Efficient Algorithms for Matching Problems Under Preferences with Ties

A Research Proposal for Doctoral Program in the Department of Computing Science, University of Glasgow.

Supervisor - David Manlove

Sofiat Olaosebikan (sofiat@aims.edu.gh)

April 9, 2016

1 Introduction

Many real-world problems can be modelled as bipartite matching problems, which mostly comprise a set of participants, set of offers, and the preference list of participants over offers or vice versa. In this setting, the goal is to find an optimal allocation of participants to offers. However, due to the number of participants that appear in practical scenarios, it is not feasible to make manual allocation or use brute-force techniques; hence the need to design and implement efficient matching algorithms. Matching students to courses/projects, applicants to job posts and medical residents to hospitals are some example applications.

Problems arising in this setting could be one-sided or two-sided. For instance, suppose a University department want to allocate students to projects based on their (the students) ranked preference, with a bound on the number of students that can be assigned to a project. This scenario is an application of many-to-one bipartite matching problems with one-sided preference. Henceforth, we will refer to the problem of allocating students to projects as the Student-Project Allocation Problem (SPA) [1].

In some variants of (SPA), comprising set of students, projects and lecturers, lecturers are also permitted to rank students they are willing to supervise in order of preference, and there is always constraints on the number of students a lecturer can supervise. This reduces to a bipartite matching problem with two-sided preference. A generalization of (SPA) occurs when lecturers (students) cannot provide a strict ranking of all their preferred students (projects), as the case may be. For instance, lecturers might be happier to rank their favourite students, and then group together the remainder at the tail of their list. We shall refer to this instance as the Student-Project Allocation Problem with Ties (SPAT) [1]. Stability of matching in this model, where no (student, lecturer) pair would prefer to be assigned together than remain in their current assignment, is very important.

The (SPA) model and its variants can be generalised to other scenarios different from Student-Project Allocation, for example, allocation of applicants to posts at large organisations with several departments. Although a considerable amount of research has been done over the years to tackle matching problems of this kind, many open problems are yet to be solved. Some of these problems and their important applications in large-scale centralized matching schemes motivate this research proposal. This proposal will briefly discuss existing literature on variants of (SPA), describe my previous work and motivation for research. It will also outline various open problems that have been identified in (SPAT) and similar cases, and discuss a research approach to tackle some of them.

2 Literature review

The problem of allocating students to projects considering preferences and capacity constraints was described as the Student-Project Allocation problem (SPA) in [2]. Here, we discuss existing work that has been done in variants of the (SPA) model.

Given an instance of (SPA) with lecturers preference over students (SPA-S), Manlove *et al.* [2] presented two optimal linear-time algorithms that could be employed to construct a stable matching of students to projects, with respect to the preference information, optimality and capacity constraints. The stable matching produced by the first algorithm is student-oriented (that is, students have the best-possible projects) while the one produced by the second algorithm is lecturer-oriented (that is, lecturers have the best-possible students).

For an instance of (SPA) with lecturers preference over projects (SPA-P), the stable matchings can have different sizes. It was proved in [4] that the problem of finding a maximum cardinality stable matching under this setting is NP-hard, and a 2−approximation algorithm was presented. O'Malley [11] later proved that this problem is approximable within $\frac{3}{2}$ [1] . He presented a linear-time algorithm to construct a strongly stable matching, if one exist, or report that none exist.

A setting where lecturers preference lists admit student-project pair (SPA-(S,P)) is a generalization of (SPA-S) and (SPA-P). Manlove *et al.* [2] suggested the study, and Abu El-Atta and Moussa [12] formally defined the problem [1]. In their paper [12], they extended the student-oriented algorithm for (SPA-S) to the (SPA-(S,P)) setting. They proved that for every instance of (SPA-(S,P)), we can construct a stable matching. In addition, such a matching can be found in $O(m)$ time, where m is the total length of the students' preference lists.

3 Previous work and motivation for research

During the research phase of my Masters program, I reviewed a known theoretical procedure that guarantees the decomposition of the edge set of an arbitrary bipartite graph $G(X, Y, E)$ into three disjoint sets: E_0, the set of edges that appear in no maximum matching; E_w, the set of edges that appear in at least one maximum matching, but not all; and E_1, the set of edges that appear in all maximum matching. In this project, I designed an algorithm that runs in time $O(n + m)$. It improves upon that presented by Costa [18], which runs in time $O(nm)$, where n is the number of vertices and m the number of edges. In addition, I implemented the Hopcroft-Karp algorithm [19] to find a maximum matching in bipartite graphs, and I submitted it on the Python Package Index `https://pypi.python.org/pypi/hopcroftkarp/1.2.4.`. Judging from the number of downloads and feedbacks, I believe many researchers have found it useful.

I recently designed and implemented an algorithm to assign students to courses for presentation at the African Institute for Mathematical Sciences, Ghana. There were 18 available courses, 48 students and each student ranked three of their most preferred courses according to order of preference. The goal was to maximize the number of students matched to their first choice, then second choice, and third choice. Also, to ensure that the presentations were well spread out across the fields, no more than three students can be assigned to one particular course. An instance $\mathcal{I}$ of the problem consist of a set $S = \{s_1, s_2, \dots, s_{48}\}$ of students, set $C = \{c_1, c_2, \dots, c_{18}\}$ of courses, subset of courses $P(s_i) = \{c_{\alpha_1}, c_{\alpha_2}, c_{\alpha_3}\}$ that student s_i finds acceptable, and capacity 3 for each c_k's. The algorithm is described in pseudocode form in Algorithm 1.

After invoking the algorithm on $\mathcal{I}$, the matching obtained allocated $30, 8$ and 6 students to their first, second and third choice respectively, while 4 students were unassigned. These students were offered the available courses to select from. One one hand, the algorithm satisfied the goal of the problem - assign students to their top ranked courses according to how their response was received, and if the course is oversubscribed, assign them their next preferred course. On the other hand, it could be further improved, to produce a student-optimal and maximum matching.

I have a strong background in Mathematics, a keen interest in Computer Programming and I am enthusiastic about research. I am passionate about combining my intellectual ability and skills set to investigate theoretical approaches that will aid the design, analysis and implementation of efficient algorithms which could be applied in practical scenarios to find solutions to real-world matching problems.

Algorithm 1 Algorithm to allocate students to courses

Input: Instance $\mathcal{I}$ of student-course allocation
Output: student-optimal matching M in $\mathcal{I}$

```
M ← ∅
A ← ∅ # students that could not be assigned
for s_i ∈ S do
    j = 1
    while j ≠ 4 do
        if c_{α_j} is under-subscribed then
            M ← M ∪ {(s_i, c_{α_j})}
            break # pick the next student on the list
        else j ← j + 1
        end if
    end while
    if s_i cannot be assigned then
        add s_i to A
    end if
end for
return M
```

4 Formal definition of (SPAT) model

As mentioned earlier, a natural generalization of (SPA) arises when participants are allowed to express ties in their ranked preference list. We refer to this problem as Student-Project Allocation problem with Ties (SPAT). An instance of (SPAT) comprises a set of students $S = \{s_1, s_2, \ldots, s_{n_1}\}$, a set of projects $P = \{p_1, p_2, \ldots, p_{n_2}\}$ and a set of lecturers $L = \{l_1, l_2, \ldots, l_{n_3}\}$, each lecturer l_j having a capacity $d_j \in \mathbb{Z}^+$, indicating the number of students he is willing to supervise, and each project p_k having a capacity $c_k \in \mathbb{Z}^+$ indicating the maximum number of students that could be assigned to it. Each student (lecturer) has a preference list consisting of a subset of the projects (students) he finds acceptable, listed in order of preference.

Ties in the Preference Lists: The students (lecturers) preference lists may contain one or more ties, each consisting of two or more projects (students) of equal preference. When ties appear in preference lists, we can define three levels of stability (as in the classical Hospitals/Residents Problem with Ties (HRT) [5] [6]), namely weak, strong and super-stability.

Weak, Strong and Super-Stability: Let M be a matching of an instance of (SPAT). Given a pair $(s_i, p_k) \in S \times P$ (that is, s_i finds p_k acceptable), we define the following notations:

- $s_i \sim p_k$ implies s_i is either unassigned in M or strictly prefers p_k to $M(s_i)$.
- $p_k \sim s_i$ implies l_j (the lecturer who offers p_k) is either undersubscribed in M or strictly prefers s_i to at least one member of $M(p_k)$.
- $s_i \asymp p_k$ implies s_i either is unassigned in M or strictly prefers p_k to $M(s_i)$ or is indifferent between them.
- $p_k \asymp s_i$ implies l_j is either undersubscribed in M or strictly prefers s_i to at least one member of $M(p_k)$ or is indifferent between them.

s_1: $(p_1\ p_2)$	l_1 offers p_1 and p_2 : $(s_1\ s_2)$
s_2: $(p_1\ p_3)$	l_2 offers p_3 and p_4 : $s_3\ (s_2\ s_4)$
s_3: p_3	
s_4: $(p_3\ p_4)$	

Table 1: An instance of (SPAT) with $d_j = 2$ and $c_k = 1$.

M is *weakly stable* if there is no blocking pair $(s_i, p_k) \in S \times P \setminus M$, such that $s_i \sim p_k$ and $p_k \sim s_i$. M is *super-stable* if there is no blocking pair (s_i, p_k), such that $s_i \asymp p_k$ and $p_k \asymp s_i$. M is *strongly stable* if there is no blocking pair (s_i, p_k), such that either (i) $s_i \sim p_k$ and $p_k \asymp s_i$; or (ii) $s_i \asymp p_k$ and $p_k \sim s_i$. For an instance of SPAT described in Table 1, matchings $M_1 = \{(s_1, p_1), (s_2, p_3), (s_4, p_4)\}$ and $M_2 = \{(s_1, p_1), (s_4, p_3)\}$ are weakly stable because they admit a blocking pair (s_3, p_3). $M_3 = \{(s_1, p_1), (s_3, p_3), (s_4, p_4)\}$ is a super-stable matching because (s_2, p_1) is a blocking pair. Clearly a super-stable matching is strongly stable and a strongly stable matching is weakly stable.

5 Proposed research problems

We discuss several interesting problems we may investigate in order of priority, .

1. Attempt to extend the algorithmic procedure described in [1] for obtaining a super-stable matching in (HRT) to the (SPAT) setting. Construct new efficient algorithms for finding stable matching in variants of (SPAT), under the strong stability and super-stability, or report that none exist. Implement the (SPAT) stability-based algorithms, and empirically evaluate the properties of and relationship between the stable matchings produced, based on their size (cardinality) and profile (number of participants matched with their first choice, second choice and so on). Using real data and randomly generated data, compare the results obtained from applying profile-based optimality criteria. Investigate how the nature of the preference list (including length, size and position of the ties) affect the stable matching produced.

2. For a given instance of (SPAT), it is clear that by breaking preference ties arbitrarily, we can achieve stable matching of different sizes, under the weak stability criteria [7]. However, the problem of achieving a maximum matching in this setting is NP-hard. This has been proved for a similar case (the Stable Marriage Problem with Incomplete Lists) in [17]. Attempt to describe approximation algorithms for this case, with a performance guarantee better than $\frac{3}{2}$. Consider special cases in which ties in preference lists are restricted to participants on one side, and there is at most one tie per list (each tie could probably be of length 2). Construct efficient algorithms to find all weakly stable matchings in (SPAT).

3. We note that coping with the hardness of finding maximum weakly stable matching in (SPAT) is not restricted to describing approximation algorithms only. Another common technique for solving matching problems is Integer Programming (IP). Attempt to generate IP models for variants of (SPAT). Consider some special cases, say, with restrictions on the positions or size of ties. Investigate how the model can be applied to extensions of (SPAT) with different nature of preference lists, including incomplete lists and partial orders.

4. After the size and profile of the stable matching for an instance of (SPAT) has been considered, it is also equally important to achieve a load-balancing matching (that is, ensure the workload of supervising students is evenly spread out across the lecturing staff). One way to achieve this with respect to lecturers is to introduce lower quotas. A lower quota on lecturer l_j is the minimum number of students that must be assigned to l_j in any feasible solution. In an instance of (SPAT), is a feasible solution guaranteed? Can we construct new efficient algorithms for finding stable matching in this setting, if one exists. Investigate other existing ways and explore new ways of achieving load-balancing that will result in a feasible solution.

5. An instance I of (SPA) can also be extended to admit lower quotas on the projects, that is, project p_k cannot run unless at least a particular number of students are assigned to it. Attempting an open problem in [1], construct a polynomial-time algorithm for finding a stable matching in I or report that none exist, where no project can be closed. Can we extend this to the (SPAT) model?

6. Considering an instance of (SPA) that restricts lecturers preference and admit ties in students preference list. Stability in this setting is no longer relevant, optimality properties like Pareto optimality (a matching to which there is no alternative matching where some participants are better off and no participant worse off), popularity (a matching M is popular if there is no other matching M^* such that majority of the participants prefers M^* to M than M to M^*), profile (a vector, whose $i-th$ component is the number of students with their $i-th$ choice) and size of the matching become important factors to consider. Investigate existing algorithms to solve variants of this problem and construct improved efficient algorithm that finds the best optimal matching, based on these optimality criteria. Attempt other open problems in this setting.

7. Attempt to define the problem of ties/incomplete lists appearing in an instance of (SPA) with lecturers preference over (student-project) pairs (SPA-(S,P)). Investigate if it admits a stable matching, under the three levels of stability mentioned earlier, and other open problems in the (SPA-(S,P)) context.

6 Research approach and milestones

I intend to devote a large part of the research period to theoretical investigation of the proposed problems in the (SPA) and (SPAT). I plan to approach Problem 1 by analysing the resident-oriented and hospital-oriented algorithm presented in [1] and [16] respectively for a given instance of (HRT) to construct a super-stable matching, or report that none exist. The idea is to extend the resident-oriented and hospital-oriented algorithms to the student-oriented and lecturer-oriented algorithms respectively, in a given instance I of (SPAT) under the super-stability criteria. In [13], Kavitha *et al.* presented an efficient algorithm for finding a strongly stable matching, if one exist, for an instance of the Stable Marriage Problem with Ties and Incomplete Lists (SMTI), solvable in time $O(nm)$. They also extended their algorithm to the (HRT) context. I intend to explore these algorithms, investigate if it can be further improved in both context, and finally extend it to the (SPAT) context. These ideas will be explored with the view of determining if a further modification of existing approaches or a completely new approach will yield improved efficient algorithms for variants of (SPAT).

As discussed earlier, matchings $M_1 = \{(s_1, p_1), (s_2, p_3), (s_4, p_4)\}$ and $M_2 = \{(s_1, p_1), (s_4, p_3)\}$ for an instance of (SPAT) described in Table 1 are weakly stable and of different sizes. The problem of finding a maximum cardinality matching, as mentioned in Problem 2, is NP-hard. Hence, the need to describe approximation algorithms with a good performance guarantee. I intend to approach this problem by analysing the approximation algorithms Király [15] and Paluch [14] independently derived with a performance guarantee of $\frac{3}{2}$ to find a weakly stable matching of maximum size for (SMTI) and (HRT) (with ties occurring on both sides) [1]. It would also be worth investigating if there is an approximation algorithm with a performance guarantee better than $\frac{3}{2}$, for finding a weakly stable maximum matching in both context. This will provide a great insight on how these approximation algorithms could be extended to the (SPAT) setting. If not, knowledge acquired from analysing them would be useful in describing new approximation algorithms with a performance guarantee less or equal to $\frac{3}{2}$, for finding a weakly maximum matchings in (SPAT).

Finally, I intend to investigate if there is a reasonable algorithm to find all weakly matchings in (SPAT). An approach could be to construct two bipartite graphs, one with set of lecturers and students as vertices, and the edges corresponding to whether a lecturer prefers a student. The other with set of students and projects as vertices, and the edge corresponding to whether a student prefers a project. After which I will invoke the procedure of edge decomposition using the Dulmage and Mendelsohn decomposition from my masters research on these graphs. It would be worth exploring if this approach will yield new and improved results and algorithms.

In view of the amount of support and expertise available within the department, I intend to implement some of these new efficient algorithms and empirically evaluate them using real data where available or randomly-generated data, by varying parameters such as the number of participants, the capacity constraints, the length of the preference lists, the number, position and sizes of the ties. I also intend to test their correctness, performance and scalability in terms of time and space. In addition, I will carry out the development and evaluation of user interfaces for the algorithmic implementation to ensure the solutions provide maximum benefits to the target users. I will review existing usability measurement frameworks, find one that is appropriate, and analyse the usability of these developed interfaces, determining how well they satisfy user needs.

I hope to be able to carry out this research work within three years of diligent work. Below is a proposed time line.

Academic Year	**Period**	**Task to achieve**
First	First half	General reading. Attending relevant classes, especially in the area of algorithms and computational complexity.
	Second Half	1. Extension of super-stable matching algorithm for (HRT) to (SPAT). 2. Efficient algorithm to find strongly stable matching for an instance of (SPAT), should one exist. 3. $c-$approximation algorithms for finding maximum weakly stable matching in (SPAT) and similar context, where $c \leq \frac{3}{2}$.

Second	First quarter	Efficient algorithm to find all weakly stable matchings in (SPAT).
	Second quarter	Integer programming formulation for variants of (SPAT).
	Third quarter	(SPA) approximation algorithms with lower quotas on lecturers and/or projects in the general case and when ties are allowed in the preference lists.
	Fourth quarter	1. Improved efficient algorithms for an instance of (SPA) that only permit ties in the students preference lists. 2. Efficient algorithms for finding stable matchings in the (SPA-(S,P)) context, under the three stability criteria.
Third	First Half	1. Implementation and evaluation of some of the algorithms developed. 2. Modification of existing user interface and empirical study of the implemented algorithms.
	Second half	Writing up Ph.D. Thesis.

7 Related Work

As stated in [17], Ronn [[9] [10]] was possibly the first to study stable matching problems with ties in the preference lists from an algorithmic point of view. In addition to other things, he proved that the Stable Roommates problem (the non-bipartite extension of Stable Marriage) becomes NP-complete when ties are permitted.

The classical Hospitals/Residents Problem (HR) is similar to (SPA). The two models are many-to-one matching problems in which participants in both sets are permitted to have preference lists. A common application of (HR) is the matching of medical residents to hospitals where each resident ranks a set of hospitals and each hospital ranks a set of residents in order of preference. Generalisations of the HR problem may arise depending on the application. Two such examples include the Hospital/Residents Problem with Ties (HRT), where elements in preference lists may have equal rank, and the Hospital/Residents Problem with Couples (HRC) where couples wish to be matched to hospitals generally close to each other [8].

The House Allocation Problem (HA) is also similar to the variant of (SPA) where preference list is restricted to one set of participants. As discussed in one of the open problems, stability is no longer relevant in this context as it requires participants in both sets to have preferences. Optimality properties like Pareto optimality, popularity, profile and cardinality of the matching become important factors to consider. These factors introduce variants of the HA problem with different algorithms being developed to solve them. In [3], an $O(\sqrt{n}m)$ algorithm is described to find a maximum cardinality Pareto optimal matching (where n is the number of agents and m is the total length of the preference lists). Applications of the (HA) include campus housing allocation, matching reviewers to conference paper submissions, allocating professors to offices, clients to servers, and so on.

8 Conclusion

Modifications to the input and required output criteria in existing applications of matching problems has led to continuous research in the area. New applications have also been identified in various context which have left matching problems an interesting area of research. From a theoretical and algorithmic perspective, my research will aim to tackle some open problems that has been identified in variants of (SPAT), perhaps some other matching problems involving ties and/or incomplete lists. New efficient algorithms for finding stable matchings in this setting based on various optimality criteria still need to be developed, and we will aim to construct them. Where it is known that achieving a maximum stable matching is NP-hard, we will aim to describe approximation algorithms with improved performance. We will also implement these algorithms and test their correctness and performance with real and randomly generated data. Developing or modifying existing user interfaces for these algorithms and optimising them to maximize usability will improve on the practical usefulness of the developed algorithms in large-scale matching schemes where variants of this model appears. To sum up, results from my research will present several additions to knowledge, advancing theoretical understanding and applications of matching problems.

References

[1] D. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

[2] D. F. Manlove, D. J. Abraham and R. Irving. Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms*, 5 (1):73 – 90, 2007.

[3] D. J. Abraham, K. Cechlárová, D. F. Manlove and K. Mehlhorn. Pareto Optimality in House Allocation Problems. In: ISAAC 2004, LNCS. vol. 3341, pp. 3–15, 2004.

[4] D. F. Manlove and G. O'Malley. Student-project allocation with preferences over projects. *Journal of Discrete Algorithms*, 6:553 – 560, 2008.

[5] R.W. Irving, D.F. Manlove and S. Scott. The hospitals/residents problem with ties. In: *Proceedings of SWAT 2000: The 7th Scandinavian Workshop on Algorithm Theory, in: Lecture Notes in Computer Science*, vol. 1851, Springer-Verlag, 2000, pp. 259 – 271.

[6] R.W. Irving, D.F. Manlove, S. Scott. Strong stability in the hospitals/residents problem. In: *Proceedings of STACS 2003: The 20th Annual Symposium on Theoretical Aspects of Computer Science, in: Lecture Notes in Computer Science*, vol. 2607, Springer-Verlag, 2003, pp. 439 – 450

[7] R.W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261 – 272, 1994.

[8] E. J. McDermid and D. F. Manlove. Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization*, 19:3, pp 279 – 303. 2009.

[9] E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11:285 – 304, 1990.

[10] E. Ronn. On the complexity of stable matchings with and without ties. PhD thesis, Yale University, 1986.

[11] G. O'Malley. Algorithmic Aspects of Stable Matching Problems. PhD thesis, University of Glasgow, Department of Computing Science, 2007.

[12] A.H Abu El-Atta and M.I Moussa. Student project allocation with preference lists over (student,project) pairs. In *Proceedings of ICCEE 09: the 2nd International Conference on Computer and Electrical Engineering (IEEE)*, vol. 94, pages 375 – 379, 2009.

[13] T. Kavitha, K. Mehlhorn, D. Michail and K. Paluch. Strongly stable matchings in time $O(nm)$ and extension to the Hospitals-Residents problem. In *ACM Transactions on Algorithms*, **3**, 2, article number 15, 2007.

[14] Paluch, K. Faster and simpler approximation of stable matchings, in *Proceedings of WAOA '11: 9th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science*, Vol. 7164 (Springer), pp. 176 – 187, 2012

[15] Z. Király. Better and simpler approximation algorithms for the stable marriage problem *Algorithmica* 60, pp. 3 – 20, 2011.

[16] R.W. Irving, D.F. Manlove and G. O'Malley. Stable marriage with ties and bounded length preference lists, *Journal of Discrete Algorithms* 7, 2, pp. 213 – 219, 2009.

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

[18] M.C. Costa. Persistency in maximum cardinality bipartite matchings. *Operations Research Letters*, 15:143 – 149, 1994.

[19] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2:146 – 160, 1972.