Proceedings of ALGORITMY 2020
pp. 161–170

FINDING LOW-RANK SOLUTIONS IN FINANCIAL FACTOR MODELS

TERÉZIA FULOVÁ *

Abstract.
In financial factor models based on the structure of a correlation matrix, the rank of the correlation matrix should be equal to the number of factors. However, it is not so rare to obtain a high-rank correlation matrix from the given data in practical applications. Therefore, it is necessary to find the nearest low-rank correlation matrix to the computed one. If we take the Frobenius norm to measure the "nearness" of two matrices, we will show that this problem can be formulated in the form of a rank-constrained semidefinite program. Although this kind of problem is considered to be NP-hard, there are some rank reduction techniques to deal with this non-convex rank constraint.

Key words. low-rank correlation matrix, rank-constrained semidefinite program, rank reduction algorithms

AMS subject classifications. 90C22, 90C26, 91-08

1. Introduction. As stated in [8] and [15], the problem of finding a low-rank correlation matrix to the given approximate correlation matrix arose as a part of the calibration of the so-called *multi-factor market model of interest rates*. Financial institutions use this model for pricing their interest rate derivatives portfolio. The variables are the interest rates assumed to follow log-normal stochastic processes. With historical data, a correlation structure of interest rates can be extracted. The idea of the model is then to implant the correlation structure into the stochastic processes for the interest rates, so that the model can appropriately describe the dynamics of interest rates and pricing can be more accurate.

If the model works with k factors, it is evident that the rank of the correlation matrix should not exceed k. If the rank is higher than the number of factors, which is almost always the case, the correlation matrix cannot be used. Therefore, a low-rank correlation matrix is necessary that is the best approximation to the given correlation matrix under the Frobenius norm. That is why the problem of finding the nearest low-rank correlation matrix (2.1) is so significant.

Let us note that the number of interest rates in the model can be enormous in practical applications, i.e. the dimension of the correlation matrix is very high. On the other hand, the term structure of interest rates is driven by multiple factors (four or more) but definitely not as many as the dimension of the matrix.

We will show how to formulate this problem in the form of a rank-constrained semidefinite problem, and consequently, we will be able to apply some rank reduction algorithms to solve it.

1.1. Notation. In this paper, the symbol $\mathcal{S}^n$ denotes the linear subspace of $n \times n$ symmetric matrices. If $X \in \mathcal{S}^n$ is positive semidefinite, we write $X \succeq 0$. For two symmetric matrices A and B of order n we say $A \succeq B$ if $A - B \succeq 0$, which is the definition of the so called Loewner partial ordering ([5, §A.3]).

*Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia (Terezia.Fulova@fmph.uniba.sk).

The Schur complement of a matrix X in block matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is defined as the matrix $D - CA^{-1}B$ assuming matrix A is invertible and according to [4, §A.5.5] it holds

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \Leftrightarrow A \succeq 0 \ \wedge \ D - CA^{-1}B \succeq 0. \tag{1.1}$$

The rank of a matrix X is denoted by $rank(X)$, and the rank of diagonalizable matrices is equal to the number of their non-zero eigenvalues. The symbol λ_i denotes the eigenvalue of a given matrix with the i-th largest eigenvalue. In case the given matrix is positive semidefinite, λ_i is always real.

The Frobenius norm of a matrix X is defined as

$$\|X\|_F = \sqrt{tr(XX^T)}, \tag{1.2}$$

where $tr(.)$ denotes the trace of a matrix. As stated in [14, §6], Loewner partial ordering has the following property

$$A \succeq B \Rightarrow tr(A) \geq tr(B) \tag{1.3}$$

if A and B are positive semidefinite.

We denote by $\mathcal{N}(X)$ the nullspace of a matrix X and by I the identity matrix, whose dimension should be apparent from the context.

2. Problem formulation. Given a real symmetric matrix C of order n, the task is to find the nearest low-rank correlation matrix. If we take the Frobenius norm to measure the "nearness" of two matrices, this problem can be formulated as a rank-constrained optimization problem

$$\begin{array}{lrcl} \min\limits_{X \in \mathcal{S}^n} & \|C - X\|_F^2 & & \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0, \\ & rank(X) & \leq & k, \end{array} \tag{2.1}$$

where $X \in \mathcal{S}^n$ is the matrix variable and $k \in N$ is the desired rank. The first two constraints of the problem force the matrix variable to satisfy the definition of the correlation matrix, i.e. to have unit diagonal and to be positive semidefinite. The last constraint guarantees that its rank is at most k, where k represents the number of factors in the model.

If the rank constraint is ommited from the formulation (2.1), we obtain a convex relaxation of the original problem in the form

$$\begin{array}{lrcl} \min\limits_{X \in \mathcal{S}^n} & \|C - X\|_F^2 & & \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0. \end{array} \tag{2.2}$$

The problem (2.2) represents finding the nearest correlation matrix X to the given symmetric matrix C disregarding its rank. Equivalently, we can see the problem (2.2) as the rank-constrained problem (2.1) when choosing $k = n$.

3. Literature review. Because of its importance in finance, the problem (2.1) was recognized by many researchers. In the next, we mention the most known algorithms for solving this problem.

There are several specific methods for this particular problem. Grubisic and Pietersz introduced a geometric optimization algorithm in [8], based on parametrizing the constraint set by the Cholesky manifold and using and applying standard algorithms over the manifold. Its disadvantage is a large number of iterations. The article [15] offers a Lagrange multiplier algorithm. However, it was shown that the convergence of this method is not guaranteed. In [12], majorization is suggested as a suitable rank reduction method, but in general, majorization is not considered to be effective enough for practical computation. In [2], the authors introduced the concept of the alternating projections algorithm, which is based on the idea to project the given symmetric matrix onto the cone of matrices with the specified rank.

The problem (2.2) is usually solved by the alternating projections algorithm introduced in [9]. Another approach is to apply the preconditioned Newton method, which was originally introduced by Qi and Sun in [13] and modified by Borsdorf and Higham in [3]. There were also efforts to formulate the problem in the form of a semidefinite program. However, the expression of the Frobenius norm via vectorization of the matrix leads to a large-scale problem, for which the standard solvers for semidefinite programming problems were not sufficient, as mentioned in [9].

4. Semidefinite formulation. In this section, we introduce the technique of how to convert the problem (2.2) into a semidefinite program in its standard form

$$\begin{array}{lccl} \min\limits_{X\in\mathcal{S}^n} & tr(DX) & & \\ & tr(A_iX) & = & b_i, \quad i=1,...,m, \\ & X & \succeq & 0, \end{array} \tag{4.1}$$

where $D, A_1, ..., A_m \in \mathcal{S}^n$ and $b \in \mathcal{R}^m$.

After adding the rank constraint to the formulation (4.1), the problem (2.1) can be analogically converted into a rank-constrained semidefinite problem of the form

$$\begin{array}{lccl} \min\limits_{X\in\mathcal{S}^n} & tr(DX) & & \\ & tr(A_iX) & = & b_i, \quad i=1,...,m, \\ & X & \succeq & 0, \\ & rank(X) & \leq & k, \end{array} \tag{4.2}$$

where $D, A_1, ..., A_m \in \mathcal{S}^n$, $b \in \mathcal{R}^m$ and $k \in N$ is the desired rank of matrix variable X.

The problem (4.1) is known as a semidefinite relaxation of the rank-constrained semidefinite problem (4.2). However, standard solvers for semidefinite programs are proved to converge to a high-rank optimal solution despite the existence of low-rank solutions. Therefore, there arises the need to search for rank reduction methods.

As we mentioned in the previous section, the problem (2.2) can be reformulated as a semidefinite program using the vectorization of matrix $C-X$, but the solvers cannot handle this vectorization. Therefore we offer a new way of handling the Frobenius norm in order to solve a semidefinite program, which can be solved efficiently using interior-point methods.

Using the definition of the Frobenius norm (1.2) we can rewrite the problem (2.2) into the form

$$\begin{array}{rrcl} \min\limits_{X \in \mathcal{S}^n} & tr\left[(C-X)(C-X)^T\right] & & \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0. \end{array} \tag{4.3}$$

After introducing a new matrix variable Z satisfying

$$Z \succeq (C-X)(C-X)^T \tag{4.4}$$

and thanks to the trace property (1.3) we obtain $tr(Z) \geq tr\left[(C-X)(C-X)^T\right]$ and the problem (4.3) can be equivalently reformulated as follows

$$\begin{array}{rrcl} \min\limits_{X \in \mathcal{S}^n} & tr(Z) & & \\ & Z & \succeq & (C-X)(C-X)^T, \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0. \end{array} \tag{4.5}$$

Using the Schur complement property (1.1), the inequality (4.4) is expressed as a linear matrix inequality (LMI) of the form

$$\left[\begin{array}{cc} I & (C-X)^T \\ C-X & Z \end{array} \right] \succeq 0. \tag{4.6}$$

In the last step, we obtain the semidefinite problem

$$\begin{array}{rrcl} \min\limits_{X \in \mathcal{S}^n} & tr(Z) & & \\ & \left[\begin{array}{cc} I & (C-X)^T \\ C-X & Z \end{array} \right] & \succeq & 0, \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0. \end{array} \tag{4.7}$$

Applying the above procedure, the problem (2.1) can be equivalently reformulated as a rank-constrained semidefinite problem of the form

$$\begin{array}{rrcl} \min\limits_{X \in \mathcal{S}^n} & tr(Z) & & \\ & \left[\begin{array}{cc} I & (C-X)^T \\ C-X & Z \end{array} \right] & \succeq & 0, \\ & X_{ii} & = & 1, \quad i = 1, ..., n, \\ & X & \succeq & 0, \\ & rank(X) & \leq & k. \end{array} \tag{4.8}$$

The semidefinite formulation of the problem (2.1) enables using rank reduction algorithms to solve the problem (4.8) since they were developed especially for the semidefinite problems, as explained in the following chapter.

5. Rank reduction algorithms.

5.1. Rank reduction via convex iteration. In [5, §4.4.2], it is proposed that the rank-constrained problem (4.2) can be expressed as an iteration of semidefinite problems where in the t-th iteration the semidefinite problems sequence (5.1) and (5.2) is solved:

$$\begin{aligned} X_t^* = \underset{X\in\mathcal{S}^n}{\operatorname{argmin}} \quad & tr(DX) + \omega_t\, tr(U_{t-1}^* X) \\ tr(A_i X) &= b_i, \quad i = 1, ..., m, \\ X &\succeq 0, \end{aligned} \tag{5.1}$$

and

$$\begin{aligned} U_t^* = \underset{U\in\mathcal{S}^n}{\operatorname{argmin}} \quad & tr(UX_t^*) \\ & 0 \preceq U \preceq I, \\ & tr(U) = n - k, \end{aligned} \tag{5.2}$$

where $\omega \geq 0$ is the relative weight and U_{t-1} is the so-called direction matrix found in the previous iteration. Usually $U_0 = 0$ so that in the first iteration the problem (5.1) becomes equivalent to the semidefinite relaxation (4.1) of the rank-constrained problem (4.2).

In this algorithm the optimal objective values of (5.2) should satisfy

$$tr(U_1^* X_1^*) \geq tr(U_2^* X_2^*) \geq tr(U_3^* X_3^*) \geq ..., \tag{5.3}$$

where the objective function of the semidefinite problem (5.2) represents the sum of $n-k$ smallest eigenvalues of X_t^* as introduced in [1]. This optimal value is reached for the matrix U_t^* that consequently enters the problem (5.1) in the $(t+1)$-th iteration. Therefore, the iteration guarantees that it holds

$$\sum_{j=k+1}^{n} \lambda_j(X_1^*) \geq \sum_{j=k+1}^{n} \lambda_j(X_2^*) \geq \sum_{j=k+1}^{n} \lambda_j(X_3^*) \geq ... \tag{5.4}$$

Our natural effort is to achieve a matrix X_t^* (for some t) with zero sum of its $n-k$ smallest eigenvalues, i.e.

$$tr(U_t^* X_t^*) = \sum_{j=k+1}^{n} \lambda_j(X_t^*) = 0. \tag{5.5}$$

This would also guarantee that the rank of matrix X_t^* is at most k since symmetric positive semidefinite matrices are diagonalizable. Theoretically, the equation (5.5) does not have to be reached since the inequalities in (5.4) are not strict.

When we realize that the problem (4.8) can be expressed in its standard form

$$\begin{aligned} \min_{Y\in\mathcal{S}^{3n}} \quad & tr(Y) \\ Y_{ii} &= 1, \quad i = 1, ..., n, \\ Y = \begin{bmatrix} X & 0 & 0 \\ 0 & I & (C-X)^T \\ 0 & C-X & Z \end{bmatrix} &\succeq 0, \\ rank(Y) &\leq n + k, \end{aligned} \tag{5.6}$$

we will consequently solve the semidefinite problems sequence (5.7) and (5.8):

$$
\begin{aligned}
Y_t^* = \ & \underset{Y \in \mathcal{S}^{3n}}{\operatorname{argmin}} \quad tr(Y) + \omega_t \, tr(U_{t-1}^* Y) \\
& \qquad Y_{ii} = 1, \quad i = 1, ..., n, \\
& Y = \begin{bmatrix} X & 0 & 0 \\ 0 & I & (C-X)^T \\ 0 & C-X & Z \end{bmatrix} \succeq 0
\end{aligned} \tag{5.7}
$$

and

$$
\begin{aligned}
U_t^* = & \underset{U \in \mathcal{S}^{3n}}{\operatorname{argmin}} \quad tr(Y^* U) \\
& \quad 0 \preceq U \preceq I, \\
& \quad tr(U) = 2n - k,
\end{aligned} \tag{5.8}
$$

where t is the number of iteration and $\omega_t \geq 0$ is the relative weight.

5.2. Rank reduction algorithm for semidefinite problems. In [10], there is introduced a rank reduction algorithm for the semidefinite problem (4.1). Given a semidefinite matrix X, the task is to find such matrix X^+ so that $rank(X^+) < rank(X)$, or equivalently $\mathcal{N}(X^+) \supset \mathcal{N}(X)$.

This rank reduction algorithm guarantees to find a solution whose rank satisfies the upper bound on rank

$$
rank(X) \leq \left\lfloor \frac{\sqrt{8m+1} - 1}{2} \right\rfloor, \tag{5.9}
$$

where m is the number of linear equality constraints in the semidefinite problem (4.1). Such a solution is guaranteed to exist as proved in [10].

Let us describe the algorithm. First of all, for any semidefinite matrix there exists a matrix $V \in \mathcal{R}^{n \times r}$, where $r = rank(X)$, such that

$$
X = VV^T. \tag{5.10}
$$

Since X is a symmetric positive semidefinite matrix, its spectral decomposition can be expressed in the form $X = Q\Lambda Q^T = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})^T$, where $QQ^T = I$ and $\Lambda = diag(\lambda_1, ..., \lambda_n)$. When we take the first r column of the matrix $Q\Lambda^{\frac{1}{2}}$, we obtain matrix V from decomposition (5.10).

If we search for matrix X^+ in the form

$$
X^+ = V(I + \alpha\Delta)V^T, \tag{5.11}
$$

where $\Delta \in \mathcal{S}^n$ is a direction matrix and $\alpha \in \mathcal{R}$ is a step size, we can see (5.11) as a shift of matrix X

$$
X^+ = X + \alpha V \Delta V^T. \tag{5.12}
$$

Our intention is to choose α and Δ such that X^+ is a solution of (4.1) and $rank(X^+) < rank(X)$. It is obvious that $\alpha \neq 0$ so that $X^+ \neq X$.

In order to maintain optimality, we require that $tr(DX^+) = tr(DX)$. After substituting X^+ with (5.12), we obtain condition

$$
tr(V^T D V \Delta) = 0. \tag{5.13}
$$

In order to maintain feasibility, X^+ needs to satisfy equality constraints $tr(A_i X) = b_i,\ i = 1, ..., m$, from where we have condition

$$tr(V^T A_i V \Delta) = 0,\ i = 1, ..., m \tag{5.14}$$

and X^+ has to stay positive semidefinite which is fulfilled if and only if

$$I + \alpha\Delta \succeq 0. \tag{5.15}$$

Since we want X^+ to be a shift of matrix X in the direction of matrices with a lower rank, the matrix $I + \alpha\Delta$ has to be also singular. To achieve this we take $\alpha = -\frac{1}{\lambda_1}$, where λ_1 is a maximum-magnitude eigenvalue of Δ.

Considering the rank-constrained semidefinite problem (4.8), we have to undergo this procedure for matrix Y. This rank reduction algorithm is summarized below.

Algorithm 1: Rank reduction for semidefinite programs

Input: a solution X of a semidefinite problem (4.1);
while $rank(X) > k$ **do**
  compute the factorization $X = Q\Lambda Q^T$ using Matlab function `svd(X)`;
  take the first r columns of matrix $Q\Lambda^{\frac{1}{2}}$ as columns of matrix V;
  solve a feasibility problem with constraints (5.13) and (5.14) to find a nonzero Δ;
  find eigenvalues of Δ using Matlab function `eig(Δ)`;
  denote the maximum-magnitude eigenvalue of Δ by λ_1;
  take $\alpha = -\frac{1}{\lambda_1}$;
  define $X = V(I + \alpha\Delta)V^T$;
end

6. Numerical results. Our experiments were performed in MATLAB R2019a [11] on an Intel Core i7-4690 CPU (3.6 GHz). To solve semidefinite programming problems, we used solver SDPT3 included in the CVX modelling system, a package for specifying and solving convex problems [6], [7].

In the first experiment, we have compared the proposed semidefinite formulation (4.7) of the problem (2.2) to the alternating projections algorithm and the preconditioned Newton method. Using Matlab function `gallery('randcorr',n)` we generated 100 random correlation matrices C of order $n = 10$, $n = 20$ and $n = 50$. The results of these experiments, displayed in Table 6.1, have shown that for bigger order n, the alternating projections algorithm requires too many iterations. Therefore, it is more effective to apply the preconditioned Newton method, as declared in [3].

Our semidefinite approach is worse than these two standard approaches concerning computation time which is caused by the necessity to initialize the CVX modelling system. Therefore, for the convex problem (2.2) (without the rank constraint), it is more effective to apply the preconditioned Newton method. However, the semidefinite program (4.7) found an optimal solution with the same (or even slightly lower) value of the objective than the other two approaches, which was the purpose of this comparison. We emphasize in each of 100 experiments this deviation was at the level of 10^{-7} on behalf of the semidefinite approach.

The most beneficial advantage of the proposed semidefinite formulation (4.7) of the problem (2.2) and using the SDPT3 solver (and the CVX modelling system) is that it allows the rank-constrained semidefinite reformulation (4.8) of the original rank-constrained problem (2.1). Recall that our primary goal is to find the nearest correlation matrix of the desired rank and the rank reduction algorithms included in Section 5 were designed for rank-constrained semidefinite programming problems of the form (4.2).

n	time (AP)	time (N)	time (SDP)	$\|X_{ap} - C\|_F$	$\|X_n - C\|_F$	$\|X_{sdp} - C\|_F$
10	0.005 s	0.005 s	0.107 s	2.585818121	2.585818303	2.585818104
20	0.008 s	0.005 s	0.172 s	6.634680585	6.634680745	6.634680550
50	x	0.005 s	0.107 s	x	20.3988005	20.398780

TABLE 6.1

Comparison between finding a solution to the problem (2.2) by the alternating projections algorithm (AP), the preconditioned Newton method (N) and semidefinite programming (SDP). There are average values of computation time and objective function counted from 100 generated experiments.

In the next experiment, we generated a random correlation matrix of order 100, and without loss of generality, we considered a three-factor model, which means we solved the problem (4.8) for $k = 3$. In order to determine the rank of the solution, we computed its eigendecomposition using Matlab function `eig(.)`. Considering the eigenvalues lower than 10^{-6} to be zero, the rank was established as the number of its non-zero eigenvalues. Let us note that considering another tolerance for zero eigenvalues, the results would look slightly different, but the comparison of the methods would not change.

In the first step, we solved the rank-constrained semidefinite problem (4.8) via the convex iteration (5.7) and (5.8). Since one of the iterating problems is bi-criteria, it was necessary to deal with the choice of relative weight ω_t. We set this weight to be an increasing sequence of the number of iteration t and compared the behaviour of the solution. Figure 6.1 and Table 6.2 show that for a faster increasing sequence, the convex iteration requires fewer iterations, and leads to a solution with the comparable value of the objective function.

Consequently, we applied the rank-reduction algorithm to solve the problem (4.8). As displayed in Fig. 6.1 and Table 6.2, the rank reduction algorithm led to a much better low-rank solution of the problem (4.8) than the convex iteration. The increment of the value of the objective function of (2.1) in the convex iteration is caused by the low weight put on the minimized Frobenius norm in the bi-criteria problem (5.7). From this point of view, the rank reduction algorithm should be preferred when solving a rank-constrained semidefinite problem because of its ability to control the value of its objective.

	Convex Iter. $\omega = 2^t$	Convex Iter. $\omega = 10^t$	Convex Iter. $\omega = t$	Rank Reduction
# iterations	8	3	72	8
$\|X - C\|_F$	55.90	56.09	55.80	11.18

TABLE 6.2

Comparison between the rank reduction algorithm and the convex iteration for $\omega_t = 2^t$, $\omega = 10^t$ and $\omega = t$ with t being the number of iteration, $n = 100$ and $k = 3$.

7. Conclusions. We have proposed the semidefinite reformulation (4.7) of the problem (2.2) and demonstrated that the semidefinite approach leads to compara-

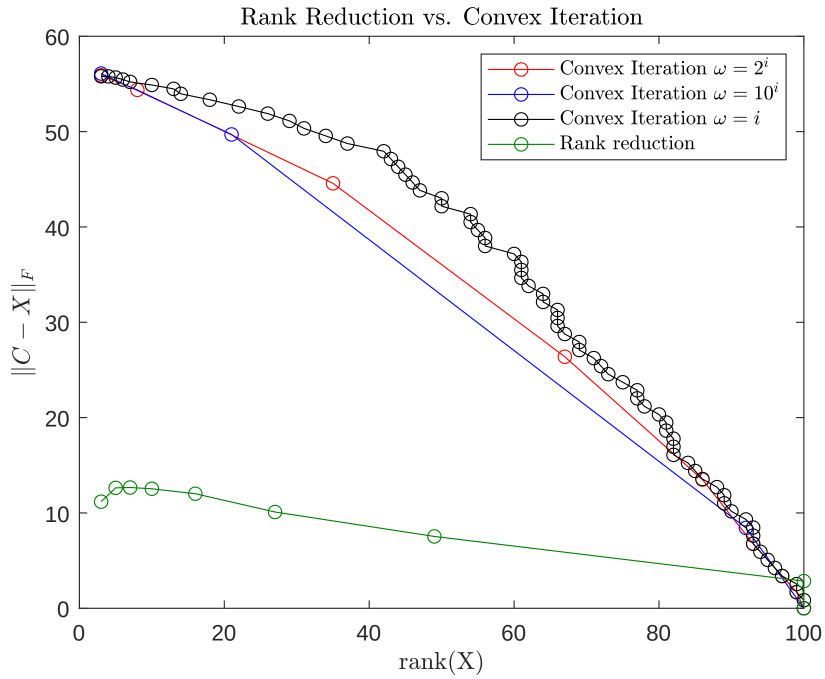

FIG. 6.1. *Trade-off between the rank and the objective function value of the solution found by the rank reduction algorithm (green) and the convex iteration for relative weights $\omega_t = 2^t$ (red), $\omega_t = 10^t$ (blue) and $\omega_t = t$ (black) with t being the number of iteration, where $n = 100$ and $k = 3$. The points represent solutions in particular iterations. The starting point $[100, 0]^T$ represents the given full-rank correlation matrix C.*

ble solutions with the existing approaches by running several experiments. This semidefinite reformulation enabled us to equivalently reformulate the original nonconvex problem (2.1) into the rank-constrained semidefinite problem (4.8), which can be solved by rank reduction algorithms. When applying the convex iteration (5.7) and (5.8), the Frobenius norm that measures the "nearness" of the low-rank correlation matrix to the given matrix, is relatively large because of the low ability of this algorithm to control the value of the objective sufficiently. Nevertheless, the rank reduction algorithm (Algorithm 1) provided a much better solution and can be recommended to solve rank-constrained semidefinite problems. It is an object for further research if there exists a correlation matrix of the given rank which would be "nearer" to the given correlation matrix than the one found in this work. It has shown that it is reasonable to deal with finding new methods for solving rank-constrained semidefinite problems.

Acknowledgements. I want to thank Mária Trnovská for helpful comments. This work was supported by Comenius University Grant for doctoral students No. UK/42/2020.

REFERENCES

[1] Alizadeh, F.: *Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization*, SIAM Journal on Optimization 5 (1995), 13–51

[2] Borsdorf, R., Higham, N. J., Raydan, M.: *Computing a nearest correlation matrix with factor structure*, International Edition III, MeBoo USA, 2001-2010
[3] Borsdorf, R., Higham, N.: *A preconditioned Newton Algorithm for the Nearest Correlation Matrix*, IMA J. Numer. Anal., 30 (2010), 94-107
[4] Boyd, S., Vandenberghe, L.: *Convex Optimization*, Cambridge University Press, New York, 2004
[5] Dattorro, J.: *Convex Optimization & Euclidean Distance Geometry*, International Edition III, MeBoo USA, 2001-2010
[6] Grant, M., Boyd, S.: *CVX: Matlab software for disciplined convex programming*, version 2.0 beta, http://cvxr.com/cvx, September 2013
[7] Grant, M., Boyd, S.: *Graph implementations for nonsmooth convex programs*, Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer, 2008, 95-110, http://stanford.edu/∼boyd/graph_dcp.html.
[8] Grubišić, I., Pietersz, R.: *Efficient rank reduction of correlation matrices*, Linear Algebra and its Applications, 422 (2007), 629-653
[9] Higham, N.: *Computing the nearest correlation matrix – a problem from finance*, IMA Journal of Numerical Analysis, 22 (2002), 329-343
[10] Lemon, A., So, A. M., Ye, Y.: *Low-Rank Semidefinite Programming: Theory and Applications*, Foundations and Trends in Optimization, 2 (2015), 1-156
[11] MATLAB R2019a, The Mathworks, Inc., MATLAB version 9.6.0.1150989, Massachusetts, 2019
[12] Pietersz, R., Groenen, P. J. F.: *Rank Reduction of Correlation Matrices by Majorization*, Quantitative Finance, 2004, 649-662
[13] Qi, H., Sun, D.: *A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), 360-385
[14] Zhang, F.: *Matrix theory - Basic Results and Techniques*, Springer Science+Business Media, New York, 1999
[15] Zhang, Z., Wu, L.: *Optimal low-rank approximation to a correlation matrix*, Linear Algebra and its Applications, 364 (2003), 161-187