Short Research Article

Implementation of Stable Marriage Algorithm in Student Project Allocation

Abstract

Project allocation is an annual challenge for lecturers and students. The process of allocating project involves matching preferences of student over project and with of staff over the student, and is thus an instance of stable marriage problem from theoretical computer science aspect. The aim is to find a stable allocation of project to students, such that it is impossible to find a project swap that would make all involved parties (both students, both staff) happier. This paper investigated efficacy of stable marriage algorithm and deployed basic Gale Sharply algorithm into the process of allocating student project. A system was developed using ruby and MySQL to handle the task. The result showed that the algorithm was able to improve the process by enhancing the stability involved. .

Keywords: stable marriage, preferences, allocation, algorithm and project.

1. Introduction

The allocation of final year student a project is continuous process that attracts a lot of attention at the end of every academic session. The task involves assigning each student a project topic for their research work as part of the requirement of their programme. The projects are proposed either by the student or the lecturer, after which both parties negotiate on the scope of the project. This paper is about deploying basic Gale Sharply stable marriage algorithm in the process of allocating student project. Where each supervisor and student will develop preference list from which project are allocated automatically when the algorithm is run.

1.2 Statement of the research problem

At present, most institution does not have complete resources for managing process of final year student project allocation. The current manual system of allocating project to student by the project coordinators tend to be inefficient as the student can be allocated to supervisor that they do not prefer. Equally, supervisor might not be able to select student that they can work with effectively. Thus in this approach, students or supervisors proposed a project whilst project coordinator handles the allocation process. It is most likely that a student might be allocated to a topic or a supervisor in an area that he/she is not interested in. Similarly, supervisor's proposed topic might be allocated to student who is not capable of undertaking it, thus posing a great challenge in the process.

Comment [DSA1]: Most institutions do not have...

1.3 Brief Overview of Basic Gale Shapley Stable Marriage Algorithm

Matching between two set of elements is a natural phenomenon that is of significant interest to researchers. The most aspect of human nature involves pairing between two set such as man to women, doctors to a hospital, student to a project and so on. And thus, this matching needs to be smooth and stable. The concept of stable marriage was initially studied in (1962) by Gale and Shapley (Gonczarowski et al., 2018). The aim was to solve the problem of matching between equal number of men and women (Teo, Sethuraman and Tan, 2001). The stable marriage problem deals with finding a stable pairing between two equally sized sets of groups, from preference order for each element in the group (Sanfoundry, 2013). The Gale-Shapley algorithm requires each element from one set in the matching to provide a complete set of preference ordered list of other opposite set in the matching. In Gale-Shapley algorithm, no incomplete preference is accepted. Which means both the two set most, be of the same size and are ranked to each other (Iwama and Miyazaki, 2008). Generally, it can be argued that stability is the key aspect that determine the success of each matching, and according to Gale and Shapley (Gale and Shapley, 1962) there always exist at least one stable matching in an instance of the stable marriage algorithm (Lightfoot, 2016).

Comment [DSA2]: Who authored the study in 1962? The same authors?

Sanfoundry (2013) argued that the Gale Shapley algorithm could be implemented programmatically as shown in the figure below:

```
function stableMatching {
    Initialize all m ∈ M and w ∈ W to free
    while ∃ free man m who still has a woman w to propose to {
       w = m's highest ranked such woman to whom he has not yet proposed
       if w is free
         (m, w) become engaged
       else some pair (m', w) already exists
         if w prefers m to m'
           (m, w) become engaged
           m' becomes free
         else
           (m', w) remain engaged
    }
}
```

A Pseudocode of Gale Shapley algorithms (Sanfoundry, 2013).

1.4 Stable marriage problem and student project allocation

Generally, the criteria for allocating projects to students are much similar to the stable marriage pairing. Matching different entities from two set of elements to each other usually invoke the need for stability since individual's shows preferences over one another. Allocating fixed number of student to a fixed number of the project has much in common to the coupling of n men and n women, in terms of the problems that may evolve. To this vein, it is apparent that deploying stable marriage problem and some of its solutions will have a great impact on the process of allocating student a project. During the process of allocating project the main aim for both student and staff is to have a happier working partner, it is argued that the basic Gale and Shapley algorithm terminate with stable set of engaged couple in which each pair is happy with each other and no any possibility for any swap that will result to happier couple than initially formed (W. Irving and Gusfield, 1989).

Comment [DSA3]: Remove

The convention in the process of allocating student project was, student always making a request to the supervisors project and supervisor response to the request with an offer. This is exactly in line to the idea of basic Gale-Shapley algorithm which involves sequential proposal from men part to the women (Gale and Shapley, 1962). However, some extension of Gale-Shapley algorithm has the view that a woman can make a request to man and can accept two or more men with the same rank (Tetsuo, Toshinori and Michio, 1999). Never the less, it can be said that stable marriage problem and its solutions, is still feasible to project allocation problem.

Moreover, allowing individual's (both student and staff) to create preferences, in the process of the allocating project, is vital for the performance of the student during the research. Stable marriage problem is strictly based on order list of preferences for the two parties involved. It is argued that matching is always stable between two set if it's resulted from their preferences to each other (F. Manlove and O'Malley, 2008).

So, therefore, it is said that stable marriage problem have much in common to the process of allocating project and the algorithm will provide the best solution to this process. Finally, it can be evident that Basic Gale-Shapley algorithm is applicable to the process of allocating project

2. Methodology

The method adopted in this paper was to design an allocation algorithm based on the criteria and the requirement of Gale Sharply algorithm. We start with creating student and supervisor's preferences then design the algorithm. The final system was developed using ruby programming language and MySQL as the database and local server for implementation.

3. System design

3.1 Student preferences design

This involves allowing the student to enter their preferences to projects they are interested. To achieve this goal, it is also necessary to consider the requirement of the stable marriage algorithm that was deployed in the design of this system. The algorithm requires that each student in the system should rank each project available in a strictly ordered way (Wan-Hong, 2017). This implies a student preferences list is required to include all available project ranked in a decreasing order of importance. So that, first project in their list is preferable than the subsequent one in that order (El-Atta and Moussa, 2009).

3.2 Supervisor preferences design

From reviewed existing system, it is understood that, during the period of project allocation, the project coordinator or admin allocate a number of the project to be proposed and supervised by each staff. Supervisors also show interest and need to create a preference of the student requesting to take their project. This resulted in staff making preference list of student willing to offer their project.

The design of supervisor's preferences list also fulfils the requirement of stable marriage algorithm, as the second entity in the matching. The algorithm requires that all the other entities (students) must be ranked by each supervisor (project).

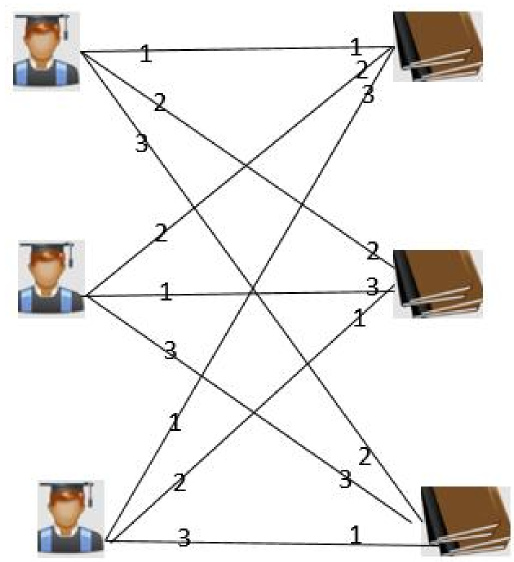


Fig.1 student and supervisors (project) preferences

3.3 Student-project allocation algorithm

The student project allocation algorithm was designed based on the basic Gale-Shapley stable marriage algorithm and some other related stable marriage problems derived from the review of other extensions of the Gale-Shapley algorithm. Some of the extension from the basic algorithm requires that no complete ranking of both partners is needed (Fleiner, 2014).

The final system was developed with little extension of the basic Gale algorithm. The system is implemented such that it automate the ranking process when the number of involved parties grow larger. When the student rank some number out of the large number of project or supervisor rank some students out of the list, the system upon the execution of the algorithm automate the ranking and add to each user preference.

Pseudocode for student-project allocation algorithm

Begin:

```
Initialization:
     Each student=nil project
     Each project= nil student
     While some student S is unmatched from student list
          (Students making request to   projects)
          P= 1st project in S preference list not requested
          S = P for each s and p (s and p could be set of
          students and projects respectively)
```

```
          Add pair S-P to the matching set
               (Project making decision of acceptance)
          If P is matched to two or more S || P prefer S' than
     S from s
               Remove pair S-P from matching set and add to
          the Unmatched set
               Match S'-P add the pair to the matching set
               Add S to the unmatched set
          End if
               Iterate the loop again
     End while
Return the matched set
Stop.
```

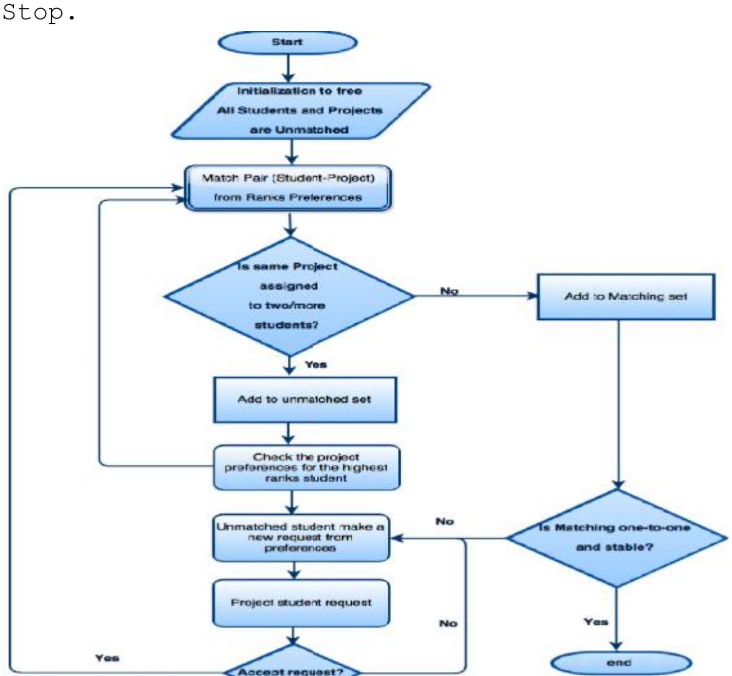


Fig.2 Flowchart for student-project allocation algorithm

Flowchart for student-project matching algorithm

A flowchart represents pictorially the step by step follow in the execution of an algorithm (Aler, 2010). Figure 5 shows the steps of execution of student project algorithm, the flowchart start by initializing both student and project to be unpaired. The next step involves pairing. For a pairing to be successful and added to the matching set, it must satisfy the condition which checks no single project allocated to two students. If the condition failed, the pair is added to the unmatched set. Then unmatched student makes a new request from the unassigned project. If the request is accepted the set are paired and added to the matching set. The cycle continues until all pairing is stable and one-to-one before the algorithm terminate.

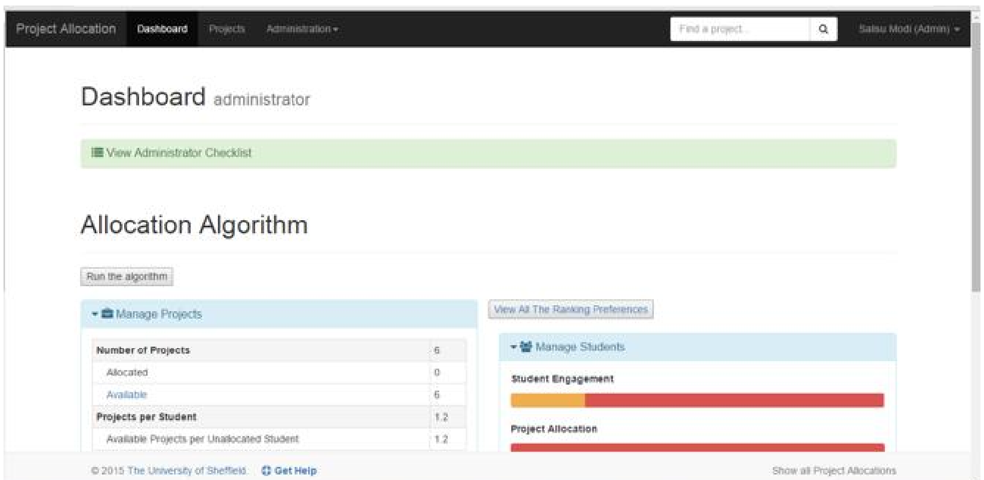
Comment [DSA4]: Which Figure 5 are you referring to?

3.4 Input for student project allocation algorithm

The students in the system individually create their preferences, from the available project of their interest. Similarly, the staff creates their rank preferences from the student in the system. Student allocation algorithm requires those overall preferences as input, in a certain constraint order. This requires n number of students and n number projects to be ranked to each other.

The developed system consists of three (3) dashboard: the student, project supervisor and administrator. The student login into the system to submit project topic, create ranking preferences, and to receive update about the allocation. The supervisor login to submit a propose topic and also create preference of the student. The administrator manage the allocation process as well as run the allocation algorithm.

The final system was implemented and the admin dashboard is shown below:



Fiq.3 admin dashboard of the implemented system

4. Result

To test the feasibility of the algorithm in the allocation process, a system is being developed for the allocation with the algorithm implemented in it. The system provide an interface for the student to enter their preferences to the available projects and supervisor (project) to the available students. The algorithm take as input the two preference list and allocate each student to his/her most appropriate project from the perspective of both ranking. The system is tested with the data and result below:

Example:

Sample students (University username)

Acp14jlr, acp14sh, acp14msa, acp14xw, and acp14hat

Sample project topics

Listening to Sheffield (LTS), decision support system (DSS), student placement portal (SPP), privacy of information (PI), and project allocation system (PAS).

Ranking Preferences

Each student rank the available project from highest to the lowest left to right. Likewise, the supervisor (project) rank the students from highest to the lowest in same order in the table below:

Table 1: students and supervisors ranking preferences.

Student preferences	**Supervisor (project) preferences**
Acp14sh =>DSS, SPP, PI, LTS, PAS Acp14msa=>SPP, PAS, LTS, PI, DSS Acp14xw=> PI, SPP, PAS, DSS, LTS Acp14hat=> PI, PAS, LTS, SPP, DSS	PAS=>acp14jlr, acp14hat, acp14sh, acp14msa, acp14xw DSS=>acp14msa, acp14sh, acp14hat, acp14jlr, acp14xw PI=>acp14msa, acp14xw, acp14hat, acp14sh, acp14jlr LTS=>acp14xw, acp14msa, acp14jlr, acp14sh, acp14hat

This page shows the sample ranking from the implemented system. The student created a rank preferences of the available project.

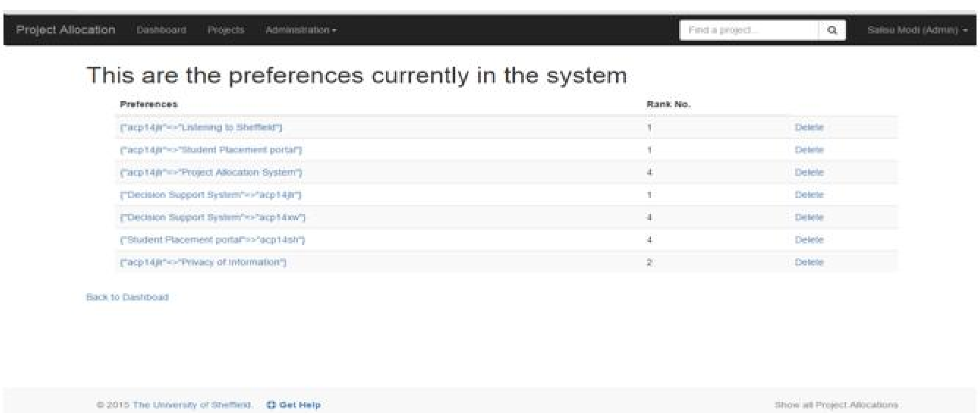


Fig.4 sample ranking from the system

The case where the number of student or the project grow large, and the student or the supervisor could not rank all the other partner. The system implement a function which automate the ranking of unranked partner.

The result after running the system with the above data is shown below:

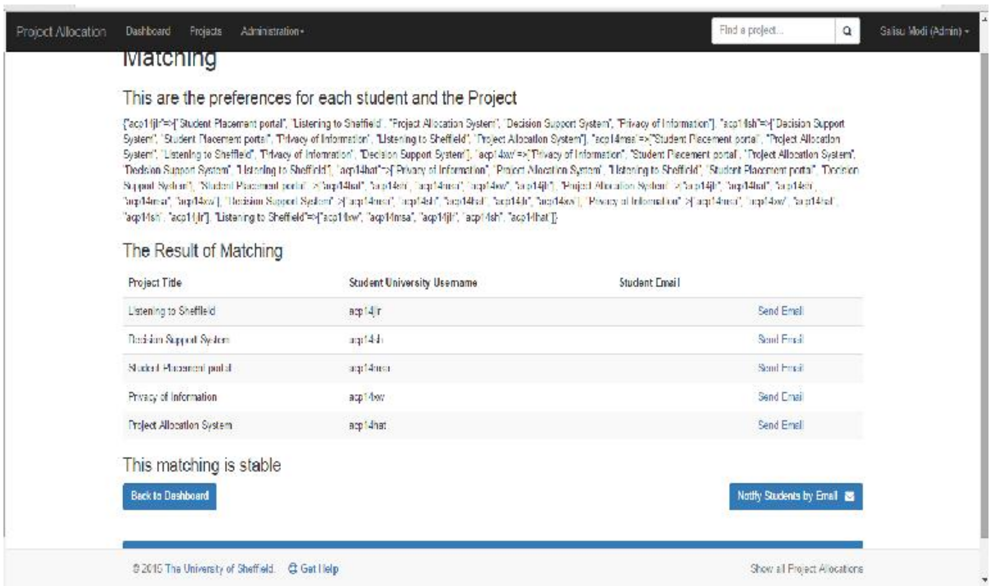


Fig. 5 sample test result from implemented system

From the above result of matching, student acp14jlr was allocated to LTS project which happened to be his/her second choice. The student cannot get his /her first choice because the student was rank fourth by the supervisor of the project. And LTS supervisor cannot get his/her first choice student (acp14xw) because was ranked the last by the student. In thus order, the algorithm makes all the remaining allocation.

At the end of matching the student to project from the preferences from both sets, a set containing each student with allocated partner was returned. It was argued that matching entities from two set of the element with preferences from both set always resulted to individuals in the set been paired with one another (F. Manlove and O'Malley, 2008).

The result of running student project allocation algorithm, student and project instances are returned. Those instances have a number of properties which include allocated partners. Each student has a partner (project) assigned to him/her. This project was at least the first or at most the last project from the preference list of the student, depending on the rank position the student was in the preferences order of the project.

From the result of running the algorithm as applied to some number of student and project, it can be concluded that, no swap between any pair will result to happier matching than the initial one since all pairing resulted from the preferences that the student or the project created and accepted before the pairing (Aderanti et all, 2016). The final result of student project allocation algorithm returns a matching set with each student in the system allocated to one project.

Comment [DSA5]: See comment above

Comment [DSA6]: et al.

5. Performance of the algorithm

The strength this algorithm is in its ability to provide stability in matching n equal number of involved parties (Moussa and Abu El-Atta, 2011). Csaba (2018) argued that in any matching that is two-sided and both the involved parties need to rank each other, then the solution that handled the matching, the stability is the main requirement for such solution. The Gale Shapley algorithm all ways result to a matching set when two parties are matched from result to a matching set when two parties are matched from their preferences (Deng, Panigrahi and Waggoner, 2017)

6. Time complexity and correctness of the algorithm

The whole idea in Gale Shapley algorithm is matching n number of one party with n number of other party (where n is any counting number). In this implementation, n number of student is matched with n number of project from their preferences. The process involved iterating through all free students while there is still unallocated student. Each free student goes to all project topic in his preference list in an orderly manner. And for each supervisor (project) student goes to, he checks if the project is unallocated, if yes the allocation (student-project) is performed. If the project is allocated to someone else, then the project (supervisor) chooses either to remain allocated (reject the current student request) or dumps the current allocation (reject the current student request) from preference list of the project. The process continues iteratively until no more unallocated student. Intuitively this algorithm involved matching N X N items iteratively, and hence the time complexity Gale Shapley algorithm is O (n^2).

The correctness of this algorithm can be viewed from two perspective; stability and perfection. In the stability aspect, Gale Shapley claim that at the end of matching no swap between pairs that will make a happier match than the initial matching will be possible. And the perfection of the algorithm is such that all the member of two parties involved most be matched to a partner (Chiarandini, Fagerberg and Gualandi, 2017).

7. Conclusion

The goal has been to investigate the different concept of stable marriage problem algorithm and how they can be deployed in student project allocation process. This goal has been achieved as we have succeeded in developing a system using an algorithm base on Gale Sharply algorithm that is capable of handling project allocation.

The system was tested with some sample data of students and supervisors. The algorithm was supplied with input (students and supervisors preferences) and the output was produced by running the algorithm as shown in previous sections.

Therefore, it can be deducted from this research that, stability in allocating student project will result to a quality of the research since student are allocated project from their preferences. We also show that deploying stable marriage algorithm in student project allocation could help in reducing the annual challenges experience during this process.

Further research can be conducted to extend the allocation algorithmm to enable monitoring the number of project that each supervisor could be assign to avoid overloading, since problem arise when the number of students who have chosen the supervisor as their first choice exceeds the supervisor's supervision capacity (Salami and Mamman, 2016).

References

Aderanti F. A., Amosa R.T., and Olawatobiloba A. A. (2016). Development of student project allocation system using matching algorithm. *International conference of science, engineering & environmental technology. 1(22),* 153-160.

Azevedo, E. (2016). A Supply and Demand Framework for Two-Sided Matching Markets. *Journal of political economy*, [online] 124(5). Available at: https://www.journals.uchicago.edu/doi/pdfplus/10.1086/687476 [Accessed 7 Aug. 2018].

Chiarandini, M., Fagerberg, R. and Gualandi, S. (2017). Handling preferences in student-project allocation. *Annals of Operations Research*.

Deng, Y., Panigrahi, D. and Waggoner, B. (2017). The complexity of stable matchings under substitutable preferences. *Association for the advancement of artificial intelligence*. [online] Available at: http://www.aaai.org [Accessed 1 Aug. 2018].

El-Atta, A. and Moussa, M. (2009). Student Project Allocation with Preference Lists over (Student, Project) Pairs. *2009 Second International Conference on Computer and Electrical Engineering*.

F. Manlove, D. and O'Malley, G. (2008). Student-Project Allocation with preferences over Projects. *Journal of Discrete Algorithms*, [online] 6(4), pp.553–560. Available at: http://www.sciencedirect.com/science/article/pii/S1570866708000476 [Accessed 11 Aug. 2015].

Fleiner, T. (2014). On Stable Matchings and Flows. *Algorithms*, 7(1), pp.1-14.

Gonczarowski, Y., Nisan, N., Ostrovsky, R. and Rosenbaum, W. (2018). *A Stable Marriage Requires Communication*. [ebook] Available at: https://arxiv.org/pdf/1405.7709.pdf [Accessed 5 Aug. 2018].

Iwama, K. and Miyazaki, S. (2008). A Survey of the Stable Marriage Problem and Its Variants. *International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*.

Lightfoot, S. (2016). *Solving the student project allocation problem*. MSc Dissertation. The University of Manchester.

Moussa, M. and Abu El-Atta, A. (2011). A Visual implementation of student project allocation. *International Journal of computer science and engineering*, vol.3(1793-8201).

Salami, H. and Mamman, E. (2016). A Genetic Algorithm for Allocating Project Supervisors to Students. *International Journal of Intelligent Systems and Applications*, 8(10), pp.51-59.

Sanfoundry, (2013). *Java Program to Implement Gale Shapley Algorithm - Sanfoundry*. [Online] Available at: http://www.sanfoundry.com/java-program-gale-shapley-algorithm [Accessed 10 Aug. 2015].

Comment [DSA7]: Remove. Check for accuracy for all the references. Use a uniform referencing style

Teo, C., Sethuraman, J. and Tan, W. (2001). Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications. *Management science*, [online] 47(9), pp.1252-1267. Available at: http://dx.doi.org/10.1287/mnsc.47.9.1252.9784 [Accessed 10 Aug. 2015].

Tetsuo, H., Toshinori, Y. and Michio, K. (1999). New fast iteration algorithm for the solution of generalized stable marriage problem. *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference*, 6, pp.1051 - 1056.

Wan-Hong, I. (2017). *A web-based project allocation system.* MSc project. The University of Hong Kong.

W. Irving R, Gusfield D. Amazon.com: The Stable Marriage Problem: Structure and Algorithms (Foundations of Computing) (9780262515528): Dan Gusfield, Robert W. Irving: Books. *Amazoncom*. I989. Available at: http://www.amazon.com/The-Stable-Marriage-Problem-Foundations/dp/0262515520. Accessed June 17, 2015.

Comment [DSA8]: See comment above