



**ISE**

Industrial and  
Systems Engineering

## **Multi-start Approach to Global Conic Optimization**

Tamás Terlaky  
Lehigh University

R. Sotirov  
Tilburg University

Report: 10T-001

# Multi-start Approach to Global Conic Optimization

R. Sotirov\*      T. Terlaky†

February 3, 2010

## Abstract

In this paper we present a multi-start approach to solve a determinant maximization problem of an integer matrix variable that has a constant trace. In order to enforce that elements of the matrix variable are integer, we add a nonlinear repeller term to the objective function. To solve this nonlinear optimization problem we implement our sequential conic trust region based algorithm within a multi-start framework. Computational results demonstrate the potential of our approach.

**Keywords:** determinant maximization, trust-region algorithm, conic optimization, multi-start framework

## 1 Introduction

The determinant maximization problem can be considered as a generalization of semidefinite programming (SDP). In addition to having SDP as a special case, max-det problems also arise in many fields, including crystallography, data mining, statistics, computational geometry, information and communication theory, etc. For an extensive account of applications of max-det problems see [27] and the references therein. Due to the theoretical and practical importance, the max-det problem is a well studied problem, see [19, 22, 24, 27]. Also, there are interior point methods (IPM) based softwares, e.g., SDPT3 [25] and MAXDET [27], that can efficiently solve determinant optimization problems with linear matrix inequality constraints.

In this paper we consider a determinant optimization problem for which a matrix variable is integer and has a constant trace. We show that if we relax the integrality constraints, the optimal solution of this problem is always a diagonal matrix. More precisely, we show that the determinant of a positive definite

---

\*Department of Econometrics and Operations Research, Tilburg University, The Netherlands. r.sotirov@UvT.nl

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA, terlaky@Lehigh.edu. Partially supported by a start-up package of Lehigh University, The Canada Research Chair Program and NSERC.

matrix with constant trace is maximal for some diagonal matrix. In order to enforce integrality constraints to the matrix variable in the relaxed problem, we add a nonlinear term to the objective function that is known as the repeller term, see [2]. To solve this nonlinear determinant maximization problem we implement a sequential trust region algorithm. Since the sequential algorithm converges locally, it is very important to choose a “good” starting point. Due to the structure of the problem, it is not clear which feasible point should be taken as an initial one. Therefore we set our sequential trust region algorithm into a multi-start framework.

Multi-start methods have recently turned out to be a very efficient tool for global optimization, see e.g., [3, 6, 7, 18, 20, 26]. In general, as searching for a global optimum a multi-start method restarts a local search procedure from different initial points. While traditional multi-start algorithms apply a local search repeatedly to random feasible points, adaptive multi-start techniques use more sophisticated constructions of initial points, see e.g., [3, 4, 6, 7, 20]. In the latter approach, one explores information from the previous sampling points, or some properties of the feasible set in order to find “better” initial points. In our multi-start procedure, we combine ideas from both approaches. Namely, we first start our sequential algorithm from a randomly chosen feasible point. If a solution matrix obtained from the sequential algorithm does not satisfy integrality constraints, we apply the gradient ascent approach to all noninteger matrix elements in order to find a feasible point outside of a valley of the local optimum. Since a so obtained matrix does not need to satisfy integrality constraints and is not a local optimum, we restart from that point our sequential algorithm. In this way we further explore the neighborhood of the initial random point. We repeat exchanging the sequential algorithm and the gradient ascent method till there is no more improvement in the objective value, or some of the stopping criteria are satisfied. After the described process stops, we restart the whole procedure from a new random initial point. Finally, for the solution of the determinant optimization problem we take the best integral solution among all computed local optimum. Although the multi-start techniques are used to solve global optimization problems successfully, this is for the first time that they are combined with an interior point solver for conic optimization.

To illustrate our approach, we use a determinant maximization model from crystallography where one tends to maximize a determinant of the so called Karle-Hauptman (KH) matrix [9, 11, 23]. We also verify computational results obtained by our algorithm for randomly taken data matrices.

The paper is organized as follows. In the next section we present the determinant optimization problem of our interest. In Section 3, we describe our adaptive multi-start sequential conic-trust region algorithm and in Section 4 we provide numerical results. Concluding remarks are give in Section 5.

## Notation

The space of  $k \times k$  symmetric matrices is denoted by  $\mathcal{S}_k$ , and the space of

$k \times k$  symmetric positive semidefinite matrices by  $\mathcal{S}_k^+$ . For  $A, B \in \mathcal{S}_k$ ,  $A \succeq 0$  (resp.  $A \succ 0$ ) denotes positive semidefiniteness (resp. positive definiteness), and  $A \geq 0$  elementwise nonnegative matrix. We use  $\text{tr}(A)$  to denote the trace of a square matrix  $A$ ,  $\text{diag}(X)$  the vector of the diagonal elements of matrix  $X$ , and  $I_n$  is the identity matrix of order  $n$ .

The Hadamard product of two matrices  $U = (u_{ij})$  and  $V = (v_{ij})$  of the same size is denoted by  $U \circ V$ , where  $(U \circ V)_{ij} := u_{ij} \cdot v_{ij}$  for all  $i, j$ .

## 2 Determinant maximization problem

We consider the following determinant optimization problem

$$\min -\log \det(B \circ X) \quad (1)$$

$$\text{s.t. } \text{tr}(X) = \alpha \quad (2)$$

$$B \circ X \succ 0 \quad (3)$$

$$X \in \mathcal{S}_n \quad (4)$$

$$x_{ij} \in \{-1, 1\}, \quad \forall i, j, \quad i \neq j, \quad (5)$$

where  $B \in \mathcal{S}_n$ ,  $B \geq 0$  is a given matrix, and  $\alpha \geq 0$  is a given number. By relaxing constraints (5) to

$$-1 \leq x_{ij} \leq 1, \quad \forall i, j, \quad i \neq j, \quad (6)$$

a solution of the corresponding relaxed problem is a scalar multiple of the identity matrix. Namely, it follows from Hadamard inequality [12] that the determinant of a positive definite matrix is maximized when the matrix is diagonal. This was shown in [19], but in the sequel we provide another proof.

**Lemma 1.** *Among the  $n \times n$  positive definite matrices  $X$  with trace  $\text{tr}(X) = \alpha$ , the determinant is maximized when  $X = \frac{\alpha}{n} I_n$ .*

*Proof.* The claim follows from Hadamard's inequality (see [12], pg. 477) and from the fact that the optimum of the following convex optimization problem

$$\min \left\{ -\log \left( \prod_{i=1}^n \lambda_i \right) : \sum_{i=1}^n \lambda_i = \alpha, \lambda_i > 0, i = 1, \dots, n \right\}$$

is  $X = \frac{\alpha}{n} I_n$ . □

Lemma 1 further implies:

**Lemma 2.** *Let  $B \in \mathcal{S}_n$ ,  $B \geq 0$ . Then, among the  $n \times n$  positive definite matrices  $X$  with constant trace  $\text{tr}(X) = \alpha$  and s.t.  $B \circ X \succ 0$ , the determinant  $\det(B \circ X)$  is maximized when  $X = \frac{\alpha}{n} I_n$ .*

*Proof.* Follows trivially from Lemma 1. □

In order to enforce constraints (5) to the relaxed problem (1)–(4), (6) we add to the objective function a *repeller* term. Namely, for every  $x_{ij}$ , ( $i \neq j$ ) we introduce the following function

$$f(x_{ij}) = -\log(x_{ij}^2). \quad (7)$$

Function  $f(\cdot)$  is convex on the intervals  $(-\infty, 0)$  and  $(0, \infty)$ , and obtains large values for  $|x_{ij}| \ll 1$ ,  $x_{ij} \neq 0$ . Thus, function  $f(\cdot)$  acts as a *repeller* for the off-diagonal elements of the matrix variable  $X$  that are *attracted* to zero in the optimization process. Similar attractor–repeller approach is applied in Anjos and Vannelli [2], but their model does not contain linear matrix inequalities. Our *attractor–repeller* SDP relaxation is

$$\left. \begin{array}{l} \min \quad -\log \det(B \circ X) - \sum_{i \neq j} \log(x_{ij}^2) \\ \text{s.t.} \quad \text{tr}(X) = \alpha \\ \quad \quad B \circ X \succ 0 \\ \quad \quad -1 \leq x_{ij} \leq 1, \quad \forall i, j, \quad i \neq j \\ \quad \quad X \in \mathcal{S}_n. \end{array} \right\} \text{(AR)}$$

Note that in (AR) one can not require  $X \succ 0$ , since  $B$  does not need to be positive definite. (AR) is a nonlinear semidefinite problem (NL–SDP) and there are no solvers that can directly solve it. Therefore, we develop our own algorithm for solving (AR) which is presented in the sequel.

The following lemma is crucial for interpreting numerical results.

**Lemma 3.** *Let  $X^*$  be a solution to the problem (1)–(5) and the corresponding objective be  $\mu^* = -\log \det(B \circ X^*)$ . Then, there are  $2^n$  matrices equivalent to  $X^*$  for which the objective value is also  $\mu^*$ .*

*Proof.* It is clear that a value of the determinant of  $B \circ X^*$  does not change if the  $i$ -th row and column of  $X^*$  are multiplied by  $-1$ . Also, the determinant of  $B \circ X^*$  does not change if two rows/column of  $X^*$  are multiplied by  $-1$ , etc.  $\square$

### 3 A multi-start sequential algorithm

Our aim is to solve the determinant problem (AR) by solving a sequence of conic trust region subproblems within a multi-start framework.

In Section 3.1 we derive a conic trust region subproblem from the nonlinear attractor–repeller model (AR). In Section 3.2 the conic subproblem is integrated into a sequential framework. Sequential methods for semidefinite programming are analyzed and successfully applied in [1, 8, 14]. In Subsection 3.3 we incorporate our sequential conic trust–region algorithm within a multi-start setting. Multi-start algorithms are used for solving global optimization problems. To date, in most of the multi-start approaches [3, 18, 20, 26], heuristics or gradient-based solvers are used to solve the corresponding subproblems. To the best of

our knowledge, we are the first to implement an IPM solver within the multi-start framework.

### 3.1 A conic trust region subproblem

The papers [13, 28, 29] present penalty-based and IPM algorithms for solving nonlinear SDP and/or SOCO problems, however there is no publicly available software that can solve a determinant maximization nonlinear conic problem. Therefore we develop our own solver for solving (AR). Since it is well known that IPMs solve linear conic problems efficiently (see e.g., [21, 25]), we introduce an IPMs based algorithm that solves the derived determinant optimization conic trust region subproblem.

For a given point, the subproblem is derived in the following way. The repeller term (7) in the objective of (AR) is expanded into Taylor series up to order two about the given point, and a trust region around the given point is added as a constraint.

Let  $\bar{X} = (\bar{x}_{ij})$  be the given point and  $h_{ij} \in \mathbb{R}$  be the displacement in  $\bar{x}_{ij}$  for all  $i, j, i \neq j$ . The conic trust region subproblem derived from (AR) is:

$$\left. \begin{array}{l} \min \quad -\log \det(B \circ X) - \sum_{i \neq j} \left[ \log(\bar{x}_{ij}^2) + \frac{2}{\bar{x}_{ij}} h_{ij} - \frac{1}{\bar{x}_{ij}^2} h_{ij}^2 \right] \\ \text{s.t.} \quad \text{tr}(X) = \alpha \\ (B \circ X) \succ 0 \\ x_{ij} = \bar{x}_{ij} + h_{ij}, \quad \forall i, j, \quad i \neq j \\ -1 \leq \bar{x}_{ij} + h_{ij} \leq 1, \quad \forall i, j, \quad i \neq j \\ X \in \mathcal{S}_n \\ \sum_{i \neq j} h_{ij}^2 \leq \Delta^2, \end{array} \right\} (\text{AR}_{\text{trust}})$$

where  $\Delta$  is the trust region radius. Note that for every fixed  $\bar{x}_{ij}$ , the following function

$$\bar{f}(h_{ij}) := - \left( \log(\bar{x}_{ij}^2) + \frac{2}{\bar{x}_{ij}} h_{ij} - \frac{1}{\bar{x}_{ij}^2} h_{ij}^2 \right)$$

acts as a *repeller* for the small off diagonal elements. Since the trust region constraint

$$\sum_{i,j} h_{ij}^2 \leq \Delta^2$$

is a second-order cone constraint (SOCO), it follows that  $(\text{AR}_{\text{trust}})$  is a nonlinear determinant maximization SDP-SOCO problem. Therefore, in the remaining part of the paper we refer to  $(\text{AR}_{\text{trust}})$  as the conic trust region subproblem. Note that if we have any additional conic constraint in the original problem, we can apply the same approach.

**Remark 4.** *Ellipsoidal trust region could be also used. The ellipsoid could be defined as the “Dikin ellipsoid”:*

$$\sum_{i,j} \left( \frac{h_{ij}}{\min\{1 - x_{ij}, 1 + x_{ij}\}} \right)^2 \leq \Delta.$$

### 3.2 A sequential trust region algorithm for solving (AR)

In this subsection we describe our trust region based algorithm that solves (AR) by solving a sequence of conic trust region subproblems (AR<sub>trust</sub>). A variant of our sequential trust region algorithm is successfully applied in [1] for solving a nonlinear SDP problem arising in magnetic resonance imaging. In the sequel we describe our algorithm and present the corresponding pseudo code in Figure 1.

*Initialization step.* As a *starting point* we take a symmetric random matrix that is feasible for (AR). More precisely, a matrix  $X^0 \in \mathcal{S}_n$  whose elements are normally distributed random numbers from the interval  $(-1, 1)$ ,  $\text{tr}(X^0) = \alpha$ ,  $(B \circ X^0) \succ 0$ . If the data matrix  $B$  has zero elements, then we set the corresponding elements of  $X^0$  to zero as well. We specify the initial *trust-region radius* to be  $\Delta^0 = 1$ .

*General step.* To describe a general step of the algorithm, we assume to have a *current feasible point*  $\bar{X}$ . At the current point, let

$$\begin{aligned} \mathbf{det}_{\bar{X}} &:= -\log \det(B \circ \bar{X}) \\ \mathbf{detLn}_{\bar{X}} &:= -\log \det(B \circ \bar{X}) - \sum_{i \neq j} \log(\bar{x}_{ij}^2) \\ \mathbf{trust}_{\bar{X}} &:= -\log \det(B \circ \bar{X}) - \sum_{i \neq j} \left[ \log(\bar{x}_{ij}^2) + \frac{2}{\bar{x}_{ij}} h_{ij} - \frac{1}{\bar{x}_{ij}^2} h_{ij}^2 \right]. \end{aligned}$$

Optimization of (AR<sub>trust</sub>) around the current point  $\bar{X}$  and with respect to the corresponding trust region radius  $\Delta$  gives a *new candidate point*  $X = (x_{ij})$ ,  $x_{ij} = \bar{x}_{ij} + h_{ij}$  and the corresponding value of the objective function  $\mathbf{trust}_X$ . Note that the diagonal elements in  $X$  are specified by  $\text{tr}(X) = \alpha$ ,  $(B \circ X) \succ 0$ . For the new candidate point we also compute  $\mathbf{detLn}_X$  and the *trust region ratio* in the following way:

$$\varrho = \frac{\mathbf{detLn}_X - \mathbf{detLn}_{\bar{X}}}{\mathbf{trust}_X - \mathbf{trust}_{\bar{X}}}. \quad (8)$$

Note that the closer the value of  $\varrho$  is to one, the better approximation of (AR) by the model (AR<sub>trust</sub>) is realized. A negative or very small  $\varrho$  indicates a poor approximation, and therefore the point is rejected and the trust region radius is reduced as described in (9). If a sufficient reduction in the objective function is obtained at the candidate point, then that point is accepted as the next iterate and the trust-region radius is expanded or kept the same, as specified below

$$\Delta := \begin{cases} c_1 \Delta, & \text{if } \varrho < r_1 \\ c_2 \Delta, & \text{if } \varrho > r_2 \\ \Delta, & \text{otherwise.} \end{cases} \quad (9)$$

In our computations we set  $c_1 = 0.2$ ,  $c_2 = 0.95$ ,  $r_1 = 0.2$ ,  $r_2 = 1.5$ . The choice of these parameters is made after extensive testing and benchmarking. For more information on trust-region algorithms see e.g., [5].

*Stopping criteria.* There are several conditions for terminating Algorithm I. We stop the algorithm when the trust region becomes very small ( $\leq 10^{-6}$ ), or too large ( $> 4\Delta^0$ ). Another criterion is when after a pre-specified number of iterations there is no significant improvement with respect to the objective. We stop the algorithm if  $\det_{\bar{X}} \leq 10^{-8}$ . That means that at the current point  $\bar{X}$ , matrix  $B \circ \bar{X}$  has at least one eigenvalue that is close to zero. More precisely, this matrix is close to the boundary of the semidefinite cone, and the values  $\det_{\bar{X}}$  and  $\text{trust}_{\bar{X}}$  are large which is due to the repeller term in the objective. Note that in an optimal point, the values of  $\text{trust}_{\bar{X}}$  and  $\det_{\bar{X}}$  are equal.

**Remark 5.** *In our algorithm, we explore the fact that data matrix  $B$  is sparse. Namely, for  $B_{ij} = 0$  we set  $X_{ij}^0 = 0$  and do not compute corresponding displacement  $h_{ij}$  in any step of the algorithm. Therefore all matrices  $\bar{X}$  computed by Algorithm I have the same sparsity pattern. This results with fast computations for sparse problems.*

### 3.3 An adaptive multi-start framework

Our sequential trust-region algorithm has the following drawbacks. Algorithm I, like most of the trust-region algorithms, force convergence to a local optima from a remote starting point, and the computed optima does not have to satisfy integrality conditions. The following two things can be done in order to improve a local optima. *First*, is to further explore the neighborhood of the nonintegral local optima by applying a gradient ascent approach. *Second*, if the local optima is integer or there are no improvements in the neighborhood of the nonintegral local optima, one can restart Algorithm I from a new randomly chosen starting point. The best among the integral local solutions is considered to be the solution of problem (1)–(5). In the remaining part of this section we describe in more detail the gradient ascent and the multi-start approach of our adaptive multi-start framework. A pseudo code of our adaptive multi-start algorithm is given in Figure 2.

*The gradient ascent approach.* Suppose that for a given initial feasible point, Algorithm I returns matrix  $\bar{X}$  that contains elements whose absolute values are not equal to one. Since  $(B \circ \bar{X})$  has an eigenvalue that is very close to zero, we backtrack from Algorithm I to a point that is in the interior of the semidefinite cone, say  $\tilde{X}$ . Now, we apply the method of steepest ascent (see e.g., [16]) to all noninteger off-diagonal matrix elements of  $\tilde{X}$  while holding the integer elements fixed. In the following lemma we derive the partial derivatives that are used in the steepest ascent method.



---

**Algorithm I**

---

**Input:**  
a starting point:  $X^0$ ;  
initial trust region radius:  $\Delta^0 = 1$ ;  
input parameters:  $r_1 = 0.2, r_2 = 1.5, c_1 = 0.2, c_2 = 0.95$ ;

**begin**  
 $\Delta \leftarrow \Delta^0$ ;  
 $\bar{X} \leftarrow X^0$ ;

**while** one of the *stopping criteria* is satisfied (see page 7)  
solve (AR<sub>trust</sub>): return **trust** <sub>$X$</sub>  &  $(h_{ij}, \forall i, j, i \neq j)$ ;  
new candidate point:  $x_{ij} = \bar{x}_{ij} + h_{ij}, \forall i, j, i \neq j$ ;  
compute  $\varrho$  from (8);  
**if**  $\varrho < r_1$   
 $\Delta \leftarrow \Delta \cdot c_1$ ;  
**else**  
update  $\Delta$  according to (9);  
 $\bar{X} \leftarrow X$ ;  
**end**  
**end**  
return  $\bar{X}$ , **trust** <sub>$\bar{X}$</sub> , **detLn** <sub>$\bar{X}$</sub> , **det** <sub>$\bar{X}$</sub> ;  
**end**

Figure 1: Sequential trust-region algorithm.

**Lemma 6.** Let  $B, X \in S_n$ ,  $B = (b_{ij})$ ,  $B \geq 0$ ,  $X = (x_{ij})$  and  $C = B \circ X$ . Then,

$$\frac{\partial \det(C)}{\partial x_{kl}} = d_{lk} b_{kl} \det(C), \quad (10)$$

where  $D = (d_{ij})$  is the inverse of  $C$ .

*Proof.* From the chain rule and the relations

$$\frac{\partial \det(C)}{\partial c_{kl}} = d_{lk} \det(C) \quad \text{and} \quad \frac{\partial c_{kl}}{\partial x_{kl}} = b_{kl}$$

formula (10) follows.  $\square$

We iterate the steepest ascent algorithm, while taking care that the absolute value of each off-diagonal element does not exceed one, the current point is feasible, and till no improvement in the objective. Note that when applying the element-wise steepest ascent method, we move away the current point from the local optimum that was computed by Algorithm I. Thus, by applying the steepest ascent method we “climb the hill” in the neighborhood of the local

minimum. Since after the element-wise steepest ascent algorithm the computed matrix does not have to have all  $\pm 1$  elements and is not a local minimum, we restart Algorithm I from that point. The algorithm repeats this procedure until a stopping criteria is satisfied, i.e., when an optimal integer point is found or if there is no improvement in the objective after a prescribed number of steepest ascent rounds. We call our algorithm *adaptive* since a randomly chosen starting point is adapted by the steepest ascent method. For details on the steepest ascent approach see Figure 2.

*Multi-start approach.* Our adaptive multi start algorithm attempts to find a global minimum to problem (1)–(5) by restarting the conic trust–region algorithm from multiple starting points. A starting point is a random feasible matrix, or a matrix obtained from a local optimum from which the element-wise steepest ascent algorithm is applied. Every computed integral solution is stored, and among them the one that corresponds to the lowest objective value is reported as a solution to problem (1)–(5). Note that larger number of runs, increases the probability of finding a global solution.

For each data matrix  $B$ , there are several equivalent matrices for which the value of the objective function is the same, see Lemma 3. For an optimal point it is easy to identify all the equivalent optimal points, if needed.

## 4 Numerical results

In this section we present our computational results on solving problem (1)–(5) by our multi-start sequential algorithm. We first present numerical results obtained by solving the maximum determinant problem that appears in crystallography, and then present our results for randomly generated matrices.

We implement our multi-start sequential algorithm in MATLAB on a Pentium IV, 3.4 GHz dual-core processor. To solve a sequence of conic trust region subproblems we use the SDPT3 [25] software and the YALMIP [15] interface. We restrict our search to 25 random point restarts, and for each restart to 7 local search loops.

*Determination of crystal structures using the Karle–Hauptman matrix.*

In order to determine a crystal structure from a single crystal  $x$ -ray diffraction experiment, one needs to solve a determinant optimization problem, see [9, 23]. To form this optimization problem, it is necessary to construct a Karle–Hauptman (KH) matrix [10, 23]. The KH matrix is a symmetric matrix that is derived in the following way.

From the single crystal  $x$ -ray diffraction experiment one records  $M$  data points that are associated with the Miller index  $H_k$  and *diffraction intensity*  $E_{H_k} \geq 0$ ,  $k = 1, \dots, M$ . To construct a KH matrix  $A = (a_{ij})$  of order  $n \leq M$ , a subset of  $n - 1$  data points must be chosen. We assume w.l.g. that elements

---



---

**Algorithm II**

---



---

**Input:**  
data matrix:  $B$ ;  
 $\beta = 0.9$ ;  
 $k \leftarrow 1$ ;  
**repeat** for a given  $\#$  of starting points  
a random starting point  $X^0$   
start **Algorithm I** from  $X^0$ : return  $\bar{X}$ ,  $\det_{\bar{X}}$   
br  $\leftarrow 1$ ;  
**if**  $\bar{X}$  is not integral  
**while** br  $\leq$  the prescribed number of local searches  
**do** *steepest ascent* for the noninteger elements  
till one of the stopping criteria is satisfied (see page 7)  
 $\tilde{x}_{ij} \leftarrow \tilde{x}_{ij} + \beta \cdot \text{inv}(B \circ \tilde{X})_{ij} \cdot \det(B \circ \tilde{X}) B_{ij}$ ;  
**end**  
start **Algorithm I** from  $\tilde{X}$ : return  $\bar{X}$ ,  $\det_{\bar{X}}$   
br  $\leftarrow \text{br} + 1$ ;  
**end**  
**end**  
**if**  $\bar{X} \in \{-1, 1\}^n$  &  $\det(B \circ X) > 0$ , then  $X_k \leftarrow \bar{X}$ ;  $k \leftarrow k + 1$ ; **end**  
**end**  
 $X^* = \min_k \{-\log \det(B \circ X_k)\}$

Figure 2: The adaptive multi-start algorithm.

in the subset are numbered from 2 to  $n$ , and redefine

$$H_{1k} := H_k, \quad k = 2, \dots, n.$$

The diffraction intensities which are corresponding to the data points from the subset of  $n - 1$  elements, are used to generate the first row of the KH matrix in the following way:

$$a_{1k} = E_{H_{1k}} \cdot \exp(i \cdot \phi_{H_{1k}}), \quad k = 2, \dots, n, \quad (11)$$

where  $i$  is the imaginary unit, and  $\phi_{H_{1k}}$  an unknown variable. The expression on the right hand side in (11) is known as a *structure factor*, and the unknown variable as a *phase* of the structure factor. All diagonal elements in the KH matrix are equal to the given value  $E_{00}$ , i.e.,

$$a_{kk} = E_{00}, \quad k = 1, \dots, n. \quad (12)$$

The remaining elements of the KH matrix are generated in the following way

$$a_{kl} = E_{H_{kl}} \exp(i \cdot \phi_{H_{kl}}), \quad k \neq l, \quad k, l \geq 2, \quad (13)$$

where

$$H_{kl} := H_{1l} - H_{1k},$$

and  $\phi_{H_{kl}}$  is an unknown phase of the corresponding structure factor. In the case that  $H_{kl}$  is not recorded  $E_{H_{kl}} = 0$ , and therefore we set  $a_{kl} = 0$ .

For a KH matrix  $A$  the following properties [9, 23] are known:

1.  $A$  is Hermitian.
2.  $A$  is positive semidefinite.
3.  $\det(A) > 0$  if  $n \leq M$ ,  $\det(A) = 0$  if  $n > M$ .
4. The maximum-determinant rule holds: when a given KH matrix  $A$  of order  $n$ , containing structure factors with known phases, is enlarged to a KH matrix  $\bar{A}$  of order  $n + 1$  by adding a row and a column containing elements with unknown phases, then

$$\det(A) \geq \det(\bar{A}).$$

The last property is a consequence of the fact that the determinant of the KH matrix is a special case of the Gram determinant, see [23].

It is proven in [23] that for a given KH matrix, which contains structure factors with unknown phases, the most probably set of phases will maximize the determinant of the KH matrix. Therefore, unknown phases can be derived from a solution of the following optimization problem, see [11]:

$$\left. \begin{array}{l} \min \quad -\log \det(A) \\ \text{s.t.} \quad A \succ 0 \\ \quad \quad A = A(\phi_{H_{kl}}) \\ \quad \quad \phi_{H_{kl}} \in [0, 2\pi], \quad k, l = 1, \dots, n, \quad k \neq l, \end{array} \right\} \quad (14)$$

where elements of  $A$  are of the form (11)–(13). In general, when a crystal structure is not specified, unknown phases  $\phi_{H_{kl}}$  can obtain any value from  $[0, 2\pi]$ . Here, we are interested in a *centrosymmetric structure* i.e., the structure for which  $\phi_{H_{kl}} \in \{0, \pi\}$ ,  $\forall k, l$ . Note that in the view of (12), we can assign

$$\phi_{H_{kk}} = 0, \quad \forall k. \quad (15)$$

Therefore, for this special case variables  $x_{kl}$ , which are defined in the following way

$$x_{kl} := \exp(i \cdot \phi_{H_{kl}}), \quad \forall k, l,$$

satisfy  $x_{kl} \in \{-1, 1\}$ . After assigning  $E_{H_{kk}} = E_{00}$ , for all  $k$ , w.r.t. (15), it follows from (12)–(13) that  $a_{kl} = E_{H_{kl}} x_{kl}$ , for all  $k, l$ .

Let  $\bar{A} := (E_{H_{kl}})$  be a matrix of diffraction intensities. Then for a centrosymmetric crystal structure and  $X = (x_{ij}) \in \mathcal{S}_n$ ,  $x_{ij} \in \{-1, 1\}$ ,  $x_{ii} = 1$ , it follows that  $A = \bar{A} \circ X$ . Therefore, an equivalent formulation to (14) is

$$\left. \begin{array}{l} \min \quad -\log \det(\bar{A} \circ X) \\ \text{s.t.} \quad \text{diag}(X) = e \\ \quad \quad (\bar{A} \circ X) \succ 0 \\ \quad \quad x_{kl} \in \{-1, 1\}, \quad k, l = 1, \dots, n, \quad k \neq l, \end{array} \right\} \quad (16)$$

where  $e$  is the all ones vector.

Clearly, determinant optimization problem (16) is of the form of (1)–(5). In our computations, we choose elements for the first row/column of a KH matrix such that the associated matrix  $\bar{A}$  has large off-diagonal elements. This is a standard practice in phase determination computations. The data matrix formulated in this way does not have to be positive definite and might be very sparse. However, Algorithm II can efficiently explore sparsity in the data matrix, see Remark 5.

In Table 1 we present computational results obtained by solving problem (16) where matrix of diffraction intensities  $\bar{A}_n$ ,  $n = 11, \dots, 19$  is rescaled so that its diagonal elements are all ones. In the first column of Table 1 the orders of the KH matrices  $A_n = \bar{A}_n \circ X_n$ ,  $n = 11, \dots, 19$  are listed, and in the second column the corresponding determinant values  $\det(\bar{A}_n \circ X_n)$  are given, where  $X_n \in \{-1, 1\}^{n \times n}$  is a solution of (16) computed by Algorithm II.

The elements of matrix  $\bar{A}_{11}$  are derived as described earlier, and  $X_{11}$  is computed by our algorithm. Matrix  $\bar{A}_{12}$  is obtained from  $\bar{A}_{11}$  by adding the 12th row and column as specified by (13), and with respect to the new data point for which  $\bar{a}_{1,12} = E_{H_{1,12}}$ . Further, by Algorithm II we only compute those elements of  $X_{12}$  that are corresponding to the 12th row/column and remaining ones are fixed values from  $X_{11}$ . Matrix  $\bar{A}_{13}$  is obtained from  $\bar{A}_{12}$  in the similar way, etc.

The results show that when the size of the KH matrix increases, the corresponding determinant value decreases. This is a well known property of KH matrices (see property 4 on page 11). It is also known (see [23]) that when the order of the KH matrix increases, the corresponding determinant values converge to zero. Our numerical results show that the determinant values decrease almost linearly with the order of the KH matrix.

There are two reasons for computing determinant values for matrices only up to order 19. First, is that the quality of a solution computed by our algorithm very likely deteriorate with the size of the problem (see the following two paragraphs). Second, is that the main objective in current crystallography packages such as CRUNCH [8] is to maximize the determinant for a number of small matrices until enough phase determination is available to compose a full density function.

$n$	$\det(A_n)$
11	0.025575
12	0.018151
13	0.010703
14	0.005099
15	0.001850
16	0.000358
17	0.000027
18	0.000003
19	3.044749e-007

Table 1: Changes in the  $\det(A_n)$  with respect to  $n$ .

*Computational results for randomly chosen data matrices.* We have tested the quality of the solution obtained by Algorithm II for problem (1)–(5), where data matrix  $B_n$  is a randomly chosen matrix of order  $n$ , and we also take  $\alpha = n$ . For each  $n = 6, 7, 8$  we formulate 10 data matrices  $B_n$  for which we compute a solution to problem (1)–(5) by our algorithm. We compare these 10 solutions with an optimal solution of the underlined problem that is obtained by exhausting search i.e., by evaluating the determinant values for all feasible  $\{-1, 1\}^{n \times n}$  matrices. The results are as follows.

For problems where the matrix variable is of order 7 or less, we obtain an optimal solution after a few rounds of Algorithm II, i.e., after restarting the algorithm from several starting points. For problems of size 8 we obtain a solution that has on average a value that is at least 85% of the optimal value of the problem. In order to compute an optimal solution for the problem of order 8, we need about 6.5 hours on our PC. Note that for  $n = 8$  there are 268,435,456 symmetric matrices with  $\{-1, 1\}$  elements. Since it is intractable to verify determinant values for all symmetric  $\{-1, 1\}$  matrices of order  $n \geq 9$  and find an optimal solution of problem (1)–(5), we can not estimate the quality of our bounds for larger problems.

## 5 Concluding Remarks

In this paper we combine a sequential conic trust region and a multi-start approach in order to solve the determinant optimization problem with an integral matrix variable. This is for the first time, to the best of our knowledge, that these two approaches are combined. Further, a crucial idea for computing an integral solution matrix is the repeller term that is added to the objective function. The role of this term is to act as a repeller for the matrix elements that are attracted to zero in the relaxed problem. Although we are not the first to use a repeller term (see [2]), we are the first to use it in the conic framework.

We test our algorithm on a problem that arises in crystallography. To demon-

strate the difficulty of finding the maximal determinant of an integral matrix, we recall that identifying the maximal determinant of a  $\{-1, 1\}$  matrix of order 19 is an open problem [17].

We believe that the approach described in this paper can be used for solving the maximal determinant problem with integer variables, when there are no restrictions to the matrix variable. The main difference between our problem and its generalization is that the latter one does not require positive definiteness of the matrix variable. One way to overcome this difficulty is to impose the positive definite constraint to the Gram matrix that corresponds to the matrix variable. Due to the expected difficulties such as additional nonlinear constraints and/or enlarged size of the problem, we leave resolving those issues for future research.

## References

- [1] C. Anand, R. Sotirov, T. Terlaky, and Z. Zheng. Magnetic resonance tissue density estimation using optimal SSFP pulse-sequence design. *Optimization and Engineering*, 8(2):215–238, 2007.
- [2] M.F. Anjos and A. Vannelli. An attractor–repeller approach to floorplanning. *Mathematical Methods of Operations Research*, 56:3–27, 2002.
- [3] K.D. Boese, A.B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16:101–113, 1994.
- [4] D. Bonachea, E. Ingberman, J. Levy, and S. McPeak. An improved adaptive multi-start approach to finding near-optimal solutions to the Euclidean TSP. In: Proceedings of the Genetic and Evolutionary Computation Conference, Las Vegas, Nevada, 143–150, 2000.
- [5] A.R. Conn, N.I.M. Goud, and P.L. Toint. *Trust Region Methods*. MPS–SIAM Series on Optimization, 2000.
- [6] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- [7] C. Fleurent and F. Glover. Improved constructive multistart strategies for the Quadratic Assignment Problem using adaptive memory. *INFORMS Journal on Computing*, 11:198–204, 1999.
- [8] R.W. Freund and F. Jarre. A sensitivity analysis and a convergence results for a sequential semidefinite programming method. *Numerical Analysis Manuscript*. No 03-4-9, Bell Laboratories, Murray Hill, New Jersey, 2003.
- [9] R. de Gelder, R.S.G. de Graaff, and H. Schenk. Automatic determination of crystal structures using Karle–Hauptman matrices. *Acta Crystallographica*, A49:287–293, 1993.

- [10] R. de Gelder, R.S.G. de Graaff, and H. Schenk. On construction of Karle-Hauptman matrices. *Acta Crystallographica*, A46:688–692, 1990.
- [11] R.A.G. de Graaff and W.J. Vermin. The use of Karle-Hauptman determinants in small-structure determinations. II. *Acta Crystallographica*, A38:464–470, 1982.
- [12] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [13] M. Kočvara and M. Stingl. PENNON: a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.
- [14] S. Kruk and H. Wolkowicz. SQ<sup>2</sup>P, Sequential quadratic constrained quadratic programming. In: *Advances in Nonlinear Programming*, Kluwer Academic, 177–204, 1998.
- [15] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. Proceedings of the CACSD Conference, Taipei, Taiwan, 2004. <http://control.ee.ethz.ch/~joloef/yalmip.php>
- [16] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in OR, 2006.
- [17] W.P. Orrick. The maximal  $\{-1, 1\}$ -determinant of order 15. *Metrika*, 62:195–219, 2005.
- [18] G. Palubeckis. Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research*, 131:259–282, 2004.
- [19] O. Popescu, C. Rose, and D.C. Popescu. Maximizing the determinant for a special class of block-partitioned matrices. *Mathematical Problems in Engineering*, 1:49–61, 2004.
- [20] A.E. Sepulveda and L. Epstein. The repulsion algorithm, a new multistart method for global optimization. *Journal of Structural and Multidisciplinary Optimization*, 11(3/4):145–152, 1996.
- [21] J.F. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002.
- [22] K.C. Toh. Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. *Computational Optimization and Applications*, 14(3):309–330, 1999.
- [23] G. Tsoucaris. A new method for phase determination. The maximum determinant rule. *Acta Crystallographica*, A26:492–499, 1970.



- [24] T. Tsuchiya and Y. Xia. An extension of the standard polynomial-time primal-dual path-following algorithm to the weighted determinant maximization problem with semidefinite constraints. *Pacific Journal of Optimization*, 3(1):165–182, 2007.
- [25] R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming B*, 95:189–217, 2003.
- [26] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.
- [27] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.
- [28] H. Yamashita and H. Yabe. A primal-dual interior point method for nonlinear optimization over second-order cones. *Optimization Methods and Software*, 24(3):407–426, 2009.
- [29] H. Yamashita, H. Yabe and K. Harada. A primal-dual interior point method for nonlinear semidefinite programming. Preprint 2007. Available at: [http://www.optimization-online.org/DB\\_FILE/2007/06/1692.pdf](http://www.optimization-online.org/DB_FILE/2007/06/1692.pdf).