The Wiener maximum quadratic assignment problem

Eranda Çela[*] Nina S. Schmuck[†] Shmuel Wimer[‡]
Gerhard J. Woeginger[§]

Abstract

We investigate a special case of the maximum quadratic assignment problem where one matrix is a product matrix and the other matrix is the distance matrix of a one-dimensional point set. We show that this special case, which we call the Wiener maximum quadratic assignment problem, is NP-hard in the ordinary sense and solvable in pseudo-polynomial time.

Our approach also yields a polynomial time solution for the following problem from chemical graph theory: Find a tree that maximizes the Wiener index among all trees with a prescribed degree sequence. This settles an open problem from the literature.

Keywords. Combinatorial optimization; computational complexity; graph theory; degree sequence; Wiener index.

1 Introduction

The *Quadratic Assignment Problem* (QAP) in Koopmans-Beckmann form [1] takes as input two $n \times n$ square matrices $A = (a_{ij})$ and $B = (b_{ij})$ with real entries, and asks to find a permutation π that minimizes (or maximizes) the objective function

$$Z_\pi(A, B) \;:=\; \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi(i)\pi(j)}\, b_{ij}. \tag{1}$$

[*]Corresponding author, cela@opt.math.tu-graz.ac.at, Phone: +43 316 873 5366, Fax: +43 316 873 105366, Institut für Optimierung und Diskrete Mathematik, TU Graz, Steyrergasse 30, A-8010 Graz, Austria

[†]nina.schmuck@tugraz.at. Institut für Optimierung und Diskrete Mathematik, TU Graz, Steyrergasse 30, A-8010 Graz, Austria

[‡]wimers@macs.biu.ac.il. School of Engineering, Bar-Ilan University, Ramat-Gan 52900, Israel

[§]gwoegi@win.tue.nl. Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands

Here π ranges over the set S_n of all permutations of $\{1, 2, \ldots, n\}$. The QAP is a hard and well-studied problem in combinatorial optimization; we refer the reader to the book [2] by Çela and the recent book by Burkard et al. [3] for more information on this problem. One branch of research on the QAP concentrates on the algorithmic behavior of strongly-structured special cases; see for instance Burkard & al [4] or Deineko & Woeginger [5] for typical results in this direction. In this paper, we will contribute a new topic to this research branch.

1.1 The Wiener Max-QAP

This special case of the QAP restricts matrix A to be a symmetric *product matrix*, which means that there are non-negative integers $\alpha_1 \le \alpha_2 \le \cdots \le \alpha_n$ such that

$$a_{ij} \;=\; \alpha_i\,\alpha_j \qquad \text{for } 1 \le i, j \le n. \tag{2}$$

The second matrix in the Wiener Max-QAP is the distance matrix of a one-dimensional point set, which means that there are integers $\beta_1 \le \beta_2 \le \cdots \le \beta_n$ such that

$$b_{ij} \;=\; |\beta_i - \beta_j| \qquad \text{for } 1 \le i, j \le n. \tag{3}$$

Throughout this paper, a matrix B of the form (3) will be called a *1D-distance matrix*. The goal in the Wiener Max-QAP is to *maximize* the objective value in (1), which can be rewritten as

$$Z_\pi(A, B) \;=\; \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{\pi(i)}\,\alpha_{\pi(j)}\,|\beta_i - \beta_j|. \tag{4}$$

In this paper, we fully determine the computational complexity of the Wiener Max-QAP. Our results are as follows. As a negative result, we will prove in Section 2 that the Wiener Max-QAP is NP-hard in the ordinary sense. On the positive side, in Section 3 we will derive a useful decomposition property, and we will show that there always exists an optimal permutation that is V-shaped. These positive results are then applied in Section 4 to derive a pseudo-polynomial time algorithm for the Wiener Max-QAP.

1.2 The Wiener index of a tree

The *Wiener index* $W(G)$ of a connected undirected graph $G = (V, E)$ is the sum of the distances between all pairs of vertices in V. The Wiener index was introduced in 1947 by Harold Wiener [6] to characterize certain molecular structure properties of saturated hydrocarbons. We refer the reader to the survey article [7] of Dobrynin, Entringer & Gutman for comprehensive information on this fundamental graph parameter. Chemists are often interested in the Wiener index of certain trees, where the vertices represent atoms and where the vertex degrees correspond to the valencies of the atoms. Entringer, Jackson & Snyder [8] show that among all r-vertex trees, the path P_r has the largest and the star $K_{1,r}$ has the smallest Wiener index. Fischermann & al [9] characterize the trees that minimize the Wiener index among all trees with r vertices and maximum degree at most Δ, and they also provide several results on the corresponding maximization question.

Wang [10] describes a simple greedy algorithm for finding a tree that minimizes the Wiener index among all trees with a prescribed degree sequence. Zhang & al [11] derive the same result independently and by different techniques. The corresponding maximization question remains open (note that the paper [10] claims a solution to it, and that the corrigendum [12] points out a crucial mistake that invalidates these claims). Wang [12] writes about the maximization question: *"While the extremal trees seem to be difficult to find and are not unique, an algorithm to find at least one of such trees may exist and may be easier to find."*

In this paper, we resolve the computational complexity of the maximization question: There exists a polynomial time algorithm that finds a tree that maximizes the Wiener index among all trees with a prescribed degree sequence. More precisely, we will show in Section 5 that this maximization question can be modeled as a special case of the Wiener Max-QAP (modulo certain minor modifications). Consequently, the machinery developed in Sections 3 and 4 can be applied to it. The pseudo-polynomial time complexity of our algorithm turns into a polynomial time complexity, since all the involved numbers are moderately small.

2 Complexity of the Wiener Max-QAP

In this section, we establish NP-hardness of the Wiener Max-QAP. The proof is done by means of a reduction from the following variant of the partition problem (see Garey & Johnson [13]) which is well-known to be NP-hard in the ordinary sense.

> Problem: Partition
>
> Input: A sequence $q_1, \ldots, q_{2k}$ of $2k$ positive integers with $\sum_{i=1}^{2k} q_i = 2Q$.
>
> Question: Does there exist $I \subset \{1, \ldots, 2k\}$ with $|I| = k$ and $\sum_{i \in I} q_i = Q$?

We construct an instance of the Wiener Max-QAP of dimension $n = 2k$. The $n \times n$ product matrix A is defined by $\alpha_i = q_i$, $\forall i = 1, 2, \ldots, n$, and hence satisfies $a_{ij} = q_i q_j$. The 1D-distance matrix B uses the points $\beta_i = 1$ for $1 \le i \le k$ and $\beta_i = 2$ for $k+1 \le i \le 2k$. Note that $b_{ij} = |\beta_i - \beta_j|$ equals 1 if one of the indices i, j lies in the range $1, \ldots, k$ whereas the other index lies in $k+1, \ldots, 2k$; in all other cases $b_{ij} = 0$.

Consider a permutation $\pi \in S_{2k}$, and let J denote the set of all i with $\pi(i) \le k$. Note that $|J| = k$. By setting $x = \sum_{i \in J} q_i$, the objective value in (4) can be rewritten as

$$\sum_{i \in J} \sum_{j \notin J} q_i q_j \;=\; \sum_{i \in J} q_i \sum_{j \notin J} q_j \;=\; x(2Q - x) \;=\; 2Qx - x^2.$$

As the concave function $f(x) = 2Qx - x^2$ is maximized at $x = Q$, it is easily seen that the QAP has objective value at least Q^2 if and only if the PARTITION instance has a positive answer. This yields the following result.

Theorem 2.1 *The Wiener Max-QAP is NP-hard in the ordinary sense.* $\square$

3 Structure of the Wiener Max-QAP

We first discuss a useful decomposition property of the Wiener Max-QAP. Consider some fixed instance with product matrix A and 1D-distance matrix B, and let $I \subset \{1, \ldots, n\}$ with $|I| = k$ be some fixed subset of the indices. Instead of optimizing over all possible permutations in (4), we only allow permutations π that map I into $J = \{1, \ldots, k\}$ and that consequently map the complement of I into the complement $\{k+1, \ldots, n\}$ of J. Intuitively speaking, this subdivided version QAP assigns the values α_i with $i \in I$ to the points $\beta_1, \ldots, \beta_k$ and the values α_i with $i \notin I$ to the remaining points $\beta_{k+1}, \ldots, \beta_n$.

Any permutation $\pi \in S_n$ induces a bijection σ that maps J into I, and another bijection τ that maps the complement of J into the complement of I. Furthermore, denote $X = \sum_{i \in I} \alpha_i$ and $Y = \sum_{i \notin I} \alpha_i$. The objective value $Z_\pi(A, B)$ in (4) can then be written as $Z_1 + Z_2 + Z_3 + Z_4$ in the following way.

$$Z_1 = \sum_{i \in J} \sum_{j \in J} \alpha_{\pi(i)} \alpha_{\pi(j)} |\beta_i - \beta_j| = \sum_{i \in J} \sum_{j \in J} \alpha_{\sigma(i)} \alpha_{\sigma(j)} |\beta_i - \beta_j| \tag{5}$$

$$Z_2 = \sum_{i \notin J} \sum_{j \notin J} \alpha_{\pi(i)} \alpha_{\pi(j)} |\beta_i - \beta_j| = \sum_{i \notin J} \sum_{j \notin J} \alpha_{\tau(i)} \alpha_{\tau(j)} |\beta_i - \beta_j| \tag{6}$$

$$\begin{aligned} Z_3 &= \sum_{i \in J} \sum_{j \notin J} \alpha_{\pi(i)} \alpha_{\pi(j)} |\beta_i - \beta_j| = \sum_{i \in J} \sum_{j \notin J} \alpha_{\sigma(i)} \alpha_{\tau(j)} (\beta_j - \beta_k + \beta_k - \beta_i) \\ &= \sum_{i \in J} \alpha_{\sigma(i)} Y (\beta_k - \beta_i) + \sum_{j \notin J} \alpha_{\tau(j)} X (\beta_j - \beta_k) \end{aligned} \tag{7}$$

$$Z_4 = \sum_{i \notin J} \sum_{j \in J} \alpha_{\pi(i)} \alpha_{\pi(j)} |\beta_i - \beta_j| = Z_3 \tag{8}$$

Note that in the resulting summations in (5)–(7), every single term does either depend on function σ or on function τ, but does never depend on both functions simultaneously. If we collect all the terms in $Z_1 + Z_2 + Z_3 + Z_4$ that solely depend on this function σ, we get

$$\sum_{i\in J}\sum_{j\in J} \alpha_{\sigma(i)}\alpha_{\sigma(j)}\,|\beta_i - \beta_j| + 2\sum_{i\in J} \alpha_{\sigma(i)} Y(\beta_k - \beta_i). \tag{9}$$

We observe that the objective function in (9) essentially corresponds to a smaller $(k+1)$-dimensional Wiener Max-QAP: The underlying 1D-distance matrix is built around the k points $\beta_1, \ldots, \beta_k$ plus a duplicated point at β_k (which corresponds to the occurrence of β_k in the right hand sum). The underlying product matrix is built around the k numbers α_i with $i \in I$ plus the number Y (which corresponds to the factor Y in the right hand sum). Furthermore, the right hand sum in (9) imposes the additional restriction that the new value Y has to be assigned to the duplicated point at β_k.

As a consequence of all this, the problem of finding the optimal function σ and the problem of finding the optimal function τ are two separate optimization problems that can be solved independently of each other. We call this the *decomposition property* of the Wiener Max-QAP. This decomposition property plays a central role in many of our arguments. As a first application of the decomposition property, we next deduce a result on the combinatorial structure of optimal permutations for the Wiener Max-QAP.

Definition 3.1 *A permutation $\pi \in S_n$ is called* V-shaped*, if there exists an index ℓ with $1 \le \ell \le n$ such that $\pi(i) > \pi(i+1)$ for $i = 1, \ldots, \ell-1$ and such that $\pi(i) < \pi(i+1)$ for $i = \ell, \ldots, n-1$.*

In other words, a V-shaped permutation π is first decreasing up to ℓ, and then increasing from ℓ onwards, where the increasing or the decreasing part can also be empty.

Theorem 3.2 *Every instance of the Wiener Max-QAP possesses an optimal solution π that is V-shaped.*

Proof. For simplicity of presentation, we will assume without much loss of generality that all α-values are pairwise distinct, and that also all β-values are pairwise distinct. The statement for the general case then follows easily from this (by locally reordering or renaming the values).

Now consider an optimal permutation π, and note that the values $\pi(1), \ldots, \pi(n)$ and $\alpha_{\pi(1)}, \ldots, \alpha_{\pi(n)}$ are ordered in the same way. Suppose for the sake of contradiction that permutation π has a local maximum at k with $\alpha_{\pi(k-1)} < \alpha_{\pi(k)}$ and $\alpha_{\pi(k+1)} < \alpha_{\pi(k)}$. By the decomposition property of the Wiener Max-QAP, the optimal permutation π induces an optimal solution to the five-dimensional problem of assigning the five values

$$L = \sum_{i=1}^{k-2} \alpha_{\pi(i)}, \quad \alpha_{\pi(k-1)}, \quad \alpha_{\pi(k)}, \quad \alpha_{\pi(k+1)}, \quad R = \sum_{i=k+2}^{n} \alpha_{\pi(i)}$$

to the five points β_{k-1}, β_{k-1}, β_k, β_{k+1}, and β_{k+1} subject to the constraint that value L is assigned to point β_{k-1} and that value R is assigned to point β_{k+1}.

Now let us switch the positions of $\alpha_{\pi(k-1)}$ and $\alpha_{\pi(k)}$ in the solution that π induces for the five-dimensional problem, such that $\alpha_{\pi(k-1)}$ goes to β_k and $\alpha_{\pi(k)}$ goes to β_{k-1}. Since this switch cannot increase the objective value, the difference between the corresponding two objective values is non-positive:

$$\big(\alpha_{\pi(k)} - \alpha_{\pi(k-1)}\big)\ (\beta_k - \beta_{k-1})\ \big(\alpha_{\pi(k+1)} + R - L\big) \ \leq\ 0 \tag{10}$$

In an analogous way, we can switch the positions of $\alpha_{\pi(k)}$ and $\alpha_{\pi(k+1)}$ in the induced solution. This then leads to the following inequality:

$$\big(\alpha_{\pi(k)} - \alpha_{\pi(k+1)}\big)\ (\beta_{k+1} - \beta_k)\ \big(\alpha_{\pi(k-1)} - R + L\big) \ \leq\ 0 \tag{11}$$

The first two factors in the left hand side of (10) and also the first two factors in the left hand side of (11) all are positive. This implies $\alpha_{\pi(k+1)} + R - L \leq 0$ and $\alpha_{\pi(k-1)} - R + L \leq 0$. By summing these two inequalities we arrive at the contradiction $\alpha_{\pi(k+1)} + \alpha_{\pi(k-1)} \leq 0$. We conclude that π cannot have any local maximum, and this yields that π indeed is V-shaped. $\Box$

4 An algorithm for the Wiener Max-QAP

In this section, we design a pseudo-polynomial time algorithm for the Wiener Max-QAP that is based on a standard dynamic programming approach. We recall that the two underlying sequences $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$ and $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ are in non-decreasing order.

Every state in the dynamic program is specified by a quadruple (k, m, L, R) of integers that satisfy the following conditions:

$$1 \leq k \leq n, \qquad 1 \leq m \leq n-k+1, \qquad 0 \leq L, R, \qquad L + R = \sum_{i=k+1}^{n} \alpha_i. \tag{12}$$

With every such state (k, m, L, R) we associate the following $(k+2)$-dimensional Wiener Max-QAP: The product matrix results from the $k+2$ non-negative integers in the sequence $\alpha_1, \ldots, \alpha_k, L, R$, and the 1D-distance matrix results from the k points $\beta_m, \beta_{m+1}, \ldots, \beta_{m+k-1}$ plus another point in β_m plus another point in β_{m+k-1}. The goal is to find the best solution to this QAP subject to the constraint that the value L is assigned to point β_m and that the value R is assigned to β_{m+k-1}. In other words, we want to find a bijection σ from $\{m, \ldots, m+k-1\}$ to $\{1, \ldots, k\}$ that maximizes the objective value

$$Z = \sum_{i=m}^{m+k-1} \sum_{j=m}^{m+k-1} \alpha_{\sigma(i)}\alpha_{\sigma(j)} |\beta_i - \beta_j| + 2LR |\beta_{m+k-1} - \beta_m| + $$
$$+ 2 \sum_{i=m}^{m+k-1} \alpha_{\sigma(i)} L |\beta_i - \beta_m| + 2 \sum_{i=m}^{m+k-1} \alpha_{\sigma(i)} R |\beta_{m+k-1} - \beta_i|$$

We use $Z(k, m, L, R)$ to denote the maximum objective value of the corresponding state. Next, we will describe how to compute and to store all the values $Z(k, m, L, R)$ step by step and in increasing order of k. For $k = 1$ the corresponding instances are trivial to solve, since they only have a single feasible solution.

Next consider some fixed state (k, m, L, R) with $k \geq 2$, and denote $M = \sum_{i=1}^{k-1} \alpha_i$. By the decomposition property discussed in Section 3 and by Theorem 3.2, there exists an optimal bijection σ that induces a V-shaped assignment of the k values $\alpha_1, \ldots, \alpha_k$ to the k points $\beta_m, \beta_{m+1}, \ldots, \beta_{m+k-1}$. This implies that α_k as the largest of the k values must be assigned either to point β_m or to point β_{m+k-1}. First consider the case where α_k is assigned to point β_m. Then the remaining $k-1$ values are assigned to $\beta_{m+1}, \ldots, \beta_{m+k-1}$, and by the decomposition property the largest possible objective value in this case is

$$Z_1 := Z(k-1, m+1, L+\alpha_k, R) + 2(L+\alpha_k)(M+R)|\beta_{m+1} - \beta_m|. \tag{13}$$

In the second case the value α_k is assigned to point β_{m+k-1}. Then the remaining $k-1$ values are assigned to $\beta_m, \ldots, \beta_{m+k-2}$, and the largest possible objective value is

$$Z_2 := Z(k-1, m, L, R+\alpha_k) + 2(L+M)(R+\alpha_k)|\beta_{m+k-1} - \beta_{m+k-2}|. \tag{14}$$

This yields $Z(k, m, L, R) = \max\{Z_1, Z_2\}$, and in this fashion one easily determines all the function values $Z(k, m, L, R)$ with $2 \leq k \leq n$. In the end, the optimal objective value of the underlying QAP instance can be found as $Z(n, 1, 0, 0)$.

Theorem 4.1 *The Wiener Max-QAP has a pseudo-polynomial time solution algorithm with time complexity* $O(n^2 \cdot \sum \alpha_i)$.

Proof. The correctness of the dynamic programming approach is clear from the above considerations. It remains to analyze the time complexity. We observe that there are only $O(n^2 \cdot \sum \alpha_i)$ different states (k, m, L, R) in the dynamic program: There are $O(n)$ possible values for k and m, respectively, and there are $O(\sum \alpha_i)$ possible values for L; note that the value of R is already fully determined by the values of k and L.

The expressions in (13) and (14) can be evaluated in constant time $O(1)$: They refer to values $Z(k-1, *, *, *)$ that are known from earlier stages of the dynamic program, and they refer to the values $M = \sum_{i=1}^{k-1} \alpha_i$ that can all be precomputed and stored in a preprocessing phase. All in all, this yields that the time complexity is proportional to the number of states and hence is $O(n^2 \cdot \sum \alpha_i)$. $\square$

Note that our approach only yields the optimal objective value. By storing appropriate auxiliary information in the states of the dynamic program, one can also compute the corresponding optimal permutation within the same time complexity. These are standard techniques, and we do not elaborate on them.

5 Maximizing the Wiener index of a tree

We now return to the Wiener index of a graph that has been introduced and discussed in Section 1.2. We will investigate the following algorithmic problem MaxWiener-Tree that was left open by Wang [10, 12]: An instance consists of a degree sequence $d_1, \ldots, d_r$ of r positive integers with $\sum_{i=1}^{r} d_i = 2r - 2$. The goal is to determine the largest possible Wiener index over all trees with degree sequence $d_1, \ldots, d_r$.

Recall that a *caterpillar* is a tree that turns into a path (the so-called *backbone* of the caterpillar) if all its leaves are removed. Shi [14] proved that for every instance of MaxWiener-Tree, all the maximizing trees are caterpillars; other proofs of this result can be found for instance in Wang [10] and Schmuck [15].

Now consider such a caterpillar T, let $v_1, \ldots, v_n$ denote the vertices ordered along the backbone of the caterpillar, and let ℓ_i with $1 \le i \le n$ denote the number of leaves adjacent to vertex v_i. We define set C_i to consist of vertex v_i together with its ℓ_i adjacent leaves. Then the vertex pairs inside C_i contribute an amount of ℓ_i^2 to the Wiener index $W(T)$, and the pairs with one vertex in C_i and one vertex in C_j $(i \neq j)$ contribute

$$\begin{aligned} & \ell_i \ell_j (|j-i|+2) \;+\; \ell_i \,(|j-i|+1) \;+\; \ell_j \,(|j-i|+1) \;+\; |j-i| \\ & \qquad = \;(\ell_i+1)\,(\ell_j+1)\,|j-i| \;+\; (2\ell_i\ell_j + \ell_i + \ell_j). \end{aligned}$$

Hence the Wiener index of this caterpillar T is

$$\begin{aligned} W(T) \;&=\; \sum_{i=1}^{n} \ell_i^2 \;+\; \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[(\ell_i+1)\,(\ell_j+1)\,|j-i| \;+\; (2\ell_i\ell_j + \ell_i + \ell_j)\right] \\ &=\; \left(\sum_{i=1}^{n} \ell_i\right)^2 + (n-1)\sum_{i=1}^{n} \ell_i \;+\; \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} (\ell_i+1)\,(\ell_j+1)\,|j-i|. \end{aligned} \tag{15}$$

Neither the first nor the second sum in (15) do depend on the way how the numbers $\ell_1, \ldots, \ell_n$ are assigned to the backbone vertices $v_1, \ldots, v_n$. The assignment in the third sum in (15) yields an instance of the Wiener Max-QAP with values $\alpha_i = \ell_i + 1$ and points $\beta_i = i$.

Observation 5.1 *The problem of finding the maximizing caterpillar for an instance of* MaxWiener-Tree *with an explicitly specified backbone $v_1, \ldots, v_n$ and an explicitly specified sequence $\ell_1, \ldots, \ell_n$ of leaf-numbers is equivalent to a Wiener Max-QAP.* □

Next, consider a degree sequence $d_1, \ldots, d_r$ that forms an instance of MaxWiener-Tree, and assume without loss of generality that $2 \le d_1 \le d_2 \le \cdots \le d_n$ and that $d_{n+1} = \cdots = d_r = 1$. (Since the case $n = 1$ is trivial, we assume from now on that $n \ge 2$.) It is straightforward to see that the backbone of the maximizing caterpillar will consist of n vertices $v_1, \ldots, v_n$. In contrast to this, it is not straightforward to write down the sequence $\ell_1, \ldots, \ell_n$ of leaf-numbers: If one of the inner backbone vertices v_i with $2 \le i \le n-1$ gets degree d_i then it is adjacent to $\ell_i = d_i - 2$ leaves, whereas if one of the two outermost backbone vertices v_i with $i = 1$ or $i = n$ gets degree d_i then it is adjacent to $\ell_i = d_i - 1$ leaves.

Motivated by the discussion in the preceding paragraph, we introduce an $(n+2)$-dimensional instance of the Wiener Max-QAP: The product matrix is built around the n numbers $\alpha_i = d_i - 1$ for $1 \le i \le n$ and the two additional numbers $\alpha_{n+1} = \alpha_{n+2} = 1$. The 1D-distance matrix is built around the n points $\beta_i = i-1$ for $2 \le i \le n+1$ and the two additional points $\beta_1 = 1$ and $\beta_{n+2} = n$. Furthermore, we impose the constraints that value $\alpha_{n+1} = 1$ must be assigned to point $\beta_1 = 1$, and that value $\alpha_{n+2} = 1$ must be assigned to point $\beta_{n+2} = n$. These additionally imposed constraints take care of the special treatment of the two outermost backbone vertices v_1 and v_n.

Now let us verify that the machinery of Sections 3 and 4 still can be applied to this variant of the Wiener Max-QAP. First of all, the decomposition property works out exactly as before. Also Theorem 3.2 continues to hold, since adding $\pi(1) = n+1$ and $\pi(n+2) = n+2$ to a V-shaped permutation $\pi(2), \ldots, \pi(n+1)$ of the numbers $1, \ldots, n$ always yields a V-shaped permutation. The dynamic program in Section 4 needs some small cosmetic changes that are caused by the additionally imposed constraints.

- All states (k, m, L, R) with $k \le n$ must satisfy $2 \le m \le n-k+2$ and $L, R \ge 1$.
- All states $(n+1, m, L, R)$ must satisfy $m = 1$, $L = 0$, and $R = 1$.
- All states $(n+2, m, L, R)$ must satisfy $m = 1$ and $L = R = 0$.

These conditions ensure that the dynamic program assigns the values α_{n+1} and α_{n+2} during the last two stages to points β_1 and β_{n+2} exactly as desired.

Furthermore, we note that in equations (13) and (14) we always have $|\beta_{m+1} - \beta_m| = 1$ and $|\beta_{m+k-1} - \beta_{m+k-2}| = 1$ for $k \le n$, and $|\beta_{m+1} - \beta_m| = 0$ and $|\beta_{m+k-1} - \beta_{m+k-2}| = |\beta_k - \beta_{k-1}|$ that for $k > n$. This implies that neither the recursive computations nor the values $Z(k, m, L, R)$ do depend on the second coordinate m, which consequently may be dropped. (This should also be clear intuitively, since this second coordinate encodes the piece of the backbone to which the first k values $\alpha_1, \ldots, \alpha_k$ are assigned. All backbone pieces of length k are paths on k vertices and thus have the same combinatorial structure.)

What about the time complexity? Exactly as in the proof of Theorem 4.1 the time complexity is proportional to the number of states (k, L, R). Since k can take $O(r)$ possible values, and since L can take $O(\sum \alpha_i) = O(r)$ possible values, the time complexity is $O(r^2)$.

Theorem 5.2 *The problem of finding a tree that maximizes the Wiener index among all trees with a prescribed degree sequence can be solved in quadratic time $O(r^2)$, where r denotes the overall number of terms in the degree sequence.* $\square$

6 Conclusions

We have introduced the Wiener Max-QAP, a special case of the quadratic assignment problem. We have provided a complete picture of the computational complexity of this special case: It is NP-hard in the ordinary sense, and it is solvable in pseudo-polynomial time. Our investigations also gave us a polynomial time algorithm for finding a tree that maximizes the Wiener index among all trees with a prescribed degree sequence, thereby settling a prominent open problem from chemical graph theory.

One obvious open problem is to bring the quadratic time complexity $O(r^2)$ in Theorem 5.2 down to $O(r \log r)$ or perhaps even down to linear time $O(r)$.

Another open problem concerns the Wiener Min-QAP, where the goal is to *minimize* the objective value in (4). It is an easy exercise to rewrite and to adapt the results of Sections 3 and 4 to the minimization version: The minimization version always has an optimal solution that is *pyramidal* (which means that the permutation is first increasing up to some value ℓ, and then decreasing from ℓ onwards). And the minimization problem can be solved by dynamic programming in pseudo-polynomial time, within the same time complexity as that in Theorem 4.1. The main gap in our knowledge concerns the complexity of the Wiener Min-QAP, and we pose the open problem of deciding whether it actually is NP-hard.

Acknowledgements

Gerhard Woeginger acknowledges support by the Netherlands Organization for Scientific Research (NWO), grant 639.033.403, by DIAMANT (an NWO mathematics cluster), and by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society).

We thank Stephan Wagner (Stellenbosch University) for bringing the problem of maximizing the Wiener Index of a tree to our attention.

References

[1] T.C. Koopmans, M.J. Beckmann, Assignment problems and the location of economic activities, *Econometrica 25* (1957) 53–76.

[2] E. Çela, *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[3] R.E. Burkard, M. Dell'Amico, S. Martello, *Assignment Problems*, SIAM, Philadelphia, 2009.

[4] R.E. Burkard, E. Çela, G. Rote, G.J. Woeginger, The quadratic assignment problem with a monotone anti-Monge and a symmetric Toeplitz matrix: Easy and hard cases, *Mathematical Programming B82* (1998) 125–158.

[5] V.G. Deineko, G.J. Woeginger, A solvable case of the quadratic assignment problem, *Operations Research Letters 22* (1998) 13–17.

[6] H. Wiener, Structural determination of paraffin boiling points, *Journal of the American Chemical Society 69* (1947) 17–20.

[7] A.A. Dobrynin, R. Entringer, I. Gutman, Wiener index of trees: theory and applications, *Acta Applicandae Mathematicae 66* (2001) 211–249.

[8] R.C. Entringer, D.E. Jackson, D.A. Snyder, Distance in graphs, *Czechoslovak Mathematical Journal 26* (1976) 283–296.

[9] M. Fischermann, A. Hoffmann, D. Rautenbach, L.A. Székely, L. Volkmann, Wiener index versus maximum degree in trees, *Discrete Applied Mathematics 122* (2002) 127–137.

[10] H. Wang, The extremal values of the Wiener index of a tree with given degree sequence, *Discrete Applied Mathematics 156* (2008) 2647–2654.

[11] X.D. Zhang, Q.Y. Xiang, L.Q. Xu, R.Y. Pan, The Wiener index of trees with given degree sequences, *MATCH Communications in Mathematical and in Computer Chemistry 60* (2008) 623–644.

[12] H. Wang, Corrigendum: The extremal values of the Wiener index of a tree with given degree sequence, *Discrete Applied Mathematics 157* (2008) 3754.

[13] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[14] R.H. Shi, The average distance of trees, *Systems Science and Mathematical Sciences 6* (1993) 18–24.

[15] N.S. Schmuck, The Wiener index of a graph, Master thesis, TU Graz, Austria, 2010.