On the job rotation problem

Peter Butkovič and Seth Lewis
School of Mathematics
The University of Birmingham
Edgbaston
Birmingham B15 2TT
United Kingdom

May 3, 2005

Abstract

The job rotation problem (JRP) is the following: Given an $n \times n$ matrix A over $\mathbb{R} \cup \{-\infty\}$ and $k \leq n$, find a $k \times k$ principal submatrix of A whose optimal assignment problem value is maximum. No polynomial algorithm is known for solving this problem if k is an input variable. We analyse JRP and present polynomial solution methods for a number of special cases.

Keywords: principal submatrix, assignment problem, job rotation problem, node disjoint cycles.

AMS-classification: 15A15, 90C27

1 Introduction

One of the classical problems in combinatorial optimization is the (*linear*) *assignment problem* which can be described as follows: A one-to-one assignment between two n-element sets of objects, say $\{A_1, ..., A_n\}$ and $\{B_1, ..., B_n\}$ has to be found. The cost c_{ij} of assigning A_i to B_j is given for every pair (A_i, B_j) and the task is to find an assignment that minimises the total cost. This problem has a convenient matrix formulation: If we store the coefficients c_{ij} in an $n \times n$ matrix C then the assignment problem means to choose n entries of C so that no two are from the same row or column, and their sum is minimal.

The assignment problem has, of course, also a maximising form in which the coefficients represent benefits and the object is to maximise the sum of the benefits. Many solution methods exist for the assignment problem [1], [6], probably the best known being the Hungarian method of computational complexity $O(n^3)$, whose many variants exist in the literature.

The *job rotation problem* is motivated by the following task: Suppose that a company with n employees requires these workers to swap their jobs (possibly on a regular basis) in order to avoid exposure to monotonous tasks (for instance manual workers at an assembly line or ride operators in a theme park). It is also required that to maintain stability of service only a certain number of employees, say k ($k < n$), actually swap their jobs. With each transition old job - new job a coefficient is associated expressing either the cost (for instance for an additional training) or the preference of the worker to this particular change. So the aim is to select k employees and to suggest a plan of the job changes between them so that the sum of the coefficients corresponding to these changes is minimum or maximum.

For any set X and positive integer n the symbol $X^{n\times n}$ will denote the set of all $n \times n$ matrices over X. In most cases we will deal with matrices over $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$. By a *principal submatrix* of a square matrix A we understand as usual any submatrix of A whose set of row indices is the same as the set of column indices. A principal submatrix of $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ is therefore any matrix of the form

$$\begin{pmatrix} a_{i_1i_1} & a_{i_1i_2} & \dots & a_{i_1i_k} \\ a_{i_2i_1} & a_{i_2i_2} & \dots & a_{i_2i_k} \\ \vdots & \vdots & & \vdots \\ a_{i_ki_1} & a_{i_ki_2} & \dots & a_{i_ki_k} \end{pmatrix}$$

where $1 \leq i_1 < ... < i_k \leq n$. This matrix will be denoted by $A(i_1, i_2, ..., i_k)$. Hence the job rotation problem is the problem to find, for a given $n \times n$ matrix A and $k < n$, a $k \times k$ principal submatrix of A for which the optimal assignment problem value is minimal or maximal (the diagonal entries can be set to $+\infty$ or $-\infty$ to avoid an assignment to the same job). For a particular A and k, we shall refer to this problem as JRP(A, k). The task of solving the job rotation problem for all k, we shall refer to as JRP(A) or just JRP. In the rest of the paper, we will discuss the maximisation version of the problem.

Note that there is also a "non-weighted" version of JRP in which it is only given which job moves are feasible. The problem is to decide if it is possible to re-assign / rotate k jobs between the employees, ($k \in N$), where job i can be assigned to job j only if (i, j) is from a given set of feasible transitions. This can obviously be represented by a $\{0, -\infty\}$ matrix C, where a 0 corresponds to a feasible move. Alternatively, this version can be represented by a (non-weighted) digraph $D = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{(v_i, v_j); c_{ij} = 0\}$.

The number of principal submatrices of order k of a matrix of order n is $\binom{n}{k}$. Therefore if k is an input variable, solving the assignment problem for all principal submatrices and then comparing the resulting values would be non-polynomial. If $k \leq n$ is fixed, then the method would be polynomial (though of a high degree in most cases). However, the total number of submatrices of all orders is $\sum_{k=1}^{n} \binom{n}{k} = 2^n - 1$ and therefore checking all of them would not solve JRP for all k in polynomial time. In fact, no polynomial method seems to be known for solving this problem, neither is it proved to be NP-complete. In this paper we present a number of cases when JRP is solvable polynomially. Note that there is a randomized polynomial algorithm for solving JRP [5]. It may be interesting to mention that the problem arising by removing the word "principal" from the formulation of the JRP is easily solvable [10].

In Section 2 we will give an overview of known results. Section 3 deals with matrices over $\mathbb{T} = \{-\infty, 0\}$ and Section 4 contains results for matrices over $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$. These include the proof that JRP(A) can be solved in polynomial time if this is true for every irreducible diagonal block of the Frobenius normal form of A.

2 An overview of known results

One class of solvable cases of the JRP is related to the fact that the optimal values of JRP(A, k) for $k = 1, 2, ..., n$ are the coefficients of the characteristic polynomial of A in max-algebra. Max-algebra is an analogue of linear algebra in which the conventional operations of addition and multiplication are replaced by $\oplus$ and $\otimes$ defined as follows: $a \oplus b = \max(a, b)$ and $a \otimes b = a + b$ for $a, b \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$. Terminology and notation in max-algebra are defined similarly to those in linear algebra: The iterated product $a \otimes a \otimes ... \otimes a$ in which the element a is used k-times will be denoted by $a^{(k)}$. If $A = (a_{ij})$, $B = (b_{ij})$ and $C = (c_{ij})$ are matrices with elements from $\overline{\mathbb{R}}$ of compatible sizes, we write $C = A \oplus B$ if $c_{ij} = a_{ij} \oplus b_{ij}$ for all i, j and $C = A \otimes B$ if $c_{ij} = \sum_{k}^{\oplus} a_{ik} \otimes b_{kj} = \max_{k}(a_{ik} + b_{kj})$ for all i, j. Similarly as for scalars the iterated product $A \otimes A \otimes ... \otimes A$ in which the square matrix A is used k-times will be denoted by $A^{(k)}$. In max-algebra the *unit matrix* I is a square matrix of an appropriate size whose diagonal elements are all 0 and whose off-diagonal elements are $-\infty$.

Note that the algebraic system $(\overline{\mathbb{R}}, \oplus, \otimes)$ offers an appropriate language to describe various operational research problems. An account on algebraic properties in max-algebra can be found in Cuninghame-Green [11] and [13]. Further surveys in this field are the monograph [2], the survey paper of Gondran and Minoux [15] and the monograph of Zimmermann [19] on optimisation in ordered algebraic structures.

Relevant to this paper is the max-algebraic characteristic polynomial or, briefly, *characteristic maxpolynomial* [12]. We now introduce this concept. Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$. The max-algebraic *permanent* of A is defined as an analogue of the classical one:

$$\text{maper}(A) = \sum_{\pi \in P_n}^{\oplus} \prod_{i \in N}^{\otimes} a_{i,\pi(i)}$$

where P_n stands for the set of all permutations of the set $N = \{1, ..., n\}$. In the conventional notation

$$\text{maper}(A) = \max_{\pi \in P_n} \sum_{i \in N} a_{i,\pi(i)}$$

which is obviously the optimal value of the assignment problem for the matrix A. The set of all optimal permutations will be denoted by $ap(A)$, that is,

$$ap(A) = \{\pi \in P_n; \text{maper}(A) = \sum_{i \in N} a_{i,\pi(i)}\}.$$

It will be useful to denote $\sum_{i \in N} a_{i,\pi(i)}$ by $w(A, \pi)$. Hence $\text{maper}(A) = \max_{\pi \in P_n} w(A, \pi)$.

The *characteristic maxpolynomial* of A is defined [12] as

$$\chi_A(x) = \text{maper}(A \oplus x \otimes I) = \text{maper} \begin{pmatrix} a_{11} \oplus x & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} \oplus x & ... & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & ... & a_{nn} \oplus x \end{pmatrix}.$$

It follows immediately from this definition that $\chi_A(x)$ is of the form

$$\delta_0 \oplus (\delta_1 \otimes x) \oplus ... \oplus (\delta_{n-1} \otimes x^{(n-1)}) \oplus x^{(n)}$$

or briefly $\sum_{i=0}^{n}{}^{\oplus} \delta_i \otimes x^{(i)}$ where $\delta_n = 0$ and, by convention, $x^{(0)} = 0$. It is also easily seen that for $k = 0, 1, ..., n-1$

$$\delta_{n-k} = \sum_{B \in A_k}{}^{\oplus} \operatorname{maper}(B),$$

where A_k is the set of all principal submatrices of A of order k. Hence we can readily compute $\delta_0 = \operatorname{maper}(A)$ and $\delta_{n-1} = \max(a_{11}, a_{22}, ..., a_{nn})$. Note that $\delta_{n-k} = -\infty$ if $\operatorname{maper}(B) = -\infty$ for all $B \in A_k$ in which case the term $\delta_{n-k} \otimes x^{(n-k)}$ may be omitted. Also, $\chi_A(x)$ may reduce to just $x^{(n)}$, see Theorem 1 below.

Because of the absorbing effect of the operation $\oplus$, some terms of a characteristic maxpolynomial may be omitted without changing it as a function. More precisely, the characteristic maxpolynomial of A written using usual algebra is

$$\chi_A(x) = \max(\delta_0, \delta_1 + x, \delta_2 + 2x, ..., \delta_{n-1} + (n-1)x, nx).$$

Hence, $\chi_A(x)$ is the upper envelope of $n+1$ affine-linear functions and thus a piecewise linear and convex function. If for some $k \in \{0, ..., n\}$ the inequality

$$\delta_k \otimes x^{(k)} \leq \sum_{i \neq k}{}^{\oplus} \delta_i \otimes x^{(i)}$$

holds for every real x then the term $\delta_k \otimes x^{(k)}$ is called *inessential*, otherwise it is called *essential*. Hence

$$\chi_A(x) = \sum_{i \neq k}{}^{\oplus} \delta_i \otimes x^{(i)}$$

holds for all $x \in \mathbb{R}$ if $\delta_k \otimes x^{(k)}$ is inessential, and therefore all inessential terms may be ignored if $\chi_A(x)$ is considered as a function.

An $O(n^2(m + n \log n))$ algorithm for finding all essential terms of the characteristic maxpolynomial of an $n \times n$ matrix where m is the number of finite entries of A is known [4]. It then follows that this method solves the JRP(A) in polynomial time when all terms are essential, as $\delta_{n-k}(A)$ is the optimal solution value of JRP(A, k). Note that the complexity bound has recently been improved to $O(n(m + n \log n))$ steps [14].

In the rest of the paper N will stand for the set $\{1, \ldots, n\}$ and if $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ we denote by $D(A)$ the digraph with the node set N and arc set $\{(i,j); a_{ij} > -\infty\}$. For a cycle $\sigma = (i_1, i_2, \ldots, i_p, i_1)$, let $V(\sigma) = \{i_1, i_2, \ldots, i_p\}$. For a digraph D, we say cycles $\sigma_1, \ldots, \sigma_t$ in D are pairwise node disjoint (PND) if and only if $V(\sigma_i) \cap V(\sigma_j) = \emptyset$ for $i, j = 1, \ldots, t$, $i \neq j$.

As another extreme it may happen that $\chi_A(x) = x^{(n)}$. This case can also be easily characterized. Since every permutation is either a cycle or a product of cycles, $\delta_{n-k}(A) \neq -\infty$ if and only if there exist PND cycles in $D(A)$ covering a total of k nodes. The statement of the theorem below then immediately follows.

Theorem 1. Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$. Then $\chi_A(x) = x^{(n)}$ if and only if $D(A)$ is acyclic.

Example 2. If

$$A = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 4 & 1 \\ 2 & 5 & 0 \end{pmatrix}$$

then

$$\chi_A(x) = \text{maper} \begin{pmatrix} 1 \oplus x & 3 & 2 \\ 0 & 4 \oplus x & 1 \\ 2 & 5 & 0 \oplus x \end{pmatrix} =$$

$$= \big((1 \oplus x) \otimes (4 \oplus x) \otimes (0 \oplus x)\big) \oplus \big(3 \otimes 1 \otimes 2\big) \oplus \big(2 \otimes 0 \otimes 5\big) \oplus \big(2 \otimes (4 \oplus x) \otimes 2\big) \oplus$$

$$\oplus \big((1 \oplus x) \otimes 1 \otimes 5\big) \oplus \big(3 \otimes 0 \otimes (0 \oplus x)\big) =$$

$$= x^{(3)} \oplus 4 \otimes x^{(2)} \oplus 6 \otimes x \oplus 8.$$

In the conventional notation,

$$\chi_A(x) = \max(3x, 4 + 2x, 6 + x, 8).$$

It is easily seen that this characteristic maxpolynomial has exactly one inessential term, namely $6 \otimes x$.

For $A \in \overline{\mathbb{R}}^{n\times n}$, we define $F = \{k \in N; \delta_{n-k}(A) \neq -\infty\}$ and k_{max} as $\max(F)$. The task of finding k_{max} for a general matrix can be solved in $O(n^3)$ time [8], however we can do better for symmetric matrices:

Theorem 3. [7] The task of finding k_{max} for a symmetric matrix $A \in \overline{\mathbb{R}}^{n\times n}$ is equivalent to the maximum cardinality matching problem in a bipartite graph with $2n$ nodes and can therefore be solved in $O(n^{2.5}/\sqrt{\log n})$ time.

Proof. Let $B(A)$ be the bipartite graph with the bipartition (U, V), where $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$, and set of arcs $\{u_i v_j; a_{ij} > -\infty\}$.

Let M be a matching of maximum cardinality in $B(A)$, $|M| = m$. Obviously $k_{max} \leq m$ because if $k = k_{max}$ then there are k independent finite entries in A, say $a_{i_r \pi(i_r)}$, $r = 1, \ldots, k$, and so there is a matching of cardinality k in $B(A)$, namely, $\{u_{i_r} v_{\pi(i_r)}; r = 1, \ldots, k\}$.

We now prove $k_{max} \geq m$. The set of arcs $H = \{(i, j); u_i v_j \in M\}$ in $D(A)$ consists of directed PND elementary paths (possibly cycles), since the outdegree and indegree of each node in (N, H) is at most one.

By symmetry, for each arc (i, j) in $D(A)$, (j, i) is also an arc ("counterarc"). Construct from H another set H' as follows: Choose one of the paths, say p. First, starting from a penultimate arc on p (or any arc in the case of a cycle), remove this arc and every other arc. Second, add a counterarc to every remaining arc from p. Repeat this process with every path in (N, H). At the end we obtain a subgraph (N, H') which consists of PND cycles of length two only.

Note that each set of cycles in $D(A)$ determines a matching in $B(A)$ whose cardinality is equal to the total number of arcs of these cycles. Thus none of the paths in (N, H) could have been of odd length, say s, as otherwise the total number of arcs on cycles constructed from this path would be $s + 1$, a contradiction with the maximality of M. Hence $|H'| = m$ and $k_{max} \geq m$. The complexity statement now follows, see [3]. □

3 JRP for special symmetric matrices over $\{0, -\infty\}$

In this section we show that JRP(A, k), for a symmetric matrix A over $\{0, -\infty\}$, and k even, can be solved in O(1) time, after finding k_{max}. We also describe some cases when this is true for odd values of k.

Let $\mathbb{T} = \{0, -\infty\}$ and $A \in \mathbb{T}^{n \times n}$. Then for all k, the unique finite value for $\delta_{n-k}(A)$ is 0. Also, $\delta_{n-k}(A) = 0$ if and only if there exist PND cycles in $D(A)$ covering a total of k nodes. Hence, deciding if $\delta_{n-k}(A) = 0$ for some matrix $A \in \mathbb{T}^{n \times n}$ is equivalent to deciding whether there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ such that $\left| \bigcup_{i=1}^{t} V(\sigma_i) \right| = k$.

Theorem 4. If $A \in \mathbb{T}^{n \times n}$ is a symmetric matrix, and $\delta_{n-k}(A) = 0$ for some odd $k \in N$, then $\delta_{n-k+1}(A) = 0$.

Proof. Let k be odd and $\delta_{n-k}(A) = 0$. Then there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ such that $\left|\bigcup_{i=1}^{t} V(\sigma_i)\right| = k$.

As k is odd, there exists a cycle $\sigma_r = (i_1, i_2, \ldots, i_p, i_1) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ for odd p. By symmetry, $(i_1, i_2, i_1), (i_3, i_4, i_3), \ldots, (i_{p-2}, i_{p-1}, i_{p-2})$, together with all $\sigma_1, \ldots, \sigma_t$ except σ_r, are PND cycles which cover $k-1$ nodes. Hence $\delta_{n-k+1}(A) = 0$. □

Theorem 5. If $A \in \mathbb{T}^{n\times n}$ is a symmetric matrix, then $\delta_{n-k}(A) = 0$ for all even $k \leq k_{max}$.

Proof. By induction on k: If k_{max} is even, then $\delta_{n-k_{max}}(A) = 0$. Else if k_{max} is odd, then $k_{max} - 1$ is even, and $\delta_{n-k_{max}+1}(A) = 0$ by Theorem 4. We can now assume that $\delta_{n-k}(A) = 0$ for an even $k \leq k_{max}$, and prove that $\delta_{n-k+2}(A) = 0$. Thus there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ such that $\left|\bigcup_{i=1}^{t} V(\sigma_i)\right| = k$.

If there exists a cycle $\sigma_r = (i_1, i_2, \ldots, i_p, i_1) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ for even p, then by symmetry $(i_1, i_2, i_1), (i_3, i_4, i_3), \ldots, (i_{p-3}, i_{p-2}, i_{p-3})$, together with all $\sigma_1, \ldots, \sigma_t$ except σ_r, are PND cycles which cover $k-2$ nodes.

Else, there exists a cycle $\sigma_r = (i_1, i_2, \ldots, i_p, i_1) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ for odd p. As k is even, there exists another cycle $\sigma_s = (j_1, j_2, \ldots, j_q, j_1) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\} - \{\sigma_r\}$ for odd q. Then by symmetry, $(i_1, i_2, i_1), (i_3, i_4, i_3), \ldots, (i_{p-2}, i_{p-1}, i_{p-2})$, $(j_1, j_2, j_1), (j_3, j_4, j_3), \ldots, (j_{q-2}, j_{q-1}, j_{q-2})$, together with all $\sigma_1, \ldots, \sigma_t$ except σ_r and σ_s, are PND cycles which cover $k-2$ nodes.

Therefore by induction $\delta_{n-k}(A) = 0$ for all even $k \leq k_{max}$. □

Remark. In the above proof, the set of covered nodes for δ_{n-k} (for even $k < k_{max}$) is a subset of the set of covered nodes for $\delta_{n-k_{max}}$.

If A has at least one zero on the main diagonal, (or equivalently, if the digraph $D(A)$ has at least one loop), then we can derive a number of properties:

Theorem 6. If $A \in \mathbb{T}^{n\times n}$ is a symmetric matrix, and there exists a $B \in A(l)$ containing l independent zeros, with at least one of these l zeros lying on the main diagonal, then $\delta_{n-k}(A) = 0$ for all $k \leq l$.

Proof. There exist PND cycles (one of which is a loop, say (j, j)) in $D(A)$, that cover l nodes, hence $\delta_{n-l}(A) = 0$. These cycles without the loop are PND cycles covering $l-1$ nodes, hence $\delta_{n-l+1}(A) = 0$. Also, $\delta_{n-1}(A) = 0$ follows directly from the existence of the loop (j, j).

If $l-1$ is even, then by Theorem 5, $\delta_{n-k}(A) = 0$ for $k = 2, 4, 6, \ldots, l-3$. For all $k \in \{2, 4, 6, \ldots, l-3\}$, there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ covering

k nodes, none of which is node j (because of the remark following Theorem 5). Therefore, the cycle (j, j) together with $\sigma_1, \sigma_2, \ldots, \sigma_t$ are PND cycles covering $k+1$ nodes, hence $\delta_{n-k-1} = 0$ for $k = 2, 4, 6, \ldots, l-3$.

Else if $l-1$ is odd, then by Theorem 4, $\delta_{n-l+2}(A) = 0$. As $l-2$ is even, we can use Theorem 5 again to give us $\delta_{n-k}(A) = 0$ for $k = 1, 3, 5, \ldots, l-4$. As before, there exist PND cycles $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ in $D(A)$ covering k nodes (none being node j). The cycle (j, j) together with $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ are PND cycles covering $k+1$ nodes, hence $\delta_{n-k-1}(A) = 0$ for $k = 1, 3, 5, \ldots, l-4$. □

If $l \in \{k_{max}, k_{max} - 1\}$ in Theorem 6, then we can completely solve the (non-weighted) JRP for this type of matrix:

Theorem 7. If $A \in \mathbb{T}^{n \times n}$ is a symmetric matrix, $l \in \{k_{max}, k_{max} - 1\}$ and there exists a $B \in A(l)$ containing l independent zeros, with at least one of these l zeros lying on the main diagonal, then $\delta_{n-k}(A) = 0$ for all $k \leq k_{max}$.

Proof. The statement immediately follows from Theorem 6 and the fact that $\delta_{n-k_{max}}(A) = 0$. □

Theorem 8. If $A = (a_{ij}) \in \mathbb{T}^{n \times n}$ is a symmetric matrix and $(\exists j)\ a_{jj} = 0$, then for $l \in \{k_{max}, k_{max} - 1\}$, there exists a $B \in A(l)$ containing l independent zeros, with at least one of these l zeros lying on the main diagonal.

Proof. We assume that (j, j) is a loop in $D(A)$. As $\delta_{n-k_{max}}(A) = 0$, there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ such that $\left|\bigcup_{i=1}^{t} V(\sigma_i)\right| = k_{max}$. We need to show there exist PND cycles $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ in $D(A)$, at least one being a loop, such that $\left|\bigcup_{i=1}^{t'} V(\sigma'_i)\right| \in \{k_{max}, k_{max} - 1\}$.

Clearly, if $(j, j) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$, then we are done. So assume not. Then $j \in \bigcup_{i=1}^{t} V(\sigma_i)$, as otherwise, (j, j) together with $\sigma_1, \sigma_2, \ldots, \sigma_t$ would form PND cycles in $D(A)$ covering $k_{max}+1$ nodes, which contradicts the definition of k_{max}. Hence there exists one cycle $\sigma_r = (j, i_2, i_3, \ldots, i_p, j) \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$.

If p is odd, then by symmetry, $(j, j), (i_2, i_3, i_2), (i_4, i_5, i_4), \ldots, (i_{p-1}, i_p, i_{p-1})$ together with all $\sigma_1, \sigma_2, \ldots, \sigma_t$ except σ_r form PND cycles covering k_{max} nodes. If instead p is even, then again by symmetry, $(j, j), (i_2, i_3, i_2), (i_4, i_5, i_4), \ldots,$ $(i_{p-2}, i_{p-1}, i_{p-2})$ together with all $\sigma_1, \sigma_2, \ldots, \sigma_t$ except σ_r form PND cycles covering $k_{max} - 1$ nodes. □

Theorem 9. If $A = (a_{ij}) \in \mathbb{T}^{n \times n}$ is a symmetric matrix and $(\exists j)\ a_{jj} = 0$, then $\delta_{n-k}(A) = 0$ for all $k \leq k_{max}$.

Proof. The statement immediately follows from Theorem 7 and Theorem 8. □

By Theorem 5, for symmetric matrices, unless $k_{max} = 1$, the smallest even $k \in F$ is 2. However, the smallest odd value in F is more tricky. We denote this value by k_{oddmin}.

Remark. If there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ such that $\left|\bigcup_{i=1}^{t} V(\sigma_i)\right|$ is odd, then at least one of the cycles is odd. Hence k_{oddmin} is the length of a shortest odd cycle. This cycle can be found polynomially [18]. Note that k_{oddmin} does not exist if there is no odd cycle in $D(A)$, and if this is the case, then $\delta_{n-k} = -\infty$ for all odd k. For the remainder of this section, we shall assume that k_{oddmin} exists.

Theorem 10. Let

1. $A \in \mathbb{T}^{n\times n}$ be a symmetric matrix,
2. $\sigma_1, \sigma_2, \ldots, \sigma_t$ be PND cycles in $D(A)$ covering k' nodes,
3. $\exists \sigma' \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ with $|V(\sigma')|$ odd.

Then $\delta_{n-k}(A) = 0$ for all odd $k \in \{l', \ldots, k'\}$, where $l' = \min\limits_{\substack{i=1,\ldots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)|$.

Proof. Without loss of generality, assume $|V(\sigma_t)| = l'$. Then $\sigma_1, \sigma_2, \ldots, \sigma_{t-1}$ are PND cycles covering $k' - l'$ nodes, hence $\delta_{n-k'+l'} = 0$. By Theorem 5, $\delta_{n-k} = 0$ for all even $k \in \{0, \ldots, k' - l'\}$. Take an arbitrary even $k \in \{0, \ldots, k' - l'\}$. So $k + l' \in \{l', \ldots, k'\}$. There exist PND cycles $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ in $D(A)$ covering k nodes other than those from $V(\sigma_t)$. Therefore, $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ and σ_t are PND cycles covering $k + l'$ nodes. Hence result. □

Corollary 11. Let

1. $A \in \mathbb{T}^{n\times n}$ be a symmetric matrix,
2. $k' \in \{k_{max}, k_{max} - 1\}$,
3. $\sigma_1, \sigma_2, \ldots, \sigma_t$ be PND cycles in $D(A)$ covering k' nodes,
4. $\exists \sigma' \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ with $|V(\sigma')|$ odd, and
5. $\min\limits_{\substack{i=1,\ldots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)| = k_{oddmin}$.

Then we can decide whether $\delta_{n-k}(A)$ is 0 or $-\infty$ for all k in linear time, after finding k_{max} and k_{oddmin}.

Proof. By Theorem 5, $\delta_{n-k} = 0$ for all even $k \leq k_{max}$. By definition, $\delta_{n-k} = -\infty$ for all odd $k < k_{oddmin}$. By Theorem 10, $\delta_{n-k} = 0$ for all odd $k \in \{k_{oddmin}, \ldots, k_{max} - 1\}$. By definition, $\delta_{n-k_{max}} = 0$. Hence result. □

Theorem 12. If $A \in \mathbb{T}^{n\times n}$ is a symmetric matrix, then $\delta_{n-k}(A) = 0$ for all odd $k \in \{k_{oddmin}, \ldots, k_{max} - k_{oddmin}\}$.

Proof. As $\delta_{n-k_{max}} = 0$, there exist PND cycles $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $D(A)$ such that $\left|\bigcup_{i=1}^{t} V(\sigma_i)\right| = k_{max}$. There exists another cycle σ in $D(A)$ such that $|V(\sigma)| = k_{oddmin}$.

Delete all nodes in $V(\sigma)$ from $\sigma_1, \sigma_2, \ldots, \sigma_t$, as well as incident arcs. As the cycles were PND and each node was incident to precisely two arcs, up to $2k_{oddmin}$ arcs have been deleted. Therefore this leaves a total of at least $k_{max}-2k_{oddmin}$ arcs within the remaining cycles and paths that have arisen from deleting the arcs from the cycles. Paths have the form $(i_1, i_2, \ldots, i_s)$. Replace all paths of this form by cycles $(i_1, i_2, i_1), (i_3, i_4, i_3), \ldots, (i_{s-2}, i_{s-1}, i_{s-2})$ if s is even, and $(i_1, i_2, i_1), (i_3, i_4, i_3), \ldots, (i_{s-1}, i_s, i_{s-1})$ if s is odd. This gives PND cycles covering at least $k_{max} - 2k_{oddmin}$ arcs, and therefore at least $k_{max} - 2k_{oddmin}$ nodes, none of which are nodes from $V(\sigma)$.

Therefore, by Theorem 5, for all even $k \leq k_{max} - 2k_{oddmin}$, there exist PND cycles $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ covering k nodes, but none from $V(\sigma)$. So for all even $k \leq k_{max} - 2k_{oddmin}$, we have PND cycles $\sigma'_1, \sigma'_2, \ldots, \sigma'_{t'}$ and σ which cover $k + k_{oddmin}$ nodes. Hence result. □

Remark. Note that $\{k_{oddmin}, \ldots, k_{max} - k_{oddmin}\} \neq \emptyset \iff k_{oddmin} \leq \dfrac{k_{max}}{2}$.

Corollary 13. Let

1. $A \in \mathbb{T}^{n\times n}$ be a symmetric matrix,
2. $k' \in \{k_{max}, k_{max} - 1\}$,
3. $\sigma_1, \sigma_2, \ldots, \sigma_t$ be PND cycles in $D(A)$ covering k' nodes,
4. $\exists \sigma' \in \{\sigma_1, \sigma_2, \ldots, \sigma_t\}$ with $|V(\sigma')|$ odd, and
5. $\min\limits_{\substack{i=1,\ldots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)| \leq k_{max} - k_{oddmin}$.

Then we can decide whether $\delta_{n-k}(A)$ is 0 or $-\infty$ for all k in linear time, after finding k_{max} and k_{oddmin}.

Proof. By Theorem 5, $\delta_{n-k} = 0$ for all even $k \leq k_{max}$. By definition, $\delta_{n-k} = -\infty$ for all odd $k < k_{oddmin}$. By Theorem 12, $\delta_{n-k} = 0$ for all odd $k \in \{k_{oddmin}, \dots, k_{max} - k_{oddmin}\}$. By Theorem 10, $\delta_{n-k} = 0$ for all odd $k \in \{\min_{\substack{i=1,\dots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)|, \dots, k_{max} - 1\}$. By definition, $\delta_{n-k_{max}} = 0$. Because we have $\min_{\substack{i=1,\dots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)| \leq k_{max} - k_{oddmin}$, we know δ_{n-k} for all k. □

Corollary 14. Let

1. $A \in \mathbb{T}^{n \times n}$ be a symmetric matrix,
2. $k' \in \{k_{max}, k_{max} - 1\}$,
3. $\sigma_1, \sigma_2, \dots, \sigma_t$ be PND cycles in $D(A)$ covering k' nodes,
4. $\exists \sigma' \in \{\sigma_1, \sigma_2, \dots, \sigma_t\}$ with odd $|V(\sigma')| \leq \dfrac{k_{max}}{2}$.

Then we can decide whether $\delta_{n-k}(A)$ is 0 or $-\infty$ for all k in linear time, after finding k_{max} and k_{oddmin}.

Proof. We have

$$\begin{aligned} \min_{\substack{i=1,\dots,t \\ |V(\sigma_i)| \text{ odd}}} |V(\sigma_i)| &\leq |V(\sigma')| \\ &\leq k_{max} - |V(\sigma')| \\ &\leq k_{max} - k_{oddmin}. \end{aligned}$$

The statement now follows from Corollary 13. □

Corollary 15. Let

1. $A \in \mathbb{T}^{n \times n}$ be a symmetric matrix,
2. $k' \in \{k_{max}, k_{max} - 1\}$,
3. $\sigma_1, \sigma_2, \dots, \sigma_t$ be PND cycles in $D(A)$ covering k' nodes,
4. $\exists \sigma', \sigma'' \in \{\sigma_1, \sigma_2, \dots, \sigma_t\}$, $\sigma' \neq \sigma''$, with $|V(\sigma')|$ and $|V(\sigma'')|$ odd.

Then we can decide whether $\delta_{n-k}(A)$ is 0 or $-\infty$ for all k in linear time, after finding k_{max} and k_{oddmin}.

Proof. Without loss of generality, assume that $|V(\sigma')| \leq |V(\sigma'')|$. Then we have

$$\begin{aligned} 2|V(\sigma')| &\leq |V(\sigma')| + |V(\sigma'')| \\ &\leq k' \\ &\leq k_{max}. \end{aligned}$$

Therefore $|V(\sigma')| \leq \dfrac{k_{max}}{2}$, and the statement now follows from Corollary 14. □

Remark. Note that solving an assignment problem for $A = (a_{ij}) \in \mathbb{T}^{n\times n}$ is equivalent to deciding whether the classical permanent of the matrix $B = (b_{ij})$ is positive where B is defined by $b_{ij} = 1$ if $a_{ij} = 0$ and $b_{ij} = 0$ otherwise. Therefore the statements in Section 3 solve in special cases the question: Given $A \in \{0, 1\}^{n\times n}$, and $k \leq n$, is there a $k \times k$ principal submatrix of A whose classical permanent is positive?

4 JRP for special matrices over $\overline{\mathbb{R}}$

Recall that a matrix $A \in \overline{\mathbb{R}}^{n\times n}$ is called *irreducible* if $D(A)$ is strongly connected or $n = 1$. If A, B are square matrices and A can be obtained from B by simultaneous permutations of the rows and columns then we say that A and B are *equivalent*, notation $A \sim B$. Clearly, $\sim$ is an equivalence relation. If $A \sim B$ then $\chi_A(x) = \chi_B(x)$. It is known [17] that every matrix A can be transformed in linear time to an equivalent matrix B in the Frobenius normal form, that is

$$B = \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1p} \\ & B_{22} & \dots & B_{2p} \\ & & \ddots & \vdots \\ -\infty & & & B_{pp} \end{pmatrix},$$

in which all diagonal blocks are irreducible.

In this section we study JRP for matrices over $\overline{\mathbb{R}}$. First we present some solvable special cases and then we show that JRP(A) for $A \in \overline{\mathbb{R}}^{n\times n}$ can be solved in polynomial time if this is true for every diagonal block of the Frobenius normal form of A.

4.1 Pyramidal matrices

If $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ and $k \in N$ then the principal submatrix $A(1, ..., k)$ is called a *main principal submatrix* of A, notation $A[k]$. If for all $i, j, r, s \in N$

$$\max(i, j) < \max(r, s) \Longrightarrow a_{ij} \geq a_{rs}, \tag{1}$$

then A is called *pyramidal*.

Theorem 16. If $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ is pyramidal then $\delta_{n-k}(A) = \text{maper}(A[k])$.

Proof. Let $A(l_1, \dots l_k)$ be an arbitrary principal submatrix, where $1 \leq l_1 < \dots < l_k \leq n$. Note that

$$i \leq l_i, \text{ for all } i \leq k.$$

Therefore

$$\max(i, j) \leq \max(l_i, l_j), \text{ for } i, j \leq k.$$

If equality does not hold for some i and j, then by (1) we have,

$$a_{ij} \geq a_{l_i,l_j}.$$

If equality does hold for some i and j, then let $l_t = \max(l_i, l_j)$. Note that $i < j \Leftrightarrow l_i < l_j$. So we have $t = \max(i, j)$ and therefore $l_t = t$. Hence $l_{t-1} = t - 1, \ \dots, \ l_1 = 1$. In this case

$$a_{ij} = a_{l_i,l_j}.$$

Either way, $a_{ij} \geq a_{l_i,l_j}$ holds. Therefore

$$\begin{aligned} \text{maper}(A(l_1, \dots l_k)) &\leq \text{maper}(A(1, \dots k)) \\ &= \text{maper}(A[k]) \\ &= \delta_{n-k}(A), \end{aligned}$$

as $A(l_1, \dots l_k)$ was arbitrary. Hence result. ∎

Example 17. Consider the matrix

$$A = \begin{pmatrix} 9 & 8 & 4 & 3 \\ 8 & 6 & 5 & 4 \\ 5 & 4 & 4 & 3 \\ 3 & 2 & 3 & 1 \end{pmatrix}.$$

The indicated lines help to check that A is pyramidal. Hence by Theorem 16 we find:

$$\begin{aligned} \delta_3(A) &= \text{maper}(A[1]) = 9 \\ \delta_2(A) &= \text{maper}(A[2]) = 16 \\ \delta_1(A) &= \text{maper}(A[3]) = 20 \\ \delta_0(A) &= \text{maper}(A[4]) = \text{maper}(A) = 22. \end{aligned}$$

Remark. Matrices that are not pyramidal, may become such after simultaneously permuting rows and columns. It follows from (1) that the diagonal entries of the matrix must be in descending order for (1) to be satisfied. Once rows and columns have been simultaneously permuted in this way, additional simultaneous row and column permutations may be needed between rows and columns which have a diagonal entry equal to another diagonal entry.

4.2 Monge and Hankel matrices

A matrix A will be called *diagonally dominant* if $id \in ap(A)$. (Note that throughout the paper id stands for the identity permutation.) A matrix $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ is called *Monge* if $a_{ij} + a_{rs} \geq a_{is} + a_{rj}$ for all $i, j, r, s \in N$, $i \leq r$, $j \leq s$. It is well known [6] that every Monge matrix A is diagonally dominant. It is also easily seen that a principal submatrix of a Monge matrix is also Monge. Hence JRP(A, k) for Monge matrices is readily solved by finding the k biggest entries of A.

For a given sequence $\{g_r \in \overline{\mathbb{R}}; r = 1, \ldots, 2n-1\}$, the *Hankel* matrix is the matrix $H = (h_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ where $h_{ij} = g_{i+j-1}$. Hankel matrices generated by convex sequences are Monge [9]. Therefore, for these matrices, JRP is readily solved. However, no efficient method seems to exist for Hankel matrices in general.

In this subsection we show that finiteness of $\delta_{n-k}(H)$ can be easily decided for any Hankel matrix H. Since Hankel matrices are symmetric, we can use some of the results of Section 3.

Theorem 18. If $\{g_r \in \overline{\mathbb{R}}; r = 1, \ldots, 2n-1\}$ is the sequence generating Hankel matrix $H = (h_{ij}) \in \overline{\mathbb{R}}^{n \times n}$ and $g_r \neq -\infty$ for some odd r, then $\delta_{n-k}(H) \neq -\infty$ for all $k \leq k_{max}$.

Proof. Let $C = (c_{ij})$ be defined by $c_{ij} = 0$ if $h_{ij} \neq -\infty$ and $c_{ij} = -\infty$ otherwise. Assume $g_r \neq -\infty$ for some odd r. So $(\exists i)$ $c_{ii} \neq -\infty$, i.e. $(\exists i)$ $c_{ii} = 0$. We now use Theorem 9 to give us $\delta_{n-k}(C) = 0$ for all $k \leq k_{max}$. Then as $\delta_{n-k}(C) = 0$ if and only if $\delta_{n-k}(H) \neq -\infty$, the theorem follows. □

Theorem 19. If a matrix $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$ is any matrix such that $a_{ij} = -\infty$ if $i + j$ is even, then $\delta_{n-k}(A) = -\infty$ for all odd k.

Proof. Assume $A = (a_{ij})$ is a matrix such that $a_{ij} = -\infty$ if $i + j$ is even. If $a_{i,j}$ is finite then $i + j$ is odd. So i and j must be of different parities.

Let $\sigma = (i_1, i_2, \ldots, i_p) \in C_n$ be any cyclic permutation of arbitrary length p such that $w(A, \sigma) \neq -\infty$.

As $w(A, \sigma) \neq -\infty$, then $a_{i_j, i_{j+1}} \neq -\infty$. So i_j and i_{j+1} must be of different parities. This means elements in the sequence $i_1, i_2, \ldots, i_p, i_1$ alternate between even and odd. This means p must be an even number, i.e. there are no cyclic permutations σ of odd length of finite weight. Hence result. □

If A is symmetric, then together with Theorem 5, this gives us:

Theorem 20. If $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$ is a symmetric matrix such that $a_{ij} = -\infty$ if $i + j$ is even, then $\delta_{n-k}(A) \neq -\infty$ for all even $k \leq k_{max}$, and $\delta_{n-k}(A) = -\infty$ for all odd k.

A certain type of Hankel matrix satisfies Theorem 20. Rewriting it for this type of matrix gives:

Theorem 21. If $\{g_r \in \overline{\mathbb{R}}; r = 1, \ldots, 2n-1\}$ is the sequence generating Hankel matrix H and $g_r = -\infty$ for all odd r, then $\delta_{n-k}(H) \neq -\infty$ for all even $k \leq k_{max}$ and $\delta_{n-k}(H) = -\infty$ for all odd k.

Combining Theorem 18 and Theorem 21 enables us to decide whether $\delta_{n-k}(H)$ is finite or not for any Hankel matrix H.

Theorem 22. If $\{g_r \in \overline{\mathbb{R}}; r = 1, \ldots, 2n-1\}$ is the sequence generating Hankel matrix H then

1. $\delta_{n-k}(H) \neq -\infty$ for all even $k \leq k_{max}$,

2. $\delta_{n-k}(H) = -\infty$ for all odd k if $g_r = -\infty$ for all odd r, and

3. $\delta_{n-k}(H) \neq -\infty$ for all odd $k \leq k_{max}$ if $g_r \neq -\infty$ for some odd r.

4.3 Block diagonal matrices

Let

$$A = \text{blockdiag}(A_1, A_2, \ldots, A_p) = \begin{pmatrix} A_1 & & & -\infty \\ & A_2 & & \\ & & \ddots & \\ -\infty & & & A_p \end{pmatrix}.$$

$D(A_i)$ is a subgraph of $D(A)$ for every $i = 1, \ldots, p$. Every $D(A_i)$ is disjoint from any $D(A_j)$, $j \neq i$. So any cycle in $D(A)$ has nodes entirely within one of these disjoint subgraphs, and it is not possible to have a cycle in $D(A)$ with arcs corresponding to elements from more than one of the matrices $A_1, \ldots, A_p$.

We now show how to solve JRP(A) for A in polynomial time, as long as we can solve JRP(A_j) in polynomial time, for $j = 1, \ldots, p$. This is shown in an algorithm called JRPBLOCKDIAG (see Figure 1).

Let $n(j)$ be the order of A_j, $j = 1, \ldots, p$. Assume that we have solved JRP(A_j). This may have been done in polynomial time if A_j is one of the special types of matrix previously mentioned in this paper.

So for $j = 1, \ldots p$ and $r = 1, \ldots, k$ we are able to find $\delta_{n(j)-r}(A_j)$ and also a principal submatrix $B_{jr} \in (A_j)_r$ (where $(A_j)_r$ is the set of all $r \times r$ principal submatrices of A_j) and permutation $\pi_{jr} \in ap(B_{jr})$ such that $w(B_{jr}, \pi_{jr}) = \delta_{n(j)-r}(A_j)$.

Let $D_j = D(A_j)$. For each block A_j, we have the following information: For $r = 1, \ldots, k$, the permutation π_{jr} in D_j gives cycles of total length r and total weight $\delta_{n(j)-r}(A_j)$.

We will use S, a set of pairs to tell us which submatrix to select and which elements from within it to select. We do this by assigning pairs (j, r) to S. A pair (j, r) tells us that by choosing B_{jr} and π_{jr} we select a total of r elements from B_{jr} and give a total sum of $\delta_{n(j)-r}(A_j)$.

There are p stages to the algorithm. At each stage information is collected and then stored within a set of triples called M_j. Each triple has the form (S, w, k), where S is as described above, w is the total weight of elements selected by using the information in S, and k is the total number of elements selected by using the information in S.

M_0 is set to $\{(\emptyset, 0, 0)\}$ at Stage 0. For $j = 1, \ldots, p$, at Stage j, the information found from A_j (i.e. $\delta_{n(j)-1}(A_j), \delta_{n(j)-2}(A_j), \ldots, \delta_0(A_j)$) and the information from Stage $j-1$ (i.e. M_{j-1}) is combined to produce M_j. We start by copying all triples from M_{j-1} to M_j. Next, if we can find a triple (S, w, k) (of the form described above) by combining the information found from A_j and M_{j-1} that is not in M_{j-1}, then we add (S, w, k) to M_j. Otherwise, if w is larger than the second coordinate of any triple in M_{j-1} having third component equal to k, then we replace that triple with (S, w, k) in M_j.

We now give the algorithm, called JRPBLOCKDIAG, and then discuss the correctness and complexity of this algorithm.

Algorithm JRPBLOCKDIAG
Input: $A = \text{blockdiag}(A_1, A_2, \ldots, A_p) \in \overline{\mathbb{R}}^{n\times n}$.
Output: For $k = 1, \ldots, n$, $\delta_{n-k}(A)$, and if $\delta_{n-k}(A)$ is finite, then also k independent entries of a $k \times k$ principal submatrix of A whose total is $\delta_{n-k}(A)$.

1. Set $M_0 = \{(\emptyset, 0, 0)\}$
2. For $j = 1$ to p :
   (a) For $r = 1$ to $n(j)$:
      i. Find $\delta_{n(j)-r}(A_j)$.
      ii. Find $B_{jr} \in (A_j)_r$ and $\pi_{jr} \in ap(B_{jr})$ such that $w(B_{jr}, \pi_{jr}) = \delta_{n(j)-r}(A_j)$.
   (b) Set $M_j = M_{j-1}$
   (c) For each element $(S, w, l) \in M_{j-1}$:
   For each $r = 1$ to $n(j) - l$ with $\delta_{n(j)-r}(A_j)$ finite :
      i. If $\nexists(S', w', l+r) \in M_j$, then add $(S \cup \{(j, r)\}, w + \delta_{n(j)-r}(A_j), l + r)$ to M_j.
      ii. If $\exists(S', w', l + r) \in M_j$ and $w' < w + \delta_{n(j)-r}(A_j)$, then remove $(S', w', l+r)$ from M_j and add $(S \cup \{(j, r)\}, w + \delta_{n(j)-r}(A_j), l+r)$ to M_j.
3. For $k = 1$ to n :
   If $\exists(S, w, k) \in M_p$, then return $\delta_{n-k}(A) = w$, and for $i = 1, \ldots, r$ and all $(j, r) \in S$, return the element of A that corresponds to the $(i, \pi_{jr}(i))$ entry of B_{jr}. Else return $\delta_{n-k}(A) = -\infty$.

Figure 1: An algorithm for solving JRP for block diagonal matrices.

Lemma 23.

1. If $(S, w, k) \in M_j$ in Step 3 of the algorithm, then

   (a) $S \subseteq (\{1, \dots, p\} \times \{1, \dots, n\})^{\{1,\dots,p\}}$,

   (b) $\sum_{(i,s) \in S} \delta_{n(i)-s}(A_i) = w$,

   (c) $\sum_{(i,s) \in S} s = k$,

   (d) If $(S', w', k) \in M_j$, then $S' = S$ and $w' = w$,

   (e) If $S' \subseteq (\{1, \dots, p\} \times \{1, \dots, n\})^{\{1,\dots,p\}}$, $w' = \sum_{(i,s) \in S'} \delta_{n(i)-s}(A_i)$ and $\sum_{(i,s) \in S'} s = k$ then $w' \leq w$.

2. If $S \subseteq (\{1, \dots, p\} \times \{1, \dots, n\})^{\{1,\dots,p\}}$, and $\sum_{(i,s) \in S} s = k \leq n$, then in Step 3 of the algorithm, $\exists (S', w', k) \in M_j$ where $w \leq w'$.

Proof. Statements 1(a)-1(c) are proved by induction on j, and hold automatically for $j = 0$. For $j > 0$, assume $(S, w, k) \in M_j$. We have two cases to consider:

Case 1: If $\nexists (j, r) \in S$, then $(S, w, k) \in M_{j-1}$, so 1(a)-1(c) follow by induction.

Case 2: If $\exists (j, r) \in S$, then $(S - \{(j, r)\}, w - \delta_{n(j)-r}(A_j), k - r) \in M_{j-1}$. Note that $r \leq n$, as the third coordinate of this lies between 0 and $n - r$. So again 1(a)-1(c) follow by induction.

To prove 1(d), we use the fact that each element of M_j has a unique third component due to the way Step 2(c) of the algorithm was constructed.

To prove 2, we use induction on $\max_{(h,s) \in S} h$. It holds for $S = \emptyset$, and for the inductive step, assume $i = \max_{(h,s) \in S} h$, and let $(i, r) \in S$. Note that $i \leq j$, so $i - 1 \leq j - 1$. Therefore $\exists (S', w - \delta_{n(j)-r}(A_j), k - r) \in M_{i-1}$ by induction. Thus in Step 2(c) of the construction of M_i, $(S' \cup \{(i, r)\}, w, k)$ was added to M_i. Then either $(S' \cup \{(i, r)\}, w, k) \in M_j$ or $\exists (S'', w'', k) \in M_j$ with $w < w''$. Either way, 2 holds.

To prove 1(e), note that by 2, $\exists (S'', w'', k) \in M_j$, with $w' \leq w''$. By 1(d), we see that $S'' = S$ and $w'' = w$, therefore $w' \leq w$, and 1(e) follows. ☐

From part 2 of Lemma 23, we see that if we have an $S \subseteq (\{1, \dots, p\} \times \{1, \dots, k\})^{\{1,\dots,p\}}$ with $\sum_{(i,s) \in S} s = k$ and $\sum_{(i,s) \in S} \delta_{n(i)-s}(A_i) = w$, then $\exists (S^*, w^*, k) \in$

M_p (which gives at least as much total weight w^* as w does). Then from part 1(e) of Lemma 23, we see that if $(S^*, w^*, k) \in M_p$, then no other first coordinate satisfying $\sum_{(i,s)\in S} s = k$ will provide a bigger total weight than w^*. Selecting elements of A that correspond to the $(i, \pi_{jr}(i))$ entry of B_{jr} for all $i = 1, \ldots, r$ and all $(j, r) \in S$ and adding them up will give w^*, which is the highest possible value, so $w = \delta_{n-k}(A)$.

Theorem 24. If $A = \text{blockdiag}(A_1, A_2, \ldots, A_p) \in \overline{\mathbb{R}}^{n\times n}$ and we can solve JRP(A_i, k) in $O(t)$ time, for all $i = 1, \ldots, p$ and $k = 1, \ldots, n$, then we can solve JRP(A) in $O(n^2(n + t))$ time.

Proof. Correctness follows from Lemma 23: For $k = 1, \ldots, n$, Step 3 of the algorithm chooses an element (S, w, k) from M_p (assuming $M_p \neq \emptyset$) with third component k. Thus for each k it follows (by 1(a) and 1(c) of Lemma 23) that the solution generated from S is feasible (i.e. if we select the elements of A that correspond to the $(i, \pi_{jr}(i))$ entries of B_{jr} for all $i = 1, \ldots, r$ and all $(j, r) \in S$, then k finite elements of A will be selected, resulting in a finite total weight). It also follows that its total weight is w (by 1(b) of Lemma 23), and there is no better solution (by 1(e) of Lemma 23). By part 2 of Lemma 23, it follows that if $M_p = \emptyset$ at Step 4 of the algorithm, then $\delta_{n-k}(A) = -\infty$.

For the time bound, notice that the size of each set M_{j-1} is no greater than n, because there is at most one element in M_{j-1} with the same third component (by 1(d) of Lemma 23). Each update operation of Step 2(c) can be done in constant time for each r and each M_{j-1}, and must be repeated for all $O(n)$ elements of M_{j-1} and $O(n)$ times for the r loop. Steps 1, and 2b require one operation each so can be performed in constant time. Assume that for each r, Step 2(a) can be performed in $O(t)$ time. The whole of Step 2 is repeated p times. It is easily seen that Step 3 can be done in $O(n^2)$ time. So algorithm JRPBLOCKDIAG runs in $O(pn(n+t))$ time. As $p \leq n$, this becomes $O(n^2(n + t))$ time. □

Corollary 25. If in Theorem 24, t is polynomial in n, that is, if JRP(A_i, k') can be solved in polynomial time, for all $i = 1, \ldots, p$ and $k' = 1, \ldots, k$, then for a block diagonal matrix A, JRP(A) can be solved in polynomial time.

Any matrix that can be obtained by permuting the rows and/or columns of the matrix containing zeros on the main diagonal and $-\infty$ elsewhere, will be called a *permutation matrix*. Any matrix that can be obtained by permuting the rows and/or columns of a matrix containing finite entries on the main diagonal and $-\infty$ elsewhere, will be called a *generalized permutation matrix*. It is known [8] that JRP(A) can be solved in $O(n^2)$ time, for a permutation matrix A.

Corollary 26. For any generalized permutation matrix $A \in \overline{\mathbb{R}}^{n\times n}$, JRP($A$) can be solved in $O(n^3)$ time.

Proof. Generalized permutation matrices are a special type of block diagonal matrix. Each block contains only one cycle, therefore we can solve JRP for each block in linear time, and hence use Theorem 24 to give the result. □

Remark. Note that this complexity can be improved to $O(n^2)$ time for generalized permutation matrices, by a slight alteration to this algorithm [16].

Any element of a matrix that does not lie on a finite cycle may be set to $-\infty$ without affecting δ_{n-k} for any $k \in N$. Hence if B is in Frobenius normal form, then we may set all elements of off-diagonal blocks in B to $-\infty$. Therefore if we define $C_i = B_{ii}$, for $i = 1, \ldots, p$, i.e.

$$C = \text{blockdiag}(C_1, C_2, \ldots, C_p),$$

then we have $\delta_{n-k}(C) = \delta_{n-k}(A)$ for all $k \in N$. We have derived the following:

Theorem 27. For any $A \in \overline{\mathbb{R}}^{n\times n}$, if we can solve JRP for all diagonal blocks of the Frobenius normal form of A in polynomial time, then we can solve JRP(A) in polynomial time (by converting it to a block diagonal matrix and using the JRPBLOCKDIAG algorithm of Figure 1).

References

[1] Ahuja, R.K., Magnanti, T., and Orlin J.B., *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, New Jersey, 1993

[2] Baccelli, F.L., G. Cohen, G.-J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity*. (J.Wiley and Sons, New York, 1992).

[3] Burkard, R.E., Selected topics in assignment problems., *Discrete Applied Mathematics* **123** (2002), 257-302.

[4] Burkard, R.E. and P. Butkovič, Finding all essential terms of a characteristic maxpolynomial, *Discrete Applied Mathematics* **130** (2003), 367-380.

[5] Burkard, R.E. and P. Butkovič, Max algebra and the linear assignment problem, *Math. Program., Ser.B* **98** (2003), 415-429.

[6] Burkard, R.E. and E. Çela, Linear Assignment Problems and Extensions, in *Handbook of Combinatorial Optimization* (P.M.Pardalos and D.-Z.Du, Eds.), Supplement Volume A, Kluwer Academic Publishers, 75-149, 1999.

[7] Burkard, R.E. and R.A. Cuninghame-Green, private communication.

[8] Butkovič, P., On the complexity of computing the coeffiecients of max-algebraic characteristic polynomial and characteristic equation. *Kybernetika* **39** (2003), No. 2, 129–136.

[9] Butkovič, P. and R.A. Cuninghame-Green, The linear assignment problem for special matrices. *IMA journal of Management Mathematics* **15** (2004), 1–12.

[10] Butkovič, P. and L. Murfitt, Calculating essential terms of a characteristic maxpolynomial. *CEJOR* **8** (2000), 237–246.

[11] Cuninghame-Green, R.A., *Minimax Algebra. Lecture Notes in Economics and Math. Systems* **166**, (Springer, Berlin, 1979).

[12] Cuninghame-Green, R.A., The characteristic maxpolynomial of a matrix. *J. Math. Analysis and Applications* **95** (1983), 110–116.

[13] Cuninghame-Green, R.A., Minimax Algebra and Applications in: *Advances in Imaging and Electron Physics*. Vol. **90**, pp. 1–121 (Academic Press, New York, 1995).

[14] Gassner, E., Variants of the Assignment and of the Transportation Problem, PhD thesis, Graz, 2004.

[15] Gondran M., and M. Minoux, Linear Algebra of Dioïds: a survey of recent results. *Annals of Discrete Mathematics* **19** (1984), 147-164.

[16] Lewis, S. C., On the Job Rotation Problem, PhD thesis, University of Birmingham, in preparation.

[17] Rosen, K. H. (ed), Handbook of discrete and combinatorial mathematics, Boca Raton; London : CRC Press, 2000.

[18] van Leeuwen, J. (ed), Graph Algorithms, in Handbook of theoretical computer science - Vol.A : Algorithms and complexity, 525-631, Amsterdam; Oxford : Elsevier; Cambridge, MA : MIT Press, 1990.

[19] Zimmermann, U., *Linear and Combinatorial Optimization in Ordered Algebraic Structures*. Annals of Discrete Mathematics **10** (North Holland, Amsterdam, 1981).