Performance-based Stable Matching using Gale-Shapley Algorithm

Conference Paper · January 2018

DOI: 10.4108/eai.23-4-2018.2277597

3 authors:

Elviwani Elviwani

12 PUBLICATIONS **26** CITATIONS

Andysah Putera Utama Siahaan

Universitas Pembangunan Panca Budi

329 PUBLICATIONS **1,789** CITATIONS

Liza Fitriana

AMIK Harapan Medan

2 PUBLICATIONS **6** CITATIONS

Performance-based Stable Matching using Gale-Shapley Algorithm

Elviwani[1], Andysah Putera Utama Siahaan[2], Liza Fitriana[3]
[1]Faculty of Computer Science and Information Technology, Universitas Sumatra Utara, Medan, Indonesia
[2]Faculty of Science and Technology, Universitas Pembangunan Panca Budi, Medan, Indonesia
[3]Faculty of Engineering and Computer Science, Universitas Harapan, Medan, Indonesia

elvialischan@gmail.com, andiesiahaan@gmail.com, lizafitriana@outlook.com

ABSTRACT

Performance is something that will be achieved at the institute. Everyone has different abilities in doing a task. Placement of an office position does not always result in satisfactory performance. Giving an assignment to people who are not qualified in their field will cause problems in the future. This problem arises from the inability of the person to overcome and complete the assignment. Position balancing is a terrible thing facing every institution. Many of the criteria that should be assessed for a person can be put in a specific position. Manual calculations will hinder the accuracy of employee placement in the company. The Gale-Shapley algorithm can solve this problem well by the expected criteria in a particular position. This algorithm determines the position of work based on the desire of the institution and the ability of the worker. The interchange between employees will occur if the weight changes. The optimal person will fill each job position in the field of work. Stable Matching will be implemented, and the institution will have qualified employees.

Keywords: *Gale-Shapley, Stable Matching, Decision Support Systems*

1. INTRODUCTION

Employee position is the determinant of the success of the company. Employees are related to the confidentiality of company information [1]. Every employee has the opportunity to occupy a position based on certain qualifications [2]. However, many companies mistakenly assign their employees to the company. It results in an imbalance of the resulting work. Employee performance appraisal results for the company have an essential role in decision making on various issues. It depends on the type of work and the company's goals. The company will obtain information on the extent to which the work achieved by employees in a particular period. High performing employees are expected to carry out their duties with full responsibility and can use all of their potentials. A qualified employee can make a positive contribution to the overall performance of the company. Achieving success work is by improving workability. Employee's ability to handle any job problems so that the job task can be completed on time. The ability of both knowledge and skills is an essential element to improve performance. Professional skills are needed to achieve success. Each employee has several capabilities than can be represented in specific variables. The value of each of these employees will be compared against the other employees. Adequate workability implies the improvement of employee

ICASI 2018, April 23-24, Medan, Indonesia

ISBN: 978-1-63190-162-1

performance to support the implementation of the task efficiently and professionally. Productivity depends on job satisfaction while job satisfaction is a pleasant or unpleasant emotional state by which employees view their work.

The main fault of the company is that many companies do not apply stable matching techniques in placing their employees in the available positions [3][4]. Positioning is a difficult thing to do. Employee background should be viewed and analyzed before being placed in an absolute position. The stable matching algorithm can help solve such cases. One of the algorithms used is Gale-Shapley. This algorithm determines the position of the variable based on the weighting value or the desire of an institution. Their abilities will place qualified employees. If there are employees who are better than before, old employees will be withdrawn and replaced with more qualified employees. The retrieved employee will be re-analyzed to be placed in a new position. The algorithm is expected to solve the company's stable matching problem.

2. THEORIES

2.1 Stable Matching

Many algorithms deal with pattern matching [5]–[9]. The stable matching algorithm is an algorithm for finding stable position problem solution [10]–[13]. By using this algorithm, it will get a bipartite graph that shows matching between stable and optimal element, either optimal in man side or can be optimal in woman side [14]. For example, in the Stable Marriage Problem, both men, and women, to be applicants, and one more type to be the recipient of the application. The type of set of applicants will be the rated for optimal measurement. If the man who acts as an applicant, the stable pair generated will be optimal relative to men; it also applies if the applicant's role is changing. The male role will be used as an applicant so that the results of this algorithm will produce optimal stable pair relative to men.

2.2 Gale-Shapley Algoritm

David Gale and Lloyd Shapley in 1962 introduced a matching study to allocate a set of pairs with stability known as the Stable Marriage Problem. The goal is to find a stable pair of instances of X and Y [15][16]. Each variable has its preference list of the paired pair. The Stable Marriage Problem was first introduced in Gale and Shapley's seminar paper "College Admissions and the Stability of Marriage" in 1962. To solve the Stable Marriage Problem problem, Gale and Shapley introduced the Gale-Shapley Algorithm to pair some n objects X with n objects Y using specific rules.

The first object is defined as n men and the second object is n women in an arranged marriage, in which both parties have preference list against each of the opposite sex. Gale-Shapley has a proposed rule of each man n to each woman. In the process of algorithm execution, each man has an alternative pair and free, but every woman must pair [17][18]. Women are definitely in pairs even if their partner can change. Men who couple more than one get a couple who have the least preference of himself [19]. When a woman receives a proposal directly will be accepted and become a temporary partner [20]. When a woman who has been in pairs receives a different application, she will compare with the previous application and reject the man who has a smaller criteria fit against the female preference. Each man is applying to women according to the criteria in order until later in pairs. If the application is rejected by one of the women on the list, then he is free again, and continue the application sequence on his list. The algorithm ends when everything has been paired.

The following pseudocode is Gale-Shapley algorithm process [21].

```
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 = first woman on m’s list to whom m 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' becomes free
           (m, w) become engaged
         else
           (m', w) remain engaged
    }
}
```

3. METHODOLOGY

Gale-Shapley works based on criteria stored in the preferences table. This table shows how much an employee's interest in the job and the company wants to take on the employee. The establishment of the Gale-Shapley algorithm requires two preference tables. It is formed with the same number of columns and rows as a stable matching. Each cell is filled with a number corresponding to the value expected by both parties. There can not be the same number in a row. These tables consist of employee qualifications and company desires. Preferences can be seen in the following table.

Table 1. Employee qualifications

	Position 1	**Position 2**	**Position 3**	**Position 4**	**Position 5**	**Position 6**
Employee 1						
Employee 2						
Employee 3						
Employee 4						
Employee 5						

Employee 6						

Table 2. Company desires

	Employee 1	**Employee 2**	**Employee 3**	**Employee 4**	**Employee 5**	**Employee 6**
Position 1						
Position 2						
Position 3						
Position 4						
Position 5						
Position 6						

Table 1 and 2 are tables of employee and company preference. Each employee will give weight to the six desired positions and vice versa the company assesses their employees with specific weights. The numbers "1" to "6" will be filled into each cell. These are the order of popularity for employees and companies.

4. RESULT AND DISCUSSION

An employee is a party who actively make positions based on their preference list so that the flow of this algorithm can be analogized as a job seeker who applies in the company vacancy. A position is a passive party but assigned to determine the employee who is eligible to fill vacant positions based on the options contained in the preference list. The pairing process between Employee and Position will experience engagement and unengagement under certain conditions. This section is a Gale-Shapley algorithm testing by giving a case example to a company in selecting employees to be placed in certain positions.

Table 3. Employee qualifications result

	Programmer	**Manager**	**Marketing**	**Mechanic**	**Post Man**	**Supervisor**
Pritz Brown	3	5	2	1	6	4

Sheryl	4	5	3	2	5	1
Robin Hood	1	2	3	5	6	4
Charles	5	2	3	4	1	6
Andysah	2	5	3	6	4	1
Keysha	6	1	4	5	3	2

Table 4. Company desires result

	Pritz Brown	Sheryl	Robin Hood	Charles	Andysah	Keysha
Programmer	1	2	4	5	3	6
Manager	5	4	2	3	1	6
Marketing	4	1	2	5	6	3
Mechanic	3	1	6	5	2	4
Post Man	1	6	4	5	3	2
Supervisor	2	3	4	6	5	1

Tables 3 and 4 show the value of each preference between employee and company. The Gale-Shapley algorithm has several rounds in finding the optimal value in this case. The number of rounds depends on how quickly the algorithm finds the solution. The following calculations may explain the workings of the Gale-Shapley algorithm.

Round 1

```
Employee = Pritz Brown       <1>
       Placed to      = Mechanic     <4>
       Status         = Permitted
       Reason         = Position Mechanic is available

Employee = Sheryl     <2>
```

```
       Placed to      = Supervisor   <6>
       Status         = Permitted
       Reason         = Position Supervisor is available

Employee = Robin Hood <3>
       Placed to      = Programmer   <1>
       Status         = Permitted
       Reason         = Position Programmer is available

Employee = Charles    <4>
       Placed to      = Post Man     <5>
       Status         = Permitted
       Reason         = Position Post Man is available

Employee = Andysah    <5>
       Placed to      = Supervisor   <6>
       New Weight     = 5
       Old Weight     = 3
       Replace        = Sheryl
       Status         = Unchanged
       Reason         = Weight Andysah is lower than Sheryl

Employee = Keysha     <6>
       Placed to      = Manager      <2>
       Status         = Permitted
       Reason         = Position Manager is available

Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [6] = Supervisor
Employee [3] = Robin Hood    Position [1] = Programmer
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [0] = -
Employee [6] = Keysha        Position [2] = Manager
```

Round 2

```
Employee = Andysah    <5>
       Placed to      = Programmer   <1>
       New Weight     = 3
       Old Weight     = 4
       Replace        = Robin Hood
       Status         = Changed
       Reason         = Weight Andysah is higher than Robin Hood

Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [6] = Supervisor
Employee [3] = Robin Hood    Position [0] = -
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [2] = Manager
```

Round 3

```
Employee = Robin Hood <3>
       Placed to      = Programmer   <1>
       New Weight     = 4
       Old Weight     = 3
       Replace        = Andysah
       Status         = Unchanged
       Reason         = Weight Robin Hood is lower than Andysah

Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [6] = Supervisor
Employee [3] = Robin Hood    Position [0] = -
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [2] = Manager
```

Round 4

```
Employee = Robin Hood <3>
       Placed to      = Manager      <2>
       New Weight     = 2
       Old Weight     = 6
       Replace        = Keysha
       Status         = Changed
       Reason         = Weight Robin Hood is higher than Keysha

Employee = Keysha     <6>
       Placed to      = Manager      <2>
       New Weight     = 6
       Old Weight     = 2
       Replace        = Robin Hood
       Status         = Unchanged
       Reason         = Weight Keysha is lower than Robin Hood

Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [6] = Supervisor
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [0] = -
```

Round 5

```
Employee = Keysha     <6>
       Placed to      = Supervisor   <6>
       New Weight     = 1
       Old Weight     = 3
       Replace        = Sheryl
       Status         = Changed
       Reason         = Weight Keysha is higher than Sheryl
```

```
Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [0] = -
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [6] = Supervisor
```

Round 6

```
Employee = Sheryl     <2>
       Placed to      = Supervisor   <6>
       New Weight     = 3
       Old Weight     = 1
       Replace        = Keysha
       Status         = Unchanged
       Reason         = Weight Sheryl is lower than Keysha

Employee [1] = Pritz Brown   Position [4] = Mechanic
Employee [2] = Sheryl        Position [0] = -
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [6] = Supervisor
```

Round 7

```
Employee = Sheryl     <2>
       Placed to      = Mechanic     <4>
       New Weight     = 1
       Old Weight     = 3
       Replace        = Pritz Brown
       Status         = Changed
       Reason         = Weight Sheryl is higher than Pritz Brown

Employee [1] = Pritz Brown   Position [0] = -
Employee [2] = Sheryl        Position [4] = Mechanic
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [6] = Supervisor
```

Round 8

```
Employee = Pritz Brown       <1>
       Placed to      = Mechanic     <4>
       New Weight     = 3
       Old Weight     = 1
       Replace        = Sheryl
       Status         = Unchanged
       Reason         = Weight Pritz Brown is lower than Sheryl
```

```
Employee [1] = Pritz Brown   Position [0] = -
Employee [2] = Sheryl        Position [4] = Mechanic
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [6] = Supervisor
```

Round 9

```
Employee = Pritz Brown       <1>
       Placed to      = Marketing   <3>
       Status         = Permitted
       Reason         = Position Marketing is available

Employee [1] = Pritz Brown   Position [3] = Marketing
Employee [2] = Sheryl        Position [4] = Mechanic
Employee [3] = Robin Hood    Position [2] = Manager
Employee [4] = Charles       Position [5] = Post Man
Employee [5] = Andysah       Position [1] = Programmer
Employee [6] = Keysha        Position [6] = Supervisor
```

The testing has resulted in nine rounds. The final result of Gale-Shapley has determined that each employee has gained an optimal position. The last round explains that "Andysah" gained the position as "Programmer," "Robin Hood" as "Manager," "Pritz Brown" as "Marketing," "Sheryl" as "Mechanic," "Charles" as "Post Man" and "Keysha" as "Supervisor."

5. CONCLUSION

This algorithm successfully matches the employee with the offered position. The result of this test is stable so that the optimal pair formed. The Gale-Shapley algorithm determines the pair based on the weight value of each employee. If the position of the couple changes, this means there is a new candidate who is more qualified. The algorithm has a preference table that contains interest between both parties, both employees, and positions. Any installation made between employees and positions in a set can be said to be the allocation of stable pairs. But not all preference list is a potential stable partner. A stable pair will be determined in the last round. A condition will result in a separate pair because a certain employee rank is higher than the one previously occupying that position. Although engagement occurs, employees who do not have a partner will someday occupy an existing position. The more preference lists, the more employees to occupy certain positions.

REFERENCES

[1] A. Putera, U. Siahaan, and R. Rahim, “Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm,” *Int. J. Secur. Its Appl.*, vol. 10, no. 8, pp. 173–180, Aug. 2016.

[2] H. A. Hasibuan, R. B. Purba, and A. P. U. Siahaan, “Productivity Assessment (Performance, Motivation, and Job Training) using Profile Matching,” *Int. J. Econ. Manag. Stud.*, vol. 3, no. 6, pp. 73–77, 2016.

[3] A. E. Roth, “The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory,” *J. Polit. Econ.*, vol. 92, no. 6, pp. 991–1016, 1984.

[4] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. New York: Cambridge University Press, 2009.

[5] R. Meiyanti, A. Subandi, N. Fuqara, M. A. Budiman, and A. P. U. Siahaan, “The recognition of female voice based on voice registers in singing techniques in real-time using hankel transform method and macdonald function,” *J. Phys. Conf. Ser.*, vol. 978, no. 1, p. 012051, Mar. 2018.

[6] S. Aryza, M. Irwanto, Z. Lubis, A. P. U. Siahaan, R. Rahim, and M. Furqan, “A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, p. 012067, 2018.

[7] R. Rahim, I. Zulkarnain, and H. Jaya, “A review: search visualization with Knuth Morris Pratt algorithm,” in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 237, no. 1, p. 012026.

[8] R. Rahim, A. S. Ahmar, A. P. Ardyanti, and D. Nofriansyah, “Visual Approach of Searching Process using Boyer-Moore Algorithm,” *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 012001, Dec. 2017.

[9] R. Rahim, S. Nurarif, M. Ramadhan, S. Aisyah, and W. Purba, “Comparison Searching Process of Linear, Binary and Interpolation Algorithm,” *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 012007, Dec. 2017.

[10] R. Anurag and A. Bhattacharya, “SMS: Stable Matching Algorithm using Skylines,” in *Proceedings of the 28th International Conference on Scientific and Statistical Database Management*, 2016.

[11] R. Rahim *et al.*, “Latent Semantic Indexing for Indonesian Text Similarity,” *Int. J. Eng. Technol.*, vol. 7, no. 2.3, pp. 73–77, 2018.

[12] R. Rahim, T. Afriliansyah, H. Winata, D. Nofriansyah, Ratnadewi, and S. Aryza, “Research of Face Recognition with Fisher Linear Discriminant,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, p. 012037, 2018.

[13] R. Rahim, D. Hartama, H. Nurdiyanto, A. S. Ahmar, D. Abdullah, and D. Napitupulu, “Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm,” *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 012008, 2018.

[14] D. Doerner, J., Evans and A. Shelat, “Secure Stable Matching at Scale,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[15] L. Dubins and D. Freedman, “Machiavelli and the Gale–Shapley Algorithm,” *Math. Mon.*, vol. 88, no. 7, pp. 485–494, 1981.

[16] R. Rahim *et al.*, “Searching Process with Raita Algorithm and its Application,” *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 012004, Apr. 2018.

[17] R. Rahim *et al.*, “Combination Base64 Algorithm and EOF Technique for Steganography,” *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 012003, Apr. 2018.

[18] C. P. Teo, J. Sethuraman, and W. P. Tan, “Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications,” *J. Manag. Sci.*, vol. 47, no. 9, pp. 1252–1267, 2001.

[19] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*. Massachusetts: MIT Press, 1989.

[20] H. M. Ritonga, A. P. U. Siahaan, and Suginam, “Marketing Strategy through Markov Optimization to Predict Sales on Specific Periods,” *Int. J. Innov. Res. Multidiscip. F.*, vol. 3, no. 8, pp. 184–190, 2017.

[21] Wikipedia, “Stable Marriage Problem,” *Wikipedia*, 2018. .