INFORMATIQUE THÉORIQUE ET APPLICATIONS

E. BAMPIS
A. GIANNAKOS
A. KARZANOV
Y. MANOUSSAKIS
I. MILIS

Perfect matching in general vs. cubic graphs : a note on the planar and bipartite cases

Informatique théorique et applications, tome 34, nº 2 (2000), p. 87-97
<http://www.numdam.org/item?id=ITA_2000__34_2_87_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **34** (2000) 87–97

PERFECT MATCHING IN GENERAL VS. CUBIC GRAPHS: A NOTE ON THE PLANAR AND BIPARTITE CASES

E. BAMPIS[1], A. GIANNAKOS[2], A. KARZANOV[3], Y. MANOUSSAKIS[4] AND I. MILIS[5]

Abstract. It is known that finding a perfect matching in a general graph is AC^0-equivalent to finding a perfect matching in a 3-regular (*i.e.* cubic) graph. In this paper we extend this result to both, planar and bipartite cases. In particular we prove that the construction problem for perfect matchings in planar graphs is as difficult as in the case of planar cubic graphs like it is known to be the case for the famous Map Four-Coloring problem. Moreover we prove that the existence and construction problems for perfect matchings in bipartite graphs are as difficult as the existence and construction problems for a weighted perfect matching of $O(m)$ weight in a cubic bipartite graph.

AMS Subject Classification. 05C70, 05C85.

INTRODUCTION AND NOTATION

Although the maximum matching problem in graphs dates back to 30's, with König's work, it was not earlier than 1965 when the first polynomial algorithm for general graphs was found by Edmonds [6]. Subsequently numerous more and more efficient algorithms were obtained, and today the fastest known algorithm is due to Micali-Vazirani [16, 26], who hold this record since 1980. Faster algorithms are today known for special classes of graphs such as bipartite [1, 7], interval [14],

[1] LaMI, Université d'Evry, boulevard des Coquibus, 91025 Evry Cedex, France; e-mail: bampis@lami.univ-evry.fr

[2] La.R.I.A., Université de Picardie-Jules Verne, 5 rue du Moulin Neuf, 80000 Amiens, France; e-mail: aristote@laria.u-picardie.fr

[3] Institute for System Analysis, 9, Prosp. 60 Let Octyabrya, 117312 Moscow, Russia.

[4] LRI, bâtiment 490, Université Paris-Sud, 91405 Orsay Cedex, France; e-mail: yannis@lri.fr

[5] Athens University of Economics and Business, Department of Informatics, 76 Patission St., 10434 Athens, Greece; e-mail: milis@aueb.gr

or circular arc graphs [18]. Note that no faster algorithm was known for bipartite graphs until 1991 [1, 7], and the same is true for planar graphs even today [15].

On the other hand, the parallel complexity of the maximum matching problem is a famous open question. Recall that the existence, construction and enumeration problems for perfect matchings are distinct on a parallel setting, constituting three versions with different degrees of "hardness".

It is well known that there is an NC reduction of the (construction and enumeration of) maximum matching problem to the perfect matching one (see for example [24]). However, even the problem of the existence of a perfect matching in a general graph is open and it is unknown whether it belongs to NC or not. For the existence, construction and enumeration of perfect matchings, membership has been affirmed for various special graph classes as: regular bipartite, planar bipartite, claw-free, line, dense, strongly chordal, co-comparability, graphs having a unique perfect matching or a polynomial number of perfect matchings. Vazirani [25] has proved that the existence and enumeration problems for a perfect matching in planar graphs is in NC [25] (in fact, he has extended this result to a somewhat larger class, that is, graphs without subdivisions of $K_{3,3}$), but the construction problem still remains open. For bipartite graphs all of the three versions of the perfect matching problem remain also open. For an excellent survey of the above results, the reader is referred to [12, 22].

Dahlhaus and Karpinski in [5], introduced a new interesting approach to explore the complexity behavior of the perfect matching problem. In fact they proved that the existence and the construction problems for a perfect matching in general graphs are AC^0-equivalent to the same problems in cubic (*i.e.* 3-regular) graphs. Their result is based on a simple and natural transformation from general to cubic graphs such that perfect matchings are preserved. This type of transformations are first discussed in [21] and are surveyed in [9].

In this paper we prove some further results in this vein, concerning planar and bipartite graphs and contributing to further understanding the complexity behavior of the (construction of) perfect matching problem in these popular graph classes.

First, we extend the result of [5] to the case of planar graphs: we prove that the construction problem for a perfect matching in planar graphs is NC-equivalent to the same problem in planar cubic graphs. Planar graphs is of independent interest since other positive results are known for this class; recall that the existence problem for a perfect matching as well as the maximum flow problem are known to be in NC for planar graphs [17] (this last result implies an NC algorithm for finding a perfect matching in planar-bipartite graphs).

Moreover, we prove a similar result for bipartite graphs. In fact we prove that the existence and construction problems for a perfect matching in bipartite graphs are as difficult as the existence and construction problems for weighted perfect matching in cubic bipartite graphs. It is known that finding a (non-weighted) perfect matching in a regular bipartite graph is in NC [13] and therefore a direct transformation from any bipartite to a cubic-bipartite graph, like in general and planar cases, would imply directly an NC algorithm for finding a perfect matching problem in any bipartite graph. On the other hand, as it has been remarked in [19], any probabilistic algorithm that uses the Isolation lemma for finding in parallel a non-weighted perfect matching, can provide, with slight modifications, a (probabilistic) algorithm for finding a minimum-weight perfect matching in a weighted graph. Also, there is a parallel algorithm for finding maximum weighted matchings in bipartite graphs, but clearly not in NC, since it takes $O(n^{\frac{2}{3}} \log^k n)$ time using $O(n^3)$ processors ([8]; see also [12]).

Notation

Formally, in the sequel all graphs considered are finite, undirected, without loops, and without multiple edges. We denote the vertex-set and the edge-set of a graph G by $V(G)$ and $E(G)$ respectively. The number of vertices and edges of G are denoted by n and m, respectively. Given a graph $G = (V, E)$, a *matching* is a set of disjoint edges of G. A *maximum matching*, *i.e.* a matching of maximum cardinality, is said to be *perfect* if it covers all vertices of the graph. The set of all perfect matchings in G will be denoted by $\mathcal{M}(G)$. Let $N(v)$ be the set of all the edges that are incident to vertex v, *i.e.* $N(v) = \{(u,v) | (u,v) \in E(G)\}$. As usually, $d(v)$ will be the degree of v in $V(G)$ and $\Delta(G)$ the maximum degree of the graph *i.e.* $\Delta(G) = \max_{v \in V(G)} d(v)$. The set of neighbors of v is denoted by $\Gamma(v)$. An edge of G is called a *cut-edge* if and only if its removal disconnects G. A graph G is *2-edge-connected* if and only if, between any pair of distinct vertices of G, there are at least two edge-disjoint paths connecting them. For the sake of simplicity, in the sequel we call the 2-edge-connected graphs simply *2-connected graphs.*

A graph is called *planar* if and only if it can be embedded into a plane. Given such an embedding the dual graph G^* of G is constructed as follows: the vertices of G^* correspond to faces of G, including the external face, and there is an edge $(p,q) \in E(G^*)$ if and only if p and q are adjacent faces of G.

1. The general cubic transformation

The transformation of a given graph into a cubic one proposed by Dahlhaus and Karpinski [5] consists of the following three successive steps:

Transformation T

Step T1 (Cut edges): For every cut-edge (v, u) of the original graph, a subgraph H with vertex set x_1, x_2, x_3, x_4 and edge set $\{(x_1, x_2), (x_1, x_3), (x_2, x_4), (x_3, x_4)\}$ is added, being connected to both v and u by edges (v, x_1) and (u, x_4) (see Fig. 1).

The obtained graph is 2-connected and hence it has no degree one vertices.

Step T2 (Degrees' reduction): In the graph obtained by Step T1, every vertex $v \in G$ of degree $d(v) > 3$ is replaced by an even length path $x_1 y_1 x_2 y_2 \cdots x_{d(v)-1}$

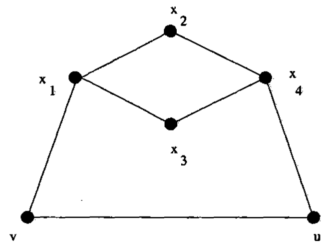

FIGURE 1. Step 1: transformation of cut-edges.

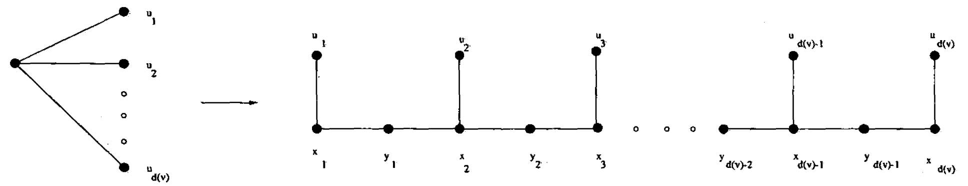

FIGURE 2. Step T2: degrees' reduction.

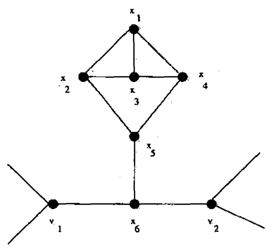

FIGURE 3. Step T3: degrees' regularization.

$y_{d(v)-1}x_{d(v)}$ and every edge (v, u_i), $i = 1, 2, ..., d(v)$, incident to v is replaced by an edge (x_i, u_i), $i = 1, 2, ..., d(v)$ (see Fig. 2).

The obtained graph has only degree two and three vertices and the number of its degree two vertices is even.

Step T3 (Degrees' regularization): In the graph obtained by Step T2, partition arbitrarily the set of its 2-degree vertices into pairs (recall that this is possible, since there is an even number of such vertices). For every pair v_1, v_2 that belongs to the partition, a subgraph C with vertex set $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ and edge set $\{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_3, x_4), (x_2, x_5), (x_4, x_5), (x_5, x_6)\}$ is added, being connected to both v_1 and v_2 by edges (v_1, x_6) and (v_2, x_6) (see Fig. 3).

The obtained graph is 3-regular, but clearly not 2-connected since every (x_5, x_6) edge is a cut edge.

Let $T(G)$ be the cubic graph obtained by the transformation T. It has been proved in [5] that each one of the three steps has the perfect matching preserving property, *i.e.* there is a surjection F: $\mathcal{M}(T(G)) \to \mathcal{M}(G)$. Hence, finding a in parallel a perfect matching in a 3-regular graph is as as difficult as in general graphs. Moreover, the next theorem holds:

Theorem 1.1. *[5] The existence and the construction problem for a perfect matching restricted to:*

1) *2-connected graphs, or*
2) *graphs of maximum degree 3, or*
3) *3-regular graphs, is AC^0 equivalent to the existence and the construction problem for a perfect matching in general graphs.*

Next theorem concerns the complexity of transformation T, and it has, formally, been proved in [2].

Theorem 1.2. *Transformation T takes*

- *$O(m)$ sequential time.*
- *$O(\log n)$ time using $O(n^2)$ processors in a CRCW PRAM.*

We conclude this section by noticing that it is enough to have the vertices of $T(G)$ appropriately marked during T, to get from a matching $M \in \mathcal{M}(T(G))$ the $F(M)$ in $O(|V(T(G)|)$, that is $O(m)$ time. Thus, the complexity of finding a perfect matching in a general graph is $O(\mathcal{C}(m))$, where $\mathcal{C}(n)$ is the complexity of finding a perfect matching in a cubic graph of size n.

2. The planar case

It is clear that Steps T1 and T2 of transformation T preserve the planarity of the input graph. In this section we shall modify the third step of T that eventually destroys planarity.

Towards this direction we proceed as follows: given a planar graph G of maximum degree three, we consider an embedding into the plane and its faces $q_1, q_2, ..., q_r$. Let V_2 be the set of degree two vertices of the graph and $V_2^{q_i}$ be the set of degree two vertices belonging to face q_i, *i.e.* $V_2 = \{v \in G \mid d(v) = 2\}$ and $V_2^{q_i} = \{v | v \text{ is a vertex on the boundary of face } q_i \text{ of } G \text{ with } d(v) = 2\}$. We consider an assignment of each degree two vertex to exactly one face, that is a partition of V_2 into sets $X^{q_1}, ..., X^{q_r}$ each one corresponding to a face of G. Clearly $X^{q_i} \subseteq V_2^{q_i}$, $X^{q_i} \cap X^{q_j} = \emptyset$, $1 \le i \ne j \le r$.

Lemma 2.1. *If V_2 can be partitioned into $X^{q_1}, ..., X^{q_r}$ such that every X^{q_i} is of even cardinality, then Step T3 of the general cubic transformation can be applied preserving planarity.*

Proof. Let a face q_i, where $|X^{q_i}|$ is even. We simply consider pairs of successive, in the boundary of q_i, vertices of degree two. Then for all these pairs Step T3 of the general cubic transformation can be applied without any edge crossings. Since X^{q_i}'s are disjoint sets, the argument applies for each face of the graph. $\square$

Our aim is, therefore, to assign an even number of degree two vertices to each face in order to be possible to apply Step T3 and to connect them "inside" the faces, thus preserving planarity as it is suggested in Lemma 2.1. This can be done by exploiting the next lemma.

Lemma 2.2. *Let G be a planar graph and $(x,y) \in E(G)$. If G' is the graph that yields, after having $(x,y) \in E(G)$ substituted with an odd length path, i.e. $G' = (V(G) \cup \{x_1, .x_2, .., x_{2l}\}, E(G) - \{(x,y)\} \cup_{i=1}^{2l-1} \{(x_i, x_{i+1})\} \cup \{(x,x_1),(x_{2l},y)\})$ for some $l \in N$, then there is a bijection $H : \mathcal{M}(G') \leftrightarrow \mathcal{M}(G)$.*

Proof. For $l = 1$, we have that $V(G') = V(G) \cup \{x_1, x_2\}$ and $E(G') = E(G) - \{(x,y)\} \cup \{(x,x_1),(x_1,x_2),(x_2,y)\}$. Clearly, for every $M \in \mathcal{M}(G)$, we have $H(M) \in \mathcal{M}(G') = M \cup \{(x_1,x_2)\}$ if $(x,y) \notin M$ and $H(M) = M - \{(x,y)\} \cup \{(x,x_1),(x_2,y)\}$ otherwise. For the induction step, let $G_k = (V(G) \cup \{x_1, .x_2, .., x_{2k}\}, E(G) - \{(x,y)\} \cup_{i=1}^{2k-1} \{(x_i,x_{i+1})\} \cup \{(x,x_1),(x_{2k},y)\})$ Suppose w.l.o.g. that $G_{k+1} = V(G) \cup \{x_1, .x_2, .., x_{2k}, x_{2k+1}, x_{2(k+1)}\}, E(G) - \{(x,y)\} \cup_{i=1}^{2k-1} \{(x_i,x_{i+1})\} \cup \{(x,x_1),(x_{2k},x_{2k+1}),(x_{2k+1},x_{2(k+1)}),(x_{2(k+1)},y)\}$. The bijection $h : \mathcal{M}(G_k) \leftrightarrow \mathcal{M}(G_{k+1})$ can be established as follows: for $M \in \mathcal{M}(G_{k+1})$, if $\{(x_{2k},x_{2k+1}), (x_{2(k+1)},y)\} \in M$ then $h^{-1}(M) = h(M) - \{(x_{2k},x_{2k+1}), (x_{2(k+1)},y)\} \cup \{(x_{2k},y)\}$, else (if $(x_{2k+1},x_{2(k+1)}) \in h(M)$), $h^{-1}(M) = h(M) - \{(x_{2k+1},x_{2(k+1)})\}$. □

Let V_2^{pq} the set of degree two vertices belonging to the common boundary of faces p and q, *i.e.* $V_2^{pq} = V_2^p \cap V_2^q$. Every degree two vertex belongs to two face boundaries, except of some vertices that eventually belong only to the external face boundary. We partition arbitrarily V_2^{pq} into V_{2p}^{pq} and V_{2q}^{pq}, of cardinality $\lfloor \frac{|V_2^{pq}|}{2} \rfloor$ and $\lceil \frac{|V_2^{pq}|}{2} \rceil$ and assign them to faces p and q respectively.

After such an arbitrary partition of the degree two vertices between all pairs of adjacent faces, it is possible that the sets of two degree vertices assigned to some faces rest odd. Let *Odd* be the set of such faces. In general this could be the case for some even number of faces. Thus, it might be some "distant", *i.e.* not belonging to adjacent faces, pairs of 2-degree vertices that have to be connected according Step T3, destroying graph's planarity.

To raise any crossing that destroys planarity we consider a "virtual way" between distant vertices of degree two that have been assigned to faces p and q of G. Notice that this virtual way corresponds to a path between vertices p and q in the dual graph G^* of G. Such paths can be found during a depth-first search in G^*, by beginning to list a new path when $q \in Odd$ is visited and the number of already visited vertices of *Odd* is even, and finish the listing when a new $q' \in Odd$ is visited and the number of already visited vertices of *Odd* is odd.

Then Lemma 2.2 can be exploited as follows: just substitute every edge that crosses the virtual way connecting a distant pair of 2-degree vertices, by paths of length three, and assign one new vertex to each of the two face boundaries where the edge belongs to. Thus, there will be no face with odd number of 2-degree vertices assigned to it, so the conditions of Lemma 2.1 will hold.

We summarize our planar cubic transformation below:

Transformation T_p
Input: A planar graph G.
Output: A planar cubic graph $T_p(G)$.

1. Step $T1$;
2. Step $T2$;
3. Embed G into the plane and construct the dual graph G^*;

{ Assign 2-degree vertices of G to its faces }

4. **For** all $q \in V(G^*)$ **do**

    4.1. $V_2^q \leftarrow \{v | v$ is a vertex of the boundary of face q in G with $d(v) = 2\}$;

    4.2. $X^q \leftarrow V_2^q$

enddo

5. **For** all $q \in V(G^*)$ **do**

    For all $p \in \Gamma(q)$ **do**

        5.1. $V_2^{pq} \leftarrow V_2^p \cap V_2^q$;
        5.2. Partition arbitrarily V_2^{pq} into V_{2p}^{pq} and V_{2q}^{pq};
        5.3. $X^p \leftarrow X^p - V_{2p}^{pq}$;
        5.4. $X^q \leftarrow X^q - V_{2q}^{pq}$;

    enddo

enddo

{ Transform odd parity faces, $|Odd| \equiv 0 \pmod 2$ }

6. $Odd \leftarrow \{q \in V(G^*) \mid |X^q| \equiv 1 \pmod 2\}$;
7. Perform a depth-first search in G^* and find $\frac{|Odd|}{2}$ paths connecting distinct pairs in Odd;

8. **For** all $\frac{|Odd|}{2}$ paths found in Step 8 **do**

    8.1. Take a path from q to p, and enumerate its vertices: $q = q_0, q_1, .., q_l, q_{l+1} = p$;

    8.2. **For** $k = 0$ to l **do**

        8.2.1. Select arbitrarily an edge in the common boundary of faces q_k and q_{k+1}, say (v, v') ;
        8.2.2. $V(G) \leftarrow V(G) \cup \{x_k, x_{k+1}\}$;
        8.2.3. $E(G) \leftarrow E(G) - \{(v, v')\} \cup \{(v, x_k), (x_k, x_{k+1}), (x_{k+1}, v')\}$;
        8.2.4. $X^{q_k} \leftarrow X^{q_k} \cup \{x_k\}$;
        8.2.5. $X^{q_{k+1}} \leftarrow X^{q_{k+1}} \cup \{x_{k+1}\}$;

    enddo

enddo

9. Connect every pair of vertices within each X^q, according to Lemma 2.3.

Let $T_p(G)$ be the cubic planar graph obtained by the transformation T_p. The next lemma follows by the Theorem 1.1 and Lemmas 2.1 and 2.2.

Lemma 2.3. *There is a surjection* $F_p : \mathcal{M}(T_p(G)) \to \mathcal{M}(G)$.

Next two theorems concern the sequential and the parallel complexity of transformation T_p respectively.

Theorem 2.4. *Transformation* T_p *takes* $O(n)$ *sequential time.*

Proof. Steps 1-3 take cost $O(m) = O(n)$ time. More specifically, G^* takes $O(n)$ time to be constructed (the reader is referred to [20] for further details). Steps 4-6 take clearly $O(n)$ time, while Step 7 can be accomplished at the cost of a depth first search in the graph, *i.e.* $O(m) = O(n)$. Finally, Step 9 takes $O(n)$ time. □

The algorithm given above for T_p can be easily converted to run in parallel and the next theorem holds.

Theorem 2.5. *Transformation* T_p *takes* $O(\log n)$ *time using* $O(n^2)$ *processors in a CRCW PRAM.*

Proof. Finding an embedding of a planar graph G and its dual G^*, in Step 3, can be done in $O(\log n)$ time using $n \log\log n / \log n$ processors [23]. A depth first search in a planar graph, in Step 7, can be done in $O(\log n)$ time using $n/\log n$ processors [11]. All other operations in the algorithm can be done in constant time using $O(n^2)$ processors. □

Thus, we can state next theorem for the planar case.

Theorem 2.6. *The existence and the construction problem for a perfect matching restricted to:*

1) *2-connected planar graphs, or*
2) *planar graphs of maximum degree 3, or*
3) *3-regular planar graphs, is* NC^1 *equivalent to the existence and the construction problem for a perfect matching in general planar graphs.*

3. The Bipartite case

The class of bipartite graphs $B = (X \cup Y, E)$ with color classes X, Y is also of independent interest (see for instance [5,10]). It is clear that a necessary condition for a bipartite graph to have a perfect matching is to be balanced, *i.e.* $|X| = |Y|$. It is also known that the construction problem for a perfect matching in regular bipartite graphs is in NC. Therefore we do not expect a direct transformation like in general and planar cases. Indeed, Steps T1 and T3 of the general cubic transformation do not preserve bipartiteness and it is not obvious how this problem can be resolved. Fortunately, Step T2 preserves bipartiteness.

It is possible to reduce the problem of finding a perfect matching in a balanced bipartite graph to the one of finding a minimum weight perfect matching in a cubic bipartite graph. To this end we assign unit weights to the edges of the

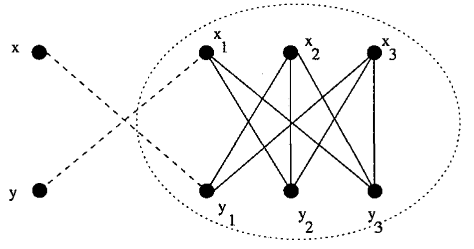

FIGURE 4. Step $T3_b$: degrees' regularization.

original bipartite graph and we modify Steps T1 (cut-edges) and T3 (degrees' regularization) of the general cubic transformation. The weights of the edges added during our transformation are either unit weighted or "heavy" c-weighted with $c = \Omega(n^3)$.

Transformation T_b

Step $T1_b$ (Degree one vertices): For every vertex v of the original graph with degree one, add a c-weighted edge (v, u) to any vertex u of the other color class. The obtained graph has no degree one vertices.

Step $T2_b$ (Degrees' reduction): In the graph obtained by Step $T1_b$ apply Step T2. For every c-weighted edge (v, u_i), assign a weight c to the corresponding edge (x_i, u_i) defined in Step T2. The obtained graph has only vertices of degree two or three. The set of its 2-degree vertices is partitioned into disjoint pairs x, y where x belongs to the one color class and y belongs to the other. This partition is possible since the number of vertices of degree two is the same in both color classes of the obtained graph.

Step $T3_b$ (Degrees' regularization): In the graph obtained after Step $T2_b$, for every pair x, y of the partition of the degree two vertices which has been done in the previous step, a bipartite C_B-component with vertex set $V(C_B) = \{x_1, x_2, x_3, y_1, y_2, y_3\}$ and edge set $E(C_B) = \{(x_1, y_2), (x_1, y_3), (x_2, y_1), (x_2, y_2), (x_2, y_3), (x_3, y_1), (x_3, y_2), (x_3, y_3)\}$ is added, being connected to both x and y by edges (x_1, y) and (y_1, x). In the obtained graph, assign c weights to (x_1, y) and (y_1, x), and unit weights to all other edges (see Fig. 4, where heavy weighted edges are shown by dashed lines).

The obtained graph is clearly 3-regular.

Every edge of the obtained (*i.e.* the transformed) graph is either unit or c-weighted. More precisely, a c-weighted edge has been added either at Step $T1_b$, in order to make the graph 2-connected, or at Step $T3_b$, in order to connect a C_B-component to the graph. Notice that there is a surjection of the set of perfect matchings without any c-weighted edges in the transformed graph, onto the set of perfect matchings in the initial graph.

On the other hand, if m is the number of edges of the initial graph, then the number of vertices of the transformed graph will be in $O(m)$. Hence, the weight of every perfect matching in the transformed graph that corresponds to some perfect matching in the initial graph, will be in $O(m)$, while the weight of every other perfect matching will be in $\Omega(n^3)$, due to the participation of c-weighted edges in it. In fact, there is a perfect matching in the initial graph, if and only if there is a minimum-weight perfect matching of $O(m)$ weight in the transformed graph. So the next theorem holds:

Theorem 3.1. *The existence and construction problems for a perfect matching in a bipartite graph are NC^1 reducible to the existence and construction problems, respectively, for a weighted perfect matching of $O(m)$ weight in a cubic bipartite graph.*

The authors wish to thank M. Karpinski for pointing out the general cubic transformation, F. Afrati for the helpful comments on this work, and the anonymous referees for their precious remarks and suggestions that led to essential changes in the original manuscript.

References

[1] H. Alt, N. Blum, K. Mehlhron and M. Paul, Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Inform. Process. Lett.* **37** (1991) 237-240.

[2] E. Bampis, A. Giannakos, A. Karzanov, I. Milis and Y. Manoussakis, *Matchings in cubic graphs are as difficult as in general graphs*, Rapport de Recherche No. 12. LaMI, Université d' Evry – Val d'Essonne (1995).

[3] R. Cole and U. Vishkin, Approximate and exact parallel scheduling, Part 1: The basic technique with applications to optimal parallel list ranking in logarithmic time. *SIAM J. Comput.* **17** (1988) 128-142.

[4] N. Chiba, T. Nishizeki and N. Saito, Applications of the planar separator theorem. *J. Inform. Process.* **4** (1981) 203-207.

[5] E. Dahlhaus and M. Karpinski, Perfect matching for regular graphs is AC^0-hard for the general matching problem. *J. Comput. System Sci.* **44** (1992) 94-102.

[6] J. Edmonds, Paths, trees and flowers. *Canad. J. Math.* **17** (1965) 449-467.

[7] T. Feder and R. Motwani, *Clique partitions, graph compression and speeding-up algorithms*, 23$^{\text{th}}$ STOC (1991) 123-133.

[8] A. Goldberg, S. Plotkin and M. Vaidya, Sublinear time parallel algorithms for matchings and related problems. *J. Algorithms* **14** (1993) 180-213.

[9] R. Greenlaw and R. Petreschi, Cubic graphs. *ACM Computing Surveys* **27** (1995) 471-495.

[10] D.Yu. Grigoriev and M. Karpinski, *The matching problem for bipartite graphs with polynomially bounded permanents is in NC*, 28$^{\text{th}}$ FOCS (1987) 166-172.

[11] T. Hagerup, Optimal parallel algorithms on planar graphs. *Inform. and Comput.* **84** (1990) 71-96.

[12] M. Karpinski and W. Rytter, *Fast parallel algorithms for graph matching problems.* Oxford University Press, preprint (to appear).