR (SDP)			22.358	0.0537	0	22.242	30.49
SA (SDP)			22.324	0.0539	3	22.324	30.54
CR (LP)	2045	0.0628	478.114	0.4219	0	-26349.725	1.17
SA (LP)			406.348	0.0628	65	406.348	5.00
CR (LP)-s			478.114	0.4219	0	-26349.725	0.01
SA (LP)-s			406.348	0.0628	65	406.348	3.78
CR (SOCP)			421.696	0.0628	0	197.423	0.07
SA (SOCP)			421.113	0.0627	3	421.113	0.25
CR (SDP)			421.497	0.0627	0	364.014	165.33
SA (SDP)			421.425	0.0628	2	421.425	165.46
CR (LP)	5050	0.0994	54.903	0.3355	0	-1137.701	10.35
SA (LP)			36.509	0.0994	50	36.509	17.44
CR (LP)-s			54.903	0.3355	0	-1137.701	0.01
SA (LP)-s			36.509	0.0994	50	36.509	7.03
CR (SOCP)			40.769	0.0995	0	15.408	0.25
SA (SOCP)			40.629	0.0995	3	40.629	0.71
CR (SDP)			40.711	0.0992	0	31.447	2164.49
SA (SDP)			40.690	0.0994	2	40.690	2164.85
CR (LP)	10100	0.0610	60.347	0.2245	0	-1395.786	46.71
SA (LP)			39.911	0.0610	62	39.911	68.54
CR (LP)-s			60.347	0.2245	0	-1395.786	0.01
SA (LP)-s			39.911	0.0610	62	39.911	21.51
CR (SOCP)			44.819	0.0610	0	10.608	0.70
SA (SOCP)			44.522	0.0610	7	44.522	3.12
CR (SDP)			21.374	0.0532	0	18.463	6750.06†
SA (SDP)			42.810	0.0610	66	42.810	6773.05†
CR (LP)	15222	0.0300	575.227	0.4318	0	-74482.507	128.21
SA (LP)			408.725	0.0300	90	408.725	185.33
CR (LP)-s			575.227	0.4318	0	-74482.507	0.02
SA (LP)-s			408.725	0.0300	90	408.725	49.14
CR (SOCP)			444.730	0.0300	0	-11.0395	1.05
SA (SOCP)			441.438	0.0300	6	441.438	4.72
CR (SDP)			309.173	0.0228	0	-65291.694	19467.30†
SA (SDP)			406.266	0.0300	92	406.266	19525.52†

† indicates numerical instability

Table 4: The comparison of GENCONT, OPSEL, CPLEX, and the proposed algorithm ($N = 50$)

Algorithm	Z	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	$f_\lambda(\boldsymbol{x})$	#chosen	time (s)
GENCONT	200	0.0334	25.290	0.0342	20.087	50	0.06
OPSEL			25.191	0.0334	25.191	50	1779.13
CPLEX			25.190	0.0334	25.190	50	4270.77
SA (SOCP)			25.090	0.0334	25.090	50	0.06
GENCONT	1050	0.0627	24.983	0.0627	24.983	48	7.91
OPSEL			24.858	0.0627	24.858	50	> 10800
CPLEX			Cannot obtain a sensible solution in 3 hours				> 10800
SA (SOCP)			24.831	0.0627	24.831	50	0.09
GENCONT	2045	0.0711	437.049	0.0694	437.049	50	88.46
OPSEL			435.826	0.0692	435.826	50	16.08
CPLEX			436.213	0.0680	436.212	50	0.37
SA (SOCP)			438.386	0.0710	438.386	50	0.09
GENCONT	5050	0.1081	42.780	0.1089	-306.701	50	1769.72
OPSEL			42.702	0.1081	42.702	50	> 10800
CPLEX			42.456	0.1066	42.456	50	2.02
SA (SOCP)			42.691	0.1080	42.691	50	0.37
GENCONT	10100	0.0701	Out of memory				
OPSEL			46.252	0.0700	46.252	50	> 10800
CPLEX			Cannot obtain a sensible solution in 3 hours				> 10800
SA (SOCP)			46.568	0.0701	46.568	50	0.87
GENCONT	15222	0.0388	Out of memory				
OPSEL			459.040	0.0388	459.040	50	> 10800
CPLEX			459.135	0.0386	459.135	50	39.20
SA (SOCP)			460.769	0.0388	460.769	50	2.56

SA (SOCP) is much shorter than SA (SDP). This was mainly because we can aggressively exploit the structure of the Wright numerator matrix $\boldsymbol{A}$ through the SOCP approach discussed in [47]. Since the objective values $\boldsymbol{g}^T\boldsymbol{x}$ of SA (SOCP) and SA (SDP) are competitive, SA (SOCP) can be considered as the most efficient approach among the three SA (LP), SA (SOCP), and SA (SDP).

Judging from these results, we chose the SOCP relaxation problem for generating the starting point when we compare Algorithm 4.2 against the existing approaches, GENCONT, OPSEL, and CPLEX. Since OPSEL and CPLEX utilized the brand-and-bound framework, we can expect their solution are close to the real optimal solution of the ED problem (2).

Tables 4 and 5 present the comparison of Algorithm 4.2, GENCONT and OPSEL (Version 2.0), CPLEX (Version 12.62) for the numbers of selected genotypes $N = 50$ and $N = 100$. We tried to execute GENCONT2, a new version of GENCONT, but its binary file did not work on our computational environment. Therefore, we used GENCONT1 for the comparison. In the tables, we evaluate $f_\lambda(\boldsymbol{x})$ in seventh column at the solution obtained from each algorithm and the eighth column reports the number of chosen genotypes $|\{i : x_i > 0\}|$. For OPSEL and CPLEX, we set the time limit as three hours and the tolerance gap as 1%. When OPSEL and CPLEX reached the time limit, we obtained a feasible solution from OPSEL, but we could not extract sensible solutions from CPLEX. GENCONT could not solve large instances ($Z = 10100$ and $Z = 15222$) due to out of memory.

From the numerical results in Tables 4 and 5, the objective values of SA (SOCP) are close to

Table 5: The comparison of GENCONT, OPSEL, CPLEX, and the proposed algorithm ($N = 100$)

Algorithm	Z	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	$f_\lambda(\boldsymbol{x})$	#chosen	time (s)
GENCONT	200	0.0258	23.640	0.0261	21.253	100	0.07
OPSEL			23.551	0.0258	23.551	100	> 10800
CPLEX			23.508	0.0258	23.508	100	1.42
SA (SOCP)			23.412	0.0258	23.412	100	0.08
GENCONT	1050	0.0539	22.749	0.0539	22.749	91	9.63
OPSEL			22.275	0.0539	22.275	100	304.89
CPLEX			Cannot obtain a sensible solution in 3 hours				> 10800
SA (SOCP)			22.321	0.0539	22.321	100	0.15
GENCONT	2045	0.0628	421.005	0.0632	392.893	100	105.40
OPSEL			419.600	0.0613	419.600	100	6.85
CPLEX			420.748	0.0619	420.748	100	0.41
SA (SOCP)			421.113	0.0627	421.113	100	0.25
GENCONT	5050	0.0995	40.692	0.0997	40.692	100	1940.43
OPSEL			40.468	0.0994	40.468	100	50.46
CPLEX			Cannot obtain a sensible solution in 3 hours				> 10800
SA (SOCP)			40.629	0.0995	40.629	100	0.71
GENCONT	10100	0.0610	Out of memory				
OPSEL			44.467	0.0696	44.467	100	> 10800
CPLEX			Cannot obtain a sensible solution in 3 hours				> 10800
SA (SOCP)			44.522	0.0610	44.522	100	3.12
GENCONT	15222	0.0300	Out of memory				
OPSEL			441.770	0.0300	441.770	100	> 10800
CPLEX			440.996	0.0290	440.99640	100	7.45
SA (SOCP)			441.438	0.0300	441.438	100	4.72

those of GENCONT, OPSEL, and CPLEX. Since the cost vector $\boldsymbol{g}$ in the objective function is usually generated from a statistical procedure, the discrepancy in the objective values make a little difference for practical use. However, GENCONT sometimes failed to satisfy the constraints; the quadratic constraint $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq 2\theta$ was violated in the $Z = 200$ or $Z = 5050$ cases, and the number of the chosen genotypes did not match the input N. Therefore, the quality of SA (SOCP) was superior to that of GENCONT.

From the viewpoint of the computation time, of SA (SOCP) is much faster than GENCONT. In particular, for the case $Z = 5050$, SA (SOCP) used less than one seconds, while GENCONT required 1700 seconds. The computation times of the branch-and-bound framework were unpredictable. In $N = 50$, OPSEL and CPLEX required longer computation for a small problem $Z = 200$ than for a large problem $Z = 2045$. It is very difficult to estimate the computation time required by OPSEL and CPLEX in advance due to a nature of the branch-and-bound framework. In contrast, SA (SOCP) consumed longer computation time for larger problems and this property is favorable for practical use.

6 Conclusion and Future Directions

In this paper, we introduced the conic relaxation approach based on LP, SOCP, and SDP for the special-case ED selection problem that is commonly encountered in tree breeding. We discussed the strength of the three conic relaxation problems, and gave the theoretical bounds of the randomized algorithm that uses the SDP relaxation problem. The fact that the theoretical bounds are not so sharp motivated us to implement the steep-ascent method so that we can acquire a suitable solution for practical usage. From the numerical results, we found that the steep-ascent method with the SOCP relaxation was the most effective among the three conic relaxation approaches, and that this outperformed the existing methods, in particular, from the viewpoint of computation time.

One of further directions is to find a better theoretical bounds for the conic relaxation problems. In the discussions of this paper, we mainly relied on the positive semidefiniteness of and non-negativity of the Wright numerator matrix $\boldsymbol{A}$. Since the specific values of the elements in this matrix strongly relate to the pedigree of the candidate pool, there is a possibility that we exploit such structures to tighten the theoretical bounds discussed in Lemma 3.2. However, we also need to reduce the computation time of the SDP relaxation problem to make the SDP approach effective.

Another research direction is to minimize inbreeding depression [18]. The objective function there is of form $(1 - (\mathrm{ID})\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x})\boldsymbol{g}^T \boldsymbol{x}$, where ID is a constant that represents a regression slope. Since the function is a cubic function with respect to the contribution $\boldsymbol{x}$, this function is not even a convex function. From the numerical results in this paper, we expect that a similar method to the steep-ascent method may perform well for solving such a problem. The minimization of the inbreeding depression will be an interesting problem to researchers in the optimization field.

References

[1] J. Ahlinder, T. Mullin, and M. Yamashita. Using semidefinite programming to optimize unequal deployment of genotypes to a clonal seed orchard. *Tree genetics & genomes*, 10(1):27–34, 2014.

[2] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[3] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.

[4] H. Y. Benson and U. Saglam. Mixed-integer second-order cone programming: A survey. *Tutorials in Operations Research*, pages 13–36, 2013.

[5] L. Bomosssoul and D. Lindgren. Optimal utilization of clones and genetic thinning of ‘ seed orchards. *Silvae Genetica*, 42:4–5, 1993.

[6] D. Charlesworth and B. Charlesworth. Inbreeding depression and its evolutionary consequences. *Annual review of ecology and systematics*, 18(1):237–268, 1987.

[7] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *Proceedings of European Control Conference*, pages 3071–3076, 2013.

[8] S. Drewes and S. Ulbrich. Subgradient based outer approximation for mixed integer second order cone programming. In *Mixed Integer Nonlinear Programming*, pages 41–59. Springer, 2012.

[9] L. Faybusovich. Euclidean jordan algebras and interior-point algorithms. *Positivity*, 1(4):331–357, 1997.

[10] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: general framework. *SIAM J. Optim.*, 11(3):647–674, 2000.

[11] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.

[12] A. Gorge, A. Lisser, and R. Zorgati. Semidefinite relaxations for mixed 0-1 second-order cone program. In *Combinatorial Optimization*, pages 81–92. Springer, 2012.

[13] J. Hallander and P. Waldmann. Optimum contribution selection in large general tree breeding populations with an application to scots pine. *Theoretical and applied genetics*, 118(6):1133–1142, 2009.

[14] S. He, Z.-Q. Luo, J. Nie, and S. Zhang. Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization. *SIAM Journal on Optimization*, 19(2):503–523, 2008.

[15] D. Hinrichs, M. Wetten, et al. An algorithm to compute optimal genetic contributions in selection programs with large numbers of candidates. *Journal of animal science*, 84(12):3212–3218, 2006.

[16] Y. Hsia, S. Wang, and Z. Xu. Improved semidefinite approximation bounds for nonconvex nonhomogeneous quadratic optimization with ellipsoid constraints. *Operations Research Letters*, 43(4):378–383, 2015.

[17] S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15(3-4):201–224, 2001.

[18] D. Lindgren, D. Danusevicius, and O. Rosvall. Unequal deployment of clones to seed orchards by considering genetic gain, relatedness and gene diversity. *Forestry*, 82(1):17–28, 2009.

[19] D. Lindgren, W. S. Libby, and F. L. Bondesson. Deployment to plantations of numbers and proportions of clones with special emphasis on maximizing gain at a constant diversity. *Theor. Appl. Genet.*, 77(6):825–831, 1989.

[20] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[21] M. Lynch and B. W. B. *Genetics and Analysis of Quantitative Traits.* Sinauer Associates, Inc., Sunderland, MA, USA, 1998.

[22] T. H. E. Meuwissen. Maximizing the response of selection with a predefined rate of inbreeding. *J. Anim. Sci.*, 75:934–940, 1997.

[23] T. Mullin and P. Belotti. Using branch-and-bound algorithms to optimize selection of a fixed-size breeding population under a relatedness constraint. *Tree Genetics & Genomes*, 12(1):1–12, 2016.

[24] T. J. Mullin. OPSEL 1.0: A computer program for optimal selection in forest tree breeding by mathematical programming. Technical Report Nr. 841-2014, Arbetsrapport från Skogforsk, 2014.

[25] T. J. Mullin, J. Hallander, O. Rosvall, and B. Andersson. Using simulation to optimise tree breeding programmes in Europe: an introduction to POPSIM. Technical Report Nr. 711-2010, Arbetsrapport från Skogforsk, 2010.

[26] K. Murota. *Discrete convex analysis.* SIAM, 2003.

[27] K. Murota. On steepest descent algorithms for discrete convex functions. *SIAM Journal on Optimization*, 14(3):699–707, 2004.

[28] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results. *Math. Program., Ser B*, 95:303–327, 2003.

[29] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9(1-3):141–160, 1998.

[30] Y. Nesterov, A. Nemirovskii, and Y. Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

[31] R. Pong-Wong and J. A. Woolliams. Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming. *Genet. Sel. Evol*, 39:3–25, 2007.

[32] S. Schmieta and F. Alizadeh. Associative and jordan algebras, and polynomial time interior-point algorithms for symmetric cones. *Mathematics of Operations Research*, 26(3):543–564, 2001.

[33] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[34] M. J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[35] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 – a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11 & 12(1-4):545–581, 1999.

[36] P. Tseng. Further results on approximating nonconvex quadratic optimization by semidefinite programming relaxation. *SIAM Journal on Optimization*, 14(1):268–283, 2003.

[37] N. Tsuchimura, S. Moriguchi, and K. Murota. Discrete convex optimization solvers and demonstration softwares. *Transactions of the Japan Society for Industrial and Applied Mathematic*, 23(2):233–252, 2013. (In Japanese).

[38] T. Tsuchiya. A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. *Optim. Methods Softw.*, 11 & 12(1-4):141–182, 1999.

[39] Y. Weng, Y. S. Park, and D. Lindgren. Unequal clonal deployment improves genetic gains at constant diversity levels for clonal forestry. *Tree genetics & genomes*, 8(1):77–85, 2012.

[40] C. G. Williams and O. Savolainen. Inbreeding depression in conifers: implications for breeding strategy. *Forest Science*, 42(1):102–117, 1996.

[41] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer Science & Business Media, 2012.

[42] J. Woolliams, P. Berg, B. Dagnachew, and T. Meuwissen. Genetic contributions and their optimization. *Journal of Animal Breeding and Genetics*, 132(2):89–99, 2015.

[43] S. Wright. Coeffcients of inbreeding and relationship. *Am. Nat.*, 56:330–338, 1922.

[44] D. Wu, A. Hu, J. Zhou, and S. Wu. A new convex relaxation for quadratically constrained quadratic programming. *Filomat*, 27(8):1511–1521, 2013.

[45] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakta, and M. Nakata. Latest developments in the SDPA family for solving large-scale SDPs. In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*, chapter 24, pages 687–714. Springer, NY, USA, 2012.

[46] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA6.0 (SemiDefinite Programming Algorithm 6.0). *Optim. Methods Softw.*, 18(4):491–505, 2003.

[47] M. Yamashita, T. J. Mullin, and S. Safarina. An efficient second-order cone programming approach for optimal selection in tree breeding. *arXiv preprint arXiv:1506.04487*, 2015.

[48] M. Yamashita and K. Nakata. Fast implementation for semidefinite programs with positive matrix completion. *Optim. Methods Softw.*, 2015. to appear.

[49] Y. Ye. Approximating global quadratic optimization with convex quadratic constraints. *Journal of Global Optimization*, 15(1):1–17, 1999.