ELSEVIER

Discrete Mathematics 192 (1998) 27–39

DISCRETE MATHEMATICS

Distances in benzenoid systems: Further developments

Victor Chepoi[a,*], Sandi Klavžar[b]

[a] *Laboratoire de Biomathématiques, Université d'Aix Marseille II, 27 Bd Jean Moulin, F-13385 Marseille Cedex 5, France*

[b] *Department of Mathematics, PEF, University of Maribor, Koroška cesta 160, 2000 Maribor, Slovenia*

Received 3 February 1997; revised 9 September 1997; accepted 17 September 1997

Abstract

In this note we present some new results on distances in benzenoids. An algorithm is presented which, for a given benzenoid system G bounded by a simple circuit Z with n vertices, computes the Wiener index of G in $O(n)$ time. Also we show that benzenoid systems have a convenient dismantling scheme, which can be derived by applying breadth-first search to their dual graphs. Our last result deals with the clustering problem of sets of atoms of benzenoids systems. We show how the k-means clustering algorithm (for points in Euclidean space) can be efficiently implemented in the case of benzenoids. © 1998 Elsevier Science B.V. All rights reserved

1. Introduction

Distance properties of molecular graphs form an important topic in chemical graph theory [27]. To justify this statement just recall the famous Wiener index which is also known as the Wiener number. This index is the first [28] but also one of the most important topological indices of chemical graphs. Its research is still very active, see recent reviews [16,23] and several new results in a volume [15] dedicated to the 50th anniversary of Wiener's paper [28].

Benzenoid systems form one of the most important class of chemical graphs [12]. Recently, Klavžar et al. [21] have shown that benzenoid systems provide so-called isometric embeddings into hypercubes and based on this fact a simple formula for the Wiener index of these graphs has been obtained. The approach was further developed in the subsequent papers [13,14]. Along these lines it was recently [5] observed that benzenoid systems can also be isometrically embedded into the Cartesian

[*] Correspondence address: SFB 343 Diskrete Strukturen in der Mathematik, Universität Bielefeld, Postfach 100131, D-33501 Bielefeld, Germany.

PII S0012-365X(98)00064-8

product of three trees. As an application it was demonstrated how to compute the diameter of a benzenoid system in optimal time. Using the results from [5,21] in [7] a simple and practical algorithm for computing the Wiener index of a benzenoid systems has been proposed (its complexity is linear in the number of vertices of a benzenoid).

Here we present some new results on distances in benzenoids. It is the main purpose of this note to give an optimal time algorithm for computing the Wiener index of a benzenoid. Also we show that benzenoid systems have a convenient dismantling scheme, which can be derived by applying breadth-first search to their dual graphs. Our third result deals with the clustering problem of sets of atoms of benzenoids systems. We show how the k-means clustering algorithm (for points in Euclidean space) can be efficiently implemented in the case of benzenoids.

2. Isometric embeddings of benzenoid systems

In this section we recall the results from [5,21] on embedding of benzenoid systems in hypercubes and products of trees.

Benzenoid systems (alias *benzenoid graphs* or *hexagonal systems*) are graphs constructed in the following manner [12]. Let H be the infinite hexagonal lattice and let Z be a circuit on it. Then a benzenoid system G is formed by the vertices and edges of H, lying on Z and in the interior of the region bounded by Z. Throughout in this note we assume that N denote the number of vertices of G and n the number of vertices on the circuit Z. The vertex set of G is denoted by $V(G)$. In a graph G the *length* of a path from a vertex v to a vertex u is the number of edges in the path. The *distance* $d_G(u,v)$ from u to v is the length of a shortest path connecting u and v.

Given two connected graphs G and H, we say that G admits an *isometric embedding* (alias distance-preserving embedding) into H if there exists a mapping

$$\beta : V(G) \to V(H)$$

such that

$$d_H(\beta(u), \beta(v)) = d_G(u,v),$$

for all vertices $u, v \in V(G)$.

The *Cartesian product* $H = H_1 \times \cdots \times H_m$ of connected graphs $H_1, \ldots, H_m$ is the graph on the vertex set

$$V(H) = \{u = (u_1, u_2, \ldots, u_m) \colon u_i \in V(H_i),\ i = 1, \ldots, m\}.$$

Two vertices $u = (u_1, u_2, \ldots, u_m)$ and $v = (v_1, v_2 \ldots, v_m)$ of H are adjacent if and only if the vectors u and v coincide in all but one position i, where u_i and v_i are adjacent in H_i. The distance between two vertices $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ of H

is given by

$$d_H(x,y) = \sum_{i=1}^{m} d_{H_i}(x_i, y_i).$$

For example, if each factor H_i is a K_2 (the connected two-vertex graph with $V(K_2) = \{0,1\}$), then H is just the m-cube (binary Hamming graph according to [21, 18]) which is equipped with the *Hamming distance* for which the distance between two binary m-tuples is equal to the number of coordinate positions in which they differ.

Proposition 1 (Klavžar, Gutman and Mohar [21]). *Every benzenoid system G has an isometric embedding into a hypercube.*

The dimension of the cube in which G embeds can be arbitrarily large (actually, it is equal to half the length of the bounding circuit Z). Instead of isometric embeddings of benzenoids into binary Hamming graphs [5] proposes such embeddings into the Cartesian product of trees (note that the graphs isometrically embeddable into products of two trees have been characterized in [3]). The main advantage is that independently of the size or of the form of the benzenoid G, there exists an isometric embedding of G into the Cartesian product of only three trees T_1, T_2, and T_3. Each of these factors is uniquely determined by parallel cuts of a given direction of G. Since we will use this embedding in the next section, we present it in more details.

Let G be a benzenoid system and let E_1, E_2, and E_3 denote the edges of G of a given direction. A *basic direction* is a direction orthogonal to one of the three edge directions of the haxagonal grid. Denote the basic directions by f_1, f_2, f_3. For $i = 1,2,3$, let G_i be the graph which is obtained from G be deleting all the edges of E_i. Note that the connected components of the graph G_i are paths. One can easily show that every such path is the unique shortest path in G between its end-vertices. Define a graph T_i whose vertices are the connected components of G_i and where two such components P' and P'' are adjacent in T_i if and only if there are vertices $u \in P'$ and $v \in P''$ which are end-vertices of an edge from E_i (see Figs. 1 and 2 for an illustration). Since G is bounded by a Jordan curve Z, every T_i is a tree (the existence of a cycle in T_i would imply that G contains a non-hexagonal interior face). This yields to the following canonical emdedding α of G into the Cartesian product $H = T_1 \times T_2 \times T_3$. For any vertex v of G put

$$\alpha(v) = (P, Q, R),$$

where P, Q, and R are the connected components of the graphs G_1, G_2, and G_3, respectively, sharing the vertex v. Moreover, as is shown in [5], α provides an isometric embedding of G into H.

To label the vertices of G one can proceed as follows. First we find the edges from each E_i, $i = 1,2,3$ (this can be done while a usual representation of G as a doubly linked list is given) and the connected components of the graph G_i. After their labeling

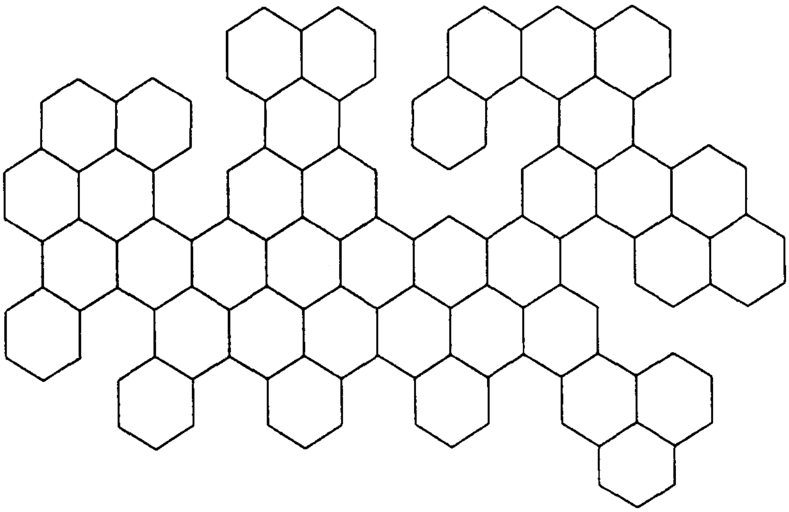

Fig. 1. A benzenoid system G.

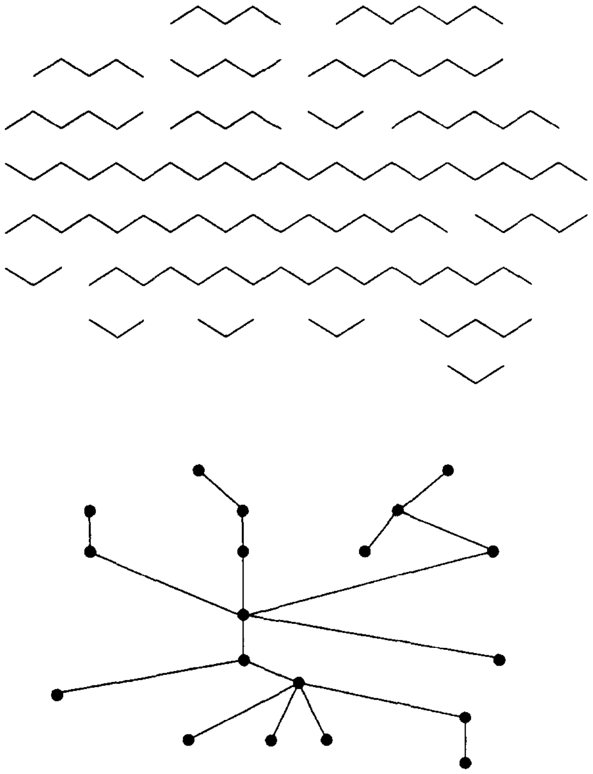

Fig. 2. G_i and T_i.

we can find the required incidence relation between them (i.e., to define the tree T_i). The ith coordinate ($i = 1,2,3$) of a vertex v of G is the label of the connected component of G_i from which v is taken. If G contains N vertices, then all these computations can be done in total $\mathrm{O}(N)$ time. The output of this algorithm consists of the trees T_1, T_2, T_3 and the labels of length three of the vertices of G. With this compact labeling of G one can work much as with points in three-dimensional space (see Section 5 for an illustration). Concluding, we obtain the following result (since trees are isometrically embeddable into hypercubes this also proves Proposition 1).

Proposition 2 (Chepoi [5]). *The cannonical embedding α provides an isometric embedding of a benzenoid system G with N vertices into the graph $H = T_1 \times T_2 \times T_3$. The trees T_1, T_2, T_3 and the corresponding labels of the vertices of G can be computed in total $\mathrm{O}(N)$ time.*

One can easily notice that during the whole construction of the tree-factors T_1, T_2, and T_3 or of labels of the vertices of G we never used any distance information. However, using this structure, after an $\mathrm{O}(N)$ time precomputing, one can answers in $\mathrm{O}(1)$ time per query questions of the form, 'What is the distance between the vertices u and v of G ?' Indeed, Proposition 2 reduces the problem of finding $d_G(u,v)$ to three similar problems on tree-factors, where we can use the algorithm of [17] for computing nearest common ancestors. Recall, that given a tree T rooted at r, the *nearest common ancestor* $nca(x,y)$ of two vertices of x and y of T is the root of the smallest subtree of T that contains both vertices x and y. Harel and Tarjan [17] presented an algorithm for computing $nca(x,y)$ of two given vertices in $\mathrm{O}(1)$ time. Since

$$d_T(x,y) = d_T(x,r) + d_T(y,r) - 2d_T(r, nca(x,y)),$$

we can find the distance between x and y in constant time. This has been used in [5] to compute the diameter of a benzenoid system in linear time.

3. Wiener index of benzenoid systems

The *Wiener index* or *Wiener number* of a (molecular) graph $G = (V,E)$ is defined as follows:

$$W(G) = \frac{1}{2} \sum_{u \in V} \sum_{v \in V} d_G(u,v).$$

First we recall the formula from [21] for computing the Wiener index of a benzenoid G with N vertices, if an isometric embedding of G into a cube of dimension q is given. Let N_i be the number of vertices v of G such that the ith coordinate of the label of u is equal to 1.

Proposition 3 (Klavžar et al. [21]). *The Wiener index of G is equal to*

$$W(G) = \sum_{i=1}^{q} N_i(N - N_i).$$

Since q is usually much smaller than the number of pairs $\{u, v\}$, this formula significantly simplifies finding the Wiener index of a benzenoid. In many cases it immediately produces the formula for computing $W(G)$. Using this formula combinatorial expressions for the Wiener index of compact pericondensed benzenoid hydrocarbons were given in [14]. However, to obtain a linear algorithm for computing the Wiener index we still need a similar formula for $W(G)$ in the case when the canonical embedding α is given. For this we next recall a concept of the Wiener index on weighted trees [20]. A *(vertex)-weighted tree* (T, π) is a tree T together with a function $\pi : V(T) \to \mathbb{N}^+$. The Wiener number $W(T, \pi)$ of (T, π) is defined as

$$W(G, w) = \frac{1}{2} \sum_{u,v \in V(G)} \pi(u)\pi(v)d_T(u, v).$$

Let G be a benzenoid system and let T_1, T_2, T_3 be the trees from the canonical embedding. For $i = 1, 2, 3$ we introduce weighted trees (T_i, π_i) as follows: for $u \in T_i$ let $\pi_i(u)$ be the number of vertices x of G such that the ith component of $\alpha(x)$ is u. In other words, $\pi_i(u)$ is just the number of vertices in the connected component of G_i which corresponds to the vertex u.

Proposition 4 (Chepoi and Klavžar [7]). *Let G be a benzenoid system and let α be the canonical embedding of G into $T_1 \times T_2 \times T_3$. Then*

$$W(G) = W(T_1, \pi_1) + W(T_2, \pi_2) + W(T_3, \pi_3).$$

By Proposition 4, an algorithm with complexity $O(N)$ for computing $W(G)$ will be provided by a linear algorithm for computing the Wiener index of a weighted tree. It is mainly the same as the linear algorithm for computing the Wiener index of an unweighted tree obtained by Mohar and Pisanski [22]. Its implementation is based on the following straightforward property (for a generalization see [20]).

Lemma 5. *Let (T, π) be a weighted tree. For an edge e of T, let T_1 and T_2 be the connected components of $T \setminus e$ and for $i = 1, 2$ set*

$$n_i(e) = \sum_{u \in T_i} \pi(u).$$

Then we have

$$W(T, w) = \sum_{e \in T} n_1(e)n_2(e).$$

Let (T,π) be a weighted tree. To find $W(T,\pi)$ we order the vertices of T so that the next vertex v is a leaf in the subtree induced by the vertices with a larger index. Let u be the neighbour of v in this subtree. Then add the factor $\pi(v)(m-\pi(v))$ to the current sum and update $\pi(u)$ by letting $\pi(u):=\pi(u)+\pi(v)$. Note finally that using Proposition 2 it is easy to obtain the weighted trees T_i in linear time $O(N)$. Therefore, $W(G)$ can be computed in $O(N)$ time [7]. The algorithm is very simple and efficient for a manual calculation, but it is not optimal, because the lower bound for this problem is $\Omega(n)$, where n is the number of vertices on the bounding circuit Z of G.

Now, we are going to present an optimal $O(n)$ time algorithm for computing $W(G)$. The main idea is that we can construct the weighted trees $(T_1,\pi_1),(T_2,\pi_2)$ and (T_3,π_3) without an explicit definiton of the facial structure of G. Namely, we need as an input the bounding circuit Z of G given in the form of a circular list (so, we can consider Z as a simple polygon). Let $\mathscr{D}$ denote the region of the plane bounded by Z. The algorithm is based on the Chazelle algorithm [4] for computing all vertex-edge visible pairs of edges of a simple polygon with n vertices. Recall, that by this algorithm in optimal $O(n)$ time one can obtain a decomposition of the $\mathscr{D}$ into strips (alias trapezoids), using parallel cuts of a given direction which pass through the vertices of Z. A straight line segment $[p,q]$ of one of the basic directions f_i is called a *cut segment* if $p,q\in Z$ and $[p,q]$ belongs to the region $\mathscr{D}$ bounded by Z.

Applying the algorithm of Chazelle separately for each of the basic directions $f_1,f_2,$ f_3, we find the subdivisions $\mathscr{D}_1$, $\mathscr{D}_2$ and $\mathscr{D}_3$ of the region $\mathscr{D}$ into strips; see Fig. 2 for an illustration (some of strips can represent triangles). Namely, every $\mathscr{D}_i$ is returned in the usual representation as a doubly linked list. (We can consider $\mathscr{D}_i$ as a planar graph with strips as interior faces and the vertices of Z as the vertex-set.) Let $\mathscr{C}_i$ the cuts of the ith direction participating in the subdivision $\mathscr{D}_i$. In $\mathscr{C}_i$ we also include the vertices of Z where both incident edges do not belong to E_i (they can be viewed as degenerated cuts). Define a new graph Γ_i whose vertices are the cuts of $\mathscr{C}_i$ and two vertices of Γ_i are adjacent if and only if they belong to a common strip of $\mathscr{D}_i$. One can easily show that each Γ_i is a tree, which can be derived from $\mathscr{D}_i$ in $O(n)$ time.

The width of strips of $\mathscr{D}_i$ takes only two values 1 and $\frac{1}{2}$. With some abuse of language, we will call an edge of Γ_i *thick* if it is defined by a strip of width 1 and *thin* if the corresponding strip has width $\frac{1}{2}$. Every cut of $\mathscr{C}_i$ is incident in Γ_i to exactly one thick edge, all remaining vertices of $\mathscr{C}_i$ being incident only to thin edges. If we remove the thick edges of Γ_i, we will get the connected subgraphs of Γ_i spanned by thin edges (we will call them thin components). Every thin component of Γ_i has the same vertices of G as some connected component of the graph G_i. In other words, if we contract all thin edges of $\mathscr{C}_i$ we will obtain the tree T_i.

Therefore, to compute the Wiener index of the weighted tree (T_i,π) one can proceed as follows. For each $c\in\mathscr{C}_i$ we compute its length l_c. Every cut c of $\mathscr{C}_i$ (degenerated or not) has exactly $(l_c/\sqrt{3})+1$ vertices of the benzenoid system G. Define $\pi'(c)=(l_c/\sqrt{3})+1$. Then $\sum_{c\in\mathscr{C}_i}\pi'(c)=N$. To find $W(T_i,\pi)$ we order the vertices of Γ_i so that the next vertex c is a leaf in the subtree induced by the vertices with a larger index. Let c^+ be the neighbour of c in this subtree. If the edge (c,c^+) is thin, then

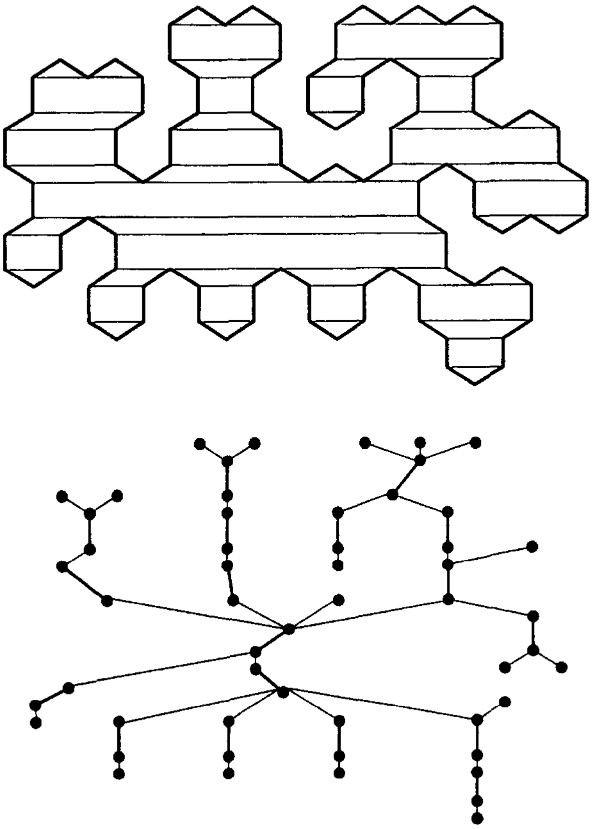

Fig. 3. $\mathscr{D}_i$ and Γ_i.

put $\pi'(c^+) := \pi'(c^+)+\pi'(c)$. Otherwise, if the edge (c,c^+) is thick, then add the factor $\pi'(c)(N-\pi'(c))$ to the current sum and update $\pi'(c^+)$ by letting $\pi'(c^+) := \pi'(c^+)+\pi'(c)$. The resulting sum will be $W(T_i,\pi_i)$. By Lemma 4 $W(G)=\sum_{i=1}^{3} W(T_i,\pi_i)$. The trees Γ_i and the weight functions π_i can be derived in total $O(n)$ time. Therefore, $W(G)$ can be computed within the same time bounds. Summarizing, we obtain the following result.

Proposition 6. *The Wiener index $W(G)$ of a benzenoid system G bounded by a circuit Z with n vertices can be computed in optimal time* $O(n)$.

4. Dismantling benzenoid systems

We say that a face F of a benzenoid system G is *pendant* in G if it includes an edge, both endvertices of which have degree 2 in G (this is a particular instance of a more general definition of a pendant cycle given in [2]). Removing this *pendant edge*

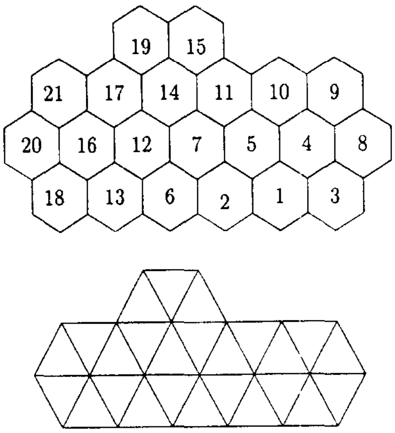

Fig. 4. G, G^* and a dismantling scheme of G^*.

from G then results in an isometric subgraph G' of G. However, G' is not necessarily a benzenoid system again. Next we will show that every benzenoid system with at least two faces has a pendant face, such that the subgraph induced by all other faces is again a benzenoid system.

A *dismantling scheme* of a benzenoid G is a linear order $F_1,\dots,F_m$ of its faces such that any F_i is a pendant face in the subgraph G_i induced by the union of faces $F_1,F_2,\dots,F_i$ and all $G=G_m,G_{m-1},\dots,G_1$ are benzenoid systems.

Let G^* denote the (inner) dual graph of a benzenoid system G. The hexagonal faces of G correspond to the vertices of G^* and two vertices of G^* are adjacent if and only if the corresponding faces share an edge of G. The graph G^* can be viewed as a subgraph of the triangular grid T (a regular tiling of the plane into triangles); see Fig. 4 for an illustration. Every vertex of T has three pairs of opposite neighbours, each of them defining a *basic line*, i.e., a line of one of three basic directions of the initial hexagonal grid H.

Lemma 7. *Let G be a benzenoid system. Then its inner dual graph G^* does not contain isometric cycles of length larger than* 3.

Proof. Assume that G^* has an isometric cycle C of length larger than 3. Let $\mathcal{R}$ be the region of the plane bounded by the circuit C. Pick a vertex x of C, and let y and z be the neighbours of x in C. Since y and z are not adjacent, there exists a basic line L, which passess through x and separates the vertices y and z (namely, y and z belong to different open halfplanes defined by L). This line intersects T along a convex set, i.e., for any vertices $u,v\in L\cap \mathrm{T}$ and $w\in \mathrm{T}$, the equality $d_{\mathrm{T}}(u,v)=d_{\mathrm{T}}(u,w)+d_{\mathrm{T}}(w,v)$ implies that w belongs to the segment $[uv]$. The line L intersects the cycle C and all intersection points are vertices of G^*. Let $t\neq x$ be the vertex of $C\cap L$ such that $[xt]\subset \mathcal{D}$. Such a vertex necessarily exists: moving along L we enter $\mathcal{R}$ in x and then we must exits $\mathscr{D}$ somewhere. Since C is a cycle of G^*, all vertices of $\mathrm{T} \cap L$ belong to G^*. Moreover, they induce the unique shortest path of T (and G^*) between $x, t \in C$. Since the vertices $y, z \in C$ do not belong to this path, we obtain a contradiction with the assumption that C is an isometric cycle of G^*. □

The graphs which do not contain isometric cycles of length greater than 3 are called *bridged* [9,25]. Anstee and Farber [1] established that any bridged graph Γ has a following cop-win ordering: *the vertices of Γ can be linearly ordered, $v_1, v_2, \ldots, v_n$, so that, for each v_i, $i > 1$, there is a neighbour v_j, $j < i$, of v_i, such that every vertex v_k, $k < i$, adjacent to v_i is also adjacent to v_j.* In [6] it is shown that *any ordering of the vertices of a bridged graph Γ produced by breadth-first search is a cop-win ordering.* This implies the following result.

Proposition 8. *Let G be a benzenoid system. Any ordering of the vertices of G^* produced by the breadth-first search is a dismantling scheme of G.*

Proof. Let $F = (u, v, w, x, y, z)$ be the last face of G in the breadth-first search ordering of G^*. It is sufficient to establish that F is a pendant face of G and that the subgraph G' of G induced by all other faces is again a benzenoid. By the result of [6] there is face F' incident with F and all other faces incident to F. This implies that F is incident with at most two other faces F_1 and F_2. If F and F' intersect along the edge (x, y), then the edge (u, v) opposite to (x, y) is pendant. Thus F is a pendant face. If F is incident only with F', then F has three pendant edges $(z, u), (u, v)$ and (v, w). Removing them we obtain again a benzenoid. If F is incident with F' and F', then F has two pendant edges (u, v) and (v, w). Then (z, u, v, w, x) is a subpath of the boundary circuit Z. Replacing in Z this subpath by (z, y, x) we obtain a new circuit Z'. Since Z' is the boundary of G', we deduce that G' is a benzenoid. Finally, if F has three neighbour faces, then Z enters F through z and exits this face through the vertex w. Replacing in Z the subpath (z, u, v, w) by (z, y, x, w) we will get a new circuit Z'. Again Z' bounds the subgraph G', i.e. G' is a benzenoid. □

5. Clustering in benzenoid systems

In this section we will adjust the well-known k-means algorithm for clustering points in Euclidean space to produce a clustering of a set of atoms (vertices) of a benzenoid system G.

The main problem in clustering consists in sorting a set $X = \{x_1, \ldots, x_r\}$ of objects into a number k of homogeneous clusters. The points in the same cluster should be as close (similar) as possible. The objects in different clusters should be as distant (dissimilar) as possible. Any method of clustering needs the formalization of such notions as a quality of the partition and the prototype of a class of the partition. Let $P = \{P_1, \ldots, P_k\}$ be a partition of X into k classes. In case when the objects of X are points in Euclidean space $\mathbb{R}^m$ the majority of the clustering methods use the dispersion function as a quality of a partition P:

$$D(P) = \sum_{i=1}^{k} \sum_{x_j \in P_i} d^2(x_j, L_i). \tag{1}$$

In this formula $L_i = (L_i^1, \ldots, L_i^m)$ is the gravity center of the class P_i, and $d(x, y) = \left(\sum_{s=1}^{m}(x^s - y^s)^2\right)^{1/2}$ is the Euclidean distance between the points $x = (x^1, \ldots, x^m)$ and $y = (y^1, \ldots, y^m)$.

The problem of finding a partition of X minimizing the dispersion function D is known to be NP-complete. Instead, some simple procedures produce partitions of X which are local minima of the function D. One of the best known of them is the k-means algorithm (see the books of Duda and Hart [8] and Jain and Dubes [19]). This algorithm starts with an arbitrary partition P' of X into k classes. Then find the gravity centers of the classes of this partition. In the next iteration each object x_i is moved to the class with gravity center closest to x_i. Denote the obtained partition by P''. If $P' = P''$, then stop, otherwise let $P' := P''$ and repeat the same procedure. The formal description of this algorithm is given below.

k-means algorithm

```
1. Choose any initial partition $P' = \{P'_1, \ldots, P'_k\}$ of $X$ into $k$ classes.
2. Compute the prototypes $L_1, \ldots, L_k$ of this partition.
3. Construct the metric partition $P'' = \{P''_1, \ldots, P''_k\}$, where
       $P''_1 = \{x_i \in X: d(x_i, L_1) = \min_{1 \leqslant j \leqslant k} d(x_i, L_j)\}$,
       $\vdots$
       $P''_k = \{x_i \in X: d(x_i, L_k) = \min_{1 \leqslant j \leqslant p} d(x_i, L_j)\}$.
4. If $P'' \neq P'$, then set $P' := P''$ and go to step 2. Else stop.
```

Returning to benzenoid systems, the following clustering problem can be formulated. Let $G = (V, E)$ be a benzenoid system with N vertices endowed with the standard distance d_G and let $X = \{x_1, \ldots, x_r\}$ be a set of atoms (vertices) of G. We wish to sort the atoms of X into k 'homogeneous' clusters according to the dispersion function D defined above. For this we can use the canonical embedding of G into the Cartesian product of three trees T_1, T_2 and T_3. Using this, the distance $d_G(x, y)$ between two given vertices $x, y \in V$ can be calculated in O(1) time. This immediately leads to efficient implementations of steps 2 and 3 of the k-means algorithm, provided the prototypes of classes are selected among vertices of G. For this we simply compute $D_i(v) = \sum_{x_j \in P_i} d_G^2(v, x_j)$ for each vertex v of G and each class P_i of the partition P of X. A prototype of a class P_i is a vertex v of G which minimizes $D_i(v)$.

However, the fact that G is an isometric subgraph of $T_1 \times T_2 \times T_3$ allows to define the prototypes of classes not in G but in a larger geometric space, in analogy to clustering procedures in $\mathbb{R}^m$. Namely, let $\mathscr{T}_s$ be a *tree-network* obtained by replacing each edge of T_s by a (solid) segment of unit length. Since any two points of $\mathscr{T}_s$ are connected in $\mathscr{T}_s$ by a unique path, its length can be regarded as the distance between selected points. The resulting metric $d_{\mathscr{T}_s}$ on $\mathscr{T}_s$ extends the metric d_{T_s}. Define $\Pi = \mathscr{T}_1 \times \mathscr{T}_2 \times \mathscr{T}_3$. Then Π is a cell complex whose cells are 3-dimensional cubes. Π can be endowed with a distance d of Euclidean type. Namely, if $x, y \in \Pi$ and $x = (x^1,x^2,x^3)$, $y = (y^1,y^2,y^3)$ with $x^s, y^s \in \mathscr{T}_s$, then define

$$d^2(x,y) = d^2_{\mathscr{T}_1}(x^1,y^1) + d^2_{\mathscr{T}_2}(x^2,y^2) + d^2_{\mathscr{T}_3}(x^3,y^3).$$

Now, we are searching for a partition P of $X = \{x_1,\dots,x_r\}$ into k classes which minimizes (at least locally) the dispersion function

$$D(P) = \sum_{i=1}^{k} \sum_{x_j \in P_i} d^2(x_j, L_i).$$

The prototypes $L_1,\dots,L_k$ of classes are selected among the points of the polyhedron Π. To implement the k-means algorithm, for a given partition $P = \{P_1,\dots,P_k\}$ of X we have to compute the prototypes $L_1,\dots,L_k$ of all classes. Suppose that $P_i = \{x_{i_1},\dots,x_{i_{p_i}}\}$, and it is necessary to compute $L_i = (L_i^1, L_i^2, L_i^3)$, $i = 1,\dots k$. As in the case of points in Euclidean space, L_i^s will be the gravity center in $\mathscr{T}_s$ of the vertices $x^s_{i_1},\dots,x^s_{i_{p_i}}$ of T_s. It is known [10,24,26] that each collection of vertices in a tree-network has a unique gravity center, which can be computed in time linear in the number of vertices of the generating tree [10,26].

Below we outline the algorithm communicated to us by A. Tamir [26]. Let $\mathscr{T}$ be a tree-network generated by a tree T with N vertices and let $v_1,\dots v_p$ be some vertices of T. We wish to find a point x of $\mathscr{T}$ minimizing the (convex) function $F(x) = \sum_{i=1}^{p} d^2_{\mathscr{T}}(x, v_i)$. For each vertex v it takes $O(N)$ time to find the directional derivatives of F with respect to all edges incident to v (see [24] for definition and properties of directional derivatives). It is known that the gravity center lies on an edge incident to a vertex v such that $F(v) \leqslant F(u)$ for all vertices u of T. It is now sufficient to show how to find such a vertex v in linear time. The algorithm is recursive. Find a centroid (simple median) of the tree T, say vertex m; this can be done in linear time by the algorithm of Goldman [10]. By computing the directional derivatives of F at edges incident to m we will find the connected component of T (obtained by removing m), which contains the optimum. Let T' denote the subtree induced by m and the vertices in the above component. By definition of the centroid, T' as well as every other component contains at most $N/2$ vertices. We continue the same procedure with the tree T'. It is clear that the entire process takes $O(N)$ time, since at each iteration we spend $O(n)$ time and reduce the number of vertices by a factor of 2.

To compute L_i^1, L_i^2, L_i^3 it is necessary to apply to each factor the algorithm described above. Note that in general the gravity center of some vertices is not located in a vertex of a tree-network. As a consequence, the prototype of a class can be an interior point of the polyhedron Π. Concluding, we obtain that for a benzenoid system G with N vertices an iteration of the k-means algorithm can be performed in $O(kN)$ time.

Acknowledgements

The work of one author (S.K.) was supported in part by the Ministry of Science and Technology of Slovenia under the grant J1-7036.

References

[1] R.P. Anstee, M. Farber, On bridged graphs and cop-win graphs, J. Combin. Theory Ser. B 44 (1988) 22–28.

[2] H.-J. Bandelt, V. Chepoi, Cellular bipartite graphs, European J. Combin. 17 (1996) 121–134.

[3] H.-J. Bandelt, V. Chepoi, Generalized rectilinear polygons, in preparation.

[4] B. Chazelle, Triangulating a simple polygon in linear time, Discrete Comput. Geom. 6 (1994) 485–524.

[5] V. Chepoi, On distances in benzenoid systems, J. Chem. Inform. Comput. Sci. 36 (1996) 1169–1172.

[6] V. Chepoi, Bridged graphs are cop-win graphs: an algorithmic proof, J. Combin. Theory Ser. B 69 (1997) 97–100.

[7] V. Chepoi, S. Klavžar, The Wiener index and the Szeged index of benzenoid systems in linear time, J. Chem. Inform. Comput. Sci. 37 (1997) 752–755.

[8] R. Duda, P. Hart, Pattern Classification and Scene Analysis (Wiley Interscience, New York, 1973).

[9] M. Farber, R.E. Jamison, On local convexity in graphs, Discrete Math. 66 (1987) 231–247.

[10] A.J. Goldman, Optimal center location in simple networks, Transport. Sci. 5 (1971) 212–221.

[11] R.L. Graham, P.M. Winkler, On isometric embeddings of graphs, Trans. Amer. Math. Soc. 288 (1985) 527–536.

[12] I. Gutman, S.J. Cyvin, Introduction to the Theory of Benzenoid Hydrocarbons, Springer, Berlin, 1989.

[13] I. Gutman, S. Klavžar, An algorithm for the calculation of the Szeged index of benzenoid hydrocarbons, J. Chem. Inform. Comput. Sci. 35, (1995) 1011–1014.

[14] I. Gutman, S. Klavžar, A Method for Calculating Wiener Numbers of Benzenoid Hydrocarbons, Models in Chemistry (Acta Chim. Hung.), in press.

[15] I. Gutman, S. Klavžar, B. Mohar, (Eds.), Fifty Years of the Wiener Index, MATCH vol. 35, 1997, in press.

[16] I. Gutman, Y.N. Yeh, S.L. Lee, Y.L. Luo, Some recent results in the theory of the Wiener number, Indian J. Chem. 32A (1993) 651–661.

[17] D. Harel, R.E. Tarjan, Fast algorithms for finding nearest common ancestors, SIAM J. Comput. 13 (1984) 338–355.

[18] W. Imrich, S. Klavžar, On the complexity of recognizing Hamming graphs and related classes of graphs, European J. Combin. 17 (1996) 209–221.

[19] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, NJ, 1988.

[20] S. Klavžar, I. Gutman, Wiener number of vertex-weighted graphs and a chemical application, submitted.

[21] S. Klavžar, I. Gutman, B. Mohar, Labeling of benzenoid systems which reflects the vertex-distance relations, J. Chem. Inform. Comput. Sci. 35 (1995) 590–593.

[22] B. Mohar, T. Pisanski, How to compute the Wiener index of a graph, J. Math. Chem. 2 (1988) 267–277.

[23] S. Nikolić, N. Trinajstić, Z. Mihalić, The Wiener index: development and applications, Croat. Chem. Acta 68 (1995) 105–129.

[24] D.R. Shier, P.M. Dearing, Optimal locations for a class of nonlinear, single-facility location problem on a network, Oper. Res. 31 (1983) 292–303.

[25] V. Soltan, V. Chepoi, Conditions for invariance of set diameters under *d*-convexification in a graph, Cybernetics 19 (1983) 750–756.

[26] A. Tamir, personal communication.

[27] N. Trinajstić, Chemical Graph Theory, CRC Press, Boca Raton, FL, 1993.

[28] H. Wiener, Structural determination of paraffin boiling points, J. Amer. Chem. Soc. 69 (1947) 17–20.