MŰEGYETEM 1782

Parameterized complexity of graph modification and stable matching problems

by Ildikó Schlotter

PhD theses

supervised by Dr. Dániel Marx

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Computer Science and Information Theory

2010

1 Introduction

In the last two decades, parameterized complexity has become a dynamic research area in theoretical computer science. Its framework was developed by Downey and Fellows [10], and its most important aim is to construct efficient algorithms for computationally hard problems.

In the parameterized approach, each input of a given problem has an integer associated with it, called the *parameter*. Thanks to this notion, we can express the running time of an algorithm as a function depending both on the length n of the input and the parameter k. Therefore, this methodology enables us to study the computational complexity of a problem in a more detailed manner than in the classical setting, where the running time of an algorithm is usually regarded only as a function of n. As a consequence, the parameterized framework offers an opportunity to investigate the intricate complexity issues of problems that formerly were only classified as being NP-hard. Since the parameterized complexity analysis leads to a better understanding of such problems, it yields more efficient algorithms.

The basic idea of the parameterized approach is to look for algorithms that have moderate running time if the parameter has small value. We say that an algorithm is *fixed-parameter tractable* or FPT, if it has running time $f(k)n^{O(1)}$ for some computable function f. Note that the running time has polynomial dependence on the length n of the input, but f can be an exponential or an even faster growing function. However, it should be emphasized that the exponential part of the running time is restricted to depend only on the parameter k. We say that a parameterized problem is FPT, if it admits an FPT algorithm. The parameterized complexity investigation of a problem might also reveal its *W[1]-hardness*, which gives us strong evidence that the problem is unlikely to be solvable by an FPT algorithm.

This dissertation contains the investigation of several problems from the parameterized viewpoint. The studied problems can be divided into two groups: problems that can be viewed as graph modification problems, and problems related to the classical Stable Marriage problem by Gale and Shapley [14]. Let us briefly introduce here the topics of these investigations.

- **Apex graphs.** Planarity is a central notion in classical graph theory. This is partly due to the fact that many problems that are NP-hard in general, admit polynomial-time algorithms [11, 19] or linear-time approximation schemes [5, 30] when restricted to planar graphs. In many cases, algorithms for planar graphs can be extended for "almost planar" graphs as well [34], therefore the study of such graphs is an important research area.

  The thesis deals with the recognizing k-*apex* graphs. We say that a graph is k-apex, if it contains k vertices such that removing these vertices results in a planar graph. The recognition of such graphs is an NP-hard problem [29], so the dissertation investigates this problem by regarding k as a parameter. Our research uses the celebrated results of Robertson and Seymour in graph minor theory [41, 40, 42, 43], as well as some other algorithmic achievements of the area [8, 38, 4].

- **Almost isomorphic graphs.** The complexity of deciding whether two graphs are isomorphic is among the most important open questions of complexity theory. Polynomial-time algorithms exist for special cases, such as the case where the input graphs are planar graphs [21] or interval graphs [31]. The thesis studies a variant of this problem, which asks whether two graphs can be made isomorphic by deleting k vertices from the larger graph. The dissertation contains the investigation of the parameterized complexity of this graph modification problem for several graph classes $\mathcal{H}$ and $\mathcal{G}$, with the parameter being the number k of vertices to delete.

- **Stable marriage with ties.**

  The Stable Marriage (also called Stable Matching) problem models situations where we are looking for a matching in a two-sided market that respects the preferences of the agents and fulfills a certain stability criterion. The classical version of the problem can be illustrated by the following example: we are given a set of men and a set of women, and each person ranks the members of the opposite sex in order of preference. The task is to find a matching (between men and women) that is *stable* in the sense that there are no man and woman who would rather form a pair with each other instead of accepting the given matching.

The thesis contains the parameterized examination of the Stable Marriage with Ties and Incomplete Lists (SMTI) problem, where the agents need not rank each member of the opposite sex, and the preference lists of the agents might also contain ties. This extension is highly motivated by practice, see e.g. the various applications in educational admission systems [6, 2, 3].

As opposed to the case when no ties are allowed, an instance of SMTI might admit stable matchings of various sizes. In such a case, the usual aim is to maximize the size of the stable matching. The resulting problem is called the Maximum Stable Marriage with Ties and Incomplete Lists (or MaxSMTI) problem. It has been proved by Iwama et al. [25] that finding a stable matching of maximum size in this situation is NP-hard. Since then, several researchers have attacked the problem, most of them presenting approximation algorithms [24, 27]. In constrast, this thesis studies the parameterized complexity of this problem. The examined parameters are the number of ties, the maximum or the total length of the ties appearing in the input instance.

- **Hospitals/Residents with couples.** A slightly more general formulation of the Stable Marriage problem is the Hospitals/Residents problem, introduced by Gale and Shapley [14] to model the "many to one " version of Stable Marriage. This can be illustrated by the following example, originating from practice: we are given a set of hospitals, each having a number of open positions, and a set of residents applying for jobs in the hospitals. Each resident has a ranking over the hospitals, and conversely, each hospital has a ranking over the residents. The task of the Hospitals/Residents problem is to assign as many residents to some hospital as possible, with the restrictions that the capacities of the hospitals are not exceeded, and the resulting assignment is stable in a certain sense.

  In the Hospitals/Residents with Couples or HRC problem, first introduced by Roth [45], residents may form couples and thus have joint rankings over the hospitals. This means that instead of ranking the hospitals individually, couples rank *pairs of hospitals* according to their preferences. This allows them to express intentions such as being applied in the same hospital, or in hospitals that are close to each other. The task of the Hospitals/Residents with Couples problem is to decide whether a stable assignment exists in such an instance. This problem has numerous practical applications such as the detailing process of the US Navy [39] or the NRMP program for assigning medical residents in the USA [45, 46].

  Given an isntance of HRC that contains no couples, we can always find a stable assignment of maximum size in linear time [14, 18]. But if we have to deal with couples, then a stable assignment need not exist, as discovered by Roth [45]. Later, Ronn [44] proved that HRC is NP-hard. We study the parameterized complexity of this problem, where we take the number of couples to be the parameter. Since this number is typically small compared to the total number of residents (and thus to the total size of the instance), this parameterization can be useful in practice.

In those cases where the obtained results show the computational hardness of some optimization problem even from the parameterized point of view, we also investigate the theoretical possibilities for applying local search methods for the given problem. Local search is a simple and extremely useful metaheuristic that is widely applied in optimization problems where an optimal solution cannot be found in polynomial time [1]. Its basic idea is to start with a feasible solution of the problem, and then iteratively improve it by searching for better solutions in the local neighborhood of the actual solution. Thus, local search algorithms explore the space of feasible solutions by moving from solution to solution. The key task of this iterative method is the following: given an instance I of the problem, a solution S for I, and an integer ℓ, find a solution S' for I in the ℓ-neighborhood of S that is better than S.

Clearly, by searching a larger neighborhood, we can hope for a better solution. The efficiency of local search can be significantly improved if this neighborhood exploration task can be carried out effectively, allowing us to search relatively large neighborhoods in a moderate time. Usually, the neighborhood exploration task turns out to be NP-hard if the radius ℓ is unbounded, so it is a natural question to ask whether this problem is FPT, if ℓ is the parameter. Such investigations have already been carried out in the literature [35, 26, 47, 28, 12] (see also [33]). We also examine this question in the framework of parameterized complexity, with connection to both MaxSMTI and HRC.

2 Research objectives

The main objective of the dissertation was to clarify the parameterized complexity of the various graph modification and stable matching problems mentioned in the previous section.

When investigating the computational complexity of a problem, the first task is to classify it either as polynomial-time solvable or as NP-hard. Since we cannot hope to give polynomial-time algorithms for NP-hard problems, our aim was to examine these problems from the parameterized approach.

Since the parameterized framework considers the complexity of a problem as a function of the parameter as well, the next natural step of such research is to identify possibly relevant parameters of a given problem that might influence its computational complexity. Given a parameterized problem, the goal is to either provide an FPT algorithm for it, or show that it is W[1]-hard.

Since the W[1]-hardness of a parameterized problem shows that it is very unlikely to admit an FPT algorithm, dealing with such problems can be very hard. Since local search algorithms are of practical importance even in those cases where exact algorithms have intractable running time, another goal of this dissertation was to investigate whether local search can be applied efficiently in those problems that turn out to be W[1]-hard.

Another possible research direction was the application of approximation algorithms. In those cases where there is no hope for giving a polynomial-time approximation algorithm, a possible objective is to give an approximation algorithm running in FPT time (with some parameterization). We carry out such investigations in a few cases as well.

3 Research methodology

3.1 Parameterized complexity

Besides using the theoretical background of classical complexity theory, the thesis applies various techniques from the area of parameterized complexity. For a comprehensive introduction to this area, refer to the monograph of Downey and Fellows [10]. For more details and a newer perspective, see also the book by Niedermeier [37] or by Flum and Grohe [13].

When dealing with an NP-hard problem, the aim of parameterized complexity is to yield algorithms that are moderately exponential in the sense that the exponential part of the running time can be restricted to a limited part of the input, called the parameter. Thus, the goal is to identify a parameter k for each input that is expected to be small in practical instances. Given this parameter k, we look for an algorithm with running time $f(k)n^{O(1)}$ where n is the input length and f is some arbitrary function that depends only on k. Such an algorithm is called *fixed-parameter tractable* or *FPT*.

The counterpart of fixed-parameter tractability is the notion of *W[1]-hardness*. Roughly, this can be thought of as the parameterized analog of NP-hardness. The W[1]-hardness of a parameterized problem can be proven by means of presenting a so-called FPT-reduction from some already known W[1]-hard problem. Most FPT-reductions in this dissertation are from the standard parameterized version of the Maximum Clique problem, where the parameter is the size of the clique we are looking for.

Each of the previous definitions of parameterized complexity can be generalized to fit into a setting where we allow two or even more integers to be parameters. This is sometimes called a *combined parameterization*. For example, if we parameterize a given problem with two parameters, then an FPT algorithm has to run in $f(k_1, k_2)n^{O(1)}$ for some function f, where n is the length of the given input, and k_1 and k_2 are the two parameters associated with the input. It is not hard to see that adding one more parameter to a given problem can make the quest for an FPT algorithm easier. Hence, if a problem turns out to be W[1]-hard with a given parameterization, then it makes sense to ask if it becomes FPT after regarding one more property of the input as parameter. Based on this argument, the thesis contains several investigations where a problem is considered with a combined parameterization.

3.2 Efficient local search

When defining the task of a *local search algorithm* for an optimization problem Q, we suppose that there is some distance $d(S_1, S_2)$ defined for each pair (S_1, S_2) of solutions for some given instance I of Q. For simplicity, we assume that Q is a maximization problem, with T denoting its objective function. The efficiency of a local search algorithm for Q depends on the fast implementation of the following key task that has to be solved iteratively.

Strict local search for Q
Input: (I, S_0, ℓ) where I is an instance of Q, S_0 is a solution for I, and $\ell \in \mathbb{N}$.
Task: If there exists a solution S for I such that $d(S, S_0) \leq \ell$ and $T(S) > T(S_0)$, then output such an S.

We call an algorithm solving this task a *strict local search* algorithm for Q. The reason why we use the adjective "strict" in the name of this problem is that in order to solve it, we are forced to find a solution better than S_0 in the ℓ-neighborhood of S_0. In some cases, this can be a hard task even if an optimal solution can be found easily (see e.g. the Vertex Cover problem on bipartite graphs [28]).

In contrast, a *permissive* local search algorithm for Q is allowed to output a solution that is not close to S_0, provided that it is better than S_0. In local search methods, such an algorithm is as useful as its strict version. Formally, its task is as follows.

Permissive local search for Q
Input: (I, S_0, ℓ) where I is an instance of Q, S_0 is a solution for I, and $\ell \in \mathbb{N}$.
Task: If there exists a solution S for I such that $d(S, S_0) \leq \ell$ and $T(S) > T(S_0)$, then output *any* solution S' for I with $T(S') > T(S)$.

Note that a strict local search algorithm for Q can be considered as a permissive local search algorithms for Q. Therefore, proving that no permissive local search algorithm exists for some problem is clearly a stronger result than proving that no strict local search algorithm exists for it.

In most cases, the above neighborhood search problems are NP-hard. However, they can usually be solved by an $O(n^\ell)$ brute force algorithm, where n is the length of the input and ℓ is the radius of the neighborhood considered. Therefore, it is natural to ask if this running time can be reduced to a form $f(\ell)n^{O(1)}$ for some function f, or in other words, whether this local search problem can be solved by an FPT algorithm, where ℓ is the parameter. We study such problems in connection to MaxSMTI and HRC.

3.3 FPT-approximation

Besides parameterized complexity, another research direction dealing with computationally hard problems is the area of approximation algorithms. Although this thesis does not concentrate on such algorithms in the first place, in some cases we investigate the possibilities for giving a so-called *FPT-approximation algorithm* for a given optimization problem. Such algorithms are similar to classical approximation algorithms in the sense that they guarantee a certain ratio describing the quality of the produced output. However, as opposed to their classical version, they do not have to run in polynomial time. Instead, they can have FPT running time with respect to some parameterization. The dissertation contains FPT-inapproximability results concerning some optimization problems connected to stable matchings.

4 New scientific results

Here I present the new results of the dissertation, grouped according to the problems considered. All the results presented as "Claims" here are new scientific achievements of the dissertation, each being an outcome of joint work together with my supervisor.

Thesis 1. Recognizing k-apex graphs

We say that a graph is k-*apex*, if it contains k vertices such that removing these vertices results in a planar graph. Given a graph G and an integer k, the k-Apex problem asks if G is k-apex. Since the family of k-apex graphs is minor-closed for each fixed k, the results of Robertson and Seymour in graph minor theory [41, 42] imply that for each fixed k there is a cubic algorithm for the k-Apex problem. However, the proof of this result is existential, and does not actually construct an FPT algorithm. In the thesis, I present an FPT algorithm for this problem that we proposed in our paper [MS07] together with my supervisor. The algorithm runs in quadratic time for each fixed k.

> **Claim 1.1** The k-Apex problem can be solved in $f(k)n^2$ time for some function f, where n denotes the number of vertices in the input graph.
> (Theorem 2.4.3 of the dissertation.)

The algorithm presented in the thesis was the first published one to solve the k-Apex problem in FPT time with k being the parameter. Ken-ichi Kawarabayashi has recently published [22] an even faster algorithm for this problem.

Thesis 2. Recognizing almost isomorphic graphs

For two graph classes $\mathcal{H}$ and $\mathcal{G}$, we define the Cleaning($\mathcal{H}, \mathcal{G}$) problem: given a pair of graphs (H, G) with $H \in \mathcal{H}$ and $G \in \mathcal{G}$, find a set of vertices S in G such that $G - S$ is isomorphic to H. We parameterize this problem by the number $k = |V(G)| - |V(H)|$ of vertices that have to be deleted. We investigated the parameterized complexity of this problem for several graph classes $\mathcal{H}$ and $\mathcal{G}$.

Since the case $k = 0$ yields exactly the famous Graph Isomorphism problem, an FPT algorithm for the general case (when both H and G can be arbitrary graphs) would solve one of the most important open questions in algorithmic graph theory by showing that Graph Isomorphism is polynomial-time solvable. Since we could not expect such a result, we considered those special cases where the Graph Isomorphism problem is known to be polynomial-time solvable.

In the dissertation, we considered the class of trees, planar graphs, 3-connected planar graphs, and interval graphs, denoted by *Tree*, *Planar*, *3-Connected-Planar*, and *Interval*, respectively. The unique class of graph containing every graph is denoted by "$-$". Using this notation, the following theorems give new results concerning both the classical and the parameterized complexity of Cleaning($\mathcal{H}, \mathcal{G}$) problem for different graph classes.

> **Claim 2.1** The Cleaning(*3-Connected-Planar, Planar*) problem for two graphs H and G can be solved in $f(k)n^2$ time for some function f, where $|V(H)| = n$ and $|V(G)| = n + k$. Here, both H and G must be planar, and H has to be 3-connected as well.
> (Theorem 3.1.6 of the dissertation.)

Graph classes $(\mathcal{H}, \mathcal{G})$	Parameter: None	Parameter: $\|V(H)\|$	Parameter: $\|V(G)\| - \|V(H)\|$
(*Tree*, *Tree*)	P [36]	FPT (trivial)	FPT (trivial)
(*Tree*, −)	NP-complete [15]	W[1]-complete [9]	FPT [Claim 2.3]
(*3-Connected-Planar*, *Planar*)	NP-complete [Claim 2.2]	FPT [11]	FPT [Claim 2.1]
(−, *Planar*)	NP-complete [15]	FPT [11]	Open
(*Interval*, *Interval*)	NP-complete [Claim 2.6]	W[1]-hard [Claim 2.5]	FPT [Claim 2.4]

Table 1: Summary of some known results for the Cleaning$(\mathcal{H}, \mathcal{G})$ problem.

Claim 2.2 The Cleaning(*3-Connected-Planar, 3-Connected-Planar*) problem is NP-complete.
(Theorem 3.1.1 of the dissertation.)

Claim 2.3 The Cleaning(*Tree*, −) problem for two graphs H and G can be solved in $f(k)n^3$ time for some function f, where $|V(H)| = n$ and $|V(G)| = n+k$. Here, H must be a tree, but G can be an arbitrary graph.
(Theorem 3.2.3 of the dissertation.)

Claim 2.4 The Cleaning(*Interval, Interval*) problem for two graphs H and G can be solved in $f(k)n^2$ time for some function f, where $|V(H)| = n$ and $|V(G)| = n + k$. Here, both H and G must be interval graphs.
(Theorem 4.3.1 of the dissertation.)

Claim 2.5 The Cleaning(*Interval, Interval*) problem is W[1]-hard when parameterized by the number of vertices of the smaller input graph.
(Theorem 4.2.1 of the dissertation.)

Claim 2.6 The Cleaning(*Interval, Interval*) problem is NP-complete.
(Theorem 4.2.1 of the dissertation.)

The new results of Thesis 2 and some already known results of the area are summarized in Table 1. These new results were published in the papers [MS09a], [MS09b], and [MS08].

Thesis 3. Investigation of the Stable Marriage with Ties and Incomplete Lists problem

The input of the Stable Marriage with Ties and Incomplete Lists (SMTI) problem is a set of men, a set of women, and a preference list for each person. The preference list of a person p is an ordering of those members of the opposite sex that are acceptable for p. This ordering is not necessarily strict, because it may contain *ties*. The task of the SMTI problem is to find a *stable matching*, i.e. a matching between men and women such that there exists no man-woman pair (m, w) (not present in the matching) such that both m and w strictly prefer each other to their partners in the matching. In the optimization version of this problem, called MaxSMTI, the task is to find a stable matching of maximum size. (The size of a matching M is the number of pairs in it.)

If no ties are present in an instance, then MaxSMTI is solvable in linear time [14, 18]. By constrast, it was proved by Manlove at el. [32] that MaxSMTI is NP-complete even in the very much restricted case when ties only occur at the end of women's preference lists, and each tie has length 2 (where the length of a tie is the number of persons contained in it). Completing these results, I settled the parameterized complexity of MaxSMTI for those parameterizations where the possible parameters are the number of ties or the total length of ties.

Claim 3.1 The MaxSMTI problem is W[1]-hard, if the parameter is the number of ties.
(Theorem 5.1.2 of the dissertation.)

Claim 3.2 The MaxSMTI problem is solvable in $O(T^T n)$ time, where T is the total length of the ties and n is the length of the input instance.
(Theorem 5.1.1 of the dissertation.)

To consider the applicability of local search for MaxSMTI, we define the distance of two different stable matchings for some instance I of MaxSMTI as the size of their symmetrical difference. Then, the task of a permissive local search algorithm for MaxSMTI is as follows: given an instance I of MaxSMTI, together with a stable matching M for I and an integer ℓ, output any stable matching for I that is greater than M, provided that there exists a stable matching for I having distance at most ℓ from M with size greater than M. (If no stable matching in the ℓ-neighborhood of M is greater than M, then the output can be arbitrary.)

My results shown below rule out the possibilities for an efficient algorithm solving this task (assuming that the standard complexity theoretic assumption W[1] $\neq$ FPT holds).

Claim 3.3 If W[1] $\neq$ FPT, then there is no permissive local search algorithm for MaxSMTI that has FPT running time with parameter ℓ denoting the radius of the search neighborhood, even if each tie has length 2.
(Theorem 5.2.2 of the dissertation.)

Claim 3.4 If W[1] $\neq$ FPT, then there is no permissive local search algorithm for MaxSMTI that has FPT running time, if the parameters are the radius of the search neighborhood and the number of ties.
(Theorem 5.2.1 of the dissertation.)

I also considered two variants of SMTI where we aim for a stable matching that is not necessarily of maximum size, but has some other desirable property expressing some kind of fairness. The key idea in both of these problems is the following: given a matching M, we define the *cost* of a person (with respect to M) as the rank of its partner in its perference list. This definition describes the satisfaction of a given person in the matching M.

Now, an *egalitarian* stable matching for an SMTI instance I is a stable matching for I that minimizes the total cost of the persons among all stable matchings for I. Similarly, a *minimum regret* stable matching for I is a stable matching for I that minimizes the maximum cost of any person. The task of the Egalitarian Stable Marriage with Ties and Incomplete Lists (or EgalSMTI) problem is to find an egalitarian stable matching for the given SMTI instance; the Minimum Regret Stable Marriage with Ties and Incomplete Lists (or MinregSMTI) problem is defined analogously.

Both these problems are hard to approximate, since it has been shown by Halldórsson [20] that for some $\delta > 0$ it is NP-hard to approximate EgalSMTI and MinregSMTI within a ratio of $\delta N(I)$, where $N(I)$ denotes the number of men in the input instance I. However, if there are no ties, then a minimum regret or an egalitarian stable matching can be found in polynomial time [23, 17]. On the one hand, I generalized this fact by giving an FPT algorithm for these problems, where the parameter is the total length of ties. On the other hand, I also showed that the problem remains inapproximable, if we only allow the number of ties to be the parameter.

Claim 3.5 EgalSMTI and MinregSMTI can be solved by an FPT algorithm, if the parameter is the total length of ties.
(Theorem 5.3.1 of the dissertation.)

	Parameters		
	$T_{\max} = 2$ (and ℓ)	T_{number} (and ℓ)	T_{total}
MaxSMTI	NP-hard [32]	W[1]-hard [Claim 3.1]	FPT [Claim 3.2]
Perm. Local Search for MaxSMTI	No FPT alg. [Claim 3.3]	No FPT alg. [Claim 3.4]	FPT [Claim 3.2]
Approximation for EgalSMTI	No poly. alg. has ratio $N^{1-\varepsilon}$ if $\varepsilon > 0$ [32]	No FPT alg. has ratio δN for some $\delta > 0$ [Claim 3.6]	FPT, exact [Claim 3.5]
Approximation for MinregSMTI	No poly. alg. has ratio $N^{1-\varepsilon}$ if $\varepsilon > 0$ [32]	No FPT alg. has ratio $N^{1-\varepsilon}$ if $\varepsilon > 0$ [Claim 3.7]	FPT, exact [Claim 3.5]

Table 2: Summary of some newly obtained and other known results connected to SMTI (assuming W[1] $\neq$ FPT and P $\neq$ NP). The parameter ℓ is only defined in the local search problem for MaxSMTI, and denotes the radius of the neighborhood search. The results hold for the permissivel version of the local search problem. We denote by T_{number}, $T_{\max}$, T_{total} the number of ties, the maximum length, and the total length of ties in a given instance. Finally, N denotes the number of men.

> **Claim 3.6** There is a $\delta > 0$ such that if W[1] $\neq$ FPT, then no approximation algorithm for EgalSMTI with ratio δN can run in FPT time, if the parameter is the number of ties. Here, N denotes the number of men in the input.
> (Theorem 5.3.2 of the dissertation.)

> **Claim 3.7** If W[1] $\neq$ FPT and $\varepsilon > 0$, then no approximation algorithm for MinregSMTI with ratio $N^{1-\varepsilon}$ can run in FPT time, if the parameter is the number of ties. Again, N denotes the number of men in the input.
> (Theorem 5.3.3 of the dissertation.)

It might worth mentioning that each hardness result presented by Thesis 3 holds even in the case where ties are restricted to be in women's preference lists. The new results of Thesis 3 and some already known results of the area are summarized in Table 2. Our results appeared in the paper [MS09c].

Thesis 4. Investigation of the Hospitals/Residents with Couples problem

In the Hospitals/Residents with Couples (HRC) problem, we are given a set of hospitals, each having a number of open positions, and a set of residents applying for jobs in the hospitals. Some residents are singles, but some of them form couples. Each hospital has a ranking over the residents, each single resident has a ranking over the hospitals, and each couple has a ranking over the pairs of hospitals. These rankings describe a strict preference order.

The task of the Hospitals/Residents with Couples problem is to find a *stable assignment* mapping residents to hospitals with the restrictions that the capacities of the hospitals are not exceeded. We say that an assignment is stable, if there is no blocking pair for it. A blocking pair may consist of a single resident and a hospital such that both of them would benefit from contracting with each other instead of accepting the given assignment, or a blocking pair may be formed by a couple of residents and a pair of hospitals, such that both hospitals and also the couple would benefit from contracting with each other as compared to the given assignment. For a more precise definition, see Chapter 6 of the dissertation.

The optimization version of HRC, called MaxHRC, asks for a stable assignment having maximum size, where the size of the assignment is the number of residents that are assigned to some hospitals.

When there are no couples in an instance of HRC, we can always find a stable assignment of maximum size in linear time [14, 18]. I determined the parameterized complexity of this problem in the case where the parameter is the number of couples.

> **Claim 4.1** The HRC problem (and hence the MaxHRC problems as well) is W[1]-hard, if the parameter is the number of couples, even if each hospital has capacity 1.
> (Theorem 6.1.1 of the dissertation.)

To apply the framework of local search in the case of MaxHRC, we say that the distance of two assignments A and A' (mapping residents to hospitals) is the number of residents that are not applied in the same hospital in A as in A'. Now, we can define the task of a strict local search algorithm for MaxHRC: given an instance I of MaxHRC, together with a stable assignment A for I and an integer ℓ, output a stable assignment A' for I that is greater than A and has distance at most ℓ from A.

The permissive version of this tasks allows us to output any stable assignment for I that has size greater than A, provided that there exists a stable assignment for I having distance at most ℓ from A and size greater than A. (If no stable assignment in the ℓ-neighborhood of A is greater than A, then the output can be arbitrary.)

As shown by Claims 4.2 and 4.3 below, I proved that a permissive local search algorithm is not likely to run in FPT time if the parameter is only the radius ℓ, but even a strict local search algorithm can be given that runs in FPT time if not only ℓ but also the number of couples is a parameter.

> **Claim 4.2** If W[1] $\neq$ FPT, then there is no permissive local search algorithm for MaxHRC that has FPT running time with parameter ℓ, denoting the radius of the search neighborhood, even if each hospital has capacity 1.
> (Theorem 6.2.1 of the dissertation.)

> **Claim 4.3** There is a strict local search algorithm for MaxHRC running in FPT time with combined parameters ℓ, denoting the radius of the search neighborhood, and the number c of couples. The randomized version of the presented algorithm runs in $O(\ell(72c)^\ell|I|)$, and gives a correct output with probability at least $(2\ell)^{-2\ell}$. Here, $|I|$ is the size of the input. The derandomized version of the algorithm runs in $O(\ell^{O(\ell)}c^\ell|I|\log|I|)$ time.
> (Theorem 6.2.2 of the dissertation.)

In the dissertation, we also investigated a simplified version of MaxHRC where we forget about preference. In this variant, we are given a set of hospitals, a set of single residents, and a set of resident couples, together with the list of acceptable residents for each hospital, a list of acceptable hospitals for each single resident, and a list of acceptable pairs of hospitals for each couple. The task of the Maximum Matching with Couples (MMC) problem is to find an assignment of residents to hospitals having maximum size that respects the acceptability criteria.

Clearly, if there are no couples, then this problem is equivalent to finding a maximum matching in a bipartite graph, and hence it is solvable in polynomial time. If couples are involved, the problem becomes hard. More precisely, the decision version of this problem is NP-complete [16, 7], even in the following special case: each hospital has capacity 2, and the acceptable hospital pairs for a couple are always of the form (h, h) for some hospital h. However, if the number of couples is small, which is a reasonable assumption in many practical applications, we proved that MMC becomes tractable, as stated in the following claim.

> **Claim 4.4** Maximum Matching with Couples can be solved in randomized FPT time, if the parameter is the number of couples.
> (Theorem 6.3.1 of the dissertation.)

Task:	Existence problem	Maximum problem	Local search algorithm with FPT running time	
Parameter:		c	ℓ	(c, ℓ)
With preferences	W[1]-hard [Claim 4.1]	W[1]-hard [Claim 4.1]	No permissive alg. [Claim 4.2]	Strict alg. (Claim 4.3]
Without preferences	P (trivial)	randomized FPT [Claim 4.4]	No permissive alg. [Claim 4.5]	Permissive alg. [Claim 4.4]

Table 3: Summary of the results of Thesis 4 (assuming W[1] $\neq$ FPT). We denote by c and ℓ the number of couples and the radius of the neighborhood search, respectively.

Claim 4.4 yields a way to cope with the MMC problem in the case where the number of couples is small. A possible way to deal with this problem when the number of couples could be large is the application of local search. Thus, I examined the possibilites for an efficient local search algorithm for MMC as well, and obtained the following result:

> **Claim 4.5** If W[1] $\neq$ FPT, then there is no permissive local search algorithm for MMC that has FPT running time with parameter ℓ, denoting the radius of the search neighborhood, even if each hospital has capacity 2.
> (Theorem 6.3.5 of the dissertation.)

The results of Thesis 4 are summarized in Table 3, and were published in [MS09d].

References

[1] E. H. L. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. Princeton University Press, Princeton, NJ, 2003. Reprint of the 1997 original [Wiley, Chichester].

[2] A. Abdulkadiroğlu, P. A. Pathak, and A. E. Roth. The New York City high school match. *American Economic Review*, 95(2):364–367, May 2005.

[3] A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The Boston public school match. *American Economic Review*, 95(2):368–371, May 2005.

[4] S. Arnborg, A. Proskurowski, and D. Seese. Monadic second order logic, tree automata and forbidden minors. In *CSL 1990: Proceedings of the 4th Workshop on Computer Science Logic*, volume 533 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 1991.

[5] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

[6] P. Biró. Student admissions in Hungary as Gale and Shapley envisaged. Technical report, University of Glasgow, Department of Computing Science, 2008.

[7] P. Biró and E. J. McDermid. Matching with sizes (or scheduling with processing set restrictions). Technical report, University of Glasgow, Department of Computing Science, 2010.

[8] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[9] Y. Chen and J. Flum. On parameterized path and chordless path problems. In *CCC 2007: 22nd Annual IEEE Conference on Computational Complexity*, pages 250–263. IEEE Computer Society, 2007.

[10] R. G. Downey and M. R. Fellows. *Parameterized complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[11] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.*, 3(3):1–27, 1999.

[12] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. A. Rosamond, S. Saurabh, and Y. Villanger. Local Search: Is brute-force avoidable? In *IJCAI 2009: Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 486–491, 2009.

[13] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, New York, 2006.

[14] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[15] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979. A Series of Books in the Mathematical Sciences.

[16] C. A. Glass and H. Kellerer. Parallel machine scheduling with job assignment restrictions. *Naval Research Logistics*, 54(3):250–257, 2007.

[17] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM J. Comput.*, 16(1):111–128, 1987.

[18] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1989.

[19] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.*, 4(3):221–225, 1975.

[20] M. M. Halldórsson, R. W. Irving, K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, and S. Scott. Approximability results for stable marriage problems with ties. *Theor. Comput. Sci.*, 306(1-3):431–447, 2003.

[21] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[22] K. ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *FOCS 2009: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 639–648, 2009.

[23] R. W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[24] R. W. Irving and D. F. Manlove. Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems. *J. Comb. Optim.*, 16(3):279–292, 2008.

[25] K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. In *Automata, Languages and Programming (Prague, 1999)*, volume 1644 of *Lecture Notes in Computer Science*, pages 443–452. Springer, Berlin, 1999.

[26] S. Khuller, R. Bhatia, and R. Pless. On local search and placement of meters in networks. *SIAM J. Comput.*, 32(2):470–487, 2003.

[27] Z. Király. Better and simpler approximation algorithms for the stable marriage problem. In *ESA '08: Proceedings of the 16th annual European symposium on Algorithms*, pages 623–634, Berlin, Heidelberg, 2008. Springer-Verlag.

[28] A. Krokhin and D. Marx. On the hardness of losing weight. In *ICALP 2008: 35rd International Colloquium on Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 662–673, Berlin, 2008. Springer.

[29] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[30] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980.

[31] G. S. Lueker and K. S. Booth. A linear time algorithm for deciding interval graph isomorphism. *J. ACM*, 26(2):183–195, 1979.

[32] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theor. Comput. Sci.*, 276(1-2):261–279, 2002.

[33] D. Marx. Local search. Parameterized Complexity News, pages 7–8, volume 3, 2008.

[34] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

[35] D. Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

[36] D. W. Matula. Subtree isomorphism in $o(n^{5/2})$. *Ann. Discrete Math.*, 2:91–106, 1978.

[37] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

[38] L. Perkovic and B. A. Reed. An improved algorithm for finding tree decompositions of small width. *Int. J. Found. Comput. Sci.*, 11(3):365–371, 2000.

[39] P. A. Robards. *Applying Two-Sided Matching Processes to the United States Navy Enlisted Assignment Process*. PhD thesis, Naval Postgraduate School Monterey CA, 2001.

[40] N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *J. Comb. Theory, Ser. B*, 41(1):92–114, 1986.

[41] N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995.

[42] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004.

[43] N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994.

[44] E. Ronn. NP-complete stable matching problems. *J. Algorithms*, 11(2):285–304, 1990.

[45] A. E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[46] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, Cambridge, MA, 1990.

[47] S. Szeider. The parameterized complexity of k-flip local search for SAT and MAX SAT. In *SAT 2009: Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing*, volume 5584 of *Lecture Notes in Computer Science*, pages 276–283. Springer, 2009.

Publications

Publication connected to Thesis 1:

[MS07] D. Marx and I. Schlotter: Obtaining a planar graph by vertex deletion. *In Proc. of 33nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2007)*, volume 4769 of *Lecture Notes in Computer Science*, pages 292–303, Berlin, Springer, 2007.

Publications connected to Thesis 2:

[MS09a] D. Marx and I. Schlotter: Parameterized graph cleaning problems.
Discrete Applied Mathematics, 157(15):3258–3267, 2009.

[MS09b] D. Marx and I. Schlotter: Cleaning interval graphs.
Manuscript, 2009.

[MS08] D. Marx and I. Schlotter: Parameterized graph cleaning problems.
In Proc. of 34nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008), volume 5344 of *Lecture Notes in Computer Science*, pages 287–299, Berlin, Springer, 2008.

Publication connected to Thesis 3:

[MS09c] D. Marx and I. Schlotter: Parameterized complexity and local search approaches for the stable marriage problem with ties.
To appear in *Algorithmica*. Published online on 23 May 2009.

Publication connected to Thesis 4:

[MS09d] D. Marx and I. Schlotter. Stable assignment with couples: parameterized complexity and local search.
In Proc. of 4th International Workshop on Parameterized and Exact Computation (IWPEC 2009), Copenhagen, Denmark, volume 5917 of *Lecture Notes in Computer Science*, pages 300–311, Springer-Verlag Berlin Heidelberg, 2009.

Egyéb publikációk:

[MS10a] K. Jansen, S. Kratsch, D. Marx and I. Schlotter: Bin packing with fixed number of bins revisited.
Accepted to *12th Scandinavian Symposium and Workshops on Algorithm Theory, Bergen, Norway (SWAT 2010).*

[MS10b] D. Marx and I. Schlotter: Parameterized complexity of the arc-preserving subsequence problem.
Manuscript, 2010, submitted.

[TRS06] G. Tassi, P. Rózsa and I. Schlotter: Matrix analysis of V- or Y-supported continuous bridge girders.
BME ÉŐK Hidak és Szerkezetek Tanszéke Tudományos Közleményei, Budapest, pages 181–192, 2006.

[SGL05] Schlotter I., Gáspár Cs., and Lukács. A: Internetes tartalmak minősítése a felhasználók modellezésével. *Híradástechnika*, 5:13–19, 2005.

[S04] Schlotter I: Hírportálok rovatainak modell alapú minősítése.
Tudományos Diákköri dolgozat, BME VIK, Távközlési és Médiainformatikai Tanszék, 2004.

Citations

[MS09a]

- R. van Bevern, H. Moser, R. Niedermeier: Kernelization through tidying—A case study based on s-plex cluster vertex deletion
  In Proc. of the 9th Latin American Theoretical Informatics Symposium (LATIN'10), volume 6034 in *Lecture Notes in Computer Science*, pages 528–539, Springer, Berlin, 2010.

[MS09c]

- ○ A. Krokhin, D. Marx: On the hardness of losing weight
  To appear in *ACM Transactions on Algorithms*.

[MS09d]

- P. Biró, E. J. McDermid: Matching with sizes (or scheduling with processing set restrictions)
  Technical Report TR-2010-307, University of Glasgow, Department of Computing Science, 2010.
- ○ A. Krokhin, D. Marx: On the hardness of losing weight
  To appear in *ACM Transactions on Algorithms*.

[MS07]

- I. Adler, M. Grohe, S. Kreutzer: Computing excluded minors
  In: *SODA 2008: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 641–650, 2008.
- M. R. Fellows, J. Guo, H. Moser, R. Niedermeier: A complexity dichotomy for finding disjoint solutions of vertex deletion problems
  In Proc. of 34th International Symposium on Mathematical Foundations of Computer Science (MFCS 2009), volume 5734 of *Lecture Notes in Computer Science*, pages 319–330, Berlin, Springer, 2009.
- P. Heggernes, F. Mancini: Dynamically maintaining split graphs
  Discrete Applied Mathematics, 157(9):2057–2069, 2009.
- Ken-ichi Kawarabayashi: Planarity Allowing Few Error Vertices in Linear Time
  In: *50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 639–648, 2009.
- M. Mahajan, V. Raman, S. Sikdar: Parameterizing above or below guaranteed values
  Journal of Computer and System Sciences, 75(2):137–153, 2009.
- F. Mancini: Minimum Fill-In and Treewidth of Split+ke and Split+kv Graphs *In Proc. of 18th International Symposium on Algorithms and Computation (ISAAC 2007)*, volume 4835 of *Lecture Notes in Computer Science*, pages 881–892, Berlin, Springer, 2007.
- F. Mancini: Graph Modification Problems Related to Graph Classes
  PhD Thesis, University of Bergen, 2008.
- G. B. Mertzios, S. D. Nikolopoulos: The lambda-cluster problem on parameterized interval graphs
  RWTH Aachen Technical Report, 2008.
- ○ D. Marx: Parameterized complexity and approximation algorithms
  The Computer Journal, 51(1):60–78, 2008.
- ○ D. Marx: Chordal deletion is fixed-parameter tractable
  To appear in *Algorithmica*.