On distances in benzenoid systems

Victor Chepoi [1]

Laboratoire de Biomathématiques
Faculté de Medecine,
Université d'Aix Marseille II
27 Bd Jean Moulin
F–13385 Marseille Cedex 5, France

We show that the molecular graph G of a benzenoid hydrocarbon admits an isometric embedding into the Cartesian product of three trees T_1, T_2, and T_3 defined by three directions of the host hexagonal grid. Namely, to every vertex v of G one can associate an ordered triplet (v_1, v_2, v_3) with v_i being a vertex of T_i $(i = 1, 2, 3)$, such that the graph–theoretic distance between two vertices u, v of G equals the sum of respective tree–distances between u_i and v_i. This labeling of the vertices of G can be obtained in $O(n)$ time. As an application of this result we present an optimal $O(n)$ time algorithm for computing the diameter of the graph G of a benzenoid system with n vertices.

1. Introduction

In chemical graph theory there is a large number of indices based on distances[1−8] (just recall the classical Wiener index or the notion of the diameter and center of chemical graphs). Our note presents an optimal $O(n)$ algorithm for computing the diameter of a benzenoid system. This is a by–product of a special labeling of benzenoid systems which reflects the vertex–distances. Most likely, our approach supplied by the corresponding algorithmical techniques and data structures can be used as a convenient structure–based method to canonically label the vertices (that is, atoms) of benzenoid systems; for this field see the papers of S.B.Elk[9,10]. The recent paper by Klavžar, Gutman and Mohar[11] was the starting point for this research, and the results presented below refine in some sense that established in ref. 11.

A *benzenoid system* (alias *hexagonal system)* G is a finite connected plane graph in which every interior face (region) is a regular hexagon of side length 1. The vertex set of G is

[1] on leave from the Universitatea de stat din Moldova, Chişinău

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. The *eccentricity* $e(v)$ of a vertex v is the maximum distance from v to any vertex in G. The *diameter* $d(G)$ is the maximum eccentricity, i.e., the largest distance between two vertices of G. Finally, let B be the circuit which is defined by the exterior face of G. Although the results of this note are valid for all benzenoid systems, in order to use an intuitive geometric terminology we will assume that G is bounded by a simple circuit B.

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

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

such that

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

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

The *Cartesian product* $H = H_1 \times \ldots \times H_m$ of connected graphs $H_1, \ldots, H_m$ is defined upon the Cartesian product of the vertex sets of the corresponding graphs (called *factors*), i.e.,

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

Two vertices $u = (u_1, \ldots, u_m)$ and $v = (v_1, \ldots, v_m)$ are adjacent in H if and only if the vectors u and v coincide except at one position i, in which we have two vertices u_i and v_i 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 coincides with 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 ref. 11) 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. The result of ref. 11 asserts that any benzenoid system admits an isometric embedding into a cube. The dimension of this cube can be arbitrarily large (actually, it is equal to half the length of the bounding circuit B). Therefore, even for relatively simple benzenoids the labels of vertices can be already quite large; see Figure 2 of ref. 11.

Instead of isometric embeddings of benzenoids into binary Hamming graphs we propose such embeddings into the Cartesian product of trees. 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 and can be constructed in a total time linear in the number n of vertices of G. Hence, we obtain a canonical labeling of the vertices of G, where all labels have length 3. With this compact labeling of G (we need only $O(n)$ space) one can work much as with points in the 3–dimensional space. Indeed, using an algorithm of Harel and Tarjan[12] after a linear time preprocessing the distance between any two given vertices of a tree can be computed in constant time $O(1)$ (i.e., using a fixed number of operations). Since we have only three tree–factors, the distance between any two given vertices of a benzenoid system G can be computed in $O(1)$, too. To continue the comparison, the Euclidean distance between points in $\mathbb{R}^3$ is computed in $O(1)$ time using a formula, while the distance between vertices of a benzenoid system can be computed within the same time bounds, but using a special algorithm. Another special algorithmic technique, namely that of matrix–searching due to Aggarwal, Klawe, Moran, Shor and Wilber[13], can be employed to compute the diameter $d(G)$ of G in $O(n)$ operations. In fact, the diameter of G can be computed in time proportional to the length of the circuit B. The algorithm can be as fast as $O(\sqrt{n})$ for more-or–less compact benzenoids (when the length of B is proportional to $\sqrt{n}$).

To give an idea of how the embedding of a benzenoid system into the Cartesian product of three trees can be obtained in Figure 1 we present an example.

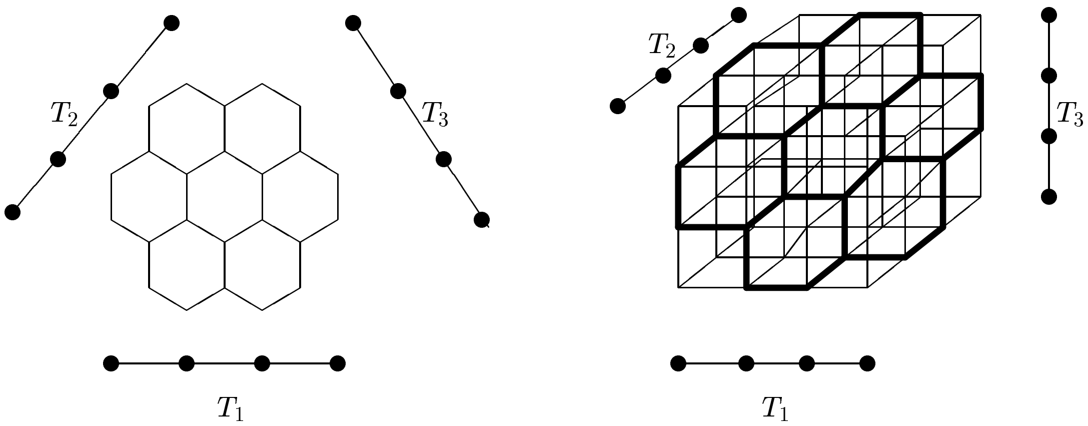

FIGURE 1. An embedding of coronene in the product of three paths.

2. Embedding into the Cartesian product of three trees

To formulate the embedding result we need some further terminology. Let G be a benzenoid bounded by a simple circuit B. By E_1, E_2, and E_3 denote the edges of G of a given direction. A straight line segment c with end points p and q is called a *cut segment* if c is orthogonal to one of the three edge directions, each p and q is the center of an edge, and the graph obtained from G by removing all edges intersected by c has exactly two connected components. In this case we say that c *separates* any two vertices from different connected components. The collection of all cut segments of G can be partitioned into three families $C_1(G)$, $C_2(G)$, and $C_3(G)$ each consisting of pairwise parallel segments. Evidently, every edge from E_i is intersected by a cut segment of $C_i(G)$ $(i = 1, 2, 3)$. Deleting all edges of E_i we obtain a graph G_i all of whose connected components are paths with end–vertices on B. One can easily show that every such path P is a shortest path. Moreover, it is the unique shortest path between the end–vertices x and y. Indeed, consider another path L between boundary vertices x and y not on any one such P from G_i. Necessarily L will contain at least one edge from E_i. On the other hand, every cut segment of C_i intersecting P also intersects an edge from L. This shows that L has more edges than P.

Define a graph T_i whose vertices are the connected components of G_i and two such components P' and P'' are adjacent in T_i if and only if there exists two adjacent vertices $u \in P'$ and $v \in P''$ such that the edge (u, v) is intersected by a cut segment from $C_i(G)$; see Figure 2 for an example. Since G is bounded by a Jordan curve B, every T_i is a tree (the existence of a cycle in T_i would imply that G contains a non–hexagonal interior face).

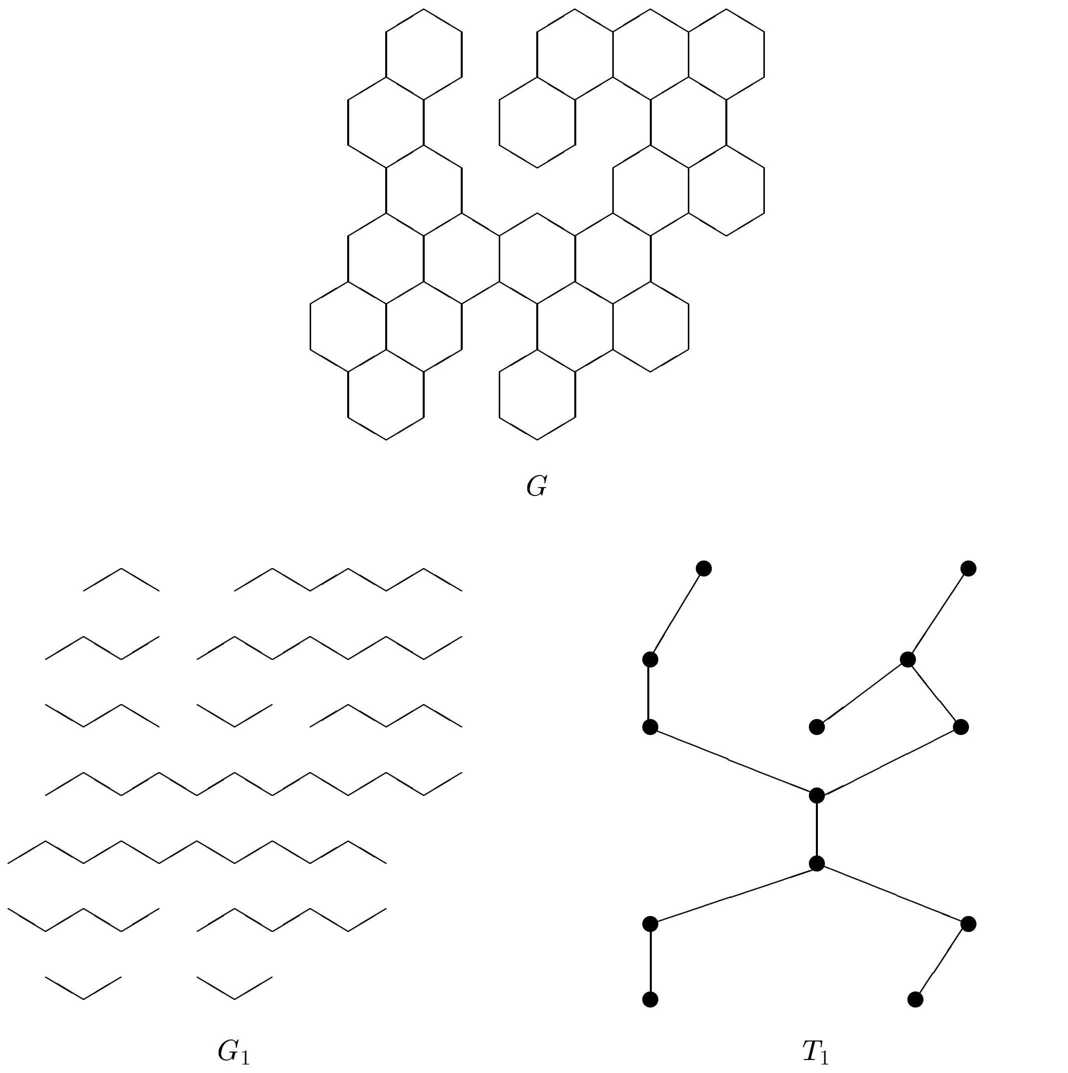

FIGURE 2. An illustration to the definition of tree–factors.

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. We claim that α provides an isometric embedding of G into H. To prove this pick two arbitrary vertices x and y of G and suppose that

$$\alpha(x) = (P', Q', R'), \qquad \alpha(y) = (P'', Q'', R'').$$

First note that the vertices x and y are separated by $d_{T_1}(P', P'')$ cut segments from C_1, exactly $d_{T_2}(Q', Q'')$ cut segments from C_2 and $d_{T_3}(R', R'')$ cut segments from C_3. (Recall, for example, that $d_{T_1}(P', P'')$ is the number of edges in the unique path of T_1 connecting P' and P''). Therefore, to complete the proof it is sufficient to show that the vertices x and y are separated by exactly $d_G(x, y)$ cut segments of G. Pick an arbitrary shortest path L between x and y. Any cut segment c separating x and y necessarily intersects at least one edge of L. If, say, $c \in C_i$ intersects two edges (u', v') and (u'', v'') of L, then we arrive at a contradiction. Indeed, granted two such intersecting edges, if the vertices u' and u'' were taken from the same connected component of the graph G_i, then u' and u'' would be connected by more than one shortest path.

Hence, α has the required distance–preserving property. To label the vertices of G one can proceed as follows. For a given edge direction i 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). More precisely, for a given cut segment $c \in C_i$ we can list all edges intersected by c in time proportional to the number of such edges. All this permits to construct the connected components of the graph G_i. After their labeling we can find the required incidence relation between them (i.e., to define the tree T_i). The i–th 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 $O(n)$ 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. Concluding, we obtain the following result.

THEOREM. *The map 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 factors T_1, T_2, T_3 as well as the corresponding labels of the vertices of G can be computed in total $O(n)$ number of operations.*

3. Computing distances

Now we outline how to compute the distance between two given vertices u and v of the benzenoid system G, provided G is represented by the data structure described in the previous section. More precisely, we wish to find $d_G(u, v)$ quickly, using just a fixed number of operations (notation $O(1)$). 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. It was only proved that the embedding α of G into the Cartesian product $H = T_1 \times T_2 \times T_3$ is isometric. Our theorem reduces the problem of finding $d_G(x, y)$ to three similar problems on factors. A major advantage is that all three factors are trees, because in this setting very efficient algorithms from ref. 12 can be applied. Thus we have to preprocess the trees T_1, T_2, and T_3 according to the algorithm described in ref. 12, in particular to make them rooted by picking arbitrary vertices as roots.

To recall the algorithmic problem solved by ref. 12 we need some preparation. Let T ($|T| = n$) be a *rooted tree* with the vertex r as a *root.* Every vertex u on a unique path connecting a vertex v with r is called an *ancestor* of v, while v is a *descendent* of u. The unique ancestor adjacent to v is the *parent* or *father* of v. Notice that the root r is an ancestor of every vertex in T. Geometrically, one can represent T in the plane so that the parent of any vertex $v \neq r$ is above v. A *subtree (with root* u) of T is an induced subgraph of T whose vertices are all descendents of vertex u of T.

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; for an illustration of this concept see Figure 3. Harel and Tarjan[12] presented an algoritm that after $O(n)$ preprocessing time answers in $O(1)$ time per query questions of the form, "What is the nearest common ancestor of vertices x and y?" To apply this algorithm for our purposes, in addition, in the preprocessing step we must compute the distances from the root r to all vertices of T (this can be done by the breadth–first search and requires additional $O(n)$ preprocessing time). Now, having a query of the form, "What is the distance between vertices x and y of T?", we have to find the nearest common ancestor $nca(x, y)$ of x and y and then to apply the formula

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

(for the justification see Figure 3). Therefore, to find the distance between two vertices of of a benzenoid G we must apply this method for each of three factors.

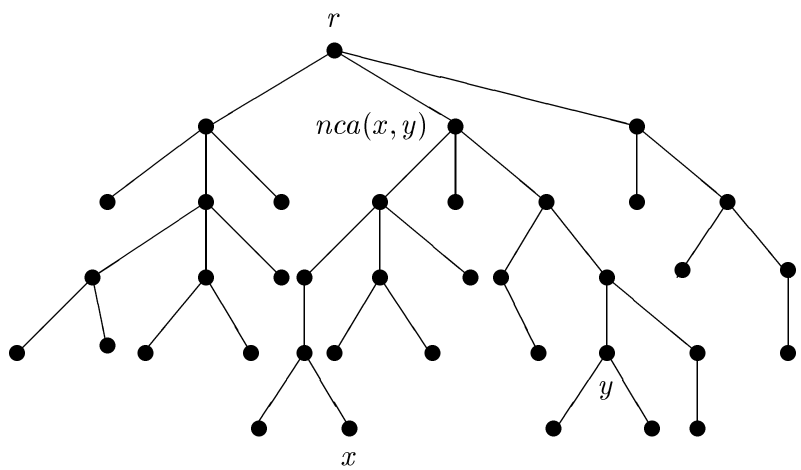

FIGURE 3. $nca(x, y)$ is the nearest common ancestor of x and y.

3. Computing the diameter

Though the data structure presented above can find some potential applications (in particular, we hope to use it for computing the center of a benzenoid system), it is in particular interesting to look at the problem of computing the diameter $d(G)$ of a benzenoid system G. Even more, we want to compute the eccentricity $e(v)$ for each boundary vertex $v \in B$ of G. This problem can be reformulated as a maxima–finding problem in the distance matrix of B. This suffices to compute $d(G)$, because it is quite obvious that the diameter of G is realized between two vertices of B. (Actually, for any vertex v of G there is a vertex $w \in B$, such that $d_G(v,w) = e(v)$.)

Denote by $v_1, \ldots, v_m$ the vertices of the cycle B with the vertices in order around the cycle, and let $d_{ij} = d_G(v_i, v_j)$, $i,j \in \{1,\ldots,m\}$. One can easily show that

$$(*) \qquad d_{ij} + d_{kl} \leq d_{ik} + d_{jl},$$

for all $1 \leq i < j < k < l \leq m$. Indeed, consider two shortest paths between the vertices i and k, and j and l, respectively. Necessarily, they must intersect in a vertex x of G. By the triangle inequality $d_{ij} \leq d_G(v_i,x) + d_G(x,v_j)$ and $d_{kl} \leq d_G(v_k,x) + d_G(x,v_l)$. Since $d_{ik} = d_G(v_i,x) + d_G(x,v_k)$ and $d_{jl} = d_G(v_j,x) + d_G(x,v_l)$, we obtain the required inequality.

Further, let D denote the $m \times m$ matrix that contains the pairwise distances between the vertices of B : more precisely, $D(i,j) = d_{ij}$ for $1 \leq i,j \leq m$. Our problem is to compute a maximum element in each row i of D : such an element is nothing else than $e(v_i)$. We use the matrix–searching technique of Aggarwal, Klawe, Moran, Shor and Wilber[13], for computing row–wise maxima of matrices with a special property called total monotonicity. The matrix D is *totally monotone* if

$$D(i,p) < D(i,q) \text{ implies } D(j,p) < D(j,q),$$

for any $i < j$ and $p < q$ (this is true in our case, because of the inequality $(*)$). Total monotonicity implies that the leftmost maxima obey the following useful property: as we move down the rows, the maximum can only move right, but never left. The matrix D is defined imlicitly: an entry is computed only when needed. The result of ref. 13 can be formulated in the following form:

The leftmost maximum in each row of a totally monotone $p \times q$ matrix can be computed using $O(p+q)$ matrix operations. If each matrix operation takes $f(p,q)$ time, then the row–maxima problem can be solved in time $O((p+q)f(p,q))$.

Applying the matrix–search algorithm to the matrix D, one can find the eccentricities of all vertices of B in total time $O(mf(m))$. If we use our data structure to compute the entries of D, then $f(m) = O(1)$, and we get an $O(m)$ time algorithm with a preprocessing step which requires $O(n)$ operations. This leads us to an optimal $O(n)$ time algorithm for computation of the diameter of a benzenoid system with n vertices.

Actually, we can build the factors T_1, T_2, and T_3 employing only the vertices of the bounding cycle B. This can be done in only $O(m)$ time, but the approach is based on the rather complicated Chazelle[14] algoritm for computing all vertex–edge visible pairs of a polygon with m vertices. Applying this algorithm separately for every direction, we will find the cuts from each family $C_1(G)$, $C_2(G)$, and $C_3(G)$. Even with the factors in hands, we can answer in $O(1)$ time only distances queries between vertices of B, and not arbitrary vertices of G. However, this suffices to run the matrix–search algorithm on D and compute the diameter of G in only $O(m)$ time.

For the details about the matrix–search algorithm consult ref. 13. The idea to use this method for the diameter problem first was applied by Hershberger and Suri[15] in case of simple polygons.

Acknowledgement. I am indebted to Dr. D.J. Klein for helpful suggestions and corrections.

References

[1] Trinajstić, N. *Chemical Graph Theory,* 2nd revised ed.; CRC Press: Boca Raton, FL, 1992.

[2] Bonchev, D.; Rouvray, D.H. *Chemical Graph Theory. Introduction and Fundamentals,* Abacus Press: New York, 1991.

[3] Bonchev, D.; Balaban, A.T.; Mekenyan, O. Generalization of the graph centre concept and derived topological centric indices. *J. Chem. Inf. Comput. Sci.* **1980,** *20,* 106–113.

[4] Bonchev, D.; Balaban, A.T.; Randić, M. The graph center concept for polycyclic graphs. *Internat. J. Quantum Chem.* **1981,** *19,* 61–82.

[5] Bonchev, D.; Balaban, A.T. Topological centric coding and nomenclature of polycyclic hydrocarbons. I. Condenced benzenoid systems (polyhexes, fusenes). *J. Chem. Inf. Comput. Sci.* **1981,** *21,* 223–229.

[6] Bonchev, D.; Mekenyan, O.; Balaban, A.T. Iterative procedure for the generalized graph center in polycyclic graphs. *J. Chem. Inf. Comput. Sci.* **1989,** *29,* 91–97.

[7] Bonchev, D.; Balaban, A.T. Central vertices versus central rings in polycyclic systems. *J. Math. Chem.* **1993,** *14,* 287–304.

[8] Balaban, A.T. Chemical graphs: looking back and glimpsing ahead. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 339–350.

[9] Elk, S.B. A canonical ordering of polybenzenes and polymantes using prime number factorization technique. *J. Math. Chem.* **1990,** *4,* 55–68.

[10] Elk, S.B. Graph theoretical algorithm to canonically name the isomers of the regular polyhedranes. *J. Chem. Inf. Comput. Sci.* **1992,** *32,* 14–22.

[11] Klavžar, S.; Gutman, I.; Mohar B. Labeling of benzenoid systems which reflects the vertex–distance relations. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 590–593.

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

[13] Aggarwal, A.; Klawe, M.; Moran, S.; Shor, P.; Wilber, R. Geometric applications of a matrix searching algorithm. *Algorithmica,* **1987,** *2,* 195–208.

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

[15] Hershberger, J.; Suri, S. Matrix searching with the shortest path metric. *Technical Report* **1993** 37 pp.

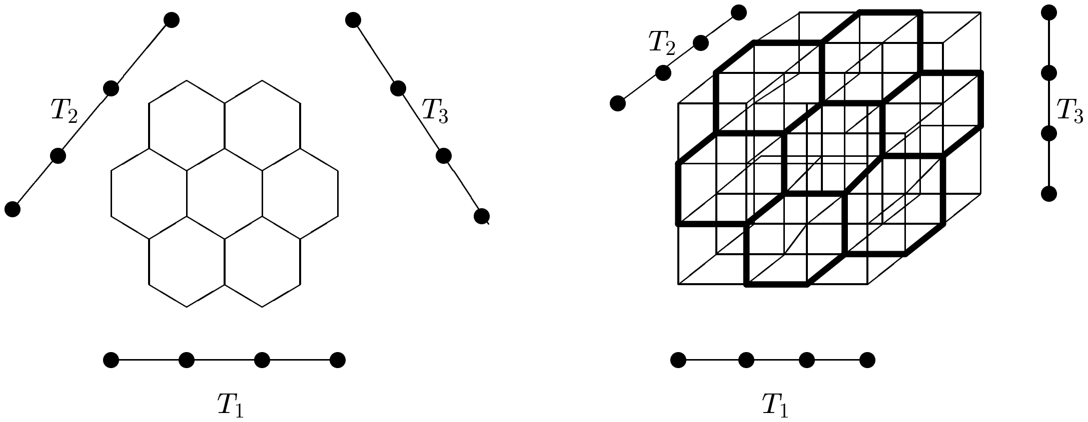

FIGURE 1. An embedding of coronene in the product of three paths.

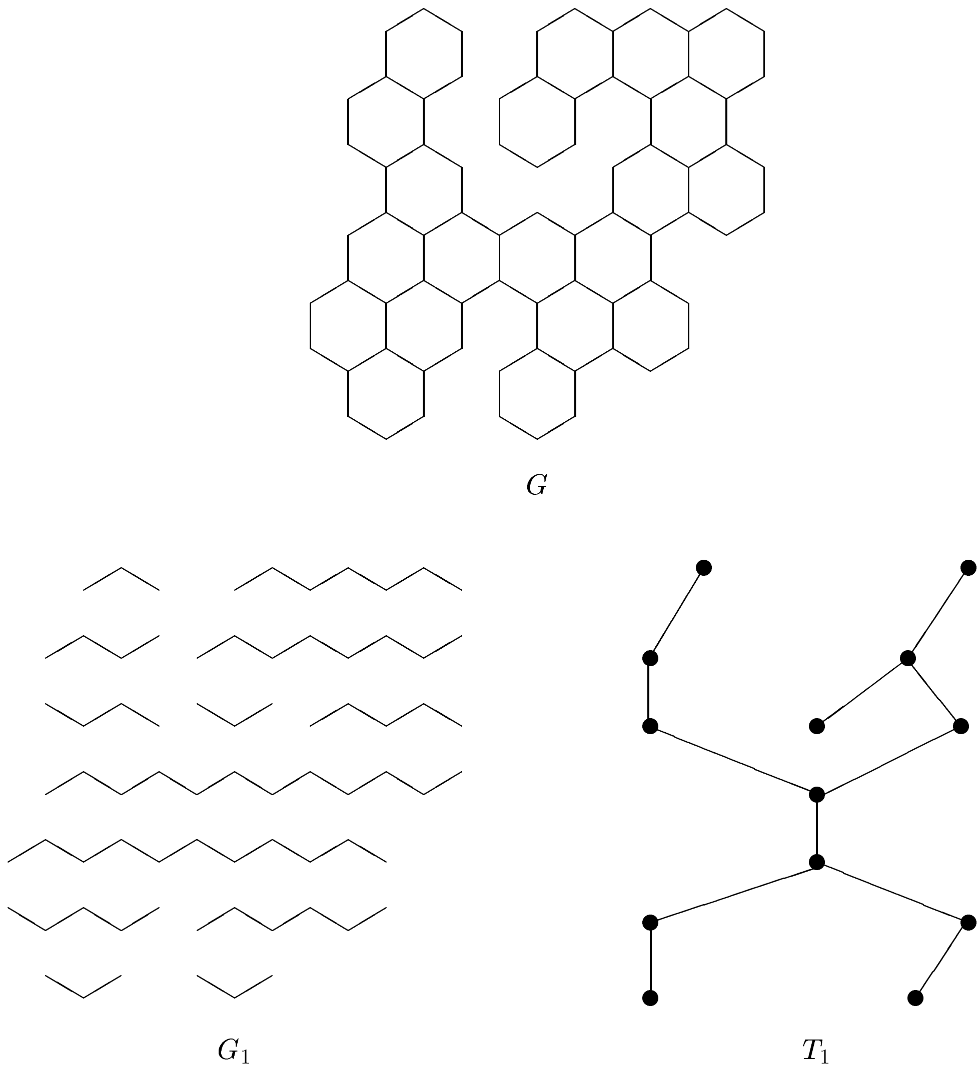

FIGURE 2. An illustration to the definition of tree–factors.

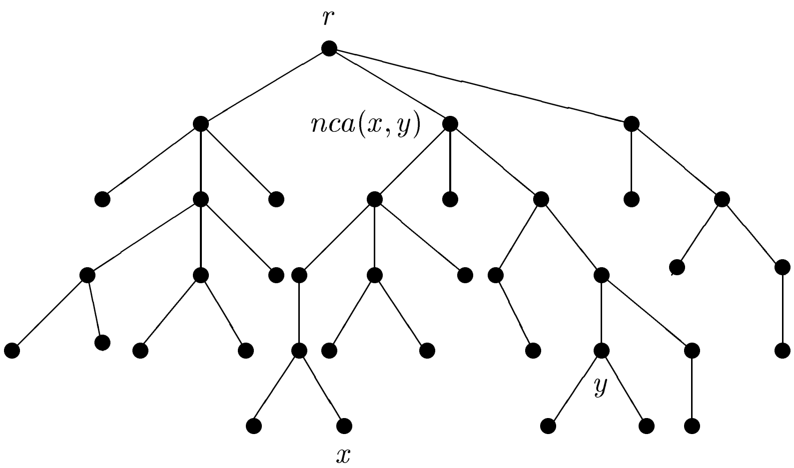

FIGURE 3. $nca(x,y)$ is the nearest common ancestor of x and y.