Electronic Colloquium on Computational Complexity, Report No. 112 (2013)

Exact Perfect Matching in Complete Graphs [1]

R. Gurjar[2] A. Korwar[2] J. Messner[3] T. Thierauf[3]

August 12, 2013

Abstract

A *red-blue graph* is a graph where every edge is colored either red or blue. The *exact perfect matching* problem asks for a perfect matching in a red-blue graph that has exactly a given number of red edges.

We show that for complete and bipartite complete graphs, the exact perfect matching problem is logspace equivalent to the perfect matching problem. Hence an efficient parallel algorithm for perfect matching would carry over to the exact perfect matching problem for this class of graphs. We also report some progress in extending the result to arbitrary graphs.

1 Introduction

The matching problem is one of the most studied problem in complexity theory. The problem is especially interesting with respect to its parallel complexity. It appears in various versions, some of them are perfect matching (*PM*), maximum matching (*MM*), and exact perfect matching (*xPM*) introduced by Papadimitriou and Yannakakis [PY82]. All of these problems are known to be solvable by efficient randomized parallel algorithms, they are in the class RNC [MVV87]. They are known to be in the class NC, i.e., solvable by efficient parallel algorithms without randomization, for some restricted classes of graphs only, for example, planar graphs [Vaz89]. These results seem to indicate that the problems are of similar complexity. However, while we know polynomial time algorithms for *PM* and *MM*, it is not known whether *xPM* is in P.

For complete graphs and complete bipartite graphs, Karzanov [Kar87] gave a characterization of when such a graph has an exact perfect matching. The characterization immediately gives an easy test for the *existence*

[1]Work supported in part by the Indo-German DST-DFG program, DFG grant TH 472/4-1 and DST grant DST/CS/20100251. The first author is partially supported by TCS research fellowship.

[2]IIT Kanpur, India

[3]Aalen University, Germany

ISSN 1433-8092

of an exact perfect matching. Karzanov also developped a polynomial-time algorithm to *construct* an exact perfect matching. Yi, Murty, and Spera [YMS02] gave a simpler construction algorithm for bipartite complete graphs. Whereas Karzanov gave separate proofs for the cases of complete and complete bipartite graphs, Geerdes and Szabó [GS11] gave a unified proof for for both cases of Karzanov's characterization theorem, but they left open a unified construction algorithm.

In this paper, we improve and extend these results as follows.

- We give unified proofs and construction algorithms for the case of bipartite and non-bipartite graphs, whereas the argument in [YMS02] is just for the bipartite case.
- Our polynomial-time algorithms are in fact logspace reduction from the exact perfect matching problem for complete graphs and complete bipartite graphs (*cxPM*) to the perfect matching problem (*PM*). This ties the complexity of *cxPM* to that of *PM*. Recall that for *PM* it is still open whether it can be solved efficiently in parallel.
- We report some progress in extending the results from complete graphs to arbitrary graphs. As in [YMS02], our algorithm has two major phases. The first phase constructs an *exact pseudo perfect matching*, which is a set of edges that comes very close to the exact perfect matching, in some sense. By adapting an argument of Yuster [Yus12], we are actually able to construct the exact pseudo perfect matching for arbitrary graphs, instead of just complete graphs as in [YMS02]. However, the second phase of our algorithm still works for complete graphs only.
- Finally, our paper might be considered as the first *complete* exposition of a proof of Karzanov's characterization theorem. Whereas Karzanov himself leaves wide parts of the proof to the reader, [YMS02] is still somewhat handwaving at crucial points. The exposition of [GS11] is very succint; we were not able to completely follow it.

In the next section, we define the problems considered in this paper. In Section 3 we show logspace reductions between these problems. The main part of the paper is Section 4. We show a logspace reduction from exact perfect matching to perfect matching for complete graphs and complete bipartite graphs. The results in Section 4 are written in a way that the mathematical arguments for the existence of an exact perfect matching are cleanly separated from the complexity considerations. A reader who is only interested in the mathematical proofs that an exact perfect matching exists can easily skip the complexity arguments.

2 Preliminaries

Graphs and Matchings. Let $G = (V, E)$ be an undirected graph. For a node $v \in V$ let $\Gamma(v) = \{ u \in V \mid (v, u) \in E \}$ be the set of neighbors of v. A *matching* in G is a set $M \subseteq E$, such that no two edges in M have a common vertex. We say that an edge $e \in M$ *covers* a vertex v if v is one of its endpoints. The number $|M|$ of edges in M is called the *size of* M. A matching M is called *maximal* if there is no edge e such that $M \cup \{e\}$ is a matching, it is called *perfect* if every vertex is covered by some edge in M. For a weight function $w : E \to \mathbb{N} = \{0, 1, 2, \dots\}$ of G the *weight of a matching* M is defined as $w(M) = \sum_{e \in M} w(e)$.

In the exact perfect matching problem defined below, every edge of graph $G = (V, E)$ is colored either red or blue. We call G *red-blue graph*. Let E_R and E_B be the red, resp. blue edges of G. By $G_R = (V, E_R)$, resp. $G_B = (V, E_B)$ we denote the subgraphs consisting of all the red, resp. blue, edges of G. We call a red-blue graph *monochromatic* if all its edges have the same color.

The *square of* G is the graph $G^2 = (V, E^2)$, where $E^2 = \{ (u, w) \mid$ there exists a $v \in V$ such that $(u, v) \in E$ and $(v, w) \in E \}$.

For a unified treatment of the general and the bipartite case, we call a graph G *full* if G is a complete graph K_n or a complete bipartite graph $K_{m,n}$. We call a full graph G *balanced* if it is a K_{2n} or a $K_{n,n}$. For a set V of nodes we also write K_V to denote the complete graph on V. Similarly, for a partition U, W of the nodes, we write $K_{U,W}$ to denote the complete bipartite graph according to partition U, W.

Problems. We define the problems considered in this paper.

- *Perfect matching (PM)*: Given a graph G, decide if G has a perfect matching.
- *Maximum matching (MM)*: Given a graph G and a number k, decide if there is a matching of size $\geq k$.
- *Weighted perfect matching (wPM)*: Given a graph G, a weight function $w : E \to \mathbb{N}$ and threshold weight W, decide if there is a perfect matching of weight $\geq W$.
- *Weighted maximum matching (wMM)*: Given a graph G, a weight function $w : E \to \mathbb{N}$ and threshold weight W, decide if there is a matching of weight $\geq W$.
- *Weighted exact perfect matching (wxPM)*: Given a graph G, a weight function $w : E \to \mathbb{N}$ and threshold weight W, decide if there is a perfect matching in G of weight exactly W.

- *Exact perfect matching (xPM)*: Given a red-blue graph G and a number r, decide if there is a perfect matching in G with exactly r red edges.
- *Complete exact perfect matching (cxPM)* is the same as *xPM*, but restricted to complete graphs K_n and $K_{n,n}$.

We assume here that the input graph G is given by its adjacency matrix. In case of a weighted or colored graph, the weights and colors are encoded in the adjacency matrix.

For the weight function $w : E \to \mathbb{N}$ in the above problems we assume a unary encoding for the weights. Equivalently one may allow a binary encoding with the restriction that weights are bounded by a fixed polynomial in $|V|$. Note that *wxPM* with unrestricted binary weights is NP-complete [PY82, GKM+11]. However *wMM* with unrestricted binary weights is in P [Edm65].

The other extreme is to allow only weights from the set $\{0, 1\}$. We indicate by $w_{01}xPM, w_{01}PM$ and $w_{01}MM$, the problems *wxPM*, *wPM* and *wMM* respectively with weights in the set $\{0, 1\}$. Observe that $w_{01}xPM$ is the same problem as *xPM* if one identifies red edges with weight 1 edges and blue edges with weight 0 edges.

For each of the above decision problems there is a corresponding *construction version*. For example, in the construction version of *MM* one has to construct a matching of size $\geq k$, or output that there is no such matching. For *MM*, *wPM*, *wMM* we also consider their *optimization versions*. Here, the constructed matching has to be of maximum cardinality, resp. weight. Hence, the threshold parameter k, resp. W, is omitted from the input.

Reductions and complexity classes. Let $\mathcal{C}$ be a class of functions and A, B be two problems. Then A is $\mathcal{C}$ *many-one reducible to* B if there is a function $f \in \mathcal{C}$ such that $x \in A \iff f(x) \in B$.

When we consider the construction versions of A and B, we say that A is $\mathcal{C}$ *many-one reducible to* B if there are two functions $f, g \in \mathcal{C}$ such that $x \in A \iff f(x) \in B$ and for any certificate w that $f(x) \in B$, we have that $w' = g(x, w)$ is a certificate for $x \in A$.

We will consider the following complexity classes:

- AC^0 is the class of circuit families with unbounded fan-in and- and or-gates, polynomial size, and constant depth.
- TC^0 is defined as AC^0 with additional unbounded threshold gates.
- NC^1 is the class of circuit families with fan-in 2 and- and or-gates, polynomial size, and logarithmic depth.

- L and P are the classes of problems that can be decided by logarithmic-space bounded resp. polynomial-time bounded Turing machines.

It is known that

$$\mathrm{AC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{P}.$$

We use the same notation for the functional versions of the classes. It will be clear from the context whether we consider functions. We will also consider *Turing reductions* to decision, construction and optimization versions. In case of Turing reductions to construction or optimization versions the answer to a query will also consist of an appropriate certificate. See [Vol99] for some more details on reductions and the considered complexity classes.

For later use, we collect some problems that can be solved within these complexity class.

Lemma 2.1. *Given a graph G. In* AC^0 *one can decide whether each connected component of G is full and, in this case, compute the components. If G is bipartite, then also the bipartition can be computed in* AC^0*.*

In TC^0 *one can decide whether all components of G are balanced.*

Proof. For each vertex v verify in parallel whether the component that contains v is full.

- The component is complete if for each $u \in \Gamma(v)$ we have $\Gamma(u) - \{v\} = \Gamma(v) - \{u\}$. Note that $\Gamma(v)$ can be read from the vth row in the adjacency matrix.
- The component is bipartite complete if $\Gamma(u) = \Gamma(v)$ for each $u \in \Gamma(v)$, and for each $w_1, w_2 \in \Gamma(v)$ we have $\Gamma(w_1) = \Gamma(w_2)$.

To compute a list of the components (and their possible bipartition) without duplicates additionally verify that v is the smallest vertex in its component. All this can be done in AC^0.

To verify whether the component that contains v is balanced, additionally check whether the number of nodes $|E(v) \cup \{v\}|$ is even, in case the component is complete, respectively check whether $|E(v)| = |E(w_1)|$, in case the component is bipartite. This can be done in TC^0. □

Lemma 2.2. *There are logspace algorithms that on input of a graph G*

1. *compute a list of its connected components,*
2. *decide whether G is bipartite and, in this case, compute a bipartition of G.*

Proof. For the first claim assume some order on the vertices of $G = (V, E)$. For each vertex $v \in V$ loop through all vertices w and test whether v is the smallest vertex reachable from v. Recall that undirected reachability is in L [Rei08]. If v is the smallest vertex reachable from v, then output all other vertices reachable from v as a connected component.

This way we obtain a list of the connected components of G, ordered by the minimal vertex they contain. Clearly, on can transform them to the corresponding subgraph in logarithmic space. Let $G_1, \ldots, G_l$ be the of connected components of G.

To decide whether G is bipartite compute for each connected component $G_i = (V_i, E_i)$ the graph G_i^2. Note that G_i is bipartite if, and only if, it is a single node or G_i^2 has exactly two components. This can be decided in logspace by part 1 of the lemma. We also obtain the components $V_{i,1},\ V_{i,2}$ of G_i^2 that form a bipartition of G_i. Then $V_j = \bigcup_{1 \leq i \leq l} V_{i,j}$, for $j \in \{1, 2\}$, is the bipartition of G. $\square$

Lemma 2.3. *Given a full graph G with n vertices and a matching M of size k in G. An extension of M to a maximum matching can be computed in* TC^0.

Proof. Let V_M be the vertices covered by M, For $G = K_n$, we extend M by $\lfloor \frac{n}{2} \rfloor - k$ pairs of nodes from $V - V_M$. These pairs are found by sorting $V - V_M$ and pairing consecutive vertices. Note that sorting can be done in TC^0, see [Vol99].

For $G = K_{n,m}$, let $n \leq m$ and $V = (U, W)$ be the bipartition of the nodes. We sort $U - V_M$ and $W - V_M$ separately and extend M by pairing the ith vertices in both sets for $1 \leq i \leq n - k$. $\square$

If the graph G in Lemma 2.3 is balanced, then the maximum matching will be a perfect matching.

3 A Chain of Reductions

In this section we prove reductions between the various matching problems which puts them in a reduction chain. The borderline with respect to complexity lies between *wPM* and *xPM*: *wPM* is in P whereas the complexity of *xPM* is still unclear. RNC is an upper bound for it [MVV87]. All reductions are logspace (in fact, AC^0) many-one reductions.

Theorem 3.1. *Via* AC^0 *many-one reductions, for both decision and construction, we have*

$$PM \equiv MM \equiv w_{01}MM \leq wMM \equiv wPM \equiv w_{01}PM \leq xPM \equiv wxPM.$$

Proof. We work our way from left to right in the chain of reductions. We just give the proof for the decision versions, the proof for the construction versions is always an obvious extension.

(i) *$PM \equiv MM \equiv w_{01}MM$.* The reduction from *PM* to *MM* is straightforward: let G be the input graph with n nodes, then $G \in PM \iff (G, \lceil n/2 \rceil) \in MM$.

To reduce *MM* to *PM*, let $G = (V, E)$ be the input graph with n nodes and $k \geq 1$. Add $n - 2k$ new vertices and add an edge between each new vertex and each node of G. Call the new graph G'. Then $(G, k) \in MM \iff G' \in PM$.

The reduction $MM \leq w_{01}MM$ is straightforward: define the weight function w to be 1 for every edge. For the reverse reduction $w_{01}MM \leq MM$, just remove the weight function and the weight 0 edges.

(ii) *$w_{01}MM \leq wMM$.* Clearly, $w_{01}MM$ is just a special case of *wMM* which means that the identity function will do as a reduction.

(iii) *$wMM \equiv wPM \equiv w_{01}PM$.* To show that $wMM \leq wPM$, add a new node v to the input graph G if the number of nodes is not even and add new edges with weight 0 to make the graph complete. Let the new graph be G' and w' be the extended weight function. Now any matching of weight $\geq W$ in G can be extended to a perfect matching of weight $\geq W$ in G', and vice versa (cf. the reduction $MM \leq wPM$ in [KUW86]). Hence $(G, w, W) \in wMM \iff (G', w', W) \in wPM$.

To show that $wPM \leq wMM$ we use a standard technique that guarantees that each matching of a certain minimum weight is perfect (see, e.g., [Yus12]). Let G have $2n$ vertices. Define a new weight function w' by $w'(e) = w(e) + nw_0$ for any $e \in E$, where $w_0 \geq \max\{\, w(e) \mid e \in E\,\}$ (for simplicity, w_0 can be the length of the unary encoded input G, w, W). According to the new weight function a non-perfect matching has weight $\leq (n-1)(w_0 + nw_0) = (n^2 - 1)w_0$. A perfect matching of weight W according to w will have weight $\geq W + n^2 w_0$ according to w'. Hence $(G, w, W) \in wPM \iff (G, w', W + n^2 w_0) \in wMM$.

To see $wPM \leq w_{01}PM$, replace each edge e in a given polynomially weighted graph G with a simple path of length $2w(e) - 1$ such that the edges on the path have weight 1 and 0 alternatingly (beginning with weight 1). Call the new 01-weighted graph G'. Then there is a direct correspondence between the perfect matchings in G and G' of the same weight. The reduction $w_{01}PM \leq wPM$ is simply an identity mapping.

(iv) *$wPM \leq wxPM$.* We first describe a weighted graph H_t that has a perfect matching of weight s for $0 \leq s \leq 2^t - 1$: H_t consists of t disjoint length 4 cycles where the i-th cycle has one edge of weight 2^{i-1} and three edges of weight 0, for $i = \{1, 2, \ldots, t\}$.

Let (G, w, W) be an input to *wPM*. Let $w_0 \geq \max\{\, w(e) \mid e \in E\,\}$ and fix $k \geq \log(|V| \cdot w_0)$. Let $G' = G \cup \bigcup_{i=0}^{k} H_t$ be the disjoint union of the graphs G and H_t for $0 \leq t \leq k$, and let w' denote the weight function w extended to the edges in H_t as described above. Then it is clear that $(G, w, W) \in wPM \iff (G', w', W + 2^t - 1) \in wxPM$.

(v) $xPM \equiv wxPM$**.** We already mentioned above that *xPM* is identical to $w_{01}xPM$ which is a special case of *wxPM*. Therefore $xPM \leq wxPM$.

For the reverse reduction $wxPM \leq w_{01}xPM$ we use the same function as for the reduction $wPM \leq w_{01}PM$ from above. □

Note that the theorem also holds if we consider the bipartite versions of all the problems. Only the proof of $MM \leq PM$ needs an adjustment: in the bipartite case, let $G = (V_1 \cup V_2, E)$, where $|V_i| = n_i$ for $i = 1, 2$. Add $n - 2k$ vertices and connect $n_1 - k$ of them with all nodes in V_2, and $n_2 - k$ of them with all nodes in V_1. Call the new graph G'. Then G' is bipartite and $(G, k) \in MM \iff G' \in PM$.

If the bipartition is given as part of the input, as we usually assume, this is an AC^0-reduction. If the bipartition is not given, we can compute it in logspace by Lemma 2.2. Consequently we get a logspace-reduction in this case.

4 *PM* vs. *xPM*

By the chain of reductions in the previous section, *PM* is reducible to *xPM*. Whether there is a polynomial-time reduction in the reverse direction is a longstanding open problem. We approach this problem. We first show that with two queries to *wPM*, one can construct in logarithmic space an *exact pseudo perfect matching* (see definition below), an intermediate problem that comes already very close to an exact perfect matching.

For this construction we use ideas from Yuster [Yus12]. Instead of a perfect matching with r red edges, Yuster constructs a matching that has r red edges but may have one edge less than a perfect matching. Our construction widely generalizes and simplifies a polynomial-time construction from Yi, Murty, and Spera [YMS02] that works only for complete bipartite graphs.

The second step in [YMS02] turns the exact pseudo perfect matching in a complete bipartite graph into an exact perfect matching in polynomial time. We use ideas from [GS11] and [Kar87] and show that this can be done uniformly for bipartite and non-bipartite complete graphs. Moreover, our construction provides a logspace reduction to *PM*.

4.1 Definitions

In the rest of this section, $G = (V, E)$ is a red-blue graph that has $|V| = 2n$ vertices. An *l-cycle* is a cycle with l edges. An *(r, b)-cycle* is a cycle with r

red edges and b blue edges.

If G has a perfect matching, we denote by M_R and M_B some perfect matching in G with the maximum number of red edges and the maximum number of blue edges, respectively. By $r_{\max}$ and $r_{\min}$ we denote the number of red edges in M_R, resp. M_B. Note that a perfect matching with r red edges will have $n - r$ blue edges. A perfect matching in G with exactly r red edges is called *r-perfect matching* or *r-pm*, for short.

A *pseudo perfect matching* P is a subset of edges of G of size $n = |V|/2$ such that at most one node is covered by exactly two edges, where one edge is red and the other is blue. This node is called the *bad node*. If there is a bad node, then there is one node that is not covered, which is called an *exposed node*. All other nodes are covered by exactly one edge. If P has r red edges, it is also called an *r-pseudo perfect matching* or *r-ppm*, for short. Note that an r-ppm has $n - r$ blue edges.

4.2 Constructing an exact pseudo perfect matching

Let G be a red-blue graph such that there are perfect matchings in G. Then there are perfect matchings M_B and M_R with $r_{\min}$ and $r_{\max}$ red edges, respectively. Clearly, we must have $r_{\min} \le r \le r_{\max}$ for an r-perfect matching to exist. Our first step is to show that there always is an r-pseudo perfect matching.

Theorem 4.1. *Let G be a red-blue graph that has a perfect matching and $r \ge 0$.*

$$r_{\min} \le r \le r_{\max} \implies G \text{ has an } r\text{-ppm } P.$$

Furthermore, P can be computed in logspace with two queries to the construction version of wPM, or with one query to the optimization version of wPM.

Proof. Since G has a perfect matching, M_R and M_B are defined. If $r = r_{\min}$ or $r = r_{\max}$, we can set $M = M_B$ or $M = M_R$, respectively, and are done. It remains to consider the case when $r_{\min} < r < r_{\max}$.

Consider the graph $M_R \triangle M_B$. The components are disjoint simple cycles $C_1, C_2, \ldots, C_k$ of even length ≥ 4, where the edges in each cycle are alternately from M_R and M_B. For convinience, the C_i's are defined here as sets of edges.

To construct the r-pseudo perfect matching, we start with $M_0 = M_B$, which has $< r$ red edges. Then we successively swap the edges on cycles $C_1, C_2, \ldots$. That is, we consider the perfect matchings

$$M_i = M_B \triangle (C_1 \cup C_2 \cup \cdots \cup C_i)$$

for $i = 0, \ldots, k$. Observe that $M_k = M_R$, which has $> r$ red edges. Hence there must be an intermediate point $i_0 < k$ such that M_{i_0} has $< r$ red edges and M_{i_0+1} has $\geq r$ red edges. In the lucky case, M_{i_0+1} has exactly r red edges and we are done. So assume that M_{i_0+1} has $> r$ red edges.

Let us denote $C = C_{i_0+1}$. We construct the r-ppm out of M_{i_0} and C. The edges of cycle C can be split into two parts, $C^0 = C \cap M_{i_0}$, which are the edges from M_B, and the remaining edges $C^1 = C - M_{i_0}$ which come from M_R. By the construction, C^1 has strictly more red edges than C^0. Therefore there must be a red edge in C^1 which is adjacent to a blue edge in C^0. Let us denote

$$\begin{aligned} C &= \{(u_0, u_1),\ (u_1, u_2),\ \ldots,\ (u_{2l-1}, u_0)\} \\ C^0 &= \{(u_0, u_1),\ (u_2, u_3),\ \cdots,\ (u_{2l-2}, u_{2l-1})\} \\ C^1 &= \{(u_1, u_2),\ (u_3, u_4),\ \cdots,\ (u_{2l-1}, u_0)\}, \end{aligned}$$

such that (u_0, u_1) is blue and (u_1, u_2) is red.

We define a ppm P_3 by adding (u_1, u_2) to M_{i_0} and removing (u_2, u_3). Then u_1 becomes the bad node and u_3 becomes exposed.

$$P_3 = (M_{i_0} \cup \{(u_1, u_2)\}) \ - \ \{(u_2, u_3)\}.$$

The number of red edges in P_3 is either the same as in M_{i_0}, if (u_2, u_3) is red, or increases by one, if (u_2, u_3) is blue. Hence the number of red edges in P_3 is $\leq r$.

Now we successively increase the C^1-part of the ppm by swapping more edges of cycle C. This results in moving the exposed vertex. The next step is to add $(u_3, u_4) \in C^1$ to P_3 and to remove $(u_4, u_5) \in C^0$. Then u_5 becomes the exposed node and we get ppm P_5,

$$P_5 = (P_3 \cup \{(u_3, u_4)\}) \ - \ \{(u_4, u_5)\}.$$

It is possible that the number of red edges decreases when going from P_3 to P_5. But because we only swap two edges, the number of red edges in P_3 and P_5 differ by ≤ 1. Continuing that way, we finally have ppm's $P_3, P_5, P_7, \ldots, P_{2l-1}$, with exposed node $u_3, u_5, u_7, \ldots, u_{2l-1}$, respectively, and the number of red egdes in successive ppm's differ by ≤ 1.

Let us consider the last ppm, P_{2l-1}. Observe that P_{2l-1} almost agrees with perfect matching M_{i_0+1}, they only differ on edges (u_0, u_1) and (u_{2l-1}, u_0),

$$M_{i_0+1} = (P_{2l-1} \cup \{(u_{2l-1}, u_0)\}) \ - \ \{(u_0, u_1)\}.$$

Recall that (u_0, u_1) is blue. Therefore the number of red edges in P_{2l-1} is either the same or one less than the number of red edges in M_{i_0+1}. Hence the number of red edges in P_{2l-1} is $\geq r$. It follows that at least one of the ppm's P_j constructed above has exactly r red edges.

Complexity: Assume first that we have given M_R and M_B. Applying Lemma 2.2 to $(V, M_R \triangle M_B)$, the cycles $C_1, C_2, \ldots, C_k$ can be computed in logspace. The remaining operations in the above argument can be performed in logspace as well.

The queries to *wPM* are as follows. Define the weight function w_R as

$$w_R(e) \quad = \quad \begin{cases} 0 & \text{if } e \text{ is red} \\ 1 & \text{if } e \text{ is blue} \end{cases}$$

Then the query $(G, w_R, n-r)$ to the construction version of *wPM* gives a perfect matching N_1 in G with $\geq n-r$ blue edges. Therefore N_1 has $r_1 \leq r$ red edges.

Similarly we define the weight function w_B as $w_B(e) = 1$, if e is red, and 0 otherwise. Then the query (G, w_B, r) to the construction version of *wPM* gives a perfect matching N_2 in G with $r_2 \geq r$ red edges.

Now observe that the above proof for the existence of ppm P works as well if we use N_1 and N_2 instead of M_B and M_R. This shows that P can be constructed with two queries to the construction version of *wPM*. Note that with r_1 and r_2 in hand, we can also verify the condition $r_{\min} \leq r \leq r_{\max}$.

To combine the two queries into one query, define the graph G' as the disjoint union of two copies of G. Define the weight function w' to be w_R on the first copy of G and w_B on the second copy. Then the single query (G', w') to the optimization version of *wPM* will give us a perfect matching in G' which consists of M_R in the first copy of G and of M_B in the second copy. $\square$

If we consider balanced graphs instead of arbitrary graphs, the complexity bound in Theorem 4.1 can be slightly improved. In a balanced graphs any matching can be extended to perfect matching. In an arbitrary graph this might not be possible. Moreover, Lemma 2.3 states that such an extension can be computed efficiently in balanced graphs.

We show that the two queries to *wPM* in Theorem 4.1 can be replaced by two queries to *MM* which in turn can be replaced by one query to *PM*.

We define

- *Complete exact pseudo perfect matching (cxPPM)*: Given a red-blue graph G and a number r, verify that G is balanced and has an r-ppm.

Corollary 4.2. *Let G be a balanced red-blue graph and $r \geq 0$.*

$$G \in cxPPM \iff r_{\min} \leq r \leq r_{\max}.$$

Furthermore, $cxPPM \leq PM$. The many-one reduction is in AC^0 *for the decision version and in logspace for the construction version.*

Proof. By Theorem 4.1 it suffices to show the direction from left to right. Let G be balanced and P be an r-ppm in G. The r red edges of P form a matching in G_R and the $n-r$ blue edges of P form a matching in G_B.

Therefore $(G_R, r), (G_B, n-r) \in MM$. As explained above, one can extend a matching with $\geq r$ red edges in the balanced graph G to a perfect matching with $\geq r$ red edges in G. Therefore we have

$$(G_R, r) \in MM \iff r_{\min} \leq r.$$

Similarly, a matching with $\geq n-r$ blue edges can be extended to a perfect matching in G with $\geq n-r$ blue edges, and hence $\leq r$ red edges. Therefore

$$(G_B, n-r) \in MM \iff r \leq r_{\max}.$$

We show $cxPPM \leq PM$. Let G be a given graph. We first check that G is full. This can be done in AC^0 by Lemma 2.1. Then we have

$$G \in cxPPM \iff (G_B, n-r), (G_R, r) \in MM \text{ and } G \in PM.$$

By Theorem 3.1, $MM \leq PM$. Hence we can compute graphs G_1 and G_2 in AC^0 such that $(G_B, n-r), (G_R, r) \in MM \iff G_1, G_2 \in PM$. Define $G' = G_1 \cup G_2 \cup G$. Then we have $G \in cxPPM \iff G' \in PM$.

To construct an r-ppm in G from a perfect matching M' in G', we split M' into perfect matchings for G_1, G_2 and G. From these one obtains matchings M_1' in G_B of size $\geq n-r$ and M_2' in G_R of size $\geq r$. By Lemma 2.3 we can extend M_1' and M_2' in TC^0 to an r_1-pm N_1 and an r_2-pm N_2 of G with $r_1 \leq r \leq r_2$. Using the construction in the proof of Theorem 4.1 we obtain an r-ppm for G. $\square$

4.3 Constructing an exact perfect matching in full graphs

We show that *cxPM* is many-one reducible to *PM* in logspace. What remains to do is to reduce the exact pseudo perfect matching from the previous subsection to exact perfect matching in logspace. We use ideas from [GS11] and [Kar87] and show that this can be done uniformly for bipartite and non-bipartite complete graphs.

The following definition partitions full graphs into four classes. The classes are already considered implicitely in Karzanov [Kar87] and are defined explicitely by Geerdes and Szabó [GS11]. Unlike [GS11], we keep the classes disjoint.

Definition 4.3. *Let G be a balanced graph. We write $G \sim (\mathrm{c}x)$ if G is of class $(\mathrm{c}x)$ for $x \in \{1, 2\mathrm{r}, 2\mathrm{b}, 3\}$, where the classes are defined as follows.*

- *Class* (c1)*: All components of G_R and G_B are full.*
- *Class* (c2r)*: $G \not\sim (\mathrm{c1})$ and all components of G_R are balanced.*
- *Class* (c2b)*: $G \not\sim (\mathrm{c1})$ and all components of G_B are balanced.*
- *Class* (c3)*: $G \not\sim (\mathrm{c1}), (\mathrm{c2r}), (\mathrm{c2b})$.*

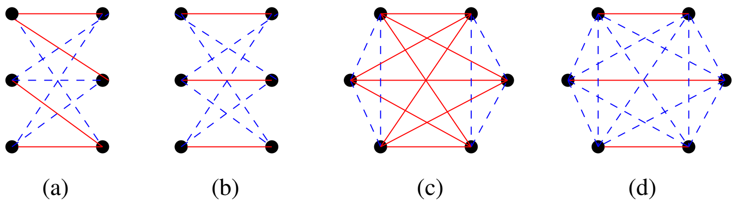

Figure 1: (a) A bipartite complete graph in class (c1). (b) A bipartite complete graph in class (c2r). (c) A complete graph in class (c1). (d) A complete graph in class (c2r). Solid and dashed lines represent red and blue edges, respectively.

See Figure 1 for some examples. By Lemma 2.1 one can determine in TC^0 to which of the classes (c1), (c2r), (c2b), or (c3) a given graph belongs.

We start by considering graphs in class (c1).

Lemma 4.4. *A balanced graph $G \sim$ (c1) is of one of the following forms.*

- *If G is bipartite with bipartition U, W then there is a partition $U = U_1 \cup U_2$ and $W = W_1 \cup W_2$ such that $G_R = K_{U_1,W_1} \cup K_{U_2,W_2}$ and $G_B = K_{U_1,W_2} \cup K_{U_2,W_1}$.*
- *If G is complete then there is a partition $V = V_1 \cup V_2$ such that one of G_R or G_B is $K_{V_1} \cup K_{V_2}$ while the other is K_{V_1,V_2}.*

Proof. Note that the sets U_i, V_i, W_i may also be empty, for $i = 1, 2$. For the correctness of the charcaterization of class (c1) observe that G_R and G_B cannot have ≥ 3 full components with at least two vertices, respectively. Assume that, say, G_B has ≥ 3 components, and let $(u_0, v_0), (u_1, v_1), (u_2, v_2)$ be edges in G_B from three different components. Then, for $i \neq j$, all edges (u_i, v_j) are red. Therefore the six nodes $u_0, v_0, u_1, v_1, u_2, v_2$ are in one component in G_R. But this component is non-full because edges (u_i, v_i) are blue, for $i = 0, 1, 2$. $\square$

As a consequence of this description we get the following lemma.

Lemma 4.5. *Every even length simple cycle in a graph $G \sim$ (c1) has an even number of red edges.*

With these observations we can completely resolve the exact perfect matching problem for graphs of class (c1). I.e., for this class we do not need the pseudo perfect matching from above. The existential part was shown by Karzanov [Kar87], see also [GS11]. We add the complexity bound on the construction.

Theorem 4.6. *Let $G \sim$ (c1).*

$$G \text{ has an } r\text{-pm} \iff r_{\min} \leq r \leq r_{\max} \text{ and } r \equiv r_{\max} \pmod 2.$$

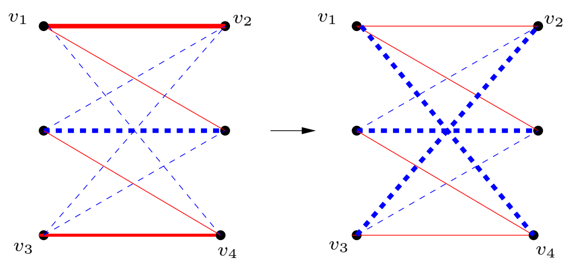

Figure 2: From a 2-pm we get a 0-pm by swapping two matched pairs.

Decision and construction of an r-perfect matching is in TC^0.

Proof. Let M be an r-perfect matching. The symmetric difference $M \triangle M_R$ consists of disjoint even simple cycles in G. By Lemma 4.5 each of these cycles has an even number of red edges. Therefore $r \equiv r_{\max} \pmod 2$.

For the reverse direction, consider the partition of the nodes of G as described above, i.e., of U and W into U_1, U_2, W_1, W_2 if G is bipartite, and of V into V_1, V_2 if G is complete. In case that any of these sets is empty, the problem becomes trivial: then we have $r_{\min} = r_{\max}$, i.e., all perfect matchings in G have the same number of red edges. In the following we assume that none of these sets is empty in either case.

We consider the case that G_R has two full components. Otherwise G_B has two full components and an analogous argument works for G_B. We start with a maximum red perfect matching M_R in G that has $r_{\max}$ red egdes. Take two red edges $e_1 = (v_1, v_2)$ and $e_3 = (v_3, v_4)$ in M_R, one from each component of G_R. Then the edges $e_2 = (v_2, v_3)$ and $e_4 = (v_4, v_1)$ are both blue. Now we swap the edges on the cycle (e_1, e_2, e_3, e_4): remove e_1, e_3 from M_R and instead add edges e_2, e_4. The resulting perfect matching has $r_{\max} - 2$ red egdes. See Figure 2 for an example.

We iterate this process until the current perfect matching has no more red egde in one of the components of G_R. Then we have reached a perfect matching a maximum number of blue edges, and therefore with $r_{\min}$ many red egdes. Hence at some intermediate stage, we have an r-perfect matching.

Complexity: We first show how to compute M_R. By Lemma 2.1 we can compute the components of G_R and G_B and check, whether we are in a trivial case, i.e., whether G_R or G_B has no edges. In this case, M_R can be computed by Lemma 2.3. Suppose now that G_R has two full components, otherwise we work with G_B instead of G_R. To obtain M_R, we first compute maximum matchings in G_R in each component of G_R by Lemma 2.3. The union of the two matchings is a maximum matching M in G_R. Then we extend M to a perfect matching M_R in G. This can be done again by Lemma 2.3, because the extension is from G_B which is a full graph.

To compute an r-perfect matching, we do the swapping of red and blue edges in the cycles in parallel. Let $k = r_{\max} - r$. Note that k is even. We choose k red edges in M_R, $k/2$ in each component of G_R. To do so, we sort the red egdes of M_R in each component of G_R and then pair up the i-th edges in each component, for $i = 1, \ldots, k/2$. Swapping the edges in all cycles in parallel gives the final r-perfect matching. $\square$

Next we consider graphs in classes (c2r) and (c2b) where a component of G_B, respectively G_R is not full. In [GS11] it is shown that these classes can be detected by looking at 4-cycles, i.e. cycles of length 4 in G_R and G_B. We give a simplified proof of this fact. Recall that an (r, b)-cycle is a cycle with r red and b blue edges.

Lemma 4.7. *Let $G = (V, E)$ be a full red-blue graph. Then*

1. *G_B has a non-full component $\iff$ there exists a $(1, 3)$-cycle in G.*
2. *G_R has a non-full component $\iff$ there exists a $(3, 1)$-cycle in G.*

Proof. We show the first statement. The proof for the second one is analogous. We start with the direction from right to left. Let (v_1, v_2, v_3, v_4) be a $(1, 3)$-cycle in G with red edge (v_1, v_4). Then v_1, v_2, v_3, v_4 all lie in the same component of G_B. In case that this component is bipartite, v_1 and v_4 lie in different partitions. Since (v_1, v_4) is red, this component of G_B can neither be a complete graph nor a complete bipartite graph.

For the reverse direction, let C_B be a component of G_B that is not full, and let G'_B be the graph induced by G_B on component C_B.

First we consider the case when G'_B is bipartite. Since G'_B is not complete, there are $u, v \in C_B$ such that the edge (u, v) is red. Because u and v are in the same component C_B, there is a shortest blue path $(u, u_1, \ldots, u_k, v)$ from u to v in G'_B. Because it is a shortest path, edges (u, u_i) are red, for all $i \in \{2, \ldots, k\}$ such that $(u_1, u_i) \in E$. Note that if G is bipartite, edge (u_1, u_i) exists in G only for odd i. Since u and v are in different partitions of G'_B, k must be even. If $k = 2$, then (u, u_1, u_2, v) is a $(1, 3)$-cycle with red egde (u, v). If $k \geq 4$, then (u, u_1, u_2, u_3) is a $(1, 3)$-cycle with red egde (u, u_3).

Now, let us consider the case when component G'_B is non-bipartite. Note that then G is also non-bipartite. Let $C = (u_1, u_2, \ldots, u_k)$ be a blue odd cycle in G'_B of smallest length. If the edge (u_1, u_i) is blue for any $i \in \{3, \ldots, k - 1\}$ then either $(u_1, u_2, \ldots, u_i)$ or $(u_1, u_i, u_{i+1}, \ldots, u_k)$ would be a blue odd cycle. This would contradict the fact that C is the smallest blue odd cycle. Hence, all edges (u_1, u_i) must be red, for $i = 3, \ldots, k - 1$. If $k \geq 5$, then (u_1, u_2, u_3, u_4) is a $(1, 3)$-cycle with red egde (u_1, u_4).

It remains to consider the case $k = 3$, i.e., $C = (u_1, u_2, u_3)$ is a triangle in G'_B. Because G'_B is not complete there must be vertices in C_B other than the triangle vertices of C.

- *Case 1*: All vertices in C_B are connected to triangle C by a blue edge. Since G'_B is not complete there are vertices $v, w \in C_B$ such that the edge (v, w) is red. Then (v, u_1, u_2, w) is a $(1, 3)$-cycle.

- *Case 2*: There is a vertex $v_0 \in C_B$ connected to triangle C by a red edge, say to u_1. If (v_0, u_2) or (v_0, u_3) is blue then (u_1, u_2, u_3, v_0), resp. (u_1, u_3, u_2, v_0), is a $(1,3)$-cycle with red egde (v_0, u_1).

  It remains the case when edges (v_0, u_2) and (v_0, u_3) are red as well. Since v_0 is in C_B there is a blue path that connects v_0 to C, say to u_3. The cases when the path ends in u_1 or u_2 instead are analogous. Let $(v_0, v_1, v_2, \ldots, v_k, u_3)$ be a shortest path in G'_B, for some $k \geq 1$. I.e., all edges (v_i, u_3) are red, for $i = 0, \ldots, k-1$, because otherwise there would be a shorter blue path from v_0 to u_3.

  - If $k = 1$ then (v_0, v_1, u_3, u_1) is a $(1,3)$-cycle with red edge (v_0, u_1).
  - If $k \geq 2$ then $(v_{k-2}, v_{k-1}, v_k, u_3)$ is a $(1,3)$-cycle with red edge (v_{k-2}, u_3).

$\square$

Recall that a graph G is in class (c2r), if G_R is full, in fact balanced, and G_B is not full. By Lemma 4.7, G must have a $(1,3)$-cycle, but no $(3,1)$-cycle. Because G_R is balanced, there must be red perfect matching in G, i.e., $r_{\max} = n$. The case $G \sim$ (c2b) is similar. On the other hand, if G neither has $(1,3)$-cycle nor a $(3,1)$-cycle, then G_R and G_B are both full, and hence G is in class (c1).

Corollary 4.8. *Let G be a balanced red-blue graph. Then*

1. $G \sim$ (c1) $\iff$ *every 4-cycle in G has an even number of red edges.*
2. $G \sim$ (c2r) $\iff$ *G has a $(1,3)$-cycle, but no $(3,1)$-cycle, and $r_{\max} = n$.*
3. $G \sim$ (c2b) $\iff$ *G has a $(3,1)$-cycle, but no $(1,3)$-cycle, and $r_{\min} = 0$.*

The next lemma shows that when graph G is not in class (c1) then an r-perfect matching always exists whenever $r_{\min} \leq r \leq r_{\max}$, barring a few special cases. Namely, the case when $r = 1$ or $r = n-1$ are handled separately in Lemma 4.10 below, and we now consider the case where $2 \leq r \leq n-2$. Yi, Murty, and Spera [YMS02] proved the lemma for bipartite complete graphs. We extend the proof to the non-bipartite case. For the complexity bound, we show that an r-perfect matching can be constructed from an r-pseudo perfect matching of G. Recall that the latter can be computed with one query to *PM*, by Corollary 4.2.

Lemma 4.9. *Let $G \not\sim$ (c1) be a balanced graph with $2n$ nodes, where $n \geq 4$. Let $r_{\min} \leq r \leq r_{\max}$ and also $2 \leq r \leq n-2$. Then G has an r-perfect matching. It can be constructed from an r-ppm of G in* AC^0.

Proof. Let $G = (V, E)$ be a balanced graph with $2n$ nodes, $n \geq 4$, $r_{\min} \leq r \leq r_{\max}$ and also $2 \leq r \leq n-2$. The proof is by induction on n.

In the base case $n = 4$, only the case $r = 2$ needs to be considered because of the restrictions on r. Let $V = \{u_1, u_2, u_3, u_4, v_1, v_2, v_3, v_4\}$. In the bipartite case, u_i's and v_i's will be the two partitions.

As $G \not\sim$ (c1), G has a $(3,1)$-cycle or a $(1,3)$-cycle by Corollary 4.8. We consider the case that G has a $(3,1)$-cycle. The case of a $(1,3)$-cycle is analogous. Let (u_1, v_1, u_2, v_2) be a $(3,1)$-cycle with (u_2, v_2) being blue. We will show that in every case how the remaining edges are colored, there is 2-pm in G.

Assume first that (u_3, v_3) is blue.

- If (u_4, v_4) is blue then $\{(u_1, v_2), (u_2, v_1), (u_3, v_3), (u_4, v_4)\}$ is a 2-pm.
- If (u_4, v_4) is red then $\{(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4)\}$ is a 2-pm.

Hence, we assume that (u_3, v_3) is red. Analogously, we may assume that (u_3, v_4), (u_4, v_3) and (u_4, v_4) are red as well. In the non-bipartite case, similarly, we may assume that the edges (u_3, u_4) and (v_3, v_4) are also red. Hence G is purely red on $\{u_3, u_4, v_3, v_4\}$.

By our assumption $r_{min} \leq 2$. If $r_{min} = 2$ then there is a 2-pm. So, let us assume that $r_{min} < 2$. This implies that there exist three independent blue edges in G. Clearly, at least one of these three edges is independent of (u_2, v_2). By symmetry, we may w.l.o.g. assume that this edge is (u_1, v_3).

Now, if (u_3, v_1) is red then $\{(u_1, v_3), (u_2, v_2), (u_3, v_1), (u_4, v_4)\}$ is a 2-pm. So, we assume that (u_3, v_1) is blue as well.

Similarly, we may assume that (u_4, v_1) is blue. Now, if (u_1, v_4) is red then $\{(u_1, v_4), (u_2, v_2), (u_3, v_1), (u_4, v_3)\}$ is a 2-pm. So, we assume that (u_1, v_4) is blue as well. Next, if (u_2, v_4) is blue then $\{(u_1, v_2), (u_2, v_4), (u_3, v_1), (u_4, v_3)\}$ is a 2-pm. So, we assume that (u_2, v_4) is red. Similarly, (u_4, v_2) is red. Now, $\{(u_1, v_3), (u_2, v_4), (u_3, v_1), (u_4, v_2)\}$ is a 2-pm. Hence, we have shown the existence of a 2-pm in every case.

In the induction step we show

$$\text{there is no } r\text{-perfect matching in } G \implies G \sim (\text{c1}).$$

Let $n \geq 5$ and assume the statement is true for balanced graphs with $2(n-1)$ nodes.

By Theorem 4.1, there exist a r-pseudo perfect matching P in G. Let $P_R = P \cap E_R$ and $P_B = P \cap E_B$ be the red and blue part of P, respectively. Let u_0 be the bad node of P and u_1 be the exposed node. Let v_0 and v_1 be the two neighbors of u_0 such that $(u_0, v_0) \in P_R$ and $(u_0, v_1) \in P_B$. Observe that in the bipartite case u_0 and u_1 are in the same partition, since G is balanced (see Figure 3).

W.l.o.g. we assume that $r \geq n - r$ and hence $r \geq 3$. Otherwise we complement the colors in G and go for an $(n-r)$-perfect matching in the

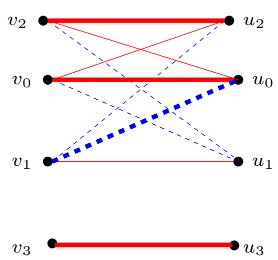

Figure 3: Showing the pseudo perfect matching P with $r = 3$. Solid and dashed lines represent red and blue edges, respectively. Bold lines represent matched edges.

complemented graph. Therefore there exist at least two further red edges $(u_2, v_2), (u_3, v_3) \in P_R$ apart from (u_0, v_0). Recall that u_1 is the exposed node in P. Hence the red egde (u_1, v_1) does not belong to P. In the bipartite case let u_2, u_3 be in the same partition as u_0, u_1.

Consider the graphs G' and G'':

$$\begin{aligned} G' &= G - \{u_2, v_2\} \\ G'' &= G - \{u_3, v_3\} \end{aligned}$$

Claim 1. $G', G'' \sim$ (c1).

Proof. We argue that the induction hypothesis applies to G'. Since G has no r-perfect matching, G' has no $(r-1)$-perfect matching. Let r'_{min} and r'_{max} be the minimum and maximum number of red edges of a perfect matching in G'. Let $P'_R = P_R - \{(u_2, v_2)\}$ and $P'_B = P_B$ and $P' = P'_R \cup P'_B$. Hence P' is an $(r-1)$-ppm in G'. Its monochromatic components P'_R and P'_B are matchings in G' of cardinality $r-1$ and $n-r$, respectively. P'_R and P'_B can be extended to perfect matchings in G' with $\geq r-1$ red edges and $\geq n-r$ blue edges, respectively. Therefore $r'_{\text{min}} \leq r-1 \leq r'_{\text{max}}$. Because $r \geq 3$ and also $r \leq n-2$ by assumption, we have $2 \leq r-1 \leq n-3$. Hence, by the induction hypothesis, $G' \sim$ (c1). Similarly $G'' \sim$ (c1). This proves Claim 1. $\square$

We collect some observations about the colors of some egdes.

Claim 2. *We have the following properties for the colors.*

(i) (u_i, v_i) *is red, for* $i \in \{0, 1, 2, 3\}$.

(ii) (u_i, v_j) *and* (u_j, v_i) *have the same color, for* $i, j \in \{0, 1, 2\}$. *In particular, because* (u_0, v_1) *is blue, also* (u_1, v_0) *is blue.*

(iii) (u_0, u_2) *and* (v_0, v_2) *have the same color.*

(iv) (u_2, v_0) *and* (u_2, v_1) *have different colors.*

Proof. Ad *(i)*: Let $N_4 = \{u_0, u_1, v_0, v_1\}$. Since G does not have an r-perfect matching by assumption, it should not be possible to modify P on N_4 to obtain an r-perfect matching. This would be the case if (u_1, v_1) would be blue, because then we could replace (u_0, v_1) by (u_1, v_1) in P.

Ad *(ii)*: Consider the 4-cycles (u_i, v_i, u_j, v_j) for $i, j \in \{0, 1, 2\}$. Since $G'' \sim$ (c1), every 4-cycle in G'' has either 2 or 4 red edges by Corollary 4.8. Because (u_i, v_i) and (u_j, v_j) are red, the other two edges must have the same color.

Ad *(iii)*: This concerns only the non-bipatite case because otherwise, these edges do not exist. In the 4-cycle (u_0, u_2, v_2, v_0) in G'', edges (u_0, v_0) and (u_2, v_2) are red. Again, the other two edges must have the same color.

Ad *(iv)*: Similarly, in the 4-cycle (u_0, v_0, u_2, v_1) in G'', the edge (u_0, v_0) is red and (u_0, v_1) is blue. Therefore the other two edges must have different colors. This proves Claim 2. □

W.l.o.g. let us fix one color, namely that (u_2, v_0) is red. In the case when (u_2, v_0) is blue, the proof is completely analogous. Then (u_0, v_2) is red as well, by Claim 2 *(ii)*, and (u_2, v_1) is blue, by Claim 2 *(iv)*. Again by *(ii)*, (u_1, v_2) is blue as well.

- (u_0, v_2) and (u_2, v_0) are red,
- (u_1, v_2) and (u_2, v_1) are blue.

Claim 3. *For all* $w \in V$

(i) (u_0, w) *and* (u_2, w) *have the same color, for* $w \neq u_0, u_2$*,*

(ii) (v_0, w) *and* (v_2, w) *have the same color, for* $w \neq v_0, v_2$*.*

Proof. We show the first claim, the proof for the second claim is analogous. Recall that in the bipartite case, the u-nodes are in the same partition. Hence either all u-nodes are connected to w, or none of them.

Consider the cycle C,

$$C = (u_0, w, u_2, v_2).$$

Suppose that (u_0, w) and (u_2, w) have different colors. Then C is a $(3, 1)$-cycle. By Corollary 4.8 applied to G'' this is not possible for $w \in G''$. Hence we have $w \in \{u_3, v_3\}$.

Let $N_8 = \{u_0, \ldots, u_3, v_0, \ldots, v_3\}$. Recall that P is a 3-ppm on the vertices of N_8. Therefore it should not be possible to modify P to a 3-pm on N_8 since otherwise one obtains an r-pm for G. Since C is a $(3, 1)$-cycle we can match its vertices with a 1-pm M_1 or a 2-pm M_2.

- If $w = v_3$, then $M_1 \cup \{(u_1, v_1), (u_3, v_0)\}$ is a 3-pm on N_8 if (u_3, v_0) is red, and $M_2 \cup \{(u_1, v_1), (u_3, v_0)\}$ is a 3-pm on N_8 if (u_3, v_0) is blue.
- If $w = u_3$, then $M_1 \cup \{(u_1, v_1), (v_3, v_0)\}$ is a 3-pm on N_8 if (v_3, v_0) is red, and $M_2 \cup \{(u_1, v_1), (v_3, v_0)\}$ is a 3-pm on N_8 if (v_3, v_0) is blue.

Hence we get a contradiction in both cases. This proves Claim 3. □

With the above claims and observations we can finally show that $G \sim$ (c1). We first consider the case when G is bipartite. Let D be a 4-cycle in G. Construct D' from D by replacing every occurence of u_2 by u_0, and every occurence of v_2 by v_0. Then D' is a 4-cycle in G'. Note that edges (u_0, u_2) and (v_0, v_2) do not exist in the bipartite case. By Claim 3, D' has the same colors on the respective edges as D. Since $G' \sim$ (c1), it follows that D', and hence D, has an even number of red edges. Therefore $G \sim$ (c1). Note that D' may be a non-simple cycle if u_0 or v_0 are in D. But the above argument ist still valid in this case.

It remains to consider the case when G is non-bipartite. Let G'_R and G'_B be the red and blue subgraph of $G' = (V', E')$, respectively. Because $G' \sim$ (c1), by Lemma 4.4, there is a partition $V' = V'_1 \cup V'_2$ such that the monochromatic parts of G' are

$$\begin{aligned} G'_1 &= K_{V'_1, V'_2} \\ G'_2 &= K_{V'_1} \cup K_{V'_2}\,. \end{aligned}$$

W.l.o.g. let us assume that $u_0 \in V'_1$.

- *Case 1*: $v_0 \in V'_1$. Then the red subgraph is not bipartite, i.e., we have $G'_1 = G'_B$ and $G'_2 = G'_R$. In particular, $K_{V'_1}$ is red. We have
  - $u_2, v_2 \in V'_1$, because (u_0, v_2) and (u_2, v_0) are red,
  - $u_1, v_1 \in V'_2$, because (u_0, v_1) and (u_1, v_0) are blue,
  - (u_0, u_2) is red, because otherwise (u_0, u_2, v_0, v_1) would be a $(1, 3)$-cycle in G''. Note that (v_0, v_1) is blue because $v_0 \in V'_1$ and $v_1 \in V'_2$.
  - (v_0, v_2) is red, because it has the same color as (u_0, u_2) by Claim 2 *(iii)*.

  Define $V_1 = V'_1 \cup \{u_2, v_2\}$ and $V_2 = V'_2$. We show that $G_B = K_{V_1} \times K_{V_2}$ and $G_R = K_{V_1, V_2}$, and therefore $G \sim$ (c1) by Lemma 4.4:

  - Edges (u_2, w) and (v_2, w) are red for every $w \in V_1$, because, by the above properies, (u_0, w) and (v_0, w) are red for every $w \in V_1$, and have the same color as (u_2, w) and (v_2, w) by Claim 3.

- Edges (u_2, w) and (v_2, w) are blue for every $w \in V_2$, because (u_0, w) and (v_0, w) are blue for $w \in V_2$ and have the same color as (u_2, w) and (v_2, w).

- *Case 2*: $v_0 \in V_2'$. Then the red subgraph is bipartite, i.e., we have $G_1' = G_R'$ and $G_2' = G_B'$. In particular, $K_{V_1', V_2'}$ is red. We have
  - $v_1 \in V_1'$, because (u_0, v_1) is blue,
  - $u_1 \in V_2'$, because (u_1, v_1) is red,
  - (u_0, u_2) is blue, because otherwise (u_0, u_2, v_0, v_1) would be a $(3,1)$-cycle in G''. Note that (v_0, v_1) is red because $v_0 \in V_2'$ and $v_1 \in V_1'$.

  Define $V_1 = V_1' \cup \{u_2\}$ and $V_2 = V_2' \cup \{v_2\}$. In an analogous way as in the first case, we get that $G_R = K_{V_1} \times K_{V_2}$ and $G_B = K_{V_1, V_2}$, and therefore $G \sim$ (c1).

Complexity: Since $G \not\sim$ (c1), G has a $(1,3)$- or $(3,1)$-cycle C which can be found in AC^0 by trying all possible 4-cycles. Let P be an r-ppm of G. If P does not have a bad node then we are done. So assume that P has one bad node and one exposed node.

Define $N \subseteq V$ as

- the nodes of C and the nodes they are matched with in P,
- the exposed node, and
- the bad node and its two neighbors in P.

Note that some of these nodes might actually be the same. With exception of the exposed node, all nodes in N are matched by an edge of P, the bad node actually twice. Hence, $|N|$ is even, $|N| \in \{4, 6, 8, 10, 12\}$.

Let P' be those edges of P which are incident on a node of N. Since the bad node is covered by a red and a blue edge, P' has ≥ 1 edge of each color. We assume that P' actually has ≥ 2 edges of each color. Otherwise add a further red, respectively blue edge from P to P'. Let N' denote the set of nodes covered by P'. Hence $|N'| \leq 14$. Let r' and b' denote the number of red, respectively blue edges of P'. We have $n' = r' + b' = |N'|/2 \leq 7$.

The graph G' induced by G on N' is a balanced graph with $2n'$ vertices. Let $r'_{\min}$ and $r'_{\max}$ be the minimum, respectively maximum number of red edges in a perfect matching in G'. We have $r'_{\min} \leq r' \leq r'_{\max}$:

- We can extend the r' red edges of P' to a perfect matching in G'. Therefore $r' \leq r'_{\max}$.
- We can extend the b' blue edges of P' to a perfect matching in G'. Therefore $r' \geq r'_{\min}$.

Since $r', b' \geq 2$, we also have $2 \leq r' \leq n' - 2$. Note that $G' \not\sim$ (c1) since it contains C. By the first part of the lemma, G' has an r'-perfect matching M'. We can find M' by trying all of the constantly many possibilities. We replace P' by M' in P, i.e., define $M = (P - P') \cup M'$. Then M is an r-perfect matching in G. □

The next lemma takes care of the case when $r = 1$ or $r = n - 1$. Again, the lemma has been proven by Yi et al. [YMS02] for bipartite graphs. We extend the proof to the non-bipartite case. For the complexity bound, we show again as in the previous lemma that an r-perfect matching can be constructed from an r-pseudo perfect matching of G which yields a reduction to *PM*, by Corollary 4.2.

Lemma 4.10. *Let G be a balanced graph.*

1. *If $r_{\max} = n$, then*

$$G \text{ has an } (n-1)\text{-perfect matching} \iff G \text{ has a } (3,1)\text{-cycle.}$$

2. *If $r_{\min} = 0$, then*

$$G \text{ has a } 1\text{-perfect matching} \iff G \text{ has a } (1,3)\text{-cycle.}$$

Let $r \in \{0, 1, n-1, n\}$. If G has an r-pm it can be constructed from an r-ppm of G in AC^0.

Proof. We show the first statement. The proof for the second statement is analogous. Let M_R be a perfect matching with $r_{\max} = n$ red edges, i.e., M_R is purely red.

Let M be an $(n-1)$-perfect matching in G, i.e., M has one blue edge. Consider $M \triangle M_R$. There is one cycle C in $M \triangle M_R$ that contains the blue edge of M. If C has length 4 we are done. So assume that $|C| > 4$.

Let $C = (v_1, v_2, \ldots, v_{2s})$ for $s > 2$, and let (v_1, v_{2s}) be the blue edge. Consider the edge (v_2, v_{2s-1}). If it is red then $(v_1, v_2, v_{2s-1}, v_{2s})$ is a $(3,1)$-cycle. If it is blue then we proceed with edge (v_3, v_{2s-2}). If it is red then again we get a $(3,1)$-cycle. We can continue so on. If the edges (v_i, v_{2s-i+1}) are all blue, for $i = 1, \ldots, s-1$, then we end up at (v_s, v_{s+1}), which is red and we have the $(3,1)$-cycle $(v_{s-1}, v_s, v_{s+1}, v_{s+2})$, where (v_{s-1}, v_{s+2}) is the blue edge.

For the reverse direction, let $C = (u_1, u_2, u_3, u_4)$ be a $(3,1)$-cycle in G with blue edge (u_1, u_4). The edges of M_R either connect nodes within C or connect a node of C with some node not in C. Let N_C be the nodes of C plus the nodes connected to C by an edge in M_R. We show that we can alter M_R on N_C such that the resulting perfect matching M has 1 blue edge.

Note that N_C contains 4, 6, or 8 vertices. We examine each case.

- $|N_C| = 4$: in this case M_R contains the red edges (u_1, u_2) and $u_3, u_4)$. We can simply swap the edges on C, i.e. we put (u_1, u_4) and (u_2, u_3) into M instead of (u_1, u_2) and (u_3, u_4).
- $|N_C| = 6$: one red edge of C is in M_R.
  - If $(u_1, u_2) \in M_R$, then there are nodes v_3, v_4 outside C such that $(u_3, v_3), (u_4, v_4) \in M_R$. If edge (v_3, v_4) is red, then we alter M_R on N_C to $\{(u_1, u_4), (u_2, u_3), (v_3, v_4)\}$. If edge (v_3, v_4) is blue, we take $\{(u_1, u_2), (u_3, u_4), (v_3, v_4)\}$.
  - The case $(u_3, u_4) \in M_R$ is symmetric to the previous case.
  - If $(u_2, u_3) \in M_R$, then there are nodes v_1, v_4 outside C such that $(u_1, v_1), (u_4, v_4) \in M_R$. If edge (v_1, v_4) is red, then we alter M_R on N_C to $\{(u_1, u_4), (u_2, u_3), (v_1, v_4)\}$. If edge (v_1, v_4) is blue, we take $\{(u_1, u_2), (u_3, u_4), (v_1, v_4)\}$.
- $|N_C| = 8$: no edge of C is in M_R. There are nodes v_1, v_2, v_3, v_4 outside C such that $(u_i, v_i) \in M$, for $i = 1, \ldots, 4$.

  If edge (v_1, v_4) is red, we put (v_1, v_4) and (u_1, u_4) into M instead of (u_1, v_1) and (u_4, v_4). So assume that (v_1, v_4) is blue.

  If edge (v_2, v_3) is blue, we put (v_2, v_3) and (u_2, u_3) into M instead of (u_2, v_2) and (u_3, v_3). So assume that (v_2, v_3) is red.

  We alter M_R on N_C to $\{(v_1, v_4), (u_1, u_2), (u_3, u_4), (v_2, v_3)\}$.

Note that the above argument works for bipartite and non-bipartite graphs.

Complexity: We consider the case $r \geq n - 1$, the case $r \leq 1$ can be handled by exchanging the colors. Let P be an r-ppm. If $r = n$ then P is also an r-pm and we are done. Otherwise $r = n - 1$ and $M = P \cap E_R$ is a matching in G_R of size $n - 1$. Let e be the edge between the vertices in G not covered by M. If e is blue, then $M \cup \{e\}$ is the required $(n - 1)$-pm. So assume that e is red. Then $M_R = M \cup \{e\}$ is a perfect matching with $r_{\max} = n$ red edges.

Next we go through all 4-cycles and search for a $(3, 1)$-cycle C. If there is no $(3, 1)$-cycle, then G has no $(n - 1)$-pm. Otherwise we construct an $(n - 1)$-pm from M_R and C as described above. Since M_R is changed only locally on constantly many edges, this can be accomplished in AC^0. $\square$

If $G \sim$ (c2r), then we have $r_{\max} = n$ and G has no $(3, 1)$-cycle by Corollary 4.8. By Lemma 4.10, G has no $(n-1)$-perfect matching. Similarly, if $G \sim$ (c2b), then G has no 1-perfect matching. If $G \sim$ (c3), then it has both, $(1, 3)$- and $(3, 1)$-cycles, and hence there is an r-perfect matching for every r such that $r_{\min} \leq r \leq r_{\max}$. In summary, we get the main theorem from Karzanov [Kar87] that characterizes the existence of an r-perfect matching for graphs in classes (c2r), (c2b), and (c3). We add the complexity bound for constructing such an r-perfect matching from an r-pseudo perfect matching.

Theorem 4.11. *Let G be a balanced graph and $r_{\min} \leq r \leq r_{\max}$. Then G has an r-perfect matching if*

- $G \sim$ (c2r) *and* $r \neq n-1$,
- $G \sim$ (c2b) *and* $r \neq 1$, *or*
- $G \sim$ (c3).

If $G \not\sim$ (c1) has an r-pm it can be constructed in AC^0 *from an r-ppm of G.*

Theorem 4.6 and 4.11 together resolve the complexity of the exact perfect matching problem for all full graphs. Recall that we can find out in TC^0 the class to which a given graph belongs to. By Corollary 4.2 we get a many-reduction to *PM*.

Theorem 4.12. *cxPM* $\leq$ *PM. The many-one reduction is* TC^0 *for decision and logspace for construction, also for the bipartite versions.*

We can now put *cxPM* into the chain of reductions from Theorems 3.1,

$$\mathit{PM} \equiv \mathit{MM} \equiv \mathit{cxPM} \leq \mathit{wMM} \equiv \mathit{wPM} \leq \mathit{xPM} \equiv \mathit{wxPM}$$

The reductions are logspace many-one reductions, also for the bipartite versions, and for both decision and construction.

Discussion

The reduction *wxPM* $\leq$ *wPM*, which would put *wxPM* in P, is still open. Our procedure for constructing an exact pseudo perfect matching works for arbitrary graphs. But the second step, which constructs an exact perfect matching from an exact pseudo perfect matching, uses the completeness of the graph at several places. It is not at all clear how to generalize the second step to arbitrary graphs.

Other open problems are whether there are NC-reductions from weighted perfect matching to perfect matching, or from the construction of a perfect matching to the decision version.

References

[Edm65] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of research of the Na tional Bureau of Standards–B. Mathematics and Mathematical Physics*, 69B:125–130, 1965.

[GKM+11] R. Gurjar, A. Korwar, J. Messner, S. Straub, and T. Thierauf. Planarizing gadgets for perfect matching do not exist. Technical Report TR11-148, Electronic Colloquium on Computational Complexity (ECCC), 2011. Revised version appears in proceedings of MFCS 2012.

[GS11] H.-F. Geerdes and J. Szabó. A unified proof for Karzanov's exact matching theorem. Technical Report QP-2011-02, Egerváry Research Group, Budapest, 2011. `www.cs.elte.hu/egres`.

[Kar87] A.V. Karzanov. Maximum matching of given weight in complete and complete bipartite graphs. *Cybernatics and Systems Analysis*, 23(1):8–13, 1987.

[KUW86] M. Karp, R. E. Upfal, and A. Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[PY82] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. ACM*, 29:285–309, April 1982.

[Rei08] O. Reingold. Undirected connectivity in log-space. *J. ACM*, 55:17:1–17:24, 2008.

[Vaz89] V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Information and computation*, 80(2):152–164, 1989.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity*. Springer, 1999.

[YMS02] T. Yi, K. G. Murty, and C. Spera. Matchings in colored bipartite networks. *Discrete Applied Mathematics*, 121(1-3):261–277, 2002.

[Yus12] R. Yuster. Almost exact matchings. *Algorithmica*, 63:39–50, 2012.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de