

# A New Perspective to Stable Marriage Problem in Profit Maximization of Matrimonial Websites

Aniket Bhatnagar\*, Varun Gambhir\*, and Manish Kumar Thakur\*

## Abstract

For many years, matching in a bipartite graph has been widely used in various assignment problems, such as stable marriage problem (SMP). As an application of bipartite matching, the problem of stable marriage is defined over equally sized sets of men and women to identify a stable matching in which each person is assigned a partner of opposite gender according to their preferences. The classical SMP proposed by Gale and Shapley uses preference lists for each individual (men and women) which are infeasible in real world applications for a large populace of men and women such as matrimonial websites. In this paper, we have proposed an enhancement to the SMP by computing a weighted score for the users registered at matrimonial websites. The proposed enhancement has been formulated into profit maximization of matrimonial websites in terms of their ability to provide a suitable match for the users. The proposed formulation to maximize the profits of matrimonial websites leads to a combinatorial optimization problem. We have proposed greedy and genetic algorithm based approaches to solve the proposed optimization problem. We have shown that the proposed genetic algorithm based approaches outperform the existing Gale-Shapley algorithm on the dataset crawled from matrimonial websites.

## Keywords

Bipartite Matching, Combinatorial Optimization, Evolutionary Computing, Genetic Algorithm, Matrimonial Websites, Stable Marriage Problem

## 1. Introduction

Matching in a bipartite graph aims to assign elements of one set to the elements of another set such that no two elements in the same set are associated. It has been successfully employed in many applications, such as, hospital-resident allocation, stable roommate problem, allocating tutorials in schools and colleges, sailor-boat problem, student project allocation, college admissions, and stable marriage problem (SMP), etc. [1-3]. The problem of stable marriage proposed by Gale and Shapely [4] is defined over equally sized sets of men and women to identify a stable matching/marriage in which each person is assigned a partner of opposite gender according to their preferences. The set of assignments or marriages is said to be unstable if there exists a man and a woman who are not married to each other but prefer each other to their assigned partners.

Gale and Shapley [4] proposed a male optimal solution using an iterative process for identifying the

\* This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received December 26, 2017; first revision April 18, 2018; accepted May 12, 2018.

Corresponding Author: Manish Kumar Thakur (mthakur.jiit@gmail.com)

\* Dept. of Computer Science Engineering and IT, Jaypee Institute of Information Technology, Noida, India ({aniket.bhatnagar0310, varungambhir95, mthakur.jiit}@gmail.com)

stable marriages with preference lists (ordered preferences) maintained for all men and women. In [5], the authors extended the work of [4] to incorporate female optimal stable solution and minimum choice stable solution. Irving et al. [6] proposed a network flow based approach to find the stable marriages with male and female optimal solutions. Further, they also generalized the problem of stable marriage according to the weighted preference lists representing the weights of numerical preferences maintained by every person in the set. In literature, the problem of stable marriage has gained a lot of attention and attracted many researchers to analyze and extend this problem with different variants [7-12].

Let  $M(1 \dots m)$  and  $W(1 \dots n)$  be the two sets of men and women respectively, where  $M$  consists of  $m$  number of men and  $W$  consists of  $n$  number of women, some of the variants of SMP are categorized as follows: (a)  $m = n$ , i.e., count of men and women are equal; (b)  $m \neq n$ , i.e., count of men and women are different; and (c) polygamy, where one person is allowed to have multiple spouses [10]. Here, some of the earlier variants of SMP utilized the preference lists maintained by individuals, whereas some of the recent contributions utilized the weighted preferences or qualitative preferences [9,11].

In these variants, SMP aims to arrange the marriages in such a way that men and women should pair up knowing the preferences of each man and woman in the populace. However, there may be several instances where the population of men and women is very large, such as marriage bureau or matrimonial websites, viz. [www.bharatmatrimony.com](http://www.bharatmatrimony.com), [www.jeevansathi.com](http://www.jeevansathi.com), and [www.shaadi.com](http://www.shaadi.com), etc., where numerous men and women register to get a suitable, satisfactory and stable match. The SMP is not applicable in these instances as it is infeasible for users to rank their partners of opposite gender to create the preference lists. Further, these matrimonial websites charge a certain amount of fee from the registered users in order to dispense the information of a suitable match [13]. This, in turn, leads to generate revenue for matrimonial websites and helps to earn profits. In the context of matrimonial websites, there is a need for a strategy which dispenses the preferred information to ensure maximum profits in terms of their ability to provide a satisfactory and stable match among the registered users. In addition, an efficient approach is needed to compute the preference lists from the information provided by the users while registering on a matrimonial website. In the process of registration, users are required to make profile by providing personal details (now onwards in this paper we call these details as features) viz. height, age, complexion, academic details, job details (viz. salary, experience, etc.), and religion, etc. While registration, users are also required to provide the necessary requirements or expectations in the partner's profile of opposite gender viz. someone might be looking for educated and good looking partner.

In this paper we have addressed these two issues: (1) computation of an individual's preference in the large populace of men and women; and (2) finding a satisfactory and stable match for registered users to ensure the maximum profit for a matrimonial website. To compute the preferences, we have used the Euclidean distance between the expectations of men and profile of women or vice versa. Further, we have proposed a combinatorial framework to maximize the profit of a matrimonial website. Several soft computing methods, such as genetic algorithm (GA) and particle swarm optimization, etc., have been successfully employed in many combinatorial problems [14,15]. To solve our proposed framework we have applied GA due to its simplicity in solving combinatorial problems [14] and also proposed a Greedy based solution. The proposed methods have been shown to outperform Gale-Shapley algorithm on the real dataset crawled from two matrimonial websites.

Rest of the paper is organized as follows: in Section 2, we present the problem formulation which includes the enhancements proposed in classical SMP. The methodology to solve the proposed formulation has been described in Section 3. Further, experiments and results are illustrated in Section 4. Finally, Section 5 concludes the work.

## 2. Problem Formulation

This section starts with defining various variables and terminologies used in this paper. In-order to meet various expectations of matrimonial websites, we have proposed enhancements for classical SMP, which has been discussed later in this section along with the problem statement(s).

### 2.1 Variables and Terminologies

Throughout the paper we have used two sets,  $M(1 \dots m)$  and  $W(1 \dots n)$ , where,  $M$  is the set of  $m$  men and  $W$  is the set of  $n$  women registered on a matrimonial website;  $m$  is the count of men;  $n$  is the count of women;  $M(a)$  is an instance/element of the set  $M$ , whereas  $W(b)$  is an instance/element of the set  $W$ ;  $F(1 \dots t)$  is the set of features (viz. height, weight, salary, etc.) describing the man or woman;  $P_M(1 \dots m, F(1 \dots t), 1 \dots t)$  and  $P_W(1 \dots n, F(1 \dots t), 1 \dots t)$  are the sets of the profile of men and women respectively which are made-up of various  $(1 \dots t)$  features and the respective quantified values/scores (at the scale of 5, where 5 represents very good, 4 represents good, 3 represents average, 2 represents below average, 1 represents bad and 0 represents very bad), viz.  $P_M(1, 1, 5)$  represents that the value of feature 1,  $F(1)$  (say salary) in the profile of man,  $M(1)$  is 5 which is the representation of a very good salary;  $E_M(1 \dots m, F(1 \dots t), 1 \dots t)$  and  $E_W(1 \dots n, F(1 \dots t), 1 \dots t)$  are the sets of expectations (or requirements) and respective quantified value of men and women respectively in their partners of opposite gender, viz.  $E_M(1, 1, 4)$  represents that man,  $M(1)$  is expecting a good (4) value of the feature,  $F(1)$ , i.e. salary in the profile of the opposite gender.

### 2.2 Proposed Enhancements in SMP

Generally a large populace of men and women register on matrimonial websites. So asking them to provide their ordered preferences of opposite gender partners seem to be a difficult task for the registered users. Further, benefits (or profits) of matrimonial websites are associated either with maximum monetary gain or with their ability to provide a suitable or satisfactory matching for each man and woman registered on the matrimonial website. If the matrimonial website does not provide satisfactory matches to its registered users, surely the website will lose its users and popularity. Hence, in this paper, we have focused on the following objective: satisfactory matching for the registered users according to their expectations to maximize the profits of matrimonial websites. In order to achieve the listed objective, we have proposed following enhancements for SMP. We refer to our proposed enhanced SMP as ESMP henceforth.

**Enhancement 1:** Instead of qualitative preference lists (ordered preferences) which are maintained by each man and woman in classical SMP, we propose to make a quantitative list (weighted preference). In the context of matrimonial websites, we propose to create this quantitative list by comparing the expectations of an individual with the profile of opposite gender partners. Here, matching of each

expectation of an individual with the profile of opposite gender partners is given a weight as some feature might be dominating over other, viz. salary may be more dominating than height and should be given more weight than height. The computed average of all the matched expectations of an individual (say a man,  $M(a) \in M$ ) with the profile of opposite gender (say a woman,  $W(b) \in W$ ) is now onwards called as *weighted matching*. The quantitative list, representing *weighted matching* for each man with all women or vice versa is now onwards called as *weighted preference*. In this paper, we have used three different types of such preferences, viz. Man Weighted Preference (MWP), Woman Weighted Preference (WWP), and Combined Weighted Preference (CWP) and are defined as follows:

$MWP(1..m, 1..n)$ : This is a *weighted preference* for each man,  $M(a) \in M$ , ( $1 < a < m$ ) with each woman,  $W(b) \in W$ , ( $1 < b < n$ ).  $MWP(a, b)$  represents the average of all the expectations of the man,  $M(a)$  matched with the profile of the woman,  $W(b)$  and defined as follows in Eq. (1).

$$MWP(a, b) = \frac{\sum_{e,p=1}^i Wt_e \times dist(E_M(M(a), F(e), V_e), P_W(W(b), F(p), V_p))}{i} \quad (1)$$

where,  $e = 1$  to  $i$  are the list of expectations of the man,  $M(a)$ ;  $Wt_e$  (at the scale of 100) is a weight associated with the  $e^{th}$  expectation; and  $dist$  is a function which returns a fractional value (between 0 and 1) depending on the Euclidean distance between the value  $V_e$  of the  $e^{th}$  expectation ( $E_M(M(a), F(e), V_e)$ ) of the man  $M(a)$  and the value  $V_p$  of the  $p^{th}$  profile ( $P_W(W(b), F(p), V_p)$ ) of the woman,  $W(b)$ , if  $F(e)$  and  $F(p)$  both represent the same feature, viz. salary.

As defined in Eq. (1), MWP of each man for each woman lies in the range between 0 and 100. Further, MWP, if represented in qualitative form, is the men's preference list as used in classical SMP.

$WWP(1..n, 1..m)$ : This is a *weighted preference* for each woman  $W(b) \in W$ , ( $1 < b < n$ ) with each man,  $M(a) \in M$ , ( $1 < a < m$ ).  $WWP(b, a)$  represents the average of all the matched expectations of the woman,  $W(b)$  with the man,  $M(a)$  and defined as follows in Eq. (2):

$$WWP(b, a) = \frac{\sum_{e,p=1}^i Wt_e \times dist(E_W(W(b), F(e), V_e), P_M(M(a), F(p), V_p))}{i} \quad (2)$$

where,  $e = 1$  to  $i$  are the list of expectations of the woman,  $W(b)$ ;  $Wt_e$  (at the scale of 100) is a weight associated with  $e^{th}$  expectation, and  $dist$  is a function which returns a fractional value (between 0 and 1) depending on the Euclidian distance between the value  $V_e$  of the  $e^{th}$  expectation ( $E_W(W(b), F(e), V_e)$ ) of the woman,  $W(b)$  and the value  $V_p$  of the  $p^{th}$  profile ( $P_M(M(a), F(p), V_p)$ ) of the man,  $M(a)$ , if  $F(e)$  and  $F(p)$  both represent the same feature, viz. salary.

As defined in Eq. (2), WWP of each woman for each man lies in the range between 0 and 100. Further, its presentation in qualitative form represents the women's preference list as used in classical SMP.

$CWP(1 \dots m, 1 \dots n)$ : This is a *weighted preference* where expectations of each man and woman are jointly explored with profiles of each woman and man respectively.  $CWP(a, b)$  represents the *weighted matching* jointly computed as the expectations of the man,  $M(a)$  matched with the profile of the woman,  $W(b)$  and expectations of that woman,  $W(b)$  matched with the profile of that man,  $M(a)$  and defined as follows in Eq. (3):

$$CWP(a, b) = \frac{MWP(a, b) + WWP(b, a)}{2} \quad (3)$$

where,  $MWP(a, b)$  and  $WWP(b, a)$  are computed according to Eq. (1) and (2), respectively. As defined in Eq. (3), CWP between each man and each woman lies in the range between 0 and 100.

An example is presented in Fig. 1 (for  $m = n = 2$ ) where, one of the CWPs is 85, i.e.  $CWP(1, 1) = 85$ . On the scale of 100, CWP of 85 is significantly large and hence suggests that almost all expectations of  $M(1)$ , viz. educated, and good looking, etc., are met in  $W(1)$ , viz. she is good looking, educated as well as almost all expectations of  $W(1)$  are met in  $M(1)$ .

	$W(1)$	$W(2)$
$M(1)$	85	81
$M(2)$	83	49

**Fig. 1.** An example of CWP.

**Enhancement 2:** Considering the objective to maximize the profits of matrimonial websites, we have modified the stability criteria of the classical SMP as follows; for a CWP, we consider pairing such men and women which maximize the average *weighted matching* (as given in Eq. (4)). Now onwards, we will call the average *weighted matching* as AWM and the maximum average *weighted matching* as MAWM.

$$\text{MAWM} = \max \left( \frac{\sum_{i=1}^{\tau} \text{CWP}(\text{pair}(i))}{\tau} \right) \quad (4)$$

where,  $\tau$  is the minimum of  $m$  and  $n$  which are the count of men and women, respectively and under inequality, minimum of the two is considered for computing the AWM;  $\text{pair}(i)$  returns the indices of a pair of men and women such that for each  $i$ , the pair of men and women is unique.

In the example of Fig. 1, there are two possible sets of pairing: (a)  $M(1)$  with  $W(1)$  and  $M(2)$  with  $W(2)$ , having AWM as 67 i.e.  $(85 + 49)/2$ ; and (b)  $M(1)$  with  $W(2)$  and  $M(2)$  with  $W(1)$ , having AWM as 82, i.e.  $(81 + 83)/2$ . Here, MAWM is achieved with the pairing given in Set b.

### 2.3 Problem Statement(s)

Considering different scenarios, following problem statements have been addressed in this paper.

**DEFINITION 1 (D1):** Given sets of  $m$  registered men and  $n$  registered women (where,  $m = n$ ), each with their profile and expectations, it is required to identify a suitable/stable pairing between men and women such that profit of the matrimonial website is maximum.

**DEFINITION 2 (D2):** Given sets of  $m$  registered men and  $n$  registered women (where,  $m \neq n$ ), each with their profile and expectations, it is required to identify a suitable/stable pairing such that profit of the matrimonial website is maximum. Here, if  $m > n$ , then some men in the list will be unpaired and if  $m < n$ , then some women will be unpaired.

## 3. Proposed Scheme(s)

As discussed in the previous section, various sets or combinations of pairings between men and women are to be explored based on their CWP. Consequently, the proposed ESMP has resulted in a combinatorial problem. This section presents our proposed greedy and GA based approaches to solve

the ESMP and achieve the desired objective optimally or near optimally. The CWP presented in Fig. 2 has been used to elaborate the algorithmic steps.

Woman →		$W(1)$	$W(2)$	$W(3)$	$W(4)$	$W(5)$	$W(6)$	$W(7)$	$W(8)$
Man ↓									
$M(1)$		85	81	43	75	61	82	69	78
$M(2)$		81	63	52	67	60	65	56	55
$M(3)$		67	56	23	70	39	72	80	82
$M(4)$		26	28	92	78	98	10	4	87
$M(5)$		85	90	24	22	13	91	88	90
$M(6)$		91	95	10	17	78	80	82	45
$M(7)$		24	34	45	56	67	10	12	45
$M(8)$		83	49	56	89	90	70	23	44

Fig. 2. CWP used to elaborate the algorithmic steps discussed in this section.

### 3.1 Greedy Based Approach (Algorithm 1)

The proposed algorithm is based on the greedy approach where a man under consideration has been paired with the best unengaged woman or vice versa.

**Step 1:** For each man,  $M(a) \in M$  in CWP, create a sorted list in decreasing order of the profile-expectation based *weighted matching* with each women,  $W(b) \in W$  using stable sorting schemes [16], i.e. maintain the order of the CWPs in the sorted list in case of the tie between CWPs of two pairs of men and women. As an example, sorted lists of men  $M(1)$  and  $M(2)$  created from the CWP of Fig. 2 are depicted in Fig. 3.

	$W(1)$	$W(6)$	$W(2)$	$W(8)$	$W(4)$	$W(7)$	$W(5)$	$W(3)$
$M(1)$	85	82	81	78	75	69	61	43

	$W(1)$	$W(4)$	$W(6)$	$W(2)$	$W(5)$	$W(7)$	$W(8)$	$W(3)$
$M(2)$	81	67	65	63	60	56	55	52

Fig. 3. Sorted lists of men,  $M(1)$  and  $M(2)$  created from the CWP of Fig. 2.

**Step 2:** Maintain a list, storing the status (initially unengaged) of each woman either as engaged or unengaged.

**Step 3:** Traverse the sorted lists (an ordered list obtained after applying the stable sorting) for each man from the maximum liked woman to the least. In each traversal, select the first woman which is yet to be engaged and make the pairing between the selected woman and the man under consideration. In case of the tie between CWPs of a man,  $M(a)$  with two or more unengaged women (say,  $W(b)$  and  $W(c)$ ), make the pair between  $W(b)$  and  $M(a)$ , if  $b < c$  (i.e. if,  $W(b)$  appears before  $W(c)$  in the ordered sorted list), otherwise make the pair between  $W(c)$  and  $M(a)$ , and change the woman’s status to engaged.

**Step 4:** Repeat Step 3 until all men (if  $m \leq n$ ) get paired or all women (if  $m > n$ ) get engaged.

In the preceding example, following pairs or matching will be formed:  $\{M(1), W(1)\}$ ,  $\{M(2), W(4)\}$ ,  $\{M(3), W(8)\}$ ,  $\{M(4), W(5)\}$ ,  $\{M(5), W(6)\}$ ,  $\{M(6), W(2)\}$ ,  $\{M(7), W(3)\}$ , and  $\{M(8), W(7)\}$ , having the AWM as 73.25. Hence in this example, the MAWM achieved by the presented algorithm (Algorithm 1) is 73.25. However, this may or may not be the actual/optimal MAWM as the proposed greedy based approach (selecting such unengaged woman to be paired with the man under consideration whose CWP is highest) may or may not converge in global maxima, i.e. optimal MAWM.

### 3.2 Genetic Algorithm Based Approach

This section starts with the discussion over the basic structure of GA and presents the proposed GA based approaches for solving the ESMP.

Usually, GA based optimization involves following: encoding of the actual real world solution space to form the computation space or genotype; chromosome, which is one of the possible solutions to a problem; population generation, which is a subset of all the possible solutions to a problem; i.e. set of chromosomes forms the population; computation of the fitness score of chromosomes; applying various operators, viz. selection, crossover and mutation to select chromosomes and generate new offspring and if suitable, replaces the existing individuals in the population.

In line with the basic structure of a GA based optimization, we present two GA based approaches to maximize the profits under two scenarios,  $m = n$ , i.e. D1; and  $m \neq n$ , i.e. D2. Both approaches differ in the process of the initial population generation. In the first approach the initial population is formed with the chromosomes generated in entirely random manner. In the second approach, the initial population is formed with the chromosomes generated in random as well as guided manner. In both approaches, the genotype is constructed using at most  $t + 1$  symbols, where,  $t = m$ . Here, each symbol represents a man index, viz. symbol 1 represents the man,  $M(1)$ , symbol 2 represents the man,  $M(2)$  and so forth. An additional symbol, '0' is used to represent dummy men to be paired with women under the scenario where  $m < n$ .

Further, a chromosome is constituted as a coded string of men-women pairing comprising of their unique identification in such an order that first woman,  $W(1)$  is assigned to the man whose index is encoded as the first gene in the chromosome; second woman,  $W(2)$  is assigned to the man whose index is encoded as the second gene in the chromosome and so on. Lengths of the chromosomes under different scenarios are as follows: (a) it is  $m$  or  $n$ , when,  $m = n$ , (b) it is  $m$ , when,  $m > n$ , where some of the men to be paired with  $(m - n)$  dummy women and not to be considered while computing the fitness score of such chromosome, (c) it is  $n$ , when,  $m < n$ , where some of the women to be paired with  $(n - m)$  dummy men encoded with the symbol, '0', and not to be considered while computing the fitness score. Fig. 4(a)–(c) present the examples of such encoding under the scenarios,  $m = n$  (with  $m$  and  $n$  as 4),  $m > n$  (with  $m = 4$  and  $n = 2$ ), and  $m < n$  (with  $m = 2$  and  $n = 4$ ), respectively. In the example of Fig. 4(a), the encoded string  $\langle 3\ 1\ 4\ 2 \rangle$  represents the following four pairing:  $\{M(3), W(1)\}$ ,  $\{M(1), W(2)\}$ ,  $\{M(4), W(3)\}$ , and  $\{M(2), W(4)\}$ . Further, in the example of Fig. 4(b) the encoded string  $\langle 3\ 1\ 4\ 2 \rangle$  represents the following two valid pairing (after discarding the pairing with dummy women):  $\{M(3), W(1)\}$ , and  $\{M(1), W(2)\}$ . Similarly, in the example of Fig. 4(c) the encoded string  $\langle 0\ 1\ 0\ 2 \rangle$  represents two dummy men encoded by the symbol '0'. In this example, four pairing

is possible (including the pairs made with dummy men). Out of the four, following two are valid pairs and contribute in computation of the fitness score:  $\{M(1), W(2)\}$  and  $\{M(2), W(4)\}$ .

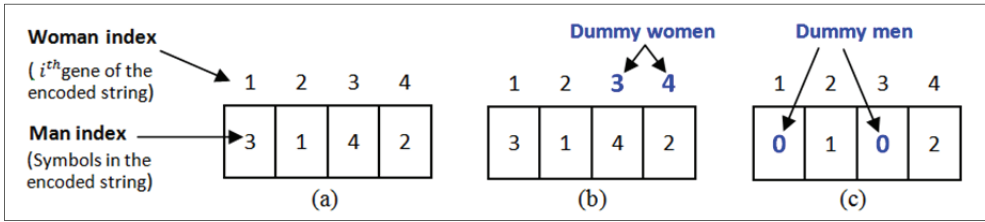


Fig. 4. Example of chromosomes under the scenarios, (a)  $m = n$ , (b)  $m > n$ , and (c)  $m < n$ .

### 3.2.1 GA based approach with random population (Algorithm 2)

This section discusses the proposed GA based approach where the initial population is created with the randomly generated chromosomes. Here we have separately handled the scenarios,  $m = n$  and  $m \neq n$ , and present two algorithms Algorithm 2.1 and Algorithm 2.2 to handle these scenarios respectively. Steps of the proposed algorithms are elaborated through the combined weighted preference, CWP given in Fig. 2.

**Algorithm 2.1:** *When,  $m = n$*

**Step 1:** Randomly make the pairing between men and women such that all men and women are uniquely paired. Apply man indices (1 to  $m$ ) to encode the pairs to constitute a chromosome and store it into a population array of length  $m$ . Repeat the process and obtain the set of chromosomes stored in different arrays (each of length  $m$ ) to generate an initial population.

Fig. 5 presents one of the chromosomes in the initial population, which is generated by making random pairs between 8 men and 8 women (whose CWP is presented in Fig. 2) and stored into a population array, *Pop* of length 8.

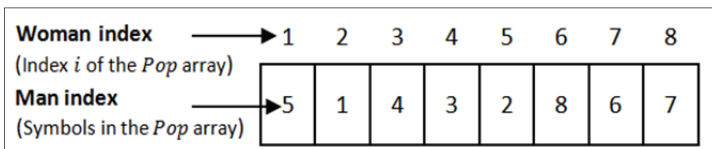


Fig. 5. An example of a chromosome in the initial population, when  $m = n = 8$ .

**Step 2:** With respect to the *weighted matching* jointly computed as expectations of a man,  $M(a)$  matched with the profile of a woman,  $W(b)$  and expectations of that woman,  $W(b)$  matched with the profile of that man and stored into the CWP, apply Eq. 5 to compute the fitness scores, *FS* of the chromosomes in the initial population as follows:

$$FS = \frac{\sum_{i=1}^m CWP(Pop(i), i)}{m} \tag{5}$$

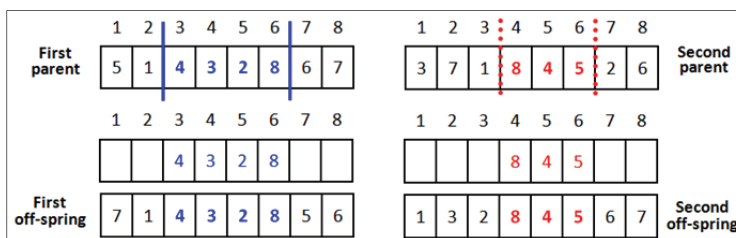


Fitness score of one of the chromosomes presented as an example in Fig. 5 is computed as  $(85 + 81 + 92 + 70 + 60 + 70 + 82 + 45) / 8$ , i.e.  $FS = 73.12$ .

**Step 3:** Apply any one of the parent selection strategies, viz. roulette wheel selection, tournament selection, etc., to select the parents (two) to be further involved in the process of mating and recombine to produce the offspring.

**Step 4:** Use the parents selected in Step 3 to undergo for the crossover operation. As it is necessary to maintain the unique pairing between all men and women in a chromosome, hence apply ordered crossover between selected parents to produce the offsprings. In this process, select a subset lying between the two crossover points in a parent (say the first parent) and add the subset to the first offspring. Further, explore the second parent to find out the symbols/values which are missing in the first offspring and add those missing symbols into the first offspring in the order they were found in the second parent. Similarly, create the second offspring by reversing the role of parents.

Fig. 6 elaborates the Step 4 with two parents selected for the mating in Step 3. Here, the subset between two crossover points, 3 and 7 are added into first offspring and remaining symbols have been added in the first offspring in the order they were found in the second parent. The second offspring is also created in the same manner.



**Fig. 6.** An example of ordered crossover to create offsprings.

**Step 5:** Randomly mutate the two offsprings created in Step 4. Because of the required unique pairing between all men and women in a chromosome, apply swap mutation to select two positions/indices randomly and interchange the values/symbols on these positions of the offsprings. Besides swap mutation, one can also select following mutation to maintain the unique pairing: scramble mutation and inversion mutation.

When we apply the swap mutation randomly at positions 4 and 8 on the first offspring (Fig. 6) and interchange their values/symbols, following chromosome will be produced:  $\langle 7\ 1\ 4\ 6\ 2\ 8\ 5\ 3 \rangle$ . Similarly,  $\langle 1\ 3\ 2\ 7\ 4\ 5\ 6\ 8 \rangle$  is the resulting chromosome, when swap mutation is randomly applied at positions 4 and 8 on the second offspring of Fig. 6.

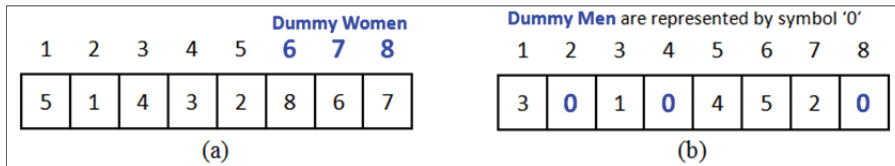
**Step 6:** Compute the fitness score of the two offsprings generated after the crossover and mutation operations. Check their fitness scores with the chromosome having the least fitness in the populace and select such chromosome to be the part of the populace having better fitness.

**Step 7:** Repeat Step 3 to Step 7 until any one of the following is not met: (a) no improvement in the population for specified  $X$  iterations or (b) total  $K$  number of generations has been produced.

**Algorithm 2.2:** *When,  $m \neq n$*

**Step 1:** Considering either  $m - n$  dummy men (if,  $m > n$ ) or  $n - m$  dummy women (if,  $n > m$ ), make the length of a chromosome as  $\max(m, n)$ . Generate a chromosome by making random groupings between  $m$  men and  $n$  women, such that all men and women are uniquely paired. These pairs might involve, either (a)  $m - n$  dummy men (if,  $m > n$ ) paired with  $m - n$  women besides  $n$  valid pairs, or (b)  $n - m$  dummy women (if,  $n > m$ ) paired with  $n - m$  men besides  $m$  valid pairs. Store all the unique pairing into the population array, *Pop*, representing a chromosome. Repeat the process and obtain the set of chromosomes stored in different population arrays to generate an initial population.

An example in Fig. 7(a) presents one of the chromosomes in the initial population. It is generated by making random pairs between 8 men and 8 women (whose CWP is presented in Fig. 2) and stored into a population array, *Pop* of length 8. In this example, we have considered 3 dummy women, viz.  $W(6)$ ,  $W(7)$ , and  $W(8)$ , which are involved in pairing with men, but the pairs involving dummy women are not considered while computing the fitness score. Similarly, the example in Fig. 7(b) presents a chromosome generated by making random pairs between 8 men (where, 3 men,  $M(6)$ ,  $M(7)$ , and  $M(8)$  are dummy men) and 8 women. These dummy men have been encoded with the symbol '0', whereas men indices (1...5) in the chromosome represents the pairing of women with actual/valid men.



**Fig. 7.** An example of chromosomes in the initial population, when (a)  $m > n$ , (b)  $m < n$ .

**Step 2:** Apply Eqs. (6) and (7) to compute the fitness scores, *FS* of the chromosomes in the initial population under the scenarios (a)  $m > n$  and (b)  $m < n$ , respectively.

$$FS = \frac{\sum_{i=1}^n CPM(Pop(i),i)}{n} \tag{6}$$

$$FS = \frac{\sum_{i=1}^n CPM(Pop(i),i)}{m}, \text{ where, } Pop(i) \geq 1 \tag{7}$$

Fitness scores of the chromosomes in preceding examples, Fig. 7(a) and (b) are computed as  $(85 + 81 + 92 + 70 + 60) / 5$ , i.e.  $FS = 77.6$  (when,  $m > n$ ) and  $(67 + 43 + 98 + 91 + 55) / 5$ , i.e.  $FS = 71$  (when  $n > m$ ), respectively.

**Step 3:** Select two parents to be involved in the process of mating and recombine to produce the offspring using any one of the parent selection strategies, viz. roulette wheel selection, tournament selection, etc.

**Step 4:** Apply the ordered crossover between selected parents to produce the offsprings. Here, some

of the men which were paired with dummy women (when,  $m > n$ ) may now be paired with actual women in the dataset.

Fig. 8 depicts the offsprings generated after the ordered crossover between the selected parents under the scenario  $m > n$ . Here three different men,  $M(1)$  and  $M(5)$  along with  $M(8)$  are respectively paired with dummy women,  $W(7)$ ,  $W(8)$ , and  $W(6)$  in the first off-spring, whereas in the second offspring, men,  $M(5)$ ,  $M(6)$ , and  $M(7)$  are respectively paired with dummy women,  $W(6)$ ,  $W(7)$ , and  $W(8)$ .

Similarly, Fig. 9 depicts the offsprings generated after the ordered crossover between the selected parents under the scenario  $m < n$ . Here, women,  $W(1)$ ,  $W(2)$ , and  $W(4)$  are paired with dummy men, represented as the symbol, '0' in the first offspring, whereas,  $W(1)$ ,  $W(3)$ , and  $W(4)$  are paired with dummy men, represented as the symbol, '0' in the second offspring.

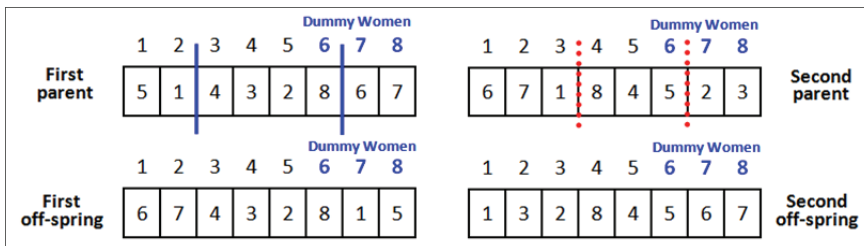


Fig. 8. An example of ordered crossover to create offsprings, when  $m > n$ .

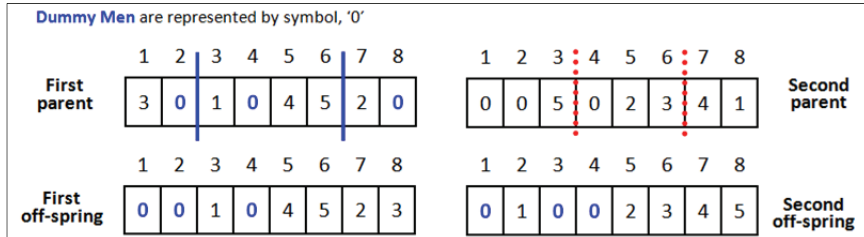


Fig. 9. An example of ordered crossover to create offsprings, when  $m < n$ .

**Step 5:** Use swap mutation to swap the randomly selected symbols/bits of the two offsprings created in previous step (Step 4). Compute the fitness scores (either using Eq. 6, if  $m > n$  or Eq. 7, if  $n > m$ ) of the two offsprings generated after the crossover and mutation operations. Check these fitness scores with the chromosome having the least fitness in the populace and select such chromosome to be the part of the populace having better fitness.

**Step 6:** Repeat, Step 3 to Step 6 until any one of the following is not met: (a) no improvement in the population for specified  $X$  iterations or (b) total  $K$  number of generations has been produced.

### 3.2.2 GA based approach with guided population (Algorithm 3)

This approach is built around the steps of Algorithm 2, discussed in the previous section. Unlike Algorithm 2, here some of the chromosomes in the initial population are generated in a guided manner

and presented subsequently when  $m = n$  and  $m \neq n$ .

**Guided Chromosome in Initial Population, when  $m = n$**

**Step (a):** Identify all such pairs of  $M(i)$ ,  $1 < i < m$  and  $W(j)$ ,  $1 < j < n$  in CWP for which  $M(i)$ 's best *weighted matching* is with the woman,  $W(j)$  and  $W(j)$ 's best *weighted matching* is with the man,  $M(i)$ . Store all such  $i$  at the  $j^{th}$  index in the population array, *Pop*.

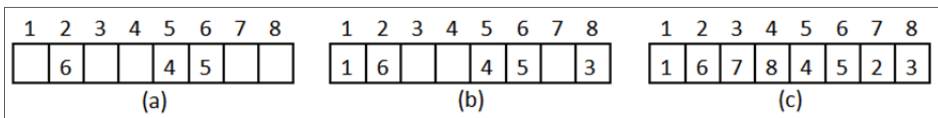
It can be seen in the CWP of Fig. 2 that,  $M(4)$ 's best *weighted matching* is with  $W(5)$  and also  $W(5)$ 's best *weighted matching* is with  $M(4)$ . Population array, *Pop*, storing all such pairing in the CWP of Fig. 2 after the Step (a) is shown in Fig. 10(a).

**Step (b):** One by one identify the pairing for remaining men by getting the best *weighted matching* of a man,  $M(i)$  with a woman  $W(j)$ . Unlike Step (a), here, the best *weighted matching* for the woman,  $W(j)$  will not be with the man,  $M(i)$ . Further, here, we may encounter a scenario where,  $W(j)$  may be the best pair for more than one man. In this scenario,  $W(j)$  is to be paired with such man,  $M(i)$  to whom  $W(j)$  is having maximum *weighted matching*.

In the preceding example, following men are still unpaired after Step (a):  $M(1)$ ,  $M(2)$ ,  $M(3)$ ,  $M(7)$ , and  $M(8)$ . Out of these unpaired men,  $M(3)$ 's best *weighted matching* is with  $W(8)$ , and there is no other man whose best *weighted matching* is with  $W(8)$ . Hence,  $M(3)$  and  $W(8)$  are paired. Further,  $M(1)$ 's and  $M(2)$ 's best *weighted matching* is with  $W(1)$ . Out of  $M(1)$  and  $M(2)$ ,  $W(1)$ 's best *weighted matching* is with  $M(1)$ . Hence,  $M(1)$  and  $W(1)$  are paired and  $M(2)$  remains unpaired after Step (b). Finally,  $M(7)$ 's and  $M(8)$ 's best *weighted matching* is with  $W(5)$  which is already paired with  $M(4)$ , and hence,  $M(7)$  and  $M(8)$  remain unpaired after Step (b).

**Step (c):** Randomly pair unpaired men and women (of the previous step) to generate the chromosome stored into the population array, *Pop*. We call this as a guided chromosome.

As shown in Fig. 10(b), men,  $M(2)$ ,  $M(7)$ , and  $M(8)$  and women,  $W(3)$ ,  $W(4)$ , and  $W(7)$  are unpaired after Step (b) and hence to be randomly paired after Step (c). The guided chromosome finally stored in the population array, *Pop* after the Step (c) is shown in Fig. 10(c).



**Fig. 10** Generation of a guided chromosome, when  $m = n$ .

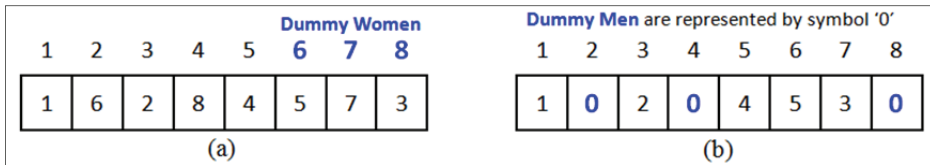
**Guided Chromosome in Initial Population, when  $m \neq n$**

Under this scenario, generate a guided chromosome in the initial population in the same manner as generated for  $m = n$ . Here, instead of actual pairing, some men might be paired with dummy women (if,  $m > n$ ) or some women might be paired with dummy men (if,  $n > m$ ).

The example presented in Fig. 11(a) shows a guided chromosome generated after applying Steps (a), (b), and (c) of the previous section on the CWP of Fig. 2 (considering the women,  $W(6)$ ,  $W(7)$ , and  $W(8)$  as dummy women). Here, Step (a) is resulted into following pairs:  $M(6)$  with  $W(2)$  and  $M(4)$

with  $W(5)$ , whereas, Step (b) makes the pairing between  $M(1)$  and  $W(1)$ . In Step (c), randomly any two men, in the current example we have considered,  $M(5)$  and  $M(8)$  out of unpaired 5 men are to be paired with 2 unpaired women,  $W(3)$  and  $W(4)$ . Finally, all the unpaired men,  $M(2)$ ,  $M(3)$ , and  $M(7)$  are randomly paired with dummy women,  $W(6)$ ,  $W(7)$ , and  $W(8)$  and not to be involved while computing the fitness score.

Similarly, the example presented in Fig. 11(b) shows a guided chromosome generated after applying Steps (a), (b), and (c) of the previous section on the CWP of Fig. 2 (considering the men,  $M(6)$ ,  $M(7)$ , and  $M(8)$  as dummy men). Here, Step (a) is resulted into following pairs:  $M(4)$  with  $W(5)$  and  $M(5)$  with  $W(6)$ , whereas, Step (b) gives the pairing between  $M(1)$  and  $W(1)$ . In Step (c), randomly any two women, in the current example, they are  $W(3)$  and  $W(7)$  out of unpaired 5 women are to be paired with 2 unpaired men,  $M(2)$  and  $M(3)$ . Finally, all the unpaired women,  $W(2)$ ,  $W(4)$ , and  $W(8)$  are randomly paired (represented as symbol '0') with dummy men,  $M(6)$ ,  $M(7)$ , and  $M(8)$  and not to be involved while computing the fitness score.



**Fig. 11.** Generation of the guided chromosomes, when (a)  $m > n$ , (b)  $m < n$ .

**Algorithm 3.1:** *When,  $m = n$*

**Step 1:** Randomly make the pairing between men and women such that they are uniquely paired and generate the set of chromosomes in the initial population. Make sure that one of the chromosomes is a guided chromosome generated using the approach discussed earlier with  $m = n$ .

**Step 2:** Apply Eq. (5) to compute the fitness scores,  $FS$  of the chromosomes in the initial population.

As an example, fitness score of one of the chromosomes presented as the guided chromosome in Fig. 10(c) is computed as  $(85 + 95 + 45 + 89 + 98 + 91 + 56 + 82) / 8$ , i.e.  $FS = 80.12$ .

**Step 3:** Select two parents from the initial population to be involved in the process of mating. Use the Steps 4 and 5 of the Algorithm 2.1 to generate the offsprings after crossover and mutation operations. Compute the fitness score of the two offsprings using Eq. (5) and check their fitness scores with the least fitted chromosome (i.e., having least fitness) in the populace. Select such chromosome to be the part of the populace having better fitness.

**Step 4:** Repeat, Step 3 until any one of the following is not met: (a) no improvement in the population for specified  $X$  iterations or (b) total  $K$  number of generations has been produced.

**Algorithm 3.2:** *When,  $m \neq n$*

**Step 1:** Randomly make the pairing between men and women such that they are uniquely paired and generate the set of chromosomes in the initial population. These pairs might involve, either (a)

$m - n$  dummy men (if,  $m > n$ ) paired with  $m - n$  women besides  $n$  actual pair, or (b)  $n - m$  dummy women (if,  $n > m$ ) paired with  $n - m$  men besides  $m$  actual pair. Make sure that in either of the case,  $m > n$  or  $n > m$ , one of the chromosomes is a guided chromosome generated using the approach discussed earlier with  $m \neq n$ .

**Step 2:** Apply Eq. (6) (when,  $m > n$ ) or Eq. (7) (when,  $m < n$ ) to compute the fitness scores,  $FS$  of the chromosomes in the initial population.

As an example, fitness score of one of the chromosomes presented as the guided chromosome in Fig. 11(a), when,  $m > n$  is computed as  $(85 + 95 + 52 + 89 + 98) / 5$ , i.e.  $FS = 83.8$ , whereas fitness score of one of the guided chromosome presented in Fig. 11(b), when,  $m > n$  is computed as  $(85 + 52 + 98 + 91 + 80) / 5$ , i.e.  $FS = 81.2$ .

**Step 3:** Use the Steps 3, 4 and 5 of the Algorithm 2.2 to select the two parents for mating and generate the off-springs after crossover and mutation operations. Compute the fitness score of the two off-springs using Eq. (6) (when,  $m > n$ ) or Eq. (7) (when,  $m < n$ ) and check their fitness scores with the least fitted chromosome (i.e., having least fitness) in the populace. Select such chromosome to be the part of the populace having better fitness.

**Step 4:** Repeat, Step 3 until any one of the following is not met: (a) no improvement in the population for specified  $X$  iterations or (b) total  $K$  number of generations has been produced.

## 4. Experiments and Results

In this section, we present the implementation details, data sets, results, and discussion over obtained results for the algorithms proposed in Section 3.

Greedy and GA based algorithms presented in the previous section had been implemented in Python besides the Gale-Shapley algorithm. Various experiments had been conducted to test and analyze the performance of the implemented algorithms. These experiments were conducted over the dataset created by crawling user's (men and women) profiles and expectations from matrimonial websites. Although the objective is to maximize the profit of a single matrimonial website, we crawled the users' profiles and expectations from various matrimonial websites to perform the experiments over a wide and diversified dataset of mixed group.

### 4.1 Dataset

As discussed earlier, we created a dataset of a mixed group of user's profile and expectations. This included the profiles of educated/uneducated individuals, long/medium/short height users, employed/unemployed/self-employed persons, etc. It also included the mixed expectations in the partners' (of opposite gender) profile, viz. looking for educated/uneducated partner, high/moderate salary, vegetarian/non-vegetarian, etc. To maintain such diversity in the experimental dataset, we crawled the user's profile and their expectations from two matrimonial websites, viz. [www.jeevansathi.com](http://www.jeevansathi.com), [www.bharatmatrimony.com](http://www.bharatmatrimony.com). In total, details (profile and expectations) of 1000 men and 1000 women were crawled from these websites. Crawling from different sources resulted in different format/structure of data and unequal set of features (viz. religion, height, salary, etc.) in individuals profile and

expectations. Therefore restructuring or cleaning of the crawled data from different sources had been done. We considered only such features (either in profile or in expectations) which were common in all data sources. In total five features had been used after the restructuring or cleaning process.

Further, we stored these restructured data into four matrices as follows:  $P_M$  (profile of men),  $E_M$  (expectations of men in his partner's profile),  $P_W$  (profile of women), and  $E_W$  (expectations of women in her partner's profile). Here,  $P_M(i, j)$  stores the numeric value scaled between -1 and 5 representing the presence (scaled between 0 and 5) / not presented (scaled to -1) of the  $j^{th}$  feature in the profile of the  $i^{th}$  man. For example, let,  $j = 2$  represents a feature salary, then  $P_M(1, 2) = 0$  is the representation of very bad earnings for the man,  $M(1)$ , whereas  $P_M(2, 2) = 5$  is the representation of very good earnings for the man,  $M(2)$ , and  $P_M(3, 2) = -1$  represents the unavailable or undisclosed information about the salary of  $M(3)$  at the matrimonial website. Similarly,  $P_W(i, j)$  represents the presence of the  $j^{th}$  feature in the profile of the  $i^{th}$  woman. Further,  $E_M(i, j)$  stores the numeric value (scaled between 0 and 5) for the  $j^{th}$  expected feature by  $M_i$  in his partner's profile. Here,  $E_M(1, 2) = 4$  suggests that  $M(1)$  is expecting a partner whose income/earning is good. Similarly,  $E_W(i, j)$  stores the numeric value for the  $j^{th}$  expected feature by  $W_i$  in her partner's profile.

Matrices discussed in previous steps had been used to create the weighted preferences, MWP and WWP as follows:

$$MWP(i, k) = \frac{\sum_{j=1}^f (x(j) \times \text{dist}(E_M(i, j), P_W(k, j)))}{t} \times 100 \quad (8)$$

where,  $i$  and  $k$  are the  $i^{th}$  and  $k^{th}$  man and woman respectively;  $f$  is the count of features;  $t$  is the count of expected features (or expectations) for the  $i^{th}$  man in his partner's profile, i.e.  $t$  is the count for which  $E_M(i, j) \geq 0$ ;  $\text{dist}$  is the distance function and returns 1 if,  $E_M(i, j) \leq P_W(k, j)$ , where,  $E_M(i, j) \neq -1$ , else returns the Euclidean distance between  $E_M(i, j)$  and  $P_W(k, j)$  scaled between 0 and 1. Further,  $x(j)$  is the weight (between 0 and 1) associated with each feature such that  $\sum_{j=1}^t x(j)$  is 1.

$$WWP(i, k) = \frac{\sum_{j=1}^f (x(j) \times \text{dist}(E_W(i, j), P_M(k, j)))}{t} \times 100 \quad (9)$$

Here, the distance function  $\text{dist}$  returns 1, if  $E_W(i, j) \leq P_M(k, j)$ , where  $E_W(i, j) \neq -1$ , otherwise returns Euclidian distance between  $E_W(i, j)$  and  $P_M(k, j)$  scaled between 0 and 1.

While computing MWP and WWP, we had considered fixed weight,  $x(j)$  for each feature. However, it can be tuned in future for further observations in the performance of proposed algorithms. Further, in context of the Indian subcontinent where inter-religion marriage is usually not in practice, we marked the *weighted matching* between  $i^{th}$  man and  $j^{th}$  woman, i.e.  $MWP(i, j)$  as 0 irrespective of the matching of other features, if any one of the two did not prefer inter-religion marriage. Similarly, under such circumstances, we made  $WWP(i, j)$  as 0. Finally, CWP was calculated by taking the mean of MWP and WWP.

## 4.2 Results and Discussion

In this section, we one by one present the results and other details for the algorithms proposed in Section 3. Performance of the proposed algorithms had been evaluated in terms of the achieved

MAWM (computed using Eq. (4)) over the CWP created in Section 4.1. Let us call this CWP as  $CWP_1$ . We run the Greedy based algorithm once, whereas, each of the GA based algorithms was run  $p$  times (10 runs) to observe the variations in the achieved MAWM. Each run of the GA based algorithms involved following in-order: population set having 50 chromosomes; computation of the fitness scores of the chromosomes using the CWP created in Section 4.1; and 10,000 iterations of selection, crossover and mutation operations to generate the offsprings.

The performance of the Greedy based approach presented in Algorithm 1 had been tested over the CWP created in Section 4.1, i.e.  $CWP_1$  (having *weighted matching* of 1000 men and 1000 women). We ran the algorithm once and successfully identified 1000 unique pairings of men and women. In the single run of Algorithm 1, the achieved MAWM was 43.64.

Further, the CWP created in Section 4.1 had been used to compute the fitness scores of the chromosomes in each of the 10 runs for Algorithm 2.1. Each run started with random selection of two chromosomes from the initial population to generate the first off-spring, i.e. in total 20 chromosomes had been selected for this purpose. Out of 20 initially selected chromosomes, the highest fitness score was 43.07, whereas the mean $\pm$ standard deviation of the initially selected 20 chromosomes was 42.62 $\pm$ 0.3. Further, in each run we successfully identified the 1000 unique pairings of men and women. Out of 10 MAWMs (one MAWM per run), the best achieved MAWM was 62.71, whereas the mean $\pm$ standard deviation of the achieved MAWMs was 62.44 $\pm$ 0.37.

A similar experimental procedure had been used to evaluate the performance of Algorithm 2.2. Here, we created another CWP (let us call,  $CWP_2$ ) by randomly removing 200 women from the CWP created in Section 4.1, i.e.  $CWP_2$  involved 1000 men and 800 women.  $CWP_2$  had been used to compute the fitness score of the chromosomes in each of the 10 runs of the Algorithm 2.2. The highest fitness score out of the 20 initially selected chromosomes, their mean $\pm$ standard deviation, out of the 10 runs the best achieved MAWM and its mean $\pm$ standard deviation have been shown in Table 1. We also conducted the experiments to observe the performance of the Algorithm 2.2 under the scenario  $m < n$ . These experiments had been conducted over the CWP created by randomly removing 200 men from the CWP created in Section 4.1. Let us call this CWP as  $CWP_3$ . Table 1 also depicts various observations (viz. best achieved MAWM out of 10 runs and its mean $\pm$ standard deviation, etc.) related to the conducted experiments under this scenario.

Next set of experiments had been conducted to evaluate the performance of Algorithm 3 under the scenarios,  $m = n$ ,  $m > n$ , and  $m < n$  using  $CWP_1$ ,  $CWP_2$ , and  $CWP_3$ , respectively. Unlike Algorithm 2, here initial populations were created with 50 chromosomes, where one of the chromosomes was generated in the guided manner. The highest fitness score out of the 10 chromosomes generated in the guided manner for 10 runs of the Algorithm 3.1, their mean $\pm$ standard deviation, out of the 10 runs the best achieved MAWM and its mean $\pm$ standard deviation have been shown in Table 1. Similar observational details related with 10 runs of Algorithm 3.2 (when  $m > n$ ) and 10 runs of Algorithm 3.2 (when  $m < n$ ) have been presented in Table 1.

Lastly, we conducted the experiments to evaluate the performance of the Gale-Shapley algorithm over same data set, i.e. MWP and WWP which had been used to create  $CWP_1$ . Here, men preference list was created for each man by replacing highest *weighted matching* in the MWP as preference 1 (i.e. most preferred woman for that man), second highest *weighted matching* as preference 2 and so on. Similarly, we identified the women preference list for each woman.



**Table 1.** Performance details of various algorithms

Algorithm	Count of man ( $m$ ), woman ( $n$ )	FS of the guided chromosome (used for Algorithm 3) or first selected chromosome (used for Algorithm 2) in the initial population of 50 chromosomes		MAWM	Mean $\pm(\sqrt{\sigma^2/p})$ for $p = 10$ runs
		Best FS of the chromosome (guided/first selected) out of $p = 10$ runs	Mean $\pm(\sqrt{\sigma^2/p})$ for $p = 10$ runs		
1 ( $m = n$ )	1000, 1000	NA	NA	43.64	NA
2.1 ( $m = n$ )	1000, 1000	43.07	42.62 $\pm$ 0.31	62.73	62.45 $\pm$ 0.38
2.2 (when $m > n$ )	1000, 800	41.54	41.10 $\pm$ 0.49	63.85	63.15 $\pm$ 0.43
2.2 (when $m < n$ )	800, 1000	42.25	41.13 $\pm$ 0.84	68.32	67.91 $\pm$ 0.28
3.1 ( $m = n$ )	1000, 1000	70.70	70.23 $\pm$ 0.28	72.93	72.81 $\pm$ 0.10
3.2 (when $m > n$ )	1000, 800	71.12	70.97 $\pm$ 0.11	73.66	73.38 $\pm$ 0.19
3.2 (when $m < n$ )	800, 1000	73.41	73.32 $\pm$ 0.10	79.81	79.55 $\pm$ 0.18
Gale-Shapley ( $m = n$ )	1000, 1000	NA	NA	35.58	NA

Each GA based algorithm had been run ten times (i.e.,  $p = 10$ ) and each run involved 10000 iterations, i.e.,  $K = 10000$ . FS=fitness score, NA=not applicable.

Further, these preference lists (preferences of 1000 men for 1000 women and vice versa) were given as inputs to the Gale-Shapley algorithm. The Gale-Shapley algorithm too successfully identified the unique pairing of each man with a woman. The *weighted matchings* against these pairs had been averaged out to compute the MAWM. The achieved MAWM by Gale-Shapley algorithm was 35.58.

Table 1 consolidates the MAWM achieved by the algorithms presented in this paper including the Gale-Shapley algorithm. As seen in Table 1, out of the four algorithms (including Gale and Shapley) applicable for the scenario, where  $m = n$ , Algorithm 3.1 (i.e. GA based approach with a guided population) had shown the best performance to maximize the profit of matrimonial websites in terms of achieved MAWM. One of the reasons behind this might be the guided chromosome in the initial population which ensured that the global/final fitness score of a chromosome is never less than the fitness score of the guided chromosome. Further, as seen from the Table 1, variations (standard deviation) in the achieved MAWM is negligible for all the 10 runs of each of the GA based algorithm, where  $m = n$ .

Similarly, as seen in Table 1, Algorithm 3.2 performed better than Algorithm 2.2 under the scenario, where  $m < n$ . Both algorithms utilized the capability of GA, however, the reason behind the better performance of Algorithm 3.2 might be the guided chromosome in the initial population. Finally, the same reason might be applicable for the better performance of Algorithm 3.2 than Algorithm 2.2 for the scenario, where  $m > n$ . Further, 10 runs of these algorithms had shown little variations (standard deviations) in the achieved MAWM.

In each of the conducted experiment for Algorithm 2 (randomly generated chromosomes), significant improvement has been observed in the fitness scores of the initially selected chromosome and the best chromosome after 10000 iterations of an experiment. Mean of the 10 fitness scores of the initially selected chromosomes corresponding to the 10 experiments conducted for Algorithm 2.1 is 42.62 $\pm$ 0.31. This mean is improved significantly to 62.45 $\pm$ 0.38 for the best chromosome after 10000 iterations of each experiment. It is evident from Table 1 that significant improvement has been observed for Algorithm 2.2 (when,  $m > n$ ) and Algorithm 2.2 (when,  $m < n$ ). Further, mean of the 10 fitness scores of the guided chromosomes corresponding to the 10 experiments conducted for Algorithm 3.1 is 70.23 $\pm$ 0.28. This mean is improved significantly to 72.81 $\pm$ 0.1 for the best chromosome

after 10000 iterations of each experiment. Similar improvements have been observed in the experiments conducted for Algorithm 3.2 (when,  $m > n$ ) and Algorithm 3.2 (when,  $m < n$ ).

## 5. Conclusion

In this paper we presented a new perspective to SMP in profit maximization of matrimonial websites. Instead of qualitative preference list, we defined the quantitative/weighted preferences computed using profile (personal details) and expectations (in partner of opposite gender) of the users registered on matrimonial websites. The objective of all the algorithms presented in this paper was to make such pairs/matching of men and women which maximizes the profit (in terms of MAWM) of matrimonial websites. Greedy and Genetic algorithm based algorithms have been proposed in this paper to solve the problem in different scenarios, viz. (a)  $m$  (count of men) =  $n$  (count of women); (b)  $m > n$ ; and (c)  $m < n$ . As discussed in Section 4, the set of algorithms (Algorithm 3) outperformed other algorithms (viz. Greedy based, and GA based where chromosomes are randomly generated) including Gale-Shapley algorithm under the scenario where,  $m = n$ .

As the future extension, the new perspective presented in this paper may also be applied to some other applications of bipartite matching viz. student project allocation, hospital resident problem, etc. As discussed in Section 4, another extension of current work is to train/tune the weights associated with the features for further observations.

## Acknowledgement

We would like to thank Yamuna P. Shukla for his assistance while writing the paper after completion of the presented work.

## References

- [1] D. Gale, "The two-sided matching problem: origin, development and current issues," *International Game Theory Review*, vol. 3, no. 2-3, pp. 237-252, 2001.
- [2] D. F. Manlove and G. O'Malley, "Student-project allocation with preferences over projects," *Journal of Discrete Algorithms*, vol. 6, no. 4, pp. 553-560, 2008.
- [3] J. Wu, "Stable matching beyond bipartite graphs," in *Proceedings of 2016 IEEE International Parallel and Distributed Processing Symposium Workshops*, Chicago, IL, 2016, pp. 480-488.
- [4] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9-15, 1962.
- [5] D. G. McVitie and L. B. Wilson, "The stable marriage problem," *Communications of the ACM*, vol. 14, no. 7, pp. 486-490, 1971.
- [6] R. W. Irving, P. Leather, and D. Gusfield, "An efficient algorithm for the "optimal" stable marriage," *Journal of the ACM*, vol. 34, no. 3, pp. 532-543, 1987.
- [7] I. Damianidis, "The stable marriage problem: optimizing different criteria using genetic algorithms," Master's thesis, University of Borås, Sweden, 2011

- [8] K. Iwama and S. Miyazaki, "A survey of the stable marriage problem and its variants," in *Proceedings of International Conference on Informatics Education and Research for Knowledge-Circulating Society*, Kyoto, Japan, 2008, pp. 131-136.
- [9] M. S. Pini, F. Rossi, B. Venable, and T. Walsh, "Stable marriage problems with quantitative preferencesm" 2010 [Online]. Available: <https://arxiv.org/abs/1007.5120>.
- [10] M. Baiou and M. Balinski, "Many-to-many matching: stable polyandrous polygamy (or polygamous polyandry)," *Discrete Applied Mathematics*, vol. 101, no. 1-3, pp. 1-12, 2000.
- [11] I. Bello and S. Lianshuan, "Genetic algorithm for the stable marriage problem (SMP)," *International Journal of Science and Research*, vol. 5, no. 6, pp. 939-944, 2016.
- [12] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Stability, optimality and manipulation in matching problems with weighted preferences," *Algorithms*, vol. 6, no. 4, pp. 782-804, 2013.
- [13] Membership option of BharatMatrimony [Online]. Available: <http://www.bharatmatrimony.com/payments/paymentoptions.php>
- [14] A. L. Corcoran and R. L. Wainwright, "Using LibGA to develop genetic algorithms for solving combinatorial optimization problems," *Practical Handbook of Genetic Algorithms*. Boca Laton, FL: CRC Press, 1995, pp. 143-172.
- [15] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Natural Computing*, vol. 7, no. 1, pp. 109-124, 2008.
- [16] Category:Stable sorts [Online]. Available: [https://en.wikipedia.org/wiki/Category:Stable\\_sorts](https://en.wikipedia.org/wiki/Category:Stable_sorts).



**Aniket Bhatnagar** <https://orcid.org/0000-0001-8259-1577>

He was one of the undergraduate students at Jaypee Institute of Information Technology, Noida, India and completed his under-graduation (B.Tech. in Computer Science Engineering) in 2017. His research interest includes the evolutionary algorithms, combinatorial optimization problems, and machine learning. He actively participates in several online/onsite competitive programming challenges and secured good ranks.



**Varun Gambhir** <https://orcid.org/0000-0002-8537-6853>

He completed his B.Tech. in Computer Science Engineering from Jaypee Institute of Information Technology, Noida, India in 2017. His research interest includes the machine learning and evolutionary algorithms for combinatorial optimization problems. Besides academics he is actively involved in various online/onsite competitive programming challenges as participant and secured good ranks.



**Manish Kumar Thakur** <https://orcid.org/0000-0003-2479-1540>

He is currently working as Assistant Professor (Senior Grade) in the Department of CSE&IT at Jaypee Institute of Information Technology (JIIT), Noida. He completed his Ph.D. in 2014 from JIIT, Noida and his M.Tech in Computer Science from Birla Institute of Technology, Mesra, Ranchi, India. His research interest includes, evolutionary algorithms, graph algorithms, parallel and distributed computing, video processing, and machine learning.