PhD Tomasz Krawczyk
Theoretical Computer Science Department
Jagiellonian University
No. 6 Łojasiewicza Street, 30-348 Cracow
krawczyk@tcs.uj.edu.pl

November 10, 2020.

Report on the PhD thesis of Pratik Ghosal
Efficient algorithms for combinatorial optimization problems related to rank-maximal matchings and rectangle tiling

Overview of the thesis

The PhD thesis of Pratik Ghosal concerns two important problems in the areas of algorithmics and combinatorial optimization. The first part of the dissertation contains the results related to rank-maximal matchings in bipartite graphs, and the second part consists of results related to the rectangle tiling problem.

Rank-maximal matchings in bipartite graphs. Problems related to matchings in graphs occupy one of the central places of algorithmic graph theory. Due to the number of various applications (auctions, resource allocations) they are intensively studied in different directions that aim to construct efficient off-line, on-line, and dynamic algorithms for various kinds of matching problems. A fairly important branch of this field is also focused on the construction of efficient algorithms searching not only for matchings of maximum size, but also for matchings optimal with respect to other parameters measuring its quality, such as those related to preferences between matched vertices.

The results contained in Chapters 3-4, related to *rank-maximal matchings* in bipartite graphs, fit very well in this line of research. An instance of the rank-maximal matching problem consists of a weighted bipartite graph $G = (A \cup P, \mathcal{E})$ with the edges $\mathcal{E}$ between *applicants* from the set A and *posts* from the set P. Every edge $e = (a, p)$ from $\mathcal{E}$ has a weight in the set $\mathbb{N} \setminus \{0\}$, called a *rank*, which determines how attractive the post p is for the applicant a. It is assumed that the lower the rank of the edge $e = (a, p)$, the more attractive the post p is for the applicant a (in particular, rank 1 edges adjacent to a lead to the first choice posts for a). A matching M in G is called *rank-maximal* if the largest possible number of applicants is matched in M to their first choice posts, and subject to this condition, the largest number of applicants are matched in M to their second choice posts, and so on. Currently the best algorithm for rank-maximal matching, by Irving, Kavitha, Mehlhorn, Michail, and Paluch, works in time $O(min(n, c\sqrt{n})m)$, where c is the maximal rank of an edge in a maximal-rank matching. Since this algorithm, hereinafter referred to as the *combinatorial algorithm*, is the starting point for the considerations of Chapters 3-4, its detailed description is given in Chapter 2.

The key concept used in the design of the combinatorial-algorithm for the rank-maximal matching is the Gallai-Edmonds decomposition of a bipartite graph G. Let M be a maximum matching in G. The Gallai-Edmonds decomposition of G partitions the vertices of G into three sets E, O, U that contain the vertices reachable by an even/odd M-alternating path in G starting in a free vertex of M (the sets E and O, respectively), and the vertices that are unreachable by such a path (the set U). It turns out that the Gallai-Edmonds decomposition is independent on the choice of M and that it has the property that every edge in any maximum matching in G joins either two vertices from U or a vertex from O and a vertex in E. The combinatorial algorithm sequentially processes the graphs $G_1, G_2, \ldots, G_r$, where G_i is the graph derived from G

TK

by restricting to the edges of rank $\leqslant i$ and r is the maximal rank of an edge in G, and finds rank-maximal matchings $M_1, \ldots, M_r$ in $G_1, \ldots, G_r$, respectively. Note that M_r is a rank-maximal matching in G. Briefly, to compute $M_1, \ldots, M_r$ the algorithm constructs the *reduced graphs* $G'_1, \ldots, G'_r$ from $G_1, \ldots, G_r$, which have the properties that the sets of rank-maximal matchings in G'_i and G_i are the same and coincide with the set of maximum matchings of G'_i arisen by the application of an augmenting path in G'_i with respect to any rank-maximal matching of G'_{i-1}. Thus, the Gallai-Edmonds decomposition of G'_i not only determines which vertices of G'_i will always belong to the rank-maximal matchings in G'_i, but it also allows to reject irrelevant edges from G_i (edges that do not belong to any rank-maximal matching in G_i) to compute the reduced graph G'_i.

Chapter 3 of the thesis is focused on the problem of maintaining a rank-maximal matching in the preference graph G that may change over time. A single modification of G may consists of addition/deletion of a single vertex with edges adjacent to it, and of addition/deletion of a single edge. The task of the algorithm is to update the rank-maximal matching after each modification of G (the algorithms that work under such a scenario are called *dynamic*). The main result of Chapter 3 is a dynamic algorithm that in time $O(min(cn, n^2) + m)$, where c is the maximal rank of an edge in the current solution, updates the rank-maximal matching after each modification of G. In Chapter 3.6 it is shown that all modifications can be handled in the same time complexity as adding a new vertex with adjacent edges to G. The dynamic algorithm of Chapter 3 maintains the same data structure as the combinatorial algorithm described in Chapter 2, consisting of the reduced graphs $G'_1, \ldots, G'_r$ of $G_1, \ldots, G_r$, their Gallai-Edmonds decompositions, and their rank-maximal matchings $M_1, \ldots, M_r$. Thus, suppose that a new applicant a with adjacent edges has been added to G. Denote the extended graph by H. The key ingredient that allows to construct the algorithm comes to a deep understanding of the differences between the reduced graphs $G'_1, \ldots, G'_r$ for G and the reduced graphs $H'_1, \ldots, H'_r$ for H, between their Gallai-Edmonds decompositions, and their rank-maximal matchings $M_1, \ldots, M_r$ and $N_1, \ldots, N_r$. The author proves, which is crucial and highly non-trivial here, that every rank-maximal matching N_i can be obtained by the application of an appropriately chosen M_i-alternating or M_i-augmenting path in H_i starting in the newly added vertex a. So, the main combinatorial result of Chapter 3 asserts that a rank-maximal matching in H can be obtained by the application of some alternating/augmenting path with respect to a rank-maximal matching in G. It is worth noting here that the presented dynamic algorithm is the first that exploits this property; the other known dynamic algorithms update the data structure gradually by applying a number of alternating/augmenting paths. Based on these ideas the dynamic algorithm can update the reduced graphs $G'_1, \ldots, G'_r$ and their Gallai-Edmonds decomposition also in time $O(min(cn, n^2) + m)$.

Chapter 4 of the thesis is focused on so-called *manipulation strategies* for an unfair applicant. In the scenarios considered in the thesis it is assumed that every applicant knows the preferences of all the remaining applicants, and that a rank-maximal matching is chosen at random by some external authority from the set of all rank-maximal matchings. It turns out that this knowledge can be used by an unfair applicant, called *manipulator*, who can falsify his list of preferences to be assigned to a post of higher (true) rank than if he had given the true list of his preferences (the highest posts for a are the posts adjacent to a by rank 1 edges). It is assumed that only the manipulator can be untruthful, other applicants show their true preference lists to the authority. It is quite easy to give an example which shows that falsifying the preference list can be beneficial to the manipulator. Chapter 4 provides three manipulations strategies, called *best non-first*, *min-max*, and *improve best*, asserting different profits to the manipulator. To discuss the first two strategies we need to introduce the notion of f-posts: a post p is an *f-post* if p is matched

Ml

to a rank 1 edge in every rank-maximal matching in the preference graph restricted to all the applicants with the manipulator removed. It is quite easy to show (Lemma 28 in Chapter 4.3) than whenever we have a rank 1 edge between the manipulator and some non-f-post, then the manipulator is adjacent to some rank 1 edge in every rank-maximal matching. In this case the manipulator has no reason to falsify his preference list. Otherwise, the external authority may match the manipulator to some non-f-post, not necessarily with the highest rank (Lemma 30 in Chapter 4.3). The simple and elegant *best non-first* strategy ensures that the authority will always assign the manipulator to a non-f-post with the highest true rank. It turns out that the strategy *best non-first* is not always optimal for the manipulator. Sometimes, the use of the strategy *min-max* may assert that the manipulator is matched to some f-post with higher true rank than the highest rank of a non-f-post. In this strategy, for all f-posts p processed in order from most attractive to least attractive, the manipulator checks whether it is possible to set his preference list so that in every rank-maximal matching he is matched to p. Obviously not every f-post p has this property. In Chapter 4.5 the author gives a very ingenious and non-trivial polynomial algorithm, based on the concept of *critical ranks*, which allows to test whether such a list exists for a fixed f-post p. The last presented, simple and elegant strategy *improve best* ensures that in some (but not every) rank-maximal matching the manipulator is assigned to a post with the highest true rank.

Chapter 5 of the thesis is devoted to the *rectangle tiling* problem, in which for a given two-dimensional array A (called *rectangle*) and a number p we are looking for a tiling of A by p non-overlapping rectangles (called *tiles*) that minimizes the weight of every rectangle, where the *weight* of a rectangle is the sum of entries of A covered by this rectangle. It is known that the rectangle tiling problem is NP-complete. Let $w(A)$ denote the weight of the rectangle A. Note that in any tiling of A there is a tile that has the weight at least $\left\lceil \frac{w(A)}{p} \right\rceil$. Hence, $\left\lceil \frac{w(A)}{p} \right\rceil$ is a natural lower bound on the weight of a tile that is used to measure the quality of approximation algorithms for the rectangle tiling problem. Thus, we say that an algorithm for the rectangle tiling problem has *approximation ratio* α if it constructs in polynomial time a tiling in which every tile has weight at most $\alpha \cdot \left\lceil \frac{w(A)}{p} \right\rceil$. The rectangle tiling problem has attracted the attention of many researchers, in particular, the upper and the lower bounds on approximation ratio have been improved many times. Eventually, in 2004, Paluch designed an algorithm with the approximation ratio $\frac{17}{8}$ and proved it is best possible. Chapter 5 of the thesis is focused on the *binary rectangle tiling* problem, in which every entry of A is from the set $\{0, 1\}$. The main result (Theorem 61) provides an $\left(\frac{3}{2} + \frac{p^2}{w(A)}\right)$–approximation algorithm for the binary rectangle tiling problem. In particular, for instances where the weight of A is much larger than p, the approximation factor approaches to $\frac{3}{2}$. The algorithm itself is based on the choice of the better of two presented tiling strategies. Both of these strategies construct special type tilings, one "from left to right" and the second "from top to bottom", with respect to so-called *boundaries and shadows*. Then it is proved, using the weak duality theorem for linear programming, that the better of these strategies yields a tiling guaranteeing the approximation factor $\left(\frac{3}{2} + \frac{p^2}{w(A)}\right)$. Moreover, Theorem 62 proves that the constant $\frac{3}{2}$ is the matching lower bound for the approximation ratio in the binary rectangle tiling problem. Indeed, the theorem provides, for every even number p and every sufficiently large W, an example of a binary matrix A of weight at least $\geqslant W$ such that any tiling of A contains a tile with the weight at least $\frac{3}{2} \cdot \frac{w(A)}{p} + 1$.

Overall evaluation

The presented thesis contains a number of deep and original results from the area of algorithmics and combinatorial optimizations. In my opinion, there are three excellent (and the most

TU

important) results: the dynamic algorithm and the manipulation strategy *mini-max* for the rank-maximal matching and the $\frac{3}{2}$−approximation algorithm for the binary rectangle tiling problem. It is worth noting that these results are significantly different, i.e., we find there the results of a technical nature as well as those based on interesting and original ideas. All this proves the author's great skills to solve difficult algorithmic and combinatorial problems.

The dynamic algorithm maintaining the rank-maximal matching, presented in Chapter 3, is technical and complicated, and its construction required a thorough understanding of the effects made by a single change in the graph on the structures that make possible to maintain the rank-maximal matching. The result itself is very deep and important in the context of the studies involving matchings in graphs with preferences. A certain drawback of this part is the fairly large number of editorial errors that make it difficult to read (the list of spotted errors is attached to the review). For example, the set $Alive(i)$, which plays the crucial role in the design of the algorithm, is not defined anywhere in the thesis. It should be noted, however, that this part of the dissertation is long and technically complicated, and the author himself put much effort to make it easier for the reader. In particular, the technical lemmas (Lemmas 14-15) used in the correctness proof (Theorem 17) of the dynamic algorithm, are very well abstracted. Additionally, the author has included many thoroughly described examples illustrating the work of the algorithm, which significantly facilitates its understanding.

Chapter 4 contains an elegant manipulation strategy *min-max*. The strategy is based on interesting and original ideas, which allow to place irrelevant posts (to which the manipulator never will be matched) on high positions on the manipulator's preference list. These ideas allow to obtain a very clever algorithm for testing whether the manipulator can be assigned to a certain f-post in every rank-maximal matching.

Chapter 5 contains a $\frac{3}{2}$−approximation algorithm for the binary rectangle tiling problem. Although the result is partly inspired by Paluch's 2004 work, the algorithm itself contains new ingredients. In particular, it is worth mentioning the tiling strategies based on the notions of *boundaries and shadows*, as well as a very elegant and nontrivial use of the weak duality theorem for linear programming for the correctness proof of the algorithm.

Among the results contained in the thesis only those concerning the manipulation strategies were published in conference materials of a very good computer science conference COCOON (Core A). The remaining work is currently under review. However, it is worth noting that Mr. Pratik Ghosal is also a coauthor of two papers published in computer science conference materials (including one excellent conference SODA) and one paper published in *Theoretical Conference Science.* As for this stage of the career, these achievements should be considered very good.

Conclusion

The presented thesis satisfies, with big margin, all the statutory and customary requirements for PhD dissertations and it certainly constitutes the basis for awarding a PhD degree in the field of computer science to Mr. Pratik Ghosal.

Tomasz Kociumaka

List of errors

Page iii, line 10, comma at the beginning of the line.
Page 12, line 15, it is: $O + |U/2|$, should be: $O + |U|/2$.
Page 15, line 4, $Alive(i)$ is not defined.
Page 23, line -8, it is: Algorithm 2, should be: Algorithm 3.
Page 23, line -2, it is: we move p_2 to AE_u, should be: we move (a_1, p_2) to AE_u.
Page 24, line 7, $Alive(i)$ is not defined, but it is used throughout the dissertation.
Page 25, line -6, line 8 in Algorithm 3 contains "else" statement.
Page 27, line -13, a is not defined.
Page 29, line -7, it is: Let P' be any even length M-alternating path starting at a_0. Should be: Let P' be any even length M-alternating path starting at a_0 and ending in $V - C$.
Page 31, line 11, it is: 1(d), should be: 1(b).
Page 34, line 1, it is: very, should be: every.
Page 35, line 15, it is: $k+1$ edges of $N \cap \mathcal{E}_2$ and k edges of $M' \cap \mathcal{E}_2$. There is probably something wrong here and throughout the paragraph. Did you mean: k edges of $N \cap \mathcal{E}_2$ and $k+1$ edges of $M' \cap \mathcal{E}_2$.
Page 36, line 14, lines pointing to Algorithm 4 are incorrect.
Page 36, line 15, it is *augmenting* or *non-augmenting*, should be: *non-augmenting* or *augmenting*.
Page 39, line 16 and line -4, it is: line 17, should be: line 16.
Page 42, line -2: two "end of proof" boxes.
Page 54, line 19, comma at the beginning of the line.
Page 64, line 8, "dot" is missing.
Page 66, line -6, it is: Figure 1, should be: Figure 4.4.
Page 70, line -3: it is quite unfortunate to denote the graph by G'. In the next line we have G'_i, which means either the reduced graph of G_i or the graph obtained from G' by taking edges of rank at most $\leqslant i$.
Page 78, line -7, it is: has, should be: have.
Page 89, line -9: is this constraint correct? We have $B'_1 \neq B_1$. Are not we in the case 2.(a).iii in this situation? Is the dual program properly defined in this example? Why we do not have y'_1, w'_2 in the objective function of the dual?
Page 93, line 7, it is: zi, j, should be: $z_{i,j}$.

Hr