Randomized and Approximation Algorithms for Blue-Red Matching

Christos Nomikos[1], Aris Pagourtzis[2,⋆], and Stathis Zachos[2,3,⋆]

[1] Department of Computer Science, University of Ioannina
cnomikos@cs.uoi.gr
[2] School of Electrical and Computer Engineering,
National Technical University of Athens
{pagour,zachos}@cs.ntua.gr
[3] CIS Department, Brooklyn College, Cuny

Abstract. We introduce the Blue-Red Matching problem: given a graph with red and blue edges, and a bound w, find a maximum matching consisting of at most w edges of each color. We show that Blue-Red Matching is at least as hard as the problem Exact Matching (Papadimitriou and Yannakakis, 1982), for which it is still open whether it can be solved in polynomial time. We present an RNC algorithm for this problem as well as two fast approximation algorithms. We finally show the applicability of our results to the problem of routing and assigning wavelengths to a maximum number of requests in all-optical rings.

1 Introduction

We define and study a matching problem on graphs with blue and red edges; we call the new problem Blue-Red Matching (BRM for short). The goal is to find a maximum matching under the constraint that the number of edges of each color in the matching does not exceed a given bound w.

We are motivated for this study by a problem that arises in all-optical networks, namely DirMaxRWA [12]. In particular, it was implicit in [12] that solving BRM exactly would imply an improved approximation ratio for DirMaxRWA in rings. Moreover, BRM can capture several interesting scenarios such as the following: Consider a team of friends that would like to play chess or backgammon. Some pairs prefer to play chess, while other pairs prefer backgammon. There could even exist pairs that would like to play either game. Now, if the number of available boards for each game is limited we need to solve BRM if we want to maximize the number of pairs that will manage to play the game of their preference.

In this work we first show that BRM is at least as hard as Exact Matching, a problem defined by Papadimitriou and Yannakakis [13], for which it is still an open question whether it can be solved in polynomial time. Therefore, an

⋆ Research supported in part by the General Secretariat of Research and Technology of Greece, through ΠΕΝΕΔ 2003 program, grant nr. ΕΔ285.

L. Kučera and A. Kučera (Eds.): MFCS 2007, LNCS 4708, pp. 715–725, 2007.

exact polynomial time algorithm for BRM would answer that question in the affirmative.

Our main result is that BRM can be solved exactly by a polynomial time randomized (in fact RNC^2) algorithm, which uses ideas from [10]. Since the sequential version of the randomized algorithm is quite slow, we also present two approximation algorithms for BRM; the first is a fast and simple greedy algorithm that achieves a $\frac{1}{2}$-approximation ratio, the second is a more involved algorithm that achieves an asymptotic $\frac{3}{4}$-approximation ratio.

We finally demonstrate the relation between BRM and DirMaxRWA in rings, by showing that an algorithm for BRM with (asymptotic) approximation ratio a results in an algorithm for DirMaxRWA in rings with (asymptotic) approximation ratio $\frac{a+1}{a+2}$. Combining all the above results we obtain as a corollary that DirMaxRWA in rings admits a randomized approximation algorithm with ratio $\frac{2}{3}$ and a (much faster) deterministic approximation algorithm with asymptotic ratio $\frac{7}{11}$.

As far as we know BRM has not been studied before. As mentioned earlier, a related problem is Exact Matching [13] which admits an RNC algorithm due to Mulmuley, Vazirani and Vazirani [10]. Polynomial time algorithms for Exact Matching are known only for special classes of graphs, e.g. complete graphs [7] and complete bipartite graphs [7][17].

2 Problem Definition and Hardness

Let $G = (V, E_{blue} \cup E_{red})$ be a graph in which each edge is colored either blue or red; E_{blue} is the set of blue edges and E_{red} the set of red edges. A matching M in G is called *w-blue-red matching* if $M \cap E_{blue} \leq w$ and $M \cap E_{red} \leq w$, that is, if it contains at most w edges of each color.

The notion of w-blue-red matching can be extended for multigraphs that may contain edges of both colors between two vertices. It is easy to see that in fact we do not have to use multigraphs; it suffices to specify a third set of initially uncolored (white) edges as follows. Let $G = (V, E_{blue} \cup E_{red} \cup E_{white})$ be a graph in which E_{blue}, E_{red}, and E_{white} are sets of blue, red, and white edges respectively. A matching M in G is called w-blue-red matching if there exists a partition $\{E_{wb}, E_{wr}\}$ of E_{white} such that $M \cap (E_{blue} \cup E_{wb}) \leq w$ and $M \cap (E_{red} \cup E_{wr}) \leq w$. In other words M is a w-blue-red matching if we can choose a color for each white edge in G so that M contains at most w edges of each color.

We define BRM to be the following optimization problem: given a graph $G = (V, E_{blue} \cup E_{red} \cup E_{white})$ and a positive integer w, find a w-blue-red matching of maximum cardinality. In the decision version of this problem, denoted by BRM(D), a bound B is also given and the question is whether G has a w-blue-red matching of cardinality at least B.

It turns out that BRM(D) is closely related to a well known problem, namely Exact Matching, defined in [13]. In this problem, the input is a graph $G = (V, E)$, a set of red edges $E' \subseteq E$ and a positive integer k and the question is whether G contains a perfect matching involving exactly k edges in E'. The next theorem shows that BRM(D) is at least as hard as EXACT MATCHING.

Theorem 1. *There is a logarithmic space reduction from* EXACT MATCHING *to* BRM(D)*.*

Proof. Consider a graph $G = (V, E)$, a set of red edges $E' \subseteq E$ and a positive integer k.

If $|V|$ is an odd number or $k > \frac{|V|}{2}$, then G does not contain a perfect matching involving exactly k edges in E'. In that case we construct a 'no' instance of BRM(D) (for example, any instance with $2w < B$).

Otherwise, let $w = \max(k, \frac{|V|}{2} - k)$ and $r = w - \min(k, \frac{|V|}{2} - k)$. Graph G^* is obtained from G by adding $2r$ new vertices $u_1, \ldots, u_r$, $v_1, \ldots, v_r$ and r edges $\{u_1, v_1\}$, …, $\{u_r, v_r\}$. The additional edges are colored blue if $k > \frac{|V|}{2} - k$, otherwise they are colored red. Furthermore, edges in $E - E'$ are colored blue and edges in E' remain red in G^*. Let $B = 2w$.

The above construction requires logarithmic space. It is not hard to check that G contains a perfect matching involving exactly k edges in E' if and only if G^* contains a w-blue-red matching of cardinality B. □

The above theorem indicates that it is probably not a trivial task to find a polynomial time (deterministic) algorithm for BRM, since this would imply polynomial time solvability for EXACT MATCHING as well. Therefore, we will restrict our attention to approximation and randomized algorithms.

3 Approximation Algorithms for Blue-Red Matching

We first observe that there exists a simple approximation algorithm for BRM, which requires time linear in the number of edges. The algorithm, which we call Greedy-BRM, constructs a w-blue-red matching M in a greedy manner: edges are examined in an arbitrary order; an edge e is added to M if both endpoints of e are unmatched and M contains fewer than w edges of the same color as e (or M contains fewer than w edges of any color if e is white). It is not hard to prove the following:

Theorem 2. *Algorithm* Greedy-BRM *returns a solution with at least* $\frac{1}{2} \cdot \mu_{OPT}$ *edges, where* μ_{OPT} *is the cardinality of an optimal solution.*

In the remaining of this section we present an approximation algorithm for BRM, which achieves asymptotic approximation ratio $\frac{3}{4}$.

The algorithm first computes a maximum cardinality matching M (Step 1). In Step 2, a color is assigned to each white edge of the graph. If after Step 2 M contains more than w edges of one color and fewer than w edges of the other color then Step 3 is executed in order to produce a more balanced matching. Finally, Step 4 eliminates superfluous edges of any of the two colors.

Algorithm Balance-BRM

Input: graph $G = (V, E_{blue} \cup E_{red} \cup E_{white})$, integer w.
Output: a w-blue-red matching of G.

1. find a maximum matching M in G
2. **for** every white edge e **do**
    color e with the color which is currently less used in M, breaking ties arbitrarily
  let E'_{red}, E'_{blue} be the sets of red and blue edges after coloring the white edges
3. **if** M contains $> w$ edges of one color and $< w$ edges of the other color **then**
  (Assume w.l.o.g. that the majority color in M is blue—the other case is symmetric)
  (a) find a maximum matching M_{red} in graph $G_{red} = (V, E'_{red})$
  (b) let G' be the graph resulting by superimposing M and M_{red}
  (c) let S be the set of all connected components in G, in which the number of edges that belong to M_{red} is greater than the number of red edges that belong to M
  (d) **while** M contains more than $w + 1$ blue edges and fewer than w red edges and S is not empty **do**
    (i) choose (arbitrarily) a connected component F in S. Let b_M, b_F be the number of blue edges in M, F respectively
    (ii) **if** $b_M - w < b_F$ **then** pick a chain F' of edges in F containing exactly $b_M - w$ blue edges, such that F' begins and ends with a blue edge
      else let $F' = F$
    (iii) delete from M all edges that belong to F'; add to M all edges in F' that belong to M_{red}
    (iv) delete F from S
4. **if** M contains more than w blue (red) edges **then**
    choose arbitrarily w of them and eliminate the rest
5. **return** M

We will next prove that algorithm Balance-BRM achieves an asymptotic $\frac{3}{4}$-approximation ratio. Let us first note that if after the first two steps there are either at most w edges of each color in matching M or at least w edges of each color in M, then M (after removing surplus edges, in the latter case, in Step 4) is an optimal solution. Therefore, it remains to examine the case in which there are more than w edges of one color and fewer than w edges of the other after Step 2. W.l.o.g. we assume that the majority color is blue. We will first give two lemmata concerning Step 3.

Each substitution in Step 3 increases the number of red edges in M. However, it may decrease the number of blue edges. In the extreme case one red edge replaces two blue edges. Therefore, we have:

Lemma 1. *If Step 3 of algorithm* Balance-BRM *decreases the number of blue edges by δ, then it increases the number of red edges by at least $\delta/2$.*

If M contains more than $w + 1$ blue edges, then Step 3 can always perform a substitution, unless the number of red edges has reached its maximum possible value. Therefore we have:

Lemma 2. *If after Step 3, M contains more than $w+1$ blue edges, then algorithm* Balance-BRM *returns an optimal solution.*

We are now ready to state the main theorem of this section.

Theorem 3. *Algorithm* Balance-BRM *returns a solution with at least $\frac{3}{4}\cdot\mu_{OPT}-\frac{1}{2}$ edges, where μ_{OPT} is the cardinality of an optimal solution.*

Proof. We prove the claim for the case in which the number of blue edges is greater than the number of red edges. The other case is symmetric.

Let μ_{SOL} be the number of edges in the solution returned by Balance-BRM, μ_r, μ_b be the number of blue and red edges respectively contained in M after Step 2, and μ_{red} be the size of M_{red}. For convenience, let us also define $z = \min(\mu_{red}, w, \mu_b + \mu_r - w)$.

All red edges in an optimal matching belong to $E_{red} \cup E_{white}$ which is equal to E'_{red} in the case in which blue is the majority color. Therefore μ_{red} is an upper bound for the number of red edges in an optimal matching, which implies $\mu_{OPT} \le w+\mu_{red}$. Since $\mu_b+\mu_r$ is the size of the maximum cardinality matching M, it also holds $\mu_{OPT} \le \mu_b + \mu_r = w + (\mu_b + \mu_r - w)$. Moreover, by definition $\mu_{OPT} \le 2w$. Combining the above inequalities we obtain:

$$\mu_{OPT} \le w + z \tag{1}$$

Lemma 2 implies that in any non-optimal solution, M contains at most $w+1$ blue edges after Step 3. Hence, in Step 3 the number of blue edges decreases by at least $\mu_b - w - 1$. By Lemma 1, the number of additional red edges is at least $\frac{(\mu_b-w-1)}{2}$. Since after Step 3 the number of blue edges in M is at least w, we get $\mu_{SOL} \ge w + \mu_r + \frac{(\mu_b-w-1)}{2}$. Using the fact that, by definition, $z \le \mu_b + \mu_r - w$, it turns out that:

$$\mu_{SOL} \ge w + \frac{\mu_b+\mu_r-w}{2} + \frac{\mu_r-1}{2} \ge w + \frac{z}{2} - \frac{1}{2} + \frac{\mu_r}{2} \ge w + \frac{z}{2} - \frac{1}{2} \tag{2}$$

From (2) and the fact that, by definition, $z \le w$ we obtain:

$$\mu_{SOL} \ge \frac{3z}{2} - \frac{1}{2} \tag{3}$$

From (1) and (2) we get that:

$$\mu_{OPT} \le \mu_{SOL} + \frac{z}{2} + \frac{1}{2} \tag{4}$$

Finally, from (3) and (4) it follows that $\mu_{OPT} \le \mu_{SOL} + \frac{1}{3}(\mu_{SOL}+\frac{1}{2}) + \frac{1}{2} = \frac{4}{3}(\mu_{SOL}+\frac{1}{2})$, which is equivalent to $\mu_{SOL} \ge \frac{3}{4}\mu_{OPT} - \frac{1}{2}$. □

It can be shown that the above asymptotic approximation ratio is tight. The complexity of the algorithm is $O(n^{2.5})$: Steps 1 and 3 require $O(n^{2.5})$ time to construct M and M_{red} and all the remaining tasks require time that is linear in the number of edges, which is at most $O(n^2)$.

4 A Randomized Algorithm for Blue-Red Matching

In this section we present a randomized polynomial time algorithm, called Random-BRM that finds an optimal solution for BRM with high probability. This algorithm makes use of some ideas proposed in [10].

Algorithm Random-BRM operates as follows: First, it augments G to a complete graph G^* by adding edges of a new color (say black). Then it assigns a random weight to each edge of G^* and constructs a variation of the Tutte matrix of G^*, in which each indeterminate is replaced by a constant value or by a monomial, depending on the weight and the color of the corresponding edge. In particular the indeterminate that corresponds to a blue (red) edge e_{ij} of weight w_{ij} is replaced by the monomial $x2^{w_{ij}}$ (resp. $y2^{w_{ij}}$). Then, the algorithm computes the Pfaffian of this matrix, which in this case is a polynomial in the variables x, y. Finally, it uses the coefficients of this polynomial in order to find a specific matching in G^*, from which it obtains an optimal solution for BRM.

The detailed algorithm is given at the end of the section; its correctness is based on a series of lemmata which are stated below, together with some necessary definitions.

Consider a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = E_{blue} \cup E_{red} \cup E_{white}$, and an positive integer w. Without loss of generality we may assume that n is even (otherwise an isolated vertex can be added to G). Let G^* be the complete graph with set of vertices V. We denote the edge $\{v_i, v_j\}$ by e_{ij}. We assume that edges not in E are colored black, i.e. $G^* = (V, E \cup E_{black})$. A perfect matching of G^* with exactly p blue and q red edges is called (p, q)-perfect matching. We denote by $\mathcal{M}$ (resp. $\mathcal{M}_{pq}$) the set of all perfect matchings (resp. (p, q)-perfect matchings) of G^*.

Perfect matchings in G^* can be used in order to obtain w-blue-red matchings in G. For fixed w, let us define a function $\mathrm{sol}_w(p, q, t) = \min(2w, \min(w, p) + \min(w, q) + t)$. The lemma below explains the use of function sol_w:

Lemma 3. *Let M be a (p, q)-perfect matching of G^* with t white edges. Then there exists a w-blue-red matching $M_w \subseteq M$ of G of cardinality $\mathrm{sol}_w(p, q, t)$.*

Proof. We can construct M_w as follows: we first select arbitrarily $\min(w, p)$ blue edges and $\min(w, q)$ red edges from M and add them to M_w; then we repeatedly select a white edge from M, color it with the color which is currently less used by edges in M_w, and add it to M_w, until the cardinality of M_w becomes $2w$ or we run out of white edges. In the latter case $M_w = \min(w, p) + \min(w, q) + t$. □

Suppose that a number s_{ij} is selected at random from $\{1, 2, \ldots, n^4\}$, for each (i, j), $1 \le i < j \le n$, and define the weight w_{ij} of e_{ij} as follows:

$$w_{ij} = \begin{cases} s_{ij} & \text{if } e_{ij} \in E \\ n^5 + s_{ij} & \text{if } e_{ij} \in E_{black} \end{cases}$$

The weight of a perfect matching M is $W_M = \sum_{e_{ij} \in M} w_{ij}$. We denote by W_{pq} the minimum weight of a matching among all matchings in $\mathcal{M}_{pq}$. The number of white edges in a (p, q)-perfect matching with weight W_{pq} can be easily computed, using the next lemma:

Lemma 4. *Let p, q be integers, with $0 \leq p, q \leq \frac{n}{2}$ and let M_{pq} be a minimum weight (p, q)-perfect matching of G^*. Then the number of white edges in M_{pq} is $\frac{n}{2} - p - q - \lfloor \frac{W_{pq}}{n^5} \rfloor$.*

The following lemma can be used to compute the number of edges in an optimal w-blue-red matching:

Lemma 5. *The number of edges in an optimal w-blue-red matching of graph G is*

$$C = \max_{(p,q): \mathcal{M}_{pq} \neq \emptyset} \mathrm{sol}_w(p, q, \frac{n}{2} - p - q - \lfloor \frac{W_{pq}}{n^5} \rfloor).$$

The Tutte matrix A of G^* is defined as follows:[1]

$$a_{ij} = \begin{cases} 0 & \text{if } i = j \\ 2^{w_{ij}} & \text{if } i < j \text{ and } e_{ij} \in E_{white} \cup E_{black} \\ x2^{w_{ij}} & \text{if } i < j \text{ and } e_{ij} \in E_{blue} \\ y2^{w_{ij}} & \text{if } i < j \text{ and } e_{ij} \in E_{red} \\ -a_{ji} & \text{if } i > j \end{cases}$$

The *canonical permutation* for a perfect matching $M \in \mathcal{M}$, denoted by π_M, is the unique permutation of $\{1, 2, \ldots, n\}$ that satisfies the following properties:

- $\{v_{\pi_M(2i-1)}, v_{\pi_M(2i)}\} \in M$, for every i, $1 \leq i \leq \frac{n}{2}$
- $\pi_M(2i - 1) < \pi_M(2i)$, for every i, $1 \leq i \leq \frac{n}{2}$
- $\pi_M(2i - 1) < \pi_M(2i + 1)$, for every i, $1 \leq i \leq \frac{n}{2} - 1$

For every matching M, let $sign(\pi_M) = (-1)^{|\{(i,j):\ i<j,\ \pi_M(i)>\pi_M(j)\}|}$ and $value(\pi_M) = \prod_{i=1}^{n/2} a_{\pi_M(2i-1),\pi_M(2i)}$.

The *Pfaffian of A* is defined as follows: $\mathcal{PF}(A) = \sum_{M \in \mathcal{M}} sign(\pi_M) \cdot value(\pi_M)$. The Pfaffian of A is a polynomial of the form: $\mathcal{PF}(A) = \sum_{p=0}^{n/2} \sum_{q=0}^{n/2} c_{pq} x^p y^q$ and it can be computed by interpolation (see [6]), using an algorithm that computes arithmetic Pfaffians [4,8] as a subroutine.

The term $x^p y^q$ of $\mathcal{PF}(A)$ corresponds to the (p, q)-perfect matchings of G^*. Therefore, if c_{pq} is nonzero, then a (p, q)-perfect matching exists in G^*. The converse does not necessarily hold: it is possible that the coefficient of c_{pq} is zero although G^* contains (p, q)-perfect matchings, in the case where the terms corresponding to these matchings are mutually cancelled. The following lemma gives a sufficient condition so that the coefficient of c_{pq} is nonzero.

Lemma 6. *Let p, q be integers, with $0 \leq p, q \leq \frac{n}{2}$ and suppose that there exists a unique minimum weight (p, q)-perfect matching M_{pq} of G^*. Then the coefficient c_{pq} of $\mathcal{PF}(A)$ is nonzero. Furthermore, W_{pq} is the maximum power of 2 that divides c_{pq}.*

[1] Strictly speaking, A is a special form of the Tutte matrix of G^*, where each indeterminate has been replaced either by a specific value or by an indeterminate multiplied by a specific value.

Proof. We have: $c_{pq} = sign(\pi_{M_{pq}}) \cdot 2^{W_{pq}} + \sum_{M \in \mathcal{M}_{pq} - \{M_{pq}\}} sign(\pi_M) \cdot 2^{W_M}$ Since M_{pq} is a unique minimum weight (p,q)-perfect matching, $W_{pq} < W_M$ for every $M \in \mathcal{M}_{pq} - \{M_{pq}\}$. Therefore $c_{pq} \bmod 2^{W_{pq}} = 0$ and $c_{pq} \bmod 2^{W_{pq}+1} = 1$, which imply that $c_{pq} \neq 0$ and that W is the maximum power 2 that divides c_{pq}. □

For every edge e_{ij} of G^* we define

$$Z_{ij} = \sum_{M \in \mathcal{M} : e_{ij} \in M} sign(\pi_M) \cdot value(\pi_M)$$

(that is, Z_{ij} is the part of $\mathcal{PF}(A)$ that involves e_{ij}). The following lemma shows how Z_{ij} can be computed up to sign.

Lemma 7. *For every edge e_{ij} of G^*, $Z_{ij} = \sigma \cdot a_{ij} \cdot \mathcal{PF}(A_{ij})$ where $\sigma \in \{-1, 1\}$ and A_{ij} is the matrix obtained from A by removing the i-th and j-th row and the the i-th and j-th column.*

If G^* has a unique minimum weight (p,q)-perfect matching M_{pq}, we can decide whether an edge belongs to M_{pq}, using the following lemma:

Lemma 8. *Suppose that G^* has a unique minimum weight (p,q)-perfect matching M_{pq}. Let c be the coefficient of $x^p y^q$ in Z_{ij}. Then $e_{ij} \in M_{pq}$ if and only if $2^{W_{pq}+1}$ does not divide c.*

In order to bound from below the probability that the algorithm returns an optimal solution, we make use of the following strong version of the Isolating Lemma:

Lemma 9. *(**Isolating Lemma [10]**) Let $B = \{b_1, b_2, \ldots, b_k\}$ be a set of elements, let $\mathcal{S} = \{S_1, S_2, \ldots, S_\ell\}$ be a collection of subsets of B and let $a_1, a_2, \ldots, a_\ell$ be integers. If we choose integer weights $w_1, w_2, \ldots w_k$ for the elements of B at random from the set $\{1, 2, \ldots, m\}$, and define the weight of set S_j to be $a_j + \sum_{b_i \in S_j} w_i$ then the probability that the minimum weight subset in $\mathcal{S}$ is not unique is at most $\frac{k}{m}$.*

Theorem 4. *Algorithm* Random-BRM *returns an optimal solution with probability at least $\frac{1}{2}$.*

Proof. If there exists a unique minimum weight element M_{pq} in every non-empty set $\mathcal{M}_{pq}$, $0 \leq p, q \leq \frac{n}{2}$ then it follows from Lemmata 3, 5, 6, 7 and 8 that the above algorithm returns an optimal w-blue-red matching.

The probability that $\mathcal{M}_{pq}$ contains at least two minimum weight elements for fixed values p and q is at most $\frac{1}{2n^2}$ by the Isolating Lemma. Thus, the probability that there exist values p, q such that $\mathcal{M}_{pq}$ contains at least two minimum weight elements is at most $(\frac{n}{2}+1)^2 \cdot \frac{1}{2n^2} \leq \frac{1}{2}$. Therefore the algorithm returns a correct solution with probability at least $\frac{1}{2}$. □

Algorithm Random-BRM

Input: graph $G = (V, E_{blue} \cup E_{red} \cup E_{white})$ with even number of vertices n, positive integer w.
Output: maximum w-blue-red matching (with probability $\geq \frac{1}{2}$).

1. augment G to a complete graph G^* by adding a set E_{black} of black edges
2. **for** every $e_{ij} \in G^*$ **do**
  – choose at random a number s_{ij} from $\{1, 2, \ldots, n^4\}$
  – **if** $e_{ij} \in E_{black}$ **then** $w_{ij} := n^5 + s_{ij}$
    else $w_{ij} := s_{ij}$
3. construct the Tutte matrix A of G^*
4. compute $\mathcal{PF}(A)$; let c_{pq} be the coefficient of $x^p y^q$ in in $\mathcal{PF}(A)$, $0 \leq p, q, \leq \frac{n}{2}$
5. **for** every $(p, q) \in \{0, 1, \ldots \frac{n}{2}\}^2$ **do**
  – **if** $c_{pq} \neq 0$ **then** let W_{pq} be the maximum power of 2 that divides c_{pq}
    else $W_{pq} := \infty$
  find $(p, q) \in \{0, 1, \ldots \frac{n}{2}\}^2$ such that $\mathrm{sol}_w(p, q, \frac{n}{2} - p - q - \lfloor \frac{W_{pq}}{n^5} \rfloor)$ is maximum
6. $M_{pq} := \emptyset$
  for every $e_{ij} \in G^*$ **do**
  – compute $|Z_{ij}| := |a_{ij} \cdot \mathcal{PF}(A_{ij})|$
  – let c be the coefficient of $x^p y^q$ in $|Z_{ij}|$
  – **if** $c \bmod 2^{W_{pq}+1} \neq 0$ **then** $M_{pq} := M_{pq} \cup \{e_{ij}\}$
7. compute a w-blue-red matching M from M_{pq}, by a greedy coloring of white edges
8. **return** M

Complexity. Sequentially, the algorithm requires $O(n^7)$ time, since the computation of the symbolic Pfaffian requires $O(n^5)$ time, using the algorithm in [4] which computes arithmetic Pfaffians in $O(n^3)$ time and Step 6 requires the computation of $O(n^2)$ minor Pfaffians. However all steps can be parallelized resulting in a RNC algorithm (in the parallel version, the algorithm from [8] is used to compute arithmetic Pfaffians). In fact, it can be shown by careful analysis that Random-BRM is an RNC^2 algorithm.

5 Application to Optical Networking

In this section we show how solving BRM can help in approximately solving the Directed Maximum Routing and Wavelength Assignment (DirMaxRWA) problem in rings.

DirMaxRWA is defined as follows [12]: Given are a directed symmetric graph G, a set of requests (pairs of nodes) R on G, and an integer w (bound on the number of available wavelengths). The goal is to find a routing and wavelength assignment to an as large as possible set of requests $R' \subseteq R$ such that any two requests routed via edge-intersecting paths receive different wavelengths and only wavelengths from $\{1, \ldots, w\}$ are used.

It can be shown that the algorithm for DirMaxRWA in rings proposed in [12] can be modified to make an explicit call to an algorithm for solving BRM (instead of implicitly solving it, as was the case originally).

The following theorem relates the approximation ratio of the modified algorithm to the approximation ratio achieved by the algorithm for BRM that is employed. The proof is an adaptation of the analysis of the algorithm presented in [12] and will appear in the full version.

Theorem 5. *An algorithm for* BRM *which returns a w-blue-red matching containing at least $a \cdot \mu_{OPT} - b$ edges, where μ_{OPT} is the size of an optimal solution and $a > 0, b \geq 0$ are constants, results in an algorithm for* DirMaxRWA *in rings that satisfies at least $\frac{a+1}{a+2} \cdot OPT - \frac{b}{a+2}$ requests, where OPT is the size of an optimal solution for* DirMaxRWA.

Therefore, by using the algorithms for BRM proposed in the previous sections (Random-BRM and Balance-BRM) we obtain the following:

Corollary 1. DirMaxRWA *in rings admits a randomized approximation algorithm with ratio $\frac{2}{3}$ and a deterministic approximation algorithm with asymptotic ratio $\frac{7}{11}$.*

Note that the $\frac{2}{3}$ approximation ratio is tight, as can be shown by appropriate examples.[2] The deterministic algorithm is slightly worse in terms of approximation ratio, but is considerably faster. An even faster deterministic approximation algorithm with ratio $\frac{3}{5}$ is obtained if we use algorithm Greedy-BRM as a subroutine. As regards time requirements, it can be shown that the complexity of the algorithm for DirMaxRWA is dominated by the complexity of the algorithm for BRM that is employed; therefore it is $O(n^7)$ if we use Random-BRM for solving BRM, while it is $O(n^{2.5})$ if we use Balance-BRM and $O(n^2)$ if we use Greedy-BRM.

References

1. Caragiannis, I.: Wavelength Management in WDM Rings to Maximize the Number of Connections. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 61–72. Springer, Heidelberg (2007)
2. Carlisle, M.C., Lloyd, E.L.: On the k-Coloring of Intervals. Discrete Applied Mathematics 59, 225–235 (1995)
3. Erlebach, T., Jansen, K.: The Complexity of Path Coloring and Call Scheduling. Theoretical Computer Science 255(1-2), 33–50 (2001)
4. Galbiati, G., Maffioli, F.: On the Computation of Pfaffians. Discrete Applied Mathematics 51(3), 269–275 (1994)
5. Garey, M., Johnson, D., Miller, G., Papadimitriou, C.: The Complexity of Coloring Circular Arcs and Chords. SIAM Journal on Algebraic Discrete Methods 1(2), 216–227 (1980)
6. Horowitz, E., Sahni, S.: On Computing the Exact Determinant of Matrices with Polynomial Entries. Journal of the ACM 22(1), 38–50 (1975)

[2] The $\frac{2}{3}$ approximation ratio improves upon the ratio obtained in [12] and has been the best known so far for DirMaxRWA in rings until very recently, when Caragiannis [1] gave a 0.708-approximation algorithm for the problem.

7. Karzanov, A.V.: Maximum Matching of Given Weight in Complete and Complete Bipartite Graphs. Kibernetika 23(1), 7–11 (1987) (English translation in CYBNAW 23(1), 8–13 (1987))
8. Mahajan, M., Subramanya, P.R., Vinay, V.: The Combinatorial Approach Yields an NC Algorithm for Computing Pfaffians. Discrete Applied Mathematics 143(1-3), 1–16 (2004)
9. Micali, S., Vazirani, V.V.: An $O(n^{2.5})$ Algorithm for Maximum Matching in General Graphs. In: Proceedings Twenty-first Annual Symposium on the Foundations of Computer Science, pp. 17–27 (1980)
10. Mulmuley, K., Vazirani, U.V., Vazirani, V.V.: Matching is as Easy as Matrix Inversion. Combinatorica 7(1), 105–113 (1987)
11. Nomikos, C., Pagourtzis, A., Zachos, S.: Satisfying a Maximum Number of Pre-Routed Requests in All-Optical Rings. Computer Networks 42(1), 55–63 (2003)
12. Nomikos, C., Pagourtzis, A., Zachos, S.: Minimizing Request Blocking in All-Optical Rings. In: Proceedings INFOCOM 2003, pp. 1771–1780 (2003)
13. Papadimitriou, C.H., Yannakakis, M.: The Complexity of Restricted Spanning Tree Problems. Journal of the ACM 29(2), 285–309 (1982)
14. Raghavan, P., Upfal, E.: Efficient Routing in All-Optical Networks. In: Proceedings of the 26th Annual ACM Symposium on the Theory of Computing STOC 1994, pp. 134–143. ACM Press, New York (1994)
15. Stamoulis, G.: Maximum Matching Problems with Constraints (in Greek). Diploma Thesis, Department of Computer Science, University of Ioannina (2006)
16. Wan, P.-J., Liu, L.: Maximal Throughput in Wavelength-Routed Optical Networks. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 46, 15–26 (1998)
17. Yi, T., Murty, K.G., Spera, C.: Matchings in Colored Bipartite Networks. Discrete Applied Mathematics 121(1-3), 261–277 (2002)