



**HAL**  
open science

# Optimizing a Generalized Gini Index in Stable Marriage Problems: NP-Hardness, Approximation and a Polynomial Time Special Case

Hugo Gilbert, Olivier Spanjaard

► **To cite this version:**

Hugo Gilbert, Olivier Spanjaard. Optimizing a Generalized Gini Index in Stable Marriage Problems: NP-Hardness, Approximation and a Polynomial Time Special Case. *Algorithmica*, 2019, 10.1007/s00453-019-00550-3 . hal-02060866

**HAL Id: hal-02060866**

**<https://hal.science/hal-02060866v1>**

Submitted on 10 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Optimizing a Generalized Gini Index in Stable Marriage Problems: NP-Hardness, Approximation and a Polynomial Time Special Case

Hugo Gilbert · Olivier Spanjaard

the date of receipt and acceptance should be inserted later

**Abstract** This paper deals with fairness in stable marriage problems. The idea studied here is to achieve fairness thanks to a Generalized Gini Index (GGI), a well-known criterion in inequality measurement, that includes both the egalitarian and utilitarian criteria as special cases. We show that determining a stable marriage optimizing a GGI criterion of agents' disutilities is an NP-hard problem. We then provide a polynomial time 2-approximation algorithm in the general case, as well as an exact algorithm which is polynomial time in the case of a constant number of non-zero weights parametrizing the GGI criterion.

**Keywords** Stable marriage problem · Fairness · Generalized Gini index · Complexity

## 1 Introduction

Since the seminal work of Gale and Shapley [1962] on stable marriages, matching problems under preferences have been extensively studied both by economists and computer scientists. These problems involve two sets of agents (also called individuals in the sequel) that should be matched with each other while taking agents' preferences into account. The results obtained in the field have a tremendous number of applications, among which the National Resident Matching Program in the US (for allocating junior doctors to hospitals), the teacher allocation in France (for allocating newly tenured teachers to schools) or the allocation of lawyers in Germany (for assigning graduating lawyers to legal internship positions). For an overview of the applications of matching models under preferences, the interested reader can refer to a recent book chapter on this topic [Biró, 2017].

The *stable marriage problem* involves  $n$  men and  $n$  women, each of whom ranks the members of the opposite sex in order of preference. The goal is to find a *stable* matching, i.e., a matching between men and women such that there is no man and woman that prefer each other to their current match. Gale and Shapley [1962] provided an algorithm that computes a stable marriage. However, it is well-known that this algorithm favours one group (men or women, according to the way the algorithm is applied) over the other.

---

H. Gilbert · O. Spanjaard  
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France  
E-mail: {hugo.gilbert,olivier.spanjaard}@lip6.fr

We are interested here in *fair* stable marriage algorithms, i.e., in procedures favouring stable marriages that fairly share dissatisfactions –also called disutilities– among individuals (irrespective of their sex), the dissatisfaction being defined for each woman (resp. man) as a function of the rank, in order of preferences, of the man (resp. woman) to whom she is paired with. Given the vector of individuals’ dissatisfactions induced by a matching, there are several ways of formalizing the notion of “fairness”. We mean here by fair stable marriage that the vector of individuals’ dissatisfactions should be well-balanced. For example, consider the following instance of the stable marriage problem.

**Example 1** *The instance consists of 10 men  $\{m_1, \dots, m_{10}\}$  and women  $\{w_1, \dots, w_{10}\}$  with the following preferences, where  $i \succ_k^m j$  (resp.  $i \succ_k^w j$ ) means that  $m_k$  (resp.  $w_k$ ) prefers  $w_i$  to  $w_j$  (resp.  $m_i$  to  $m_j$ ):*

$$\begin{aligned}
m_1 : & 1 \succ_1^m 2 \succ_1^m 3 \succ_1^m 4 \succ_1^m 5 \succ_1^m 6 \succ_1^m 7 \succ_1^m 8 \succ_1^m 9 \succ_1^m 10 \\
m_2 : & 2 \succ_2^m 1 \succ_2^m 3 \succ_2^m 4 \succ_2^m 5 \succ_2^m 6 \succ_2^m 7 \succ_2^m 8 \succ_2^m 9 \succ_2^m 10 \\
m_3 : & 3 \succ_3^m 1 \succ_3^m 2 \succ_3^m 4 \succ_3^m 5 \succ_3^m 6 \succ_3^m 7 \succ_3^m 8 \succ_3^m 9 \succ_3^m 10 \\
m_4 : & 7 \succ_4^m 1 \succ_4^m 2 \succ_4^m 3 \succ_4^m 4 \succ_4^m 5 \succ_4^m 6 \succ_4^m 8 \succ_4^m 9 \succ_4^m 10 \\
m_5 : & 6 \succ_5^m 1 \succ_5^m 2 \succ_5^m 3 \succ_5^m 4 \succ_5^m 5 \succ_5^m 7 \succ_5^m 8 \succ_5^m 9 \succ_5^m 10 \\
m_6 : & 4 \succ_6^m 1 \succ_6^m 2 \succ_6^m 3 \succ_6^m 4 \succ_6^m 5 \succ_6^m 6 \succ_6^m 7 \succ_6^m 8 \succ_6^m 9 \succ_6^m 10 \\
m_7 : & 5 \succ_7^m 1 \succ_7^m 2 \succ_7^m 3 \succ_7^m 4 \succ_7^m 5 \succ_7^m 6 \succ_7^m 7 \succ_7^m 8 \succ_7^m 9 \succ_7^m 10 \\
m_8 : & 8 \succ_8^m 4 \succ_8^m 5 \succ_8^m 6 \succ_8^m 7 \succ_8^m 8 \succ_8^m 9 \succ_8^m 10 \succ_8^m 1 \succ_8^m 2 \succ_8^m 3 \succ_8^m 10 \\
m_9 : & 10 \succ_9^m 4 \succ_9^m 5 \succ_9^m 6 \succ_9^m 7 \succ_9^m 8 \succ_9^m 9 \succ_9^m 10 \succ_9^m 1 \succ_9^m 2 \succ_9^m 3 \succ_9^m 8 \\
m_{10} : & 9 \succ_{10}^m 4 \succ_{10}^m 5 \succ_{10}^m 6 \succ_{10}^m 7 \succ_{10}^m 8 \succ_{10}^m 9 \succ_{10}^m 10 \succ_{10}^m 1 \succ_{10}^m 2 \succ_{10}^m 3 \succ_{10}^m 10
\end{aligned}$$

$$\begin{aligned}
w_1 : & 1 \succ_1^w 2 \succ_1^w 3 \succ_1^w 4 \succ_1^w 5 \succ_1^w 6 \succ_1^w 7 \succ_1^w 8 \succ_1^w 9 \succ_1^w 10 \\
w_2 : & 1 \succ_2^w 2 \succ_2^w 3 \succ_2^w 4 \succ_2^w 5 \succ_2^w 6 \succ_2^w 7 \succ_2^w 8 \succ_2^w 9 \succ_2^w 10 \\
w_3 : & 1 \succ_3^w 2 \succ_3^w 3 \succ_3^w 4 \succ_3^w 5 \succ_3^w 6 \succ_3^w 7 \succ_3^w 8 \succ_3^w 9 \succ_3^w 10 \\
w_4 : & 1 \succ_4^w 2 \succ_4^w 3 \succ_4^w 4 \succ_4^w 5 \succ_4^w 6 \succ_4^w 7 \succ_4^w 8 \succ_4^w 9 \succ_4^w 10 \\
w_5 : & 1 \succ_5^w 2 \succ_5^w 3 \succ_5^w 4 \succ_5^w 5 \succ_5^w 6 \succ_5^w 7 \succ_5^w 8 \succ_5^w 9 \succ_5^w 10 \\
w_6 : & 1 \succ_6^w 2 \succ_6^w 3 \succ_6^w 4 \succ_6^w 5 \succ_6^w 6 \succ_6^w 7 \succ_6^w 8 \succ_6^w 9 \succ_6^w 10 \\
w_7 : & 1 \succ_7^w 2 \succ_7^w 3 \succ_7^w 4 \succ_7^w 5 \succ_7^w 6 \succ_7^w 7 \succ_7^w 8 \succ_7^w 9 \succ_7^w 10 \\
w_8 : & 2 \succ_8^w 10 \succ_8^w 8 \succ_8^w 7 \succ_8^w 1 \succ_8^w 3 \succ_8^w 4 \succ_8^w 5 \succ_8^w 6 \succ_8^w 9 \\
w_9 : & 1 \succ_9^w 2 \succ_9^w 3 \succ_9^w 4 \succ_9^w 5 \succ_9^w 6 \succ_9^w 7 \succ_9^w 8 \succ_9^w 9 \succ_9^w 10 \\
w_{10} : & 1 \succ_{10}^w 2 \succ_{10}^w 3 \succ_{10}^w 4 \succ_{10}^w 5 \succ_{10}^w 6 \succ_{10}^w 7 \succ_{10}^w 8 \succ_{10}^w 9 \succ_{10}^w 10
\end{aligned}$$

*The stable marriages in this instance are:*

$$\begin{aligned}
\mathbf{x}^1 : & \{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_7), (m_5, w_6), (m_6, w_4), (m_7, w_5), \\
& (m_8, w_8), (m_9, w_{10}), (m_{10}, w_9)\} \\
\mathbf{x}^2 : & \{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_7), (m_5, w_6), (m_6, w_5), (m_7, w_4), \\
& (m_8, w_8), (m_9, w_{10}), (m_{10}, w_9)\} \\
\mathbf{x}^3 : & \{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_6), (m_5, w_7), (m_6, w_4), (m_7, w_5), \\
& (m_8, w_8), (m_9, w_{10}), (m_{10}, w_9)\} \\
\mathbf{x}^4 : & \{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_6), (m_5, w_7), (m_6, w_5), (m_7, w_4), \\
& (m_8, w_8), (m_9, w_{10}), (m_{10}, w_9)\} \\
\mathbf{x}^5 : & \{(m_1, w_1), (m_2, w_2), (m_3, w_3), (m_4, w_6), (m_5, w_7), (m_6, w_5), (m_7, w_4), \\
& (m_8, w_{10}), (m_9, w_9), (m_{10}, w_8)\}
\end{aligned}$$

*where a pair  $(m_i, w_j)$  means that  $m_i$  and  $w_j$  are matched.*

If one assumes that the dissatisfaction of an individual is equal to the rank of the partner in his/her preference list, then the dissatisfactions induced by the previous stable marriages are:

matching	vector of dissatisfactions	sum of dissatisfactions	max of dissatisfactions
$\mathbf{x}^1$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 7, 7, 7, 7, 3, 4, 10)	61	10
$\mathbf{x}^2$	(1, 1, 1, 1, 1, 5, 5, 1, 1, 1, 1, 2, 3, 4, 4, 7, 7, 3, 4, 10)	63	10
$\mathbf{x}^3$	(1, 1, 1, 5, 5, 1, 1, 1, 1, 1, 2, 3, 7, 7, 4, 4, 3, 4, 10)	63	10
$\mathbf{x}^4$	(1, 1, 1, 5, 5, 5, 5, 1, 1, 1, 1, 2, 3, 4, 4, 4, 4, 3, 4, 10)	65	10
$\mathbf{x}^5$	(1, 1, 1, 5, 5, 5, 5, 5, 5, 5, 1, 2, 3, 4, 4, 4, 4, 2, 3, 9)	74	9

where the  $i^{\text{th}}$  component of the vector is the dissatisfaction of  $m_i$  for  $i \in \{1, \dots, 10\}$ , and of  $w_{i-10}$  for  $i \in \{11, \dots, 20\}$ .

In this instance, the matching  $\mathbf{x}^4$  can be considered as inducing a *well-balanced* vector of dissatisfactions. The matchings  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  and  $\mathbf{x}^3$  indeed favour more some individuals (the men in this case) than others, while matching  $\mathbf{x}^5$  yields quite high dissatisfactions for numerous agents. The matching  $\mathbf{x}^4$  is therefore a good compromise between the *utilitarian* and the *egalitarian* viewpoints, where the utilitarian viewpoint aims at minimizing the sum of dissatisfactions while the egalitarian viewpoint aims at minimizing the dissatisfaction of the worst off individual. Both the utilitarian and egalitarian approaches have been advocated for promoting fairness in the stable marriage problem [Gusfield, 1987; Gusfield and Irving, 1989]. Other approaches aim at treating equally men and women, by minimizing the absolute difference between the total dissatisfactions of the two groups (*sex-equal* stable marriage problem [Kato, 1993; McDermid and Irving, 2014]) or by minimizing the maximum total dissatisfaction between the two groups (*balanced* stable marriage problem [Manlove, 2013]). However, note that, in the instance of Example 1, all these criteria favour either  $\mathbf{x}^1$  (utilitarian) or  $\mathbf{x}^5$  (egalitarian, sex-equal, balanced). Finally, there exists another type of approach, that is not based on assigning scores to marriages. In a first step, for each man, one lists all his possible matches in a stable marriage, in order of his preferences (this list includes as many elements as there are feasible stable marriages). In a second step, each man is matched with the median woman in the list. This procedure yields a stable marriage, which is called *median* stable marriage [Teo and Sethuraman, 1998; Cheng, 2010]. In the instance of the example, the median stable marriage is  $\mathbf{x}^4$ . Nevertheless, in this article, we focus on determining a fair stable marriage by using a *scoring rule*.

In social choice theory, a scoring rule assigns a score to each alternative by summing the scores given by every individual over the alternative. This summation principle ensures that all individuals contribute equally to the score of an alternative. An alternative is usually a candidate in an election, but it can also be an element of a combinatorial domain. For instance, in proportional representation problems [Procaccia *et al.*, 2008], where one aims at electing a committee, every feasible committee is an alternative. In the setting of stable marriage problems, every stable marriage is an alternative and the utilitarian approach is clearly a scoring rule where each individual evaluates a stable marriage by the rank of his/her match. An interesting extension of the class of scoring rules is the class of *rank dependent scoring rules* [Goldsmith *et al.*, 2014], where, instead of limiting the aggregation to a summation operation, the scores are aggregated by taking into account their ranks in the ordered list of scores. As emphasized by Goldsmith *et al.* [2014], rank dependent scoring rules can be used to favour fairness by imposing some conditions on their parameters. A well known class of rank-dependent scoring rules in inequality measurement are the *Generalized Gini Indices* (GGI) [Weymark, 1981]. Furthermore, this class of rank dependent

scoring rules circumvents both the utilitarian and egalitarian criteria. Their optimization on combinatorial domains have been studied in several settings (often under the name of *Ordered Weighted Averages*): assignment problems [Lesca *et al.*, 2018], proportional representation [Elkind and Ismaili, 2015], resource allocation [Heinen *et al.*, 2015]. To the best of our knowledge, the problem of determining a GGI optimal stable marriage has not been studied yet. This is precisely the purpose of the present work.

The paper is organized as follows. In Section 2, we introduce notations and we formally define the GGI stable marriage problem studied here. Then, in Section 3, we prove that it is NP-hard to determine an optimal stable marriage according to a GGI criterion applied to agents' disutilities. In Section 4, we provide a polynomial time 2-approximation algorithm. Finally, in Section 5, we establish a parametrized complexity result with respect to a GGI-specific parameter.

## 2 The GGI Stable Marriage Problem

Let  $\mathcal{M} = \{m_1, \dots, m_n\}$  denote the set of men, and  $\mathcal{W} = \{w_1, \dots, w_n\}$  the set of women. As in Example 1, for each  $m_k$  (resp.  $w_k$ ), a preference relation  $\succ_k^m$  (resp.  $\succ_k^w$ ) is defined on  $\mathcal{W}$  (resp.  $\mathcal{M}$ ), where  $i \succ_k^m j$  (resp.  $i \succ_k^w j$ ) means that  $m_k$  (resp.  $w_k$ ) prefers  $w_i$  to  $w_j$  (resp.  $m_i$  to  $m_j$ ). We denote by  $\text{rk}(m_i, w_j)$  the rank of woman  $w_j$  in the preference order of man  $m_i$ , and similarly for  $\text{rk}(w_j, m_i)$ .

A solution of a stable marriage problem is a matching represented by a binary matrix  $\mathbf{x}$ , where  $x_{ij} = 1$  means that  $m_i$  is matched with  $w_j$ . A matching  $\mathbf{x}$  induces a matching function  $\mu_{\mathbf{x}}$  defined by  $w_j = \mu_{\mathbf{x}}(m_i)$  and  $m_i = \mu_{\mathbf{x}}(w_j)$  if  $x_{ij} = 1$ . In a perfect matching (called indifferently matching or marriage from now on), every man (resp. woman) is matched with a different woman (resp. man). More formally, a matching is defined by:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in \{1, \dots, n\} \quad (1)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \quad (2)$$

A matching is said to be *stable* if there exists no man and woman who prefer each other to their current partner. More formally, a perfect matching is stable if the following constraints hold [Vande Vate, 1989]:

$$x_{ij} + \sum_{j' \succ_i^m j} x_{ij'} + \sum_{i' \succ_j^w i} x_{i'j} \geq 1 \quad \forall (i, j) \in \{1, \dots, n\}^2 \quad (3)$$

The set of stable marriages, i.e. binary matrices  $\mathbf{x}$  such that constraints 1, 2 and 3 hold, is denoted by  $\mathcal{X}$ . In their seminal paper, Gale and Shapley [1962] states that there always exists at least one stable marriage, which can be computed in  $O(n^2)$ .

The Gale-Shapley algorithm is based on a sequence of proposals from men to women. Each man proposes to the women following his preference order, pausing when a woman agrees to be matched with him but continuing if his proposal is rejected. When a woman receives a proposal, she rejects it if she already has a better proposal according to her preferences. Otherwise, she agrees to hold it for consideration and rejects any former proposal that she might had. Such a sequence of proposals always leads to a stable marriage called man-optimal stable marriage and denoted by  $\mathbf{x}^m$  (if the role of men and women is reversed, we obtain the woman-optimal stable marriage denoted by  $\mathbf{x}^w$ ). In the man-optimal stable marriage, each man has the best partner, and each woman has the worst partner, that is possible in any stable marriage. Contrarily, in the woman-optimal stable marriage, each

woman has the best partner, and each man has the worst partner, that is possible in any stable marriage.

Two important properties of the Gale-Shapley algorithm are that:

- if  $m$  proposes to  $w$ , then there is no stable marriage in which  $m$  has a better match than  $w$ .
- if  $m$  proposes to  $w$ , then there is no stable marriage in which  $w$  has a worse match than  $m$ .

These properties justify the notion of preference shortlists obtained through the Gale-Shapley algorithm by removing any man  $m$  from a woman  $w$ 's preference list and vice-versa, when  $w$  receives a proposal from a man she prefers to  $m$ . Note that the shortlists that are obtained at the end of the algorithm do not depend on the order in which the proposals are made.

**Example 2** For instance, with the preferences of Example 1, the Gale-Shapley algorithm leads to the following shortlists:

$$\begin{aligned}
m_1 &: 1 \succ_1^m 2 \succ_1^m 3 \succ_1^m 4 \succ_1^m 5 \succ_1^m 6 \succ_1^m 7 \succ_1^m 9 \succ_1^m 10 \\
m_2 &: 2 \succ_2^m 3 \succ_2^m 4 \succ_2^m 5 \succ_2^m 6 \succ_2^m 7 \succ_2^m 8 \succ_2^m 9 \succ_2^m 10 \\
m_3 &: 3 \succ_3^m 4 \succ_3^m 5 \succ_3^m 6 \succ_3^m 7 \succ_3^m 10 \\
m_4 &: 7 \succ_4^m 6 \succ_4^m 10 \\
m_5 &: 6 \succ_5^m 7 \succ_5^m 10 \\
m_6 &: 4 \succ_6^m 5 \succ_6^m 10 \\
m_7 &: 5 \succ_7^m 4 \succ_7^m 10 \\
m_8 &: 8 \succ_8^m 4 \succ_8^m 5 \succ_8^m 6 \succ_8^m 10 \succ_8^m 7 \\
m_9 &: 10 \succ_9^m 4 \succ_9^m 6 \succ_9^m 7 \succ_9^m 9 \succ_9^m 5 \\
m_{10} &: 9 \succ_{10}^m 8 \succ_{10}^m 10 \\
\\
w_1 &: 1 \\
w_2 &: 1 \succ_2^w 2 \\
w_3 &: 1 \succ_3^w 2 \succ_3^w 3 \\
w_4 &: 1 \succ_4^w 2 \succ_4^w 3 \succ_4^w 7 \succ_4^w 8 \succ_4^w 9 \succ_4^w 6 \\
w_5 &: 1 \succ_5^w 2 \succ_5^w 3 \succ_5^w 6 \succ_5^w 8 \succ_5^w 9 \succ_5^w 7 \\
w_6 &: 1 \succ_6^w 2 \succ_6^w 3 \succ_6^w 4 \succ_6^w 8 \succ_6^w 9 \succ_6^w 5 \\
w_7 &: 1 \succ_7^w 2 \succ_7^w 3 \succ_7^w 5 \succ_7^w 8 \succ_7^w 9 \succ_7^w 4 \\
w_8 &: 2 \succ_8^w 10 \succ_8^w 8 \\
w_9 &: 1 \succ_9^w 2 \succ_9^w 9 \succ_9^w 10 \\
w_{10} &: 1 \succ_{10}^w 2 \succ_{10}^w 3 \succ_{10}^w 4 \succ_{10}^w 5 \succ_{10}^w 6 \succ_{10}^w 7 \succ_{10}^w 10 \succ_{10}^w 8 \succ_{10}^w 9
\end{aligned}$$

These shortlists makes it possible to identify some transformations that can be applied from the man-optimal stable marriage to obtain other stable marriages (more favourable to women). These transformations are called rotations [Irving and Leather, 1986]. A rotation is a sequence  $\rho = (m_{i_0}, w_{i_0}), \dots, (m_{i_{r-1}}, w_{i_{r-1}})$  of man-woman pairs such that, for each  $i_k$  ( $0 \leq k \leq r-1$ ), (1)  $w_{i_k}$  is first in  $m_{i_k}$ 's shortlist and (2)  $w_{i_{k+1}}$  ( $k+1$  taken modulo  $r$ ) is second in  $m_{i_k}$ 's shortlist. Such a rotation is said to be exposed in the shortlists.

**Example 3** Continuing Example 1, there are two rotations exposed in the shortlists,  $\rho_1 = (4, 7), (5, 6)$  and  $\rho_2 = (6, 4), (7, 5)$ .

Given a rotation, if each  $m_{i_k}$  exchanges his current partner  $w_{i_k}$  for  $w_{i_{k+1}}$ , then the matching remains stable. Eliminating a rotation  $\rho = (m_{i_0}, w_{i_0}), \dots, (m_{i_{r-1}}, w_{i_{r-1}})$  amounts to removing all successors  $m$  of  $m_{i_{k-1}}$  in  $w_{i_k}$ 's shortlist together with the corresponding appearances of  $w_{i_k}$  in the shortlists of men  $m$ . The obtained stable marriage can then be read from the modified shortlists by matching each man with the first woman in his shortlist. In

this new stable marriage, each woman (resp. man) is better off (resp. worse off) than before eliminating the rotation.

Once an exposed rotation has been identified and eliminated, then one or more rotations may be exposed in the resulting (further reduced) shortlists. This process may be repeated, and once all rotations have been eliminated, we obtain the woman optimal stable marriage. A rotation  $\pi$  is said to be a predecessor of a rotation  $\rho$ , denoted by  $\pi < \rho$ , if  $\rho$  cannot be exposed in the men shortlists before  $\pi$  is eliminated. This notion of predecessors makes it possible to define what is called the rotation poset  $(P, \leq)$  where  $P$  is the set of all rotations and  $\leq$  is the precedence relation that we have just mentioned. A closed set in a poset  $(P, \leq)$  is a subset  $R$  of  $P$  such that  $\rho \in R, \pi < \rho \Rightarrow \pi \in R$ .

The following theorem is crucial to understand the importance of the rotation poset.

**Theorem 1** [Irving and Leather, 1986] *The stable marriages of a given stable marriage instance are in one-to-one correspondence with the closed subsets of the rotation poset.*

In this correspondence, each closed subset  $R$  represents the stable marriage obtained by eliminating the rotations in  $R$  starting from  $\mathbf{x}^m$ .

The rotation poset can be represented as a directed acyclic graph, with the rotations as nodes and an arc from  $\pi$  to  $\rho$  iff  $\pi$  is an immediate predecessor of  $\rho$  (i.e.,  $\pi < \rho$  and there is no rotation  $\sigma$  such that  $\pi < \sigma < \rho$ ). Note that this graph has at most  $n(n-1)/2$  nodes, i.e., there are at most  $n(n-1)/2$  rotations [Irving *et al.*, 1987]. Indeed, there are at most  $n^2 - n$  pairs that can be involved in rotations (the  $n$  pairs of  $\mathbf{x}^w$  cannot be involved in a rotation). Each pair belong to at most one rotation and there are at least two pairs in each rotation. We will take advantage of the rotation poset in multiple places in the paper. Importantly, note that the rotation poset (actually a subgraph whose transitive closure is the rotation poset) can be generated in  $O(n^2)$  [Gusfield and Irving, 1989].

**Example 4** *For instance, with the preferences of Example 1, the rotations and their immediate predecessors are given in the following table.*

Rotation	New pairs	Immediate predecessors
$\rho_1 = (4, 7), (5, 6)$	$(4, 6), (5, 7)$	
$\rho_2 = (6, 4), (7, 5)$	$(6, 5), (7, 4)$	
$\rho_3 = (8, 8), (9, 10), (10, 9)$	$(8, 10), (9, 9), (10, 8)$	$\rho_1, \rho_2$

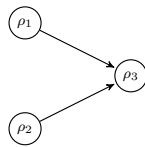


Fig. 1: Rotation poset in Example 1.

This rotation poset shows that there are (potentially many) other stable marriages than the man-optimal or woman-optimal stable marriages. These other stable marriages are likely to be fairer than  $\mathbf{x}^m$  and  $\mathbf{x}^w$  as they are both extreme cases. In order to compute a fair stable

marriage, the optimization of several aggregation functions has been investigated.

- Utilitarian approach:  $\sum_{i=1}^n \text{rk}(m_i, \mu_{\mathbf{x}}(m_i)) + \sum_{j=1}^n \text{rk}(w_j, \mu_{\mathbf{x}}(w_j))$ , which can be minimized in  $O(n^3)$  [Feder, 1994].
- Egalitarian approach:  $\max\{\text{rk}(p, \mu_{\mathbf{x}}(p)) : p \in \mathcal{M} \cup \mathcal{W}\}$ , which can also be minimized in  $O(n^2)$  [Gusfield, 1987].
- Sex-equal stable marriage:  $|\sum_{i=1}^n \text{rk}(m_i, \mu_{\mathbf{x}}(m_i)) - \sum_{j=1}^n \text{rk}(w_j, \mu_{\mathbf{x}}(w_j))|$ , the minimization of which is NP-hard [Kato, 1993].
- Balanced stable marriage:  $\max\{\sum_{i=1}^n \text{rk}(m_i, \mu_{\mathbf{x}}(m_i)), \sum_{j=1}^n \text{rk}(w_j, \mu_{\mathbf{x}}(w_j))\}$ , the minimization of which is NP-hard [Manlove, 2013].

Our contribution differs with previous works on the fair stable marriage problem. Indeed, we optimize a generalized Gini index on disutility values.

Given a matching  $\mathbf{x}$ , the *disutility*  $d(m_i, \mathbf{x})$  (also called *dissatisfaction*) of a man  $m_i$  is defined by  $d(\text{rk}(m_i, \mu_{\mathbf{x}}(m_i)))$ , where  $d : \mathbb{N} \rightarrow \mathbb{Q}^+$ , is a strictly increasing function called disutility function. The disutility values  $d(w_j, \mathbf{x})$  are defined similarly for women. Every stable marriage induces therefore a disutility vector:

$$\mathbf{d}(\mathbf{x}) = (d(m_1, \mathbf{x}), \dots, d(m_n, \mathbf{x}), d(w_1, \mathbf{x}), \dots, d(w_n, \mathbf{x}))$$

with  $N = 2n$  components. Note that the use of disutility values (often called weights) is a common way to extend the traditional framework where the aggregation function is applied on rank values (see e.g., Teo and Sethuraman [1998]; Gusfield and Irving [1989]). Using a unique disutility function for all agents guarantees that they all have the same importance in the aggregation operation. Indeed, the disutility values assigned to the ranks do not depend on the agent's identity. Note that both the egalitarian and the utilitarian variants of the stable marriage problem remain polynomially solvable if one uses disutility values.

**Example 5** We come back to Example 1. Let  $d$  be the disutility function defined by  $d(i) = (i - 1)^2$ , then the disutility values are given by the matrices  $d_M$  and  $d_W$  below where  $d_M[i][j]$  (resp.  $d_W[j][i]$ ) is the disutility of  $m_i$  (resp.  $w_j$ ) if he (resp. she) is matched with  $w_j$  (resp.  $m_i$ ).

$$d_M : \begin{pmatrix} 0 & 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 1 & 0 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 1 & 4 & 0 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 1 & 4 & 9 & 25 & 36 & 16 & 0 & 49 & 64 & 81 \\ 1 & 4 & 9 & 25 & 36 & 0 & 16 & 49 & 64 & 81 \\ 1 & 4 & 9 & 0 & 16 & 36 & 25 & 49 & 64 & 81 \\ 1 & 4 & 9 & 16 & 0 & 36 & 25 & 49 & 64 & 81 \\ 36 & 49 & 64 & 1 & 4 & 9 & 25 & 0 & 81 & 16 \\ 36 & 49 & 64 & 1 & 25 & 4 & 9 & 81 & 16 & 0 \\ 36 & 49 & 64 & 1 & 4 & 25 & 9 & 16 & 0 & 81 \end{pmatrix} \quad d_W : \begin{pmatrix} 0 & 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 0 & 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 0 & 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & 81 \\ 0 & 1 & 4 & 49 & 64 & 36 & 9 & 16 & 25 & 81 \\ 0 & 1 & 4 & 49 & 64 & 9 & 36 & 16 & 25 & 81 \\ 0 & 1 & 4 & 9 & 36 & 49 & 64 & 16 & 25 & 81 \\ 0 & 1 & 4 & 36 & 9 & 49 & 64 & 16 & 25 & 81 \\ 16 & 0 & 25 & 36 & 49 & 64 & 9 & 4 & 81 & 1 \\ 0 & 1 & 16 & 25 & 36 & 49 & 64 & 81 & 4 & 9 \\ 0 & 1 & 4 & 9 & 16 & 25 & 36 & 64 & 81 & 49 \end{pmatrix}$$

Let  $\mathbf{d} = (d_1, \dots, d_N)$  denote a disutility vector. The generalized Gini index [Weymark, 1981] is defined as follows:

**Definition 1** Let  $\lambda = (\lambda_1, \dots, \lambda_N)$  be a vector of weights such that  $\lambda_1 \geq \dots \geq \lambda_N$ . The  $\text{GGI}_{\lambda}(\cdot)$  aggregation function induced by  $\lambda$  is defined by:

$$\text{GGI}_{\lambda}(\mathbf{d}) = \sum_{i=1}^N \lambda_i d_i^{\downarrow},$$



where  $\mathbf{d}^\downarrow$  denotes the vector  $\mathbf{d}$  ordered by nonincreasing values, i.e.,  $d_1^\downarrow \geq d_2^\downarrow \geq \dots \geq d_N^\downarrow$ .

The weights of the GGI aggregation function may be defined in a variety of manner. For instance, the weights initially proposed for the Gini social-evaluation function are:

$$\lambda_i = (2(N - i) + 1)/N^2 \quad \forall i \in \{1, \dots, N\} \quad (4)$$

**Example 6** Coming back to Example 1, if the weights  $\lambda$  are defined by Equation 4 and the disutility function is defined by  $d(i) = i$ , the GGI values of the different stable marriages are (the lower the better):

matching $\mathbf{x}$	ordered vectors $\mathbf{d}^\downarrow(\mathbf{x})$	$\text{GGI}_\lambda(\mathbf{d}(\mathbf{x}))$
$\mathbf{x}^1$	(10, 7, 7, 7, 7, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	4.4525
$\mathbf{x}^2$	(10, 7, 7, 5, 5, 4, 4, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	4.4725
$\mathbf{x}^3$	(10, 7, 7, 5, 5, 4, 4, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	4.4725
$\mathbf{x}^4$	(10, 5, 5, 5, 5, 4, 4, 4, 4, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1)	4.3925
$\mathbf{x}^5$	(9, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 3, 3, 2, 2, 1, 1, 1, 1, 1)	4.74

We thus observe that using a GGI aggregation function makes it possible to obtain  $\mathbf{x}^4$  as an optimal stable marriage.

The GGI is also known in multicriteria decision making under the name of *ordered weighted average* [Yager, 1988]. This aggregation function, to minimize, is well-known to satisfy the Pigou-Dalton transfer principle if  $\lambda_1 > \lambda_2 > \dots > \lambda_N$ :

**Definition 2** An aggregation function  $F$  satisfies the transfer principle if for any  $\mathbf{d} \in (\mathbb{R}^+)^N$  and  $\varepsilon \in (0, d_j - d_i)$  where  $d_j > d_i$ :

$$F(d_1, \dots, d_i + \varepsilon, \dots, d_j - \varepsilon, \dots, d_N) < F(d_1, \dots, d_N).$$

This condition states that the overall welfare should be improved by any transfer of disutility from a “less happy” agent  $j$  to a happier agent  $i$  given that this transfer reduces the gap between the disutilities of agent  $i$  and  $j$ . We can now define the GGI Stable Marriage problem.

#### GGI Stable Marriage (GGISM)

**INSTANCE:** Two disjoint sets of size  $n$ , the men and the women; for each person, a preference list containing all the members of the opposite sex; a vector of weight parameters  $\lambda$  and a disutility function  $d$ .

**SOLUTION:** A stable marriage  $\mathbf{x}$ .

**MEASURE:**  $\text{GGI}_\lambda(\mathbf{d}(\mathbf{x}))$  (to minimize).

### 3 Complexity of the GGISM Problem

The GGISM problem extends both the egalitarian and the utilitarian approaches to the stable marriage problem. Indeed, if the weights of the GGI operator are  $\lambda = (1, \dots, 1)$ , one obtains the sum operation. If the weights are  $\lambda = (1, 0, \dots, 0)$ , one obtains the max operation. While both variants are polynomially solvable problems, the following result states that the GGISM problem is NP-hard:

**Theorem 2** The GGISM problem is NP-hard.

*Proof* We make a reduction from Minimum 2-Satisfiability, which is strongly NP-hard [Kohli *et al.*, 1994].

**Minimum 2-Satisfiability (Min 2-SAT):**

*INSTANCE:* A set  $V$  of variables, a collection  $C$  of disjunctive clauses of at most 2 literals, where a literal is a variable or a negated variable in  $V$ .

*SOLUTION:* A truth assignment for  $V$ .

*MEASURE:* Number of clauses satisfied by the truth assignment (to minimize).

To illustrate the reduction, we will use the following 2-SAT instance:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\} \quad (5)$$

$$C = \{(v_1 \vee v_2), (\neg v_2 \vee \neg v_4), (\neg v_1 \vee v_3), (v_3 \vee \neg v_4), v_2, (v_5 \vee v_6)\} \quad (6)$$

As a preliminary step, note that we can get rid of variables that are present in only one clause. Such a variable is set to true if it is present as a negative literal in the clause and to false otherwise. It can then be removed from the instance. Furthermore, we can make sure that there are exactly two literals in each clause (by duplicating literals). For example, the instance described by Equations 5 and 6 can be modified to:

$$V = \{v_1, v_2, v_3, v_4\} \quad (7)$$

$$C = \{(v_1 \vee v_2), (\neg v_2 \vee \neg v_4), (\neg v_1 \vee v_3), (v_3 \vee \neg v_4), (v_2 \vee v_2)\} \quad (8)$$

In the following we will denote by  $n_v = |V|$  the number of variables and by  $n_c = |C|$  the number of clauses. In the previous example  $n_v = 4$  and  $n_c = 5$ . Furthermore, we will denote by  $c_i$  the  $i^{\text{th}}$  clause in  $C$ .

We are now going to create an instance of the GGISM problem such that:

- There is a one-to-one correspondence between the stable marriages and the truth assignments for  $V$ .
- A stable marriage minimizing the GGI of the agent’s disutilities corresponds to a truth assignment of  $V$  minimizing the number of clauses that are satisfied.

In order to create a one-to-one correspondence between the stable marriages and the truth assignments for  $V$ , we are going to create a rotation  $\rho_i$  for each variable  $v_i \in V$ . Each of these rotations will be exposed in the shortlists from the man-optimal stable marriage for the instance under construction. Additionally, we will ensure that these rotations will be the only ones of the stable marriage instance. In other words, the rotation poset will have one vertex per variable and no edge, as illustrated in Figure 2.

We now give the “meaning” of these rotations. Let’s recall that in a stable marriage there is a one-to-one correspondence between the closed subsets of nodes of the rotation poset and the stable marriages. Now let  $\mathbf{x}$  be a stable marriage corresponding to a closed subset  $R$  of rotations, then the corresponding truth assignment over  $V$  consists in setting  $v_i = 1$  if  $\rho_i \in R$  and  $v_i = 0$  otherwise. Thus in the generated stable marriage instance, the man-optimal stable marriage (i.e.,  $R = \emptyset$ ) corresponds to a truth assignment where all variables in  $V$  are set to 0 while the woman-optimal stable marriage (i.e.,  $R = \{\rho_i | v_i \in V\}$ ) corresponds to a truth assignment where all variables in  $V$  are set to 1.

We now describe more precisely the fashion in which rotations  $\rho_i$  are generated. For each variable  $v_i$ , we create a man-woman pair  $(m_{ij}, w_{ij})$  for each clause  $c_j$  that involves



Fig. 2: Rotation poset of the stable marriage instance generated by the reduction.

$v_i$  either as a positive or negative literal. If variable  $v_i$  is present two times in a clause  $c_j$ , then two man-woman pairs  $(m_{ij}, w_{ij})$  and  $(m'_{ij}, w'_{ij})$  are created. This induces the creation of  $2n_c$  men and  $2n_c$  women in the instance. The rotation  $\rho_i$  then involves all the men and women induced by variable  $v_i$ . For example, in the instance described by Equations 7 and 8,  $\rho_2$  involves men  $m_{21}, m_{22}, m_{25}, m'_{25}$  and women  $w_{21}, w_{22}, w_{25}, w'_{25}$  as variable  $v_2$  is present in  $c_1, c_2$  and  $c_5$ . Let  $r$  denote the number of times variable  $v_i$  appears in  $C$ . The rotation  $\rho_i$  is then induced by the following patterns in the shortlists of men  $\{m_{ij} | v_i \in c_j\}$  and women  $\{w_{ij} | v_i \in c_j\}$ :

$$\begin{array}{ll}
 m_{ij_0} : w_{ij_0} \succ_{ij_0}^m w_{ij_1} & w_{ij_0} : m_{ij_{r-1}} \succ_{ij_0}^w m_{ij_0} \\
 m_{ij_1} : w_{ij_1} \succ_{ij_1}^m w_{ij_2} & w_{ij_1} : m_{ij_0} \succ_{ij_1}^w m_{ij_1} \\
 \vdots & \vdots \\
 m_{ij_{r-2}} : w_{ij_{r-2}} \succ_{ij_{r-2}}^m w_{ij_{r-1}} & w_{ij_{r-2}} : m_{ij_{r-3}} \succ_{ij_{r-2}}^w m_{ij_{r-2}} \\
 m_{ij_{r-1}} : w_{ij_{r-1}} \succ_{ij_{r-1}}^m w_{ij_0} & w_{ij_{r-1}} : m_{ij_{r-2}} \succ_{ij_{r-1}}^w m_{ij_{r-1}}
 \end{array}$$

For instance, rotation  $\rho_2$  is induced by the following pattern in the shortlists:

$$\begin{array}{ll}
 m_{21} : w_{21} \succ_{21}^m w_{22} & w_{21} : m'_{25} \succ_{21}^w m_{21} \\
 m_{22} : w_{22} \succ_{22}^m w_{25} & w_{22} : m_{21} \succ_{22}^w m_{22} \\
 m_{25} : w_{25} \succ_{25}^m w'_{25} & w_{25} : m_{22} \succ_{25}^w m_{25} \\
 m'_{25} : w'_{25} \succ_{25}^{m'} w_{21} & w'_{25} : m_{25} \succ_{25}^{w'} m'_{25}
 \end{array}$$

Note that each man  $m_{ij}$  or woman  $w_{ij}$  is involved in one and only one rotation, which is  $\rho_i$ . As a consequence, each man or woman in the generated instance has only two possible matches in a stable marriage, namely  $w_{ij_k}$  and  $w_{ij_{k+1}}$  (modulo the size  $r$  of rotation  $\rho_i$ ) for  $m_{ij_k}$ , and  $m_{ij_{k-1}}$  (modulo  $r$ ) and  $m_{ij_k}$  for  $w_{ij_k}$ . For simplicity, we will denote by  $\text{rk}^+(m_{ij})$  (resp.  $\text{rk}^-(m_{ij})$ ) the rank of the best (resp. worst) possible match for  $m_{ij}$  in a stable marriage. Notations  $\text{rk}^+(w_{ij})$  and  $\text{rk}^-(w_{ij})$  are defined similarly for women.

Given a stable marriage characterized by a set  $R$  of rotations, it is possible to determine if clause  $c_j$  is satisfied by examining which rotations belong to  $R$ . According to the form of clause  $c_j$ , columns “in” and “out” of Table 1 indicate which rotations should be included or not in  $R$  so that  $c_j$  is not satisfied. Assuming that  $c_j$  involves variables  $v_i$  and  $v_k$  (or

clause $c_j$	in	out	decisive agents
$v_i \wedge v_k$		$\rho_i, \rho_k$	$m_{ij}, m_{kj}$
$v_i \wedge \neg v_k$	$\rho_k$	$\rho_i$	$m_{ij}, w_{kj}$
$\neg v_i \wedge v_k$	$\rho_i$	$\rho_k$	$w_{ij}, m_{kj}$
$\neg v_i \wedge \neg v_k$	$\rho_i, \rho_k$		$w_{ij}, w_{kj}$
$v_i \wedge v_i$		$\rho_i$	$m_{ij}, m'_{ij}$
$\neg v_i \wedge \neg v_i$	$\rho_i$		$w_{ij}, w'_{ij}$

Table 1: Clause  $c_j$  is not satisfied iff the rotations of the second (resp. third) column are included (resp. not included) in  $R$ . Consequently, clause  $c_j$  is not satisfied iff the two agents in the last column are matched with their choices of rank  $\text{rk}^+(\cdot)$ .

possibly their negations), it is sufficient to examine the matches of two specific agents among  $m_{ij}, m_{kj}, w_{ij}, w_{kj}$  to determine if rotations  $\rho_i$  and  $\rho_k$  belong or not to  $R$ . These two specific agents are called *decisive agents of  $c_j$*  in the following. We have indeed  $\rho_i \in R$  iff the rank of the match of  $m_{ij}$  is  $\text{rk}^+(m_{ij})$ . Similarly, we have  $\rho_i \notin R$  iff the rank of the match of  $w_{ij}$  is  $\text{rk}^+(w_{ij})$ . Put another way,  $m_{ij}$  (resp.  $w_{ij}$ ) is a decisive agent of  $c_j$  if  $v_i$  (resp.  $\neg v_i$ ) belongs to  $c_j$ . The clause  $c_j$  is not satisfied iff the two decisive agents are with their match of rank  $\text{rk}^+(\cdot)$ . The decisive agents according to the form of clause  $c_j$  are given in the last column of Table 1.

For illustration, let us return to the 2-SAT instance described by Equations 7 and 8. Given the stable marriage instance generated by the reduction, and a stable marriage  $\mathbf{x}$ , clause  $v_1 \wedge v_2$  is *not* satisfied iff  $\text{rk}(m_{11}, \mu_{\mathbf{x}}(m_{11})) = \text{rk}^+(m_{11})$  and  $\text{rk}(m_{21}, \mu_{\mathbf{x}}(m_{21})) = \text{rk}^+(m_{21})$ . More generally, it is possible to count the number of clauses that are *not* satisfied by examining the ranks of the matches of the decisive agents of each clause.

We will soon explain how to use a GGI operator to count the number of clauses that are not satisfied in the 2-SAT instance. Beforehand, we need to introduce fictitious agents in order to control the positions of the decisive agents in the ordered vector of disutilities for every stable marriage. More precisely, we introduce four fictitious agents  $m_j, m'_j, w_j, w'_j$  per clause  $c_j$  such that  $m_j$  (resp.  $m'_j$ ) is the first choice of  $w_j$  (resp.  $w'_j$ ) and vice-versa. Thus  $m_j$  (resp.  $m'_j$ ) can only be matched to  $w_j$  (resp.  $w'_j$ ) in a stable marriage, and therefore the fictitious agents will not interfere with the possible matches of the other agents.

The fictitious agents are placed in the preference lists of the other agents such that  $\text{rk}^+(\cdot) = 2j + 1$  and  $\text{rk}^-(\cdot) = 2j + 2$  for the two decisive agents of clause  $c_j$ . Furthermore,  $\text{rk}^+(\cdot) = 1$  and  $\text{rk}^-(\cdot) = 2$  for the remaining (non-decisive) agents. Note that  $2j + 1 > 2$  as  $j \geq 1$  and therefore the two decisive agents of  $c_j$  are at positions  $2(n_c - j) + 1$  and  $2(n_c - j) + 2$  in the permutation that ranks the agents by non-increasing disutilities.

To achieve these properties, we position  $2j$  fictitious agents at the beginning of the preference list of the decisive agents of clause  $c_j$  (e.g.,  $m_1 m'_1 \dots m_j m'_j$  for a decisive agent  $w_{ij}$ ). These agents are positioned just before the two possible matches of the agent in a stable marriage. Regarding the non-decisive agents, their two possible matches in a stable marriage are simply placed at the beginning of their preference lists.

For illustration, in the 2-SAT instance described by Equations 7 and 8, the preference list of agent  $w_{22}$  (who is a decisive agent of  $c_2$ ) is:

$$w_{22} : m_1 \succ_{w_{22}}^w m'_1 \succ_{w_{22}}^w m_2 \succ_{w_{22}}^w m'_2 \succ_{w_{22}}^w m_{21} \succ_{w_{22}}^w m_{22} \succ_{w_{22}}^w \dots$$

and the preference list of agent  $m_{22}$  (who is *not* a decisive agent of  $c_2$ ) is:

$$m_{22} : w_{22} \succ_{m_{22}}^m w_{25} \succ_{m_{22}}^m \dots$$

$m_1 : w_1 \dots$	$w_1 : m_1 \dots$
$m'_1 : w'_1 \dots$	$w'_1 : m'_1 \dots$
$\vdots$	$\vdots$
$m_5 : w_5 \dots$	$w_5 : m_5 \dots$
$m'_5 : w'_5 \dots$	$w'_5 : m'_5 \dots$
$m_{11} : w_1 w'_1 w_{11} w_{13} \dots$	$w_{11} : m_{13} m_{11} \dots$
$m_{13} : w_1 w'_1 w_2 w'_2 w_3 w'_3 w_{13} w_{11} \dots$	$w_{13} : m_{11} m_{13} \dots$
$m_{21} : w_1 w'_1 w_{21} w_{22} \dots$	$w_{21} : m'_{25} m_{21} \dots$
$m_{22} : w_{22} w_{25} \dots$	$w_{22} : m_1 m'_1 m_2 m'_2 m_{21} m_{22} \dots$
$m_{25} : w_1 w'_1 w_2 w'_2 w_3 w'_3 w_4 w'_4 w_5 w'_5 w_{25} w'_{25} \dots$	$w_{25} : m_{22} m_{25} \dots$
$m'_{25} : w_1 w'_1 w_2 w'_2 w_3 w'_3 w_4 w'_4 w_5 w'_5 w'_{25} w_{21} \dots$	$w'_{25} : m_{25} m'_{25} \dots$
$m_{33} : w_1 w'_1 w_2 w'_2 w_3 w'_3 w_{33} w_{34} \dots$	$w_{33} : m_{34} m_{33} \dots$
$m_{34} : w_1 w'_1 w_2 w'_2 w_3 w'_3 w_4 w'_4 w_{34} w_{33} \dots$	$w_{34} : m_{33} m_{34} \dots$
$m_{42} : w_{42} w_{44} \dots$	$w_{42} : m_1 m'_1 m_2 m'_2 m_{44} m_{42} \dots$
$m_{44} : w_{44} w_{42} \dots$	$w_{43} : m_1 m'_1 m_2 m'_2 m_3 m'_3 m_4 m'_4 m_{42} m_{44} \dots$

Fig. 3: Preference lists obtained for the min 2-SAT instance of Equations 7 and 8.

This construction is illustrated in Figure 3 (where symbols  $\succ$  are omitted for readability reasons) for the Minimum 2-Satisfiability instance defined by Equations 7 and 8. The preference lists of the agents are only partially given but note that they can be completed in any consistent way that would lead to complete and transitive orders.

We now explain how to define the disutility values attributed to each rank, as well as the weights of the GGI operator, so that the number of unsatisfied clauses can be inferred from the GGI value of the stable marriage.

*Disutility values and weights of the GGI.* We first recall that each clause  $c_j$  induces 6 agents that are matched either with their first or second choices and 2 agents (the decisive ones) that are matched with their choices of rank  $2j + 1$  or  $2j + 2$ . By construction of the preference lists, note that no agent can be matched with a partner that is ranked strictly beyond  $2n_c + 2$  in his/her preference list. Therefore the values of  $d(i)$  for  $i > 2n_c + 2$  play no role, and can be fixed arbitrarily as long as they are increasing with  $i$  and strictly greater than  $d(2n_c + 2)$ .

- The increasing disutility values for ranks 1 to  $2n_c + 2$  are defined as follows (assuming that  $n_c \geq 2$ ):

$$\begin{aligned}
 d(1) &= 0 \\
 d(2) &= 1 \\
 d(2j + 1) &= j + 1, \quad \forall j \in \{1, \dots, n_c\} \\
 d(2j + 2) &= j + 1 + n_c^{-j}, \quad \forall j \in \{1, \dots, n_c\}
 \end{aligned}$$

– The non-increasing weights of the GGI are defined as follows:

$$\lambda = \left( \underbrace{n_c^{n_c+1}, n_c^{n_c}, n_c^{n_c}, n_c^{n_c-1}, \dots, n_c^3, n_c^2, n_c^2, n_c^1}_{2n_c \text{ weights}}, \underbrace{0, \dots, 0}_{6n_c \text{ weights}} \right).$$

We recall that the  $2n_c$  agents with the highest disutility values are the decisive agents (the two decisive agents of clause  $c_j$  are matched with an agent of rank  $2j + 1$  or  $2j + 2$ ) and the  $6n_c$  agents with the lowest disutility values are the *non*-decisive agents (who are matched to one of their two first choices). Consequently, the weight vector  $\lambda$  attributes a weight 0 in the GGI operator to the  $6n_c$  non-decisive agents while, for each clause  $c_j$ , it attributes a weight  $n_c^j$  (resp.  $n_c^{j+1}$ ) to the most satisfied (resp. least satisfied) of the two decisive agents of  $c_j$ .

An upper bound on the GGI value is given by  $\Delta_u = \sum_{j=1}^{n_c} (n_c^j + n_c^{j+1})d(2j + 2)$ . This would correspond to a stable marriage, where for each  $c_j$ , the two decisive agents of  $c_j$  are both matched to their choice of rank  $2j + 2$ . Similarly, a lower bound on the GGI value is given by  $\Delta_l = \sum_{j=1}^{n_c} (n_c^j + n_c^{j+1})d(2j + 1)$  (if the two decisive agents of  $c_j$  are both matched to their choice of rank  $2j + 1$ ). Simple calculations show that  $\Delta_l = \Delta_u - n_c(1 + n_c)$ .

These bounds are useful for establishing Lemma 1 below, that makes it possible to infer the number of unsatisfied clauses from the GGI value. The lower the GGI value, the higher the number of unsatisfied clauses. Hence, minimizing the GGI value amounts to maximizing the number of unsatisfied clauses, which concludes the proof.

**Lemma 1** *For the GGI stable marriage instance obtained by the method described above, a stable marriage  $\mathbf{x}$  corresponds to a truth assignment on  $V$  for which the number of unsatisfied clauses is:*

$$\left\lfloor \frac{\Delta_u - \text{GGI}_\lambda(\mathbf{d}(\mathbf{x}))}{n_c + 1} \right\rfloor$$

*Proof of Lemma 1.* We wish to show that if a stable marriage  $\mathbf{x}$  corresponds to a truth assignment on  $V$  with exactly  $k$  unsatisfied clauses then:

$$\Delta_u - (k + 1)(n_c + 1) < \text{GGI}_\lambda(\mathbf{d}(\mathbf{x})) \leq \Delta_u - k(n_c + 1),$$

from which the lemma straightforwardly follows.

Assume that  $k$  clauses  $\{c_{j_1}, \dots, c_{j_k}\}$  are unsatisfied for the truth assignment induced by  $\mathbf{x}$ . Then for each  $c_{j_l}$ , the two decisive agents of  $c_{j_l}$  are both matched to their choice of rank  $2j_l + 1$ . Hence:

$$\begin{aligned} \text{GGI}_\lambda(\mathbf{d}(\mathbf{x})) &\leq \Delta_u - \sum_{l=1}^k (n_c^{j_l} + n_c^{j_l+1})(d(2j_l + 2) - d(2j_l + 1)) \\ &= \Delta_u - \sum_{l=1}^k (n_c^{j_l} + n_c^{j_l+1})n_c^{-j_l} = \Delta_u - k(n_c + 1) \end{aligned}$$

because each decisive agent of clause  $c_{j_l}$ , for  $l \in \{1, \dots, k\}$ , has a disutility of  $d(2j_l + 1)$  and not  $d(2j_l + 2)$ .

Now, let  $\{c_{j_1}, \dots, c_{j_{n_c-k}}\}$  denote the satisfied clauses for the truth assignment induced by  $\mathbf{x}$ . Then for each  $c_{j_l}$ , at least one of the two decisive agents of  $c_{j_l}$  is matched to his/her choice of rank  $2j_l + 2$ . In the best case (w.r.t. the GGI value), only one of the two is matched

to his/her choice of rank  $2j_l + 2$ , and his/her weight in the GGI aggregation is  $n_c^{j_l+1}$  because his/her disutility is the highest among the two agents. Hence:

$$\begin{aligned}
\text{GGI}_\lambda(\mathbf{d}(\mathbf{x})) &\geq \Delta_l + \sum_{l=1}^{n_c-k} n_c^{j_l+1} (d(2j_l + 2) - d(2j_l + 1)) \\
&= \Delta_u - n_c(1 + n_c) + \sum_{l=1}^{n_c-k} n_c^{j_l+1} n_c^{-j_l} \\
&= \Delta_u - n_c(1 + n_c) + (n_c - k)n_c \\
&= \Delta_u - (k + 1)n_c \\
&> \Delta_u - (k + 1)(n_c + 1)
\end{aligned}$$

This concludes the proof of the lemma.

#### 4 A 2-approximation Algorithm

We now present a polynomial time 2-approximation algorithm for the GGI stable marriage problem.

The 2-approximation algorithm uses a linear programming formulation of the stable marriage problem, based on the rotation poset [Gusfield and Irving, 1989]. It is indeed well-known that the set of stable marriages can be characterized by the following set of inequalities where we have one binary variable  $y(\rho)$  for each rotation in the rotation poset and:

$$y(\rho') - y(\rho) \leq 0 \quad (9)$$

for each pair of rotations such that  $\rho$  precedes  $\rho'$ . Variable  $y(\rho)$  is equal to 1 if rotation  $\rho$  is included in the closed set of rotations associated to the stable marriage and 0 otherwise. Importantly, note that the extreme points of the polytope defined by constraints 9 for  $0 \leq y(\rho) \leq 1$ ,  $\forall \rho$  are in one-to-one correspondence with the stable marriages of the instance [Gusfield and Irving, 1989]. Furthermore, the stable marriage  $\mathbf{x}$  characterized by variables  $y(\rho)$  can be inferred by using Equations 10, 11 and 12 below.

To explain this point we introduce some notations. Let  $\Gamma$  denote the set of man-woman pairs included in at least one stable matching. These pairs can be found by looking at the pairs that are created and broken by each rotation. Indeed, note that for each pair  $(m, w) \in \Gamma$ , there exists exactly one rotation, denoted by  $\rho_{\text{get}}(m, w)$ , that creates this pair (unless this pair is in  $\mathbf{x}^m$ ) and exactly one rotation, denoted by  $\rho_{\text{break}}(m, w)$ , that breaks this pair (unless this pair is in  $\mathbf{x}^w$ ). Then, one can compute variables  $x_{ij}$  corresponding to a set of variables  $y(\rho)$  by using the following equations:

$$x_{ij} = 1 - y(\rho_{\text{break}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^m = 1 \quad (10)$$

$$x_{ij} = y(\rho_{\text{get}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^w = 1 \quad (11)$$

$$x_{ij} = y(\rho_{\text{get}}(i, j)) - y(\rho_{\text{break}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^m = x_{ij}^w = 0 \quad (12)$$

**Example 7** Let us come back to Example 1. The pairs in  $\Gamma$  are listed in the left column of Table 2. The rotations  $\rho_{\text{get}}(m, w)$  and  $\rho_{\text{break}}(m, w)$  for each pair  $(m, w)$  are given in the middle and right columns.

$(m, w) \in \Gamma$	$\rho_{\text{get}}(m, w)$	$\rho_{\text{break}}(m, w)$
$(m_1, w_1)$		
$(m_2, w_2)$		
$(m_3, w_3)$		
$(m_4, w_7)$		$\rho_1$
$(m_4, w_6)$	$\rho_1$	
$(m_5, w_6)$		$\rho_1$
$(m_5, w_7)$	$\rho_1$	
$(m_6, w_4)$		$\rho_2$
$(m_6, w_5)$	$\rho_2$	
$(m_7, w_5)$		$\rho_2$
$(m_7, w_4)$	$\rho_2$	
$(m_8, w_8)$		$\rho_3$
$(m_8, w_{10})$	$\rho_3$	
$(m_9, w_{10})$		$\rho_3$
$(m_9, w_9)$	$\rho_3$	
$(m_{10}, w_9)$		$\rho_3$
$(m_{10}, w_8)$	$\rho_3$	

Table 2: Rotations  $\rho_{\text{get}}(m, w)$  and  $\rho_{\text{break}}(m, w)$  in Example 1.

A mathematical programming formulation of the GGISM problem reads as follows:

$$\mathcal{P} \left\{ \begin{array}{l}
 \min_{\mathbf{d}, \mathbf{x}, \mathbf{y}} \text{GGI}_\lambda(\mathbf{d}) \\
 d_i^m = \sum_{(i,j) \in \Gamma} x_{ij} d(\text{rk}(m_i, w_j)), \quad \forall i \in \{1, \dots, n\} \\
 d_j^w = \sum_{(i,j) \in \Gamma} x_{ij} d(\text{rk}(w_j, m_i)), \quad \forall j \in \{1, \dots, n\} \\
 x_{ij} = 1 - y(\rho_{\text{break}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^m = 1 \\
 x_{ij} = y(\rho_{\text{get}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^w = 1 \\
 x_{ij} = y(\rho_{\text{get}}(i, j)) - y(\rho_{\text{break}}(i, j)), \quad \forall (i, j) \in \Gamma \text{ s.t. } x_{ij}^m = x_{ij}^w = 0 \\
 y(\rho') - y(\rho) \leq 0, \quad \forall (\rho, \rho') \text{ s.t. } \rho < \rho' \\
 d_i^m \geq 0, \quad \forall i \in \{1, \dots, n\} \\
 d_j^w \geq 0, \quad \forall j \in \{1, \dots, n\} \\
 x_{ij} \geq 0, \quad \forall (i, j) \in \Gamma \\
 y(\rho) \in \{0, 1\}, \quad \forall \rho \in P
 \end{array} \right. \quad (13) \quad (14) \quad (15) \quad (16)$$

where  $d_i^m$  (resp.  $d_j^w$ ) represents the disutility of  $m_i$  (resp.  $w_j$ ),  $\mathbf{d} = (d_1^m, \dots, d_n^m, d_1^w, \dots, d_n^w)$ , and as usual:

- $x_{ij} = 1$  (resp. 0) if  $(m_i, w_j)$  is (resp. is not) in the stable marriage  $\mathbf{x}$ ,
- $y(\rho) = 1$  (resp. 0) if  $\rho$  belongs (resp. does not) to the set of rotations characterizing  $\mathbf{x}$ ,
- $P$  is the set of all rotations.

Let us denote by  $\widehat{\mathcal{P}}$  the linear programming relaxation of  $\mathcal{P}$  where  $y(\rho) \in \{0, 1\}$  is replaced by  $0 \leq y(\rho) \leq 1$ . Importantly, note that variables  $y(\rho)$  in an optimal solution to  $\widehat{\mathcal{P}}$  are not necessarily integer because the objective function is non-linear (and therefore there does not necessarily an optimal vertex in the solution polytope).



A polynomial time 2-approximation algorithm can be obtained by rounding an optimal solution of  $\widehat{\mathcal{P}}$ . The 2-approximation algorithm writes as follows:

**ROUNDING ALGORITHM**

1. Solve  $\widehat{\mathcal{P}}$  and let  $(\widehat{\mathbf{d}}, \widehat{\mathbf{x}}, \widehat{\mathbf{y}})$  denote an optimal solution to  $\widehat{\mathcal{P}}$ ;
2. For each  $\rho \in P$ , set  $y(\rho) = 1$  if  $\widehat{y}(\rho) \geq 0.5$ , and  $y(\rho) = 0$  otherwise;
3. Return the stable marriage  $\mathbf{x}$  obtained from  $\mathbf{y}$  by using constraints 13–15 in  $\mathcal{P}$ .

**Example 8** *Coming back to Example 1, assume that the weights of the GGI operator are defined by Equation 4 and that the disutility function is defined by  $d(i) = i$ . Then, an optimal solution  $(\widehat{\mathbf{d}}, \widehat{\mathbf{x}}, \widehat{\mathbf{y}})$  to  $\widehat{\mathcal{P}}$  is characterized by  $\widehat{y}(\rho_1) = \widehat{y}(\rho_2) = 0.75$  and  $\widehat{y}(\rho_3) = 0$  (for a GGI value of 4.3075). For this instance, by ROUNDING ALGORITHM, the obtained vector  $\mathbf{y}$  is therefore  $y(\rho_1) = y(\rho_2) = 1$  and  $y(\rho_3) = 0$ . This corresponds to stable marriage  $\mathbf{x}^4$ , which is in fact an optimal solution.*

Steps 2 and 3 of the algorithm can obviously be performed in polynomial time. In step 1, solving  $\widehat{\mathcal{P}}$  can also be performed in polynomial time by using one of the linearizations of the GGI operator proposed by Ogryczak and Śliwiński [2003]. The following lemma ensures that the returned solution is a 2-approximation of an optimal solution of  $\mathcal{P}$ :

**Lemma 2** *For any feasible solution  $(\widehat{\mathbf{d}}, \widehat{\mathbf{x}}, \widehat{\mathbf{y}})$  of  $\widehat{\mathcal{P}}$ , the feasible solution  $(\mathbf{d}, \mathbf{x}, \mathbf{y})$  of  $\mathcal{P}$  obtained by setting*

$$y(\rho) = \begin{cases} 1 & \text{if } \widehat{y}(\rho) \geq 0.5, \\ 0 & \text{otherwise} \end{cases}$$

*is such that  $\widehat{\mathbf{d}} \geq \frac{1}{2}\mathbf{d}$  where  $\geq$  is taken componentwise.*

*Proof* In order to establish the result stated in the lemma, we introduce the notion of man and woman *weights* of a rotation. Given a rotation  $\rho = (m_{i_0}, w_{i_0}), \dots, (m_{i_{r-1}}, w_{i_{r-1}})$  we define the  $m_i$ -weight of that rotation by:

$$\omega_i^m(\rho) = \begin{cases} d(\mathbf{rk}(m_{i_k}, w_{i_k})) - d(\mathbf{rk}(m_{i_k}, w_{i_{(k+1) \bmod r}})) & \text{if } i \in \{i_0, \dots, i_{r-1}\} \text{ and } i = i_k \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we define the  $w_j$ -weight of that rotation by:

$$\omega_j^w(\rho) = \begin{cases} d(\mathbf{rk}(w_{i_k}, m_{i_k})) - d(\mathbf{rk}(w_{i_k}, m_{i_{(k-1) \bmod r}})) & \text{if } j \in \{i_0, \dots, i_{r-1}\} \text{ and } j = i_k \\ 0 & \text{otherwise.} \end{cases}$$

Note that a man weight of a rotation will always be negative while a woman weight of a rotation will always be positive.

Assume that  $\rho$  is a rotation that is exposed in a stable marriage  $\mathbf{x}$ , and let  $\mathbf{x}'$  be the stable marriage obtained from  $\mathbf{x}$  by eliminating  $\rho$ . Then:

$$d(m_i, \mathbf{x}') = d(m_i, \mathbf{x}) - \omega_i^m(\rho), \quad d(w_j, \mathbf{x}') = d(w_j, \mathbf{x}) - \omega_j^w(\rho).$$

Consequently, if  $\mathbf{x}$  is the stable marriage obtained from the man-optimal stable marriage  $\mathbf{x}^m$  by eliminating rotations  $\rho_1, \dots, \rho_t$ , then:

$$d(m_i, \mathbf{x}) = d(m_i, \mathbf{x}^m) - \sum_{k=1}^t \omega_i^m(\rho_k), \quad d(w_j, \mathbf{x}) = d(w_j, \mathbf{x}^m) - \sum_{k=1}^t \omega_j^w(\rho_k).$$

We now establish the result stated in the lemma. Let  $(\hat{\mathbf{d}}, \hat{\mathbf{x}}, \hat{\mathbf{y}})$  denote a feasible solution of  $\hat{\mathcal{P}}$ . The previous equations extend as follows for solutions of  $\hat{\mathcal{P}}$ :

$$\hat{d}_i^m = d(m_i, \mathbf{x}^m) - \sum_{\rho \in P} \hat{y}(\rho) \omega_i^m(\rho), \quad \hat{d}_j^w = d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} \hat{y}(\rho) \omega_j^w(\rho). \quad (17)$$

Now consider the feasible solution  $(\mathbf{d}, \mathbf{x}, \mathbf{y})$  of  $\mathcal{P}$  defined by  $y(\rho) = 1$  if  $\hat{y}(\rho) \geq 0.5$ , and 0 otherwise. The feasibility of  $(\mathbf{d}, \mathbf{x}, \mathbf{y})$  comes from the fact that  $\{\rho : \hat{y}(\rho) \geq 0.5\}$  is a closed set of rotations. Indeed, note that constraints 16 ensures that  $y(\rho') \leq y(\rho)$  for all  $\rho < \rho'$ . We have:

$$\begin{aligned} \hat{d}_i^m - \frac{1}{2}d_i^m &= d(m_i, \mathbf{x}^m) - \sum_{\rho \in P} \hat{y}(\rho) \omega_i^m(\rho) - \frac{1}{2}(d(m_i, \mathbf{x}^m) - \sum_{\rho \in P} y(\rho) \omega_i^m(\rho)) \\ &= \frac{1}{2}d(m_i, \mathbf{x}^m) - \sum_{\rho \in P} (\hat{y}(\rho) - \frac{1}{2}y(\rho)) \omega_i^m(\rho) \\ &\geq \frac{1}{2}d(m_i, \mathbf{x}^m) \geq 0 \end{aligned}$$

as  $0 \leq (\hat{y}(\rho) - \frac{1}{2}y(\rho))$  for all  $\rho \in P$  and  $\omega_i^m(\rho) \leq 0$  for all  $i \in \{1, \dots, n\}$  and  $\rho \in P$ . Hence,  $\hat{d}_i^m \geq \frac{1}{2}d_i^m$  for all  $i \in \{1, \dots, n\}$ .

Similarly, for women we have:

$$\begin{aligned} \hat{d}_j^w - \frac{1}{2}d_j^w &= d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} \hat{y}(\rho) \omega_j^w(\rho) - \frac{1}{2}(d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} y(\rho) \omega_j^w(\rho)) \\ &= \frac{1}{2}d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} (\hat{y}(\rho) - \frac{1}{2}y(\rho)) \omega_j^w(\rho) \\ &\geq \frac{1}{2}(d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} \omega_j^w(\rho)) \end{aligned}$$

as  $(\hat{y}(\rho) - \frac{1}{2}y(\rho)) \leq 0.5$  for all  $\rho \in P$  and  $\omega_j^w(\rho) \geq 0$  for all  $j \in \{1, \dots, n\}$  and  $\rho \in P$ . Since eliminating all rotations from  $\mathbf{x}^m$  leads to  $\mathbf{x}^w$ , we have that  $\frac{1}{2}(d(w_j, \mathbf{x}^m) - \sum_{\rho \in P} \omega_j^w(\rho)) = \frac{1}{2}d(w_j, \mathbf{x}^w)$ . Therefore,  $\hat{d}_j^w - \frac{1}{2}d_j^w \geq 0$  and hence,  $\hat{d}_j^w \geq \frac{1}{2}d_j^w$  for all  $j \in \{1, \dots, n\}$ .

By combining the inequalities obtained for men and women, we obtain that  $\hat{\mathbf{d}} \geq \frac{1}{2}\mathbf{d}$ , which concludes the proof.

We can now state the main result of this section:

**Theorem 3** ROUNDING ALGORITHM is a polynomial time 2-approximation algorithm for the GGI stable marriage problem, and the bound is tight.

*Proof* We first recall that all steps of ROUNDING ALGORITHM can be performed in polynomial time. Furthermore, by Lemma 2, the feasible solution  $(\mathbf{d}, \mathbf{x}, \mathbf{y})$  generated by ROUNDING ALGORITHM is such that  $\hat{\mathbf{d}} \geq \frac{1}{2}\mathbf{d}$ , where  $(\hat{\mathbf{d}}, \hat{\mathbf{x}}, \hat{\mathbf{y}})$  is an optimal solution to  $\hat{\mathcal{P}}$ . Consequently:

$$\text{GGI}_\lambda(\hat{\mathbf{d}}) \geq \frac{1}{2}\text{GGI}_\lambda(\mathbf{d})$$

because  $\text{GGI}_\lambda(\mathbf{d}) \leq \text{GGI}_\lambda(\mathbf{d}')$  for  $\mathbf{d} \leq \mathbf{d}'$  (see e.g. Fodor *et al.* [1995]) and  $\text{GGI}_\lambda(\alpha\mathbf{d}) = \alpha\text{GGI}_\lambda(\mathbf{d})$  for  $\alpha > 0$ .

For the tightness of the bound, consider the following instance of the stable marriage problem:

$$\begin{array}{ll}
m_1 : 1 \succ_1^m 2 \succ_1^m 3 & w_1 : 2 \succ_1^w 3 \succ_1^w 1 \\
m_2 : 2 \succ_2^m 3 \succ_2^m 1 & w_2 : 3 \succ_2^w 1 \succ_2^w 2 \\
m_3 : 3 \succ_3^m 1 \succ_3^m 2 & w_3 : 1 \succ_3^w 2 \succ_3^w 3
\end{array}$$

There are two rotations  $\rho_1 = (1, 1), (2, 2), (3, 3)$  and  $\rho_2 = (1, 2), (2, 3), (3, 1)$ , with  $\rho_1 < \rho_2$ , which yield three stable marriages:

- the man-optimal stable marriage  $\mathbf{x}^m$  in which each man is matched with his first choice, and which corresponds to eliminating no rotation,
- the woman-optimal stable marriage  $\mathbf{x}^w$  in which each woman is matched with her first choice, and which corresponds to eliminating both rotations,
- a “compromise” stable marriage  $\mathbf{x}^c$  in which each agent is matched with his/her second choice, and which corresponds to eliminating only  $\rho_1$ .

We use the disutility function  $d$  defined by  $d(1) = 0$ ,  $d(2) = 1 + \epsilon$  and  $d(3) = 2$  (with  $\epsilon > 0$ ) and the following GGI weights:  $\boldsymbol{\lambda} = (a, b, c, 0, 0, 0)$  with  $a \geq b \geq c > 0$ . The disutility vectors of the three stable marriages are then  $\mathbf{d}(\mathbf{x}^m) = (0, 0, 0, 2, 2, 2)$ ,  $\mathbf{d}(\mathbf{x}^w) = (2, 2, 2, 0, 0, 0)$  and  $\mathbf{d}(\mathbf{x}^c) = (1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon)$ .

By using Equation 17 from the proof of Lemma 2, the value of  $\hat{d}_i^m$  for each man  $m_i$  and the value of  $\hat{d}_j^w$  for each woman  $w_j$  are written as follows in terms of  $\hat{y}(\rho_1)$  and  $\hat{y}(\rho_2)$ :

$$\begin{aligned}
\hat{d}_i^m &= 0 + \hat{y}(\rho_1)(1 + \epsilon) + \hat{y}(\rho_2)(1 - \epsilon) \quad \forall i \in \{1, 2, 3\} \\
\hat{d}_j^w &= 2 - \hat{y}(\rho_1)(1 - \epsilon) - \hat{y}(\rho_2)(1 + \epsilon) \quad \forall j \in \{1, 2, 3\}
\end{aligned}$$

where  $\hat{y}(\rho_1) \geq \hat{y}(\rho_2)$ . We see that the three men share the same disutility value, as well as the three women. The GGI value of a feasible solution to  $\widehat{\mathcal{P}}$  is thus completely determined by the common disutility of the men or the common disutility of the women because only the three least satisfied agents are taken into account in  $\boldsymbol{\lambda}$ . Consequently, an optimal solution to  $\widehat{\mathcal{P}}$  minimizes  $\max\{\hat{d}_1^m, \hat{d}_1^w\}$  for  $\hat{y}(\rho_1) \geq \hat{y}(\rho_2)$ . Simple calculations make it possible to conclude that the only optimal solution to  $\widehat{\mathcal{P}}$  is characterized by  $\hat{y}(\rho_1) = \hat{y}(\rho_2) = 0.5$ .

For this instance, ROUNDING ALGORITHM returns therefore the woman-optimal stable marriage which has the ordered disutility vector  $(2, 2, 2, 0, 0, 0)$  and a GGI value of  $2(a + b + c)$ . However, an optimal stable marriage is the “compromise” stable marriage, which has the ordered disutility vector  $(1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon, 1 + \epsilon)$  and a GGI value of  $(1 + \epsilon)(a + b + c)$ . By taking the limit for  $\epsilon$  going to 0, we obtain the tightness of the bound.

*Remark 1* Note that the approach taken in ROUNDING ALGORITHM is valid for any aggregation criterion  $F$  on dissatisfactions of agents for which the following condition holds:

$$\begin{aligned}
F\left(\frac{1}{2}\mathbf{d}(\mathbf{x})\right) &= \frac{1}{2}F(\mathbf{d}(\mathbf{x})) \\
F(\mathbf{d}(\mathbf{x}) + \mathbf{r}) &\geq F(\mathbf{d}(\mathbf{x}))
\end{aligned}$$

where  $\mathbf{r}$  is any non-negative vector.

*Remark 2* A general approximation result for the optimization of a generalized Gini index in multiobjective optimization problems has been proposed by Kasperski and Zieliński [2015]. For the GGISM problem, it amounts to compute an optimal stable marriage according to the sum of disutilities of pairs  $(m_i, w_j)$ , where the disutility of a pair  $(m_i, w_j)$  is defined by  $\lambda_1 \max\{d(\mathbf{rk}(m_i, w_j)), d(\mathbf{rk}(w_j, m_i))\} + \lambda_2 \min\{d(\mathbf{rk}(m_i, w_j)), d(\mathbf{rk}(w_j, m_i))\}$ . This can be performed in polynomial time by linear programming. The returned solution is a  $N\lambda_1$ -approximation, provided  $\sum_{i=1}^N \lambda_i = 1$ . To obtain a better guarantee than 2, one should have  $\lambda_1 < 2/N$ . On the contrary, by taking advantage of the specific structure of the stable marriage problem, our approach yields a 2-approximation *whatever weights are used*.

## 5 The GGI Stable Marriage Problem with a Bounded Number of Non-zero Weights

In this section, we provide an algorithm whose complexity is  $O(2^K n^{2K+4})$  where  $K = \max\{i : \lambda_i > 0\}$ . Hence, the complexity is polynomial time if  $K$  is assumed to be a constant, where  $K$  is the number of non-zero weights in the GGI operator. In the parametrized complexity terminology [Niedermeier, 2006], this means that the GGI stable marriage problem belongs to class XP for parameter  $K$ .

We adopt a brute-force approach to solve the problem in  $O(2^K n^{2K+4})$ . Let

$$\mathbf{t}(\mathbf{x}) = ((d(m_1, \mathbf{x}), m_1, \mu_{\mathbf{x}}(m_1)), \dots, (d(m_n, \mathbf{x}), m_n, \mu_{\mathbf{x}}(m_n)), \\ (d(w_1, \mathbf{x}), w_1, \mu_{\mathbf{x}}(w_1)), \dots, (d(w_n, \mathbf{x}), w_n, \mu_{\mathbf{x}}(w_n)))$$

denote the vectors of triples  $(d(a_i, \mathbf{x}), a_i, \mu_{\mathbf{x}}(a_i))$  induced by stable marriage  $\mathbf{x}$ , where  $d(a_i, \mathbf{x})$  is the dissatisfaction of agent  $a_i$  when matched with  $\mu_{\mathbf{x}}(a_i)$ . We denote by  $T^\downarrow(\mathbf{x})$  the set of vectors  $\mathbf{t}^\downarrow(\mathbf{x})$  that can be obtained from  $\mathbf{t}(\mathbf{x})$  by sorting the triples in decreasing order of dissatisfactions. The projection of a vector  $\mathbf{t}^\downarrow(\mathbf{x}) \in T^\downarrow(\mathbf{x})$  on the  $K$  first components is denoted by  $\mathbf{t}_K^\downarrow(\mathbf{x})$ . We denote by  $T_K^\downarrow(\mathbf{x})$  the set  $\{\mathbf{t}_K^\downarrow(\mathbf{x}) : \mathbf{t}^\downarrow(\mathbf{x}) \in T^\downarrow(\mathbf{x})\}$ .

For instance, assume that  $\mathbf{t}(\mathbf{x}) = ((1, m_1, w_2), (2, m_2, w_1), (2, w_1, m_2), (1, w_2, m_1))$ . Then the set  $T^\downarrow(\mathbf{x})$  is

$$\{((2, m_2, w_1), (2, w_1, m_2), (1, m_1, w_2), (1, w_2, m_1)), \\ ((2, w_1, m_2), (2, m_2, w_1), (1, m_1, w_2), (1, w_2, m_1)), \\ ((2, m_2, w_1), (2, w_1, m_2), (1, w_2, m_1), (1, m_1, w_2)) \\ ((2, w_1, m_2), (2, m_2, w_1), (1, w_2, m_1), (1, m_1, w_2))\}$$

and the set  $T_2^\downarrow(\mathbf{x})$  is  $\{((2, m_2, w_1), (2, w_1, m_2)), ((2, w_1, m_2), (2, m_2, w_1))\}$ .

The idea is to enumerate all vectors in  $T_K^\downarrow = \cup_{\mathbf{x} \in \mathcal{X}} T_K^\downarrow(\mathbf{x})$  without redundancy. The polynomiality of the approach follows from the fact that  $|T_K^\downarrow| \leq (2n^2)^K$  because the number of distinct triples is upper bounded by  $2n^2$ . Note that we have:

$$\min_{\mathbf{x} \in \mathcal{X}} \text{GGI}_\lambda(\mathbf{d}(\mathbf{x})) = \min_{\mathbf{t} \in T_K^\downarrow} \text{GGI}_\lambda(\mathbf{t})$$

because  $\lambda_i = 0$  for all  $i > K$ , where, by abuse of notation, we denote by  $\text{GGI}_\lambda(\mathbf{t})$  the value of the GGI operator applied to the vector of dissatisfactions obtained from  $\mathbf{t}^\downarrow$ . Hence identifying an optimal GGI stable marriage will be performed by finding a vector  $\mathbf{t} \in T_K^\downarrow$  minimizing the GGI operator and computing a corresponding stable marriage.

**Example 9** *Coming back to the instance of Example 1, assume that  $K = 2$  and that the disutility function is defined by  $d(i) = i$ . Then, our enumeration algorithm would produce the following set  $\mathcal{T}_2^\downarrow$ :*

<sup>1</sup> Note that vector  $\mathbf{t} \in T_K^\downarrow$  is incomplete as it only has  $K$  components, but it is sufficient to apply the GGI operator because  $\lambda_i = 0$  for all  $i > K$ .

$$\begin{aligned} & \{((10, w_{10}, m_9), (7, w_4, m_6)), ((10, w_{10}, m_9), (7, w_5, m_7)), \\ & \quad ((10, w_{10}, m_9), (7, w_6, m_5)), ((10, w_{10}, m_9), (7, w_7, m_4)), \\ & \quad ((10, w_{10}, m_9), (5, m_4, w_6)), ((10, w_{10}, m_9), (5, m_5, w_7)), \\ & \quad ((10, w_{10}, m_9), (5, m_6, w_5)), ((10, w_{10}, m_9), (5, m_7, w_4)), \\ & \quad ((9, w_{10}, m_8), (5, m_4, w_6)), ((9, w_{10}, m_8), (5, m_5, w_7)), \\ & \quad ((9, w_{10}, m_8), (5, m_6, w_5)), ((9, w_{10}, m_8), (5, m_7, w_4)), \\ & \quad ((9, w_{10}, m_8), (5, m_8, w_{10})), ((9, w_{10}, m_8), (5, m_9, w_9)), \\ & \quad \quad \quad ((9, w_{10}, m_8), (5, m_{10}, w_8))\} \end{aligned}$$

For this instance, the optimal GGI value is therefore necessarily  $9\lambda_1 + 5\lambda_2$ . Note that, in most cases, the optimal GGI value depends on  $\lambda$  (it is not the case here because  $(9, 5)$  dominates componentwise all vectors of dissatisfactions obtained from  $\mathcal{T}_2^\downarrow$ ).

We now describe our enumeration algorithm. Algorithm 1 builds set  $T_K^\downarrow$  by induction using the following formula:

$$\begin{aligned} T_0^\downarrow &= \{()\} \\ T_k^\downarrow &= \{\mathbf{v} \circ t : \mathbf{v} \in T_{k-1}^\downarrow \text{ and } t \in T(\mathbf{v})\} \text{ where } T(\mathbf{v}) = \{t_k : \mathbf{t} \in T_N^\downarrow \text{ s.t. } (t_1, \dots, t_{k-1}) = \mathbf{v}\} \end{aligned}$$

The aim of Algorithm 2 is to compute  $T(\mathbf{v})$ , i.e., the set of possible triples for the  $k^{\text{th}}$  component of a vector in  $T_k^\downarrow$  starting by the  $(k-1)$ -vector  $\mathbf{v}$ . The idea is to impose restrictions on the considered stable marriages so that the least satisfied agents as well as their matches correspond to the ones in  $\mathbf{v}$ . For this purpose, we impose mandatory rotations (set  $\text{IN}_\mathbf{v}$ ) and forbidden rotations (set  $\text{OUT}_\mathbf{v}$ ). Note that, each time a rotation is made mandatory (resp. forbidden), the set of its ancestors (resp. descendants), denoted by  $\text{Anc}(\rho)$  (resp.  $\text{Desc}(\rho)$ ), are also made mandatory (resp. forbidden) so that  $\text{IN}_\mathbf{v}$  (resp.  $P \setminus \text{OUT}_\mathbf{v}$ ) remains a closed set of rotations. For each triple  $(d, a, a')$  belonging to  $\mathbf{v}$ , we ensure that agent  $a$  is matched with agent  $a'$  by making rotation  $\rho_{\text{get}}(a, a')$  mandatory and  $\rho_{\text{break}}(a, a')$  forbidden (Lines 3–5). Additionally, to ensure that the  $k$  least satisfied agents are indeed those involved in  $\mathbf{v}$ , we put a threshold on the dissatisfactions of the agents in  $\mathcal{A}_\nabla = \mathcal{M} \cup \mathcal{W} \setminus \{a : (d, a, a') \in \mathbf{v}\}$ . Note that the set  $\mathcal{A}_\nabla$  is updated in Line 3. Let  $d_{\min}(\mathbf{v})$  denote the dissatisfaction of the last triple in  $\mathbf{v}$  (i.e., the lowest level of dissatisfaction in  $\mathbf{v}$ ). The dissatisfactions of the agents in  $\mathcal{A}_\nabla$  should not be strictly greater than  $d_{\min}(\mathbf{v})$ . This condition is imposed by using again sets  $\text{IN}_\mathbf{v}$  and  $\text{OUT}_\mathbf{v}$ . More precisely, given a rotation  $\rho = (m_{i_0}, w_{i_0}), \dots, (m_{i_{r-1}}, w_{i_{r-1}})$ , we define  $d_{\max}^w(\rho) = \max_{k=0, \dots, r-1} d(w_{i_k}, m_{i_k})$  the highest dissatisfaction of a woman involved in  $\rho$  before  $\rho$  is eliminated, and  $d_{\max}^m(\rho) = \max_{k=0, \dots, r-1} d(m_{i_k}, w_{i_{k+1}})$  the highest dissatisfaction of a man involved in  $\rho$  after  $\rho$  is eliminated. To make sure that the agents in  $\mathcal{A}_\nabla$  have a dissatisfaction lower than or equal to  $d_{\min}(\mathbf{v})$ , we make mandatory (resp. forbidden) any rotation  $\rho \in P \setminus \text{OUT}_\mathbf{v}$  (resp.  $P \setminus \text{IN}_\mathbf{v}$ ) such that  $d_{\max}^w(\rho) > d_{\min}(\mathbf{v})$  (resp.  $d_{\max}^m(\rho) > d_{\min}(\mathbf{v})$ ) (Lines 6–7, resp. Lines 8–9). The enumeration of the triples in  $T(\mathbf{v})$  is performed by branching on the gender (man or woman) of the agent that will realize the  $k^{\text{th}}$  highest dissatisfaction. We denote by  $T_W(\mathbf{v})$  (resp.  $T_M(\mathbf{v})$ ) the set of triples  $(d, a, a') \in T(\mathbf{v})$  where  $a \in \mathcal{W}$  (resp.  $a \in \mathcal{M}$ ). We have of course  $T_W(\mathbf{v}) \cup T_M(\mathbf{v}) = T(\mathbf{v})$ . Algorithm 3 enumerates the triples in  $T_W(\mathbf{v})$  while Algorithm 4 enumerates the triples in  $T_M(\mathbf{v})$  (Line 10 of Algorithm 2). The validity of the approach follows from the validity of Algorithms 3 and 4.

*Validity of the approach.* The operations of Algorithms 3 and 4 are similar. They proceed in the spirit of the algorithm proposed by Gusfield [1987] for determining a minmax stable marriage. Let  $\mathbf{x}_R$  denote the stable marriage corresponding to a set  $R$  of rotations. Note that we have built sets  $\text{IN}_v$  and  $\text{OUT}_v$  such that if  $R \cap \text{IN}_v = \text{IN}_v$  and  $R \cap \text{OUT}_v = \emptyset$  then  $v \in T_{k-1}^\downarrow(\mathbf{x}_R)$ . Furthermore, the special case  $\mathbf{x}_{\text{IN}_v}$  (resp.  $\mathbf{x}_{P \setminus \text{OUT}_v}$ ) is the stable marriage compatible with  $\text{IN}_v$  and  $\text{OUT}_v$  that satisfy most the men (resp. women) as it takes as few (resp. much) rotations as allowed by sets  $\text{IN}_v$  and  $\text{OUT}_v$ . We only explain the operation of Algorithm 3, because the operation of Algorithm 4 is symmetric.

The aim of Algorithm 3 is to enumerate all triples  $(d, a, a')$  in  $T_W(v)$ . Notably, we will enumerate these triples by nonincreasing values of  $d$  by exploring carefully the set of stable marriages compatible with sets  $\text{IN}_v$  and  $\text{OUT}_v$ . More precisely, at each iteration  $i$  of the algorithm (loop `while` in Line 5) we will consider a stable marriage  $\mathbf{x}_i$  compatible with sets  $\text{IN}_v$  and  $\text{OUT}_v$  such that all women are always better off in  $\mathbf{x}_i$  than in  $\mathbf{x}_{i-1}$  for  $i \neq 0$  (with at least one woman strictly better off). At each iteration, the new triples are found by looking at set  $W_i$  that includes all women in  $\mathcal{A}_v$  whose dissatisfaction can be ranked in  $k^{\text{th}}$  position in  $\mathbf{x}_i$ , i.e., whose dissatisfaction is equal to  $d_k^\downarrow(\mathbf{x}_i)$  (Lines 3 and 13)<sup>2</sup>.

Obviously, for the women, the worst stable marriage compatible with  $\text{IN}_v$  and  $\text{OUT}_v$  is  $\mathbf{x}_{\text{IN}_v}$ . If no woman can be ranked in  $k^{\text{th}}$  position w.r.t. stable marriage  $\mathbf{x}_{\text{IN}_v}$ , then no woman can be ranked in  $k^{\text{th}}$  position for *any* stable marriage compatible with  $\text{IN}_v$ . Indeed, eliminating additional rotations would only increase the dissatisfactions of men and decrease the dissatisfactions of women. Otherwise the recurrence is initialized with  $\mathbf{x}_0 = \mathbf{x}_{\text{IN}_v}$  and stable marriage  $\mathbf{x}_{i+1}$  is obtained from  $\mathbf{x}_i$  by eliminating rotation  $\rho_{\text{break}}(m, w)$  (and all required ancestors) for all woman  $w$  in  $W_i$  so that their dissatisfactions are strictly decreased (Line 10). Loop `while` stops if one of the following conditions occurs:

- if  $W_i = \emptyset$ , it means that only men can be ranked in  $k^{\text{th}}$  position in  $\mathbf{x}_i$ ; as eliminating rotations will only improve the situation of women and deteriorate the situation of men, we can safely conclude that all triples in  $T_W(v)$  have been enumerated;
- if at least one rotation  $\rho_{\text{break}}(m, w)$  does not exist or is forbidden (i.e.,  $(m, w) \in \mathbf{x}_{P \setminus \text{OUT}_v}$ ); indeed, in this case, we can conclude that it is not possible to find a triple in  $T_W(v)$  with a dissatisfaction strictly less than the current value  $d_k^\downarrow(\mathbf{x}_i)$  (the boolean `Flag` is then set to `True` in Line 9).

*Complexity analysis and proof of termination.* In Algorithm 3, at every step  $i$  of the `while` loop, all agents in  $W_i$  share the same dissatisfaction level  $d_k^\downarrow(\mathbf{x}_i)$ . Furthermore, for all  $i \neq 0$ , we have that  $d_k^\downarrow(\mathbf{x}_i) < d_k^\downarrow(\mathbf{x}_{i-1})$ . As there are only  $n$  dissatisfaction levels (corresponding to the  $n$  possible ranks), the `while` loop necessarily terminates in  $O(n)$  iterations. The nested `for` loop also terminates in  $O(n)$  iterations because there can be at most  $n$  women in  $W_i$ . All instructions inside the `for` loop are in  $O(1)$ , except the instruction in Line 10 which is in  $O(n^2)$  (the number of rotations is upper bounded by  $n(n-1)/2$ ). Overall, Algorithm 3 is in  $O(n^4)$ . The analysis of Algorithm 4 is similar. In Algorithm 2, Lines 4 and 5 are in  $O(n^2)$ , hence the `for` loop in Line 2 is in  $O((k-1)n^2)$ , therefore in  $O(n^3)$  as  $k \leq 2n$ . Lines 6–9 are in  $O(n^4)$ . Since we have shown that both calls in Line 10 are in  $O(n^4)$ , the overall complexity of Algorithm 2 is  $O(n^4)$ . Finally, the complexity of the three nested `for` loops in Algorithm 1 is  $O(\sum_{k=1}^K (2n^2)^{k-1} (n^4 + 2n^2))$  because:

- the cardinality of set  $T_{(k-1)}^\downarrow$  in Line 4 is upper bounded by  $(2n^2)^{k-1}$  (there are at most

<sup>2</sup> We recall that  $d_k^\downarrow(\mathbf{x})$  denotes the  $k^{\text{th}}$  component of vector  $\mathbf{d}(\mathbf{x})$  when sorted by nonincreasing values.

$2n^2$  triples, and  $k - 1$  components per vector of triples in  $T_{(k-1)}^\downarrow$ );

– Line 5 is in  $O(n^4)$ ;

– Lines 6–7 are in  $O(2n^2)$ .

Overall, the complexity of Algorithm 1 thus is  $O(2^K n^{2K+4})$ .

*Final remarks.* At the end of Algorithm 1, one obtains a set  $T_K^\downarrow$  of vectors of triples. Within this set, one can choose a vector  $\mathbf{v}^*$  which realizes:

$$\min_{\mathbf{v} \in T_K^\downarrow} \text{GGI}_\lambda(\mathbf{v}) = \min_{\mathbf{x} \in \mathcal{X}} \text{GGI}_\lambda(\mathbf{d}(\mathbf{x})).$$

Given this vector  $\mathbf{v}^*$ , any stable marriage  $\mathbf{x}^*$  such that  $\mathbf{v}^* \in T_K^\downarrow(\mathbf{x}^*)$  verifies

$$\text{GGI}_\lambda(\mathbf{d}(\mathbf{x}^*)) = \min_{\mathbf{x} \in \mathcal{X}} \text{GGI}_\lambda(\mathbf{d}(\mathbf{x})).$$

Given  $\mathbf{v}^*$ , it is easy to compute a stable marriage  $\mathbf{x}^*$  such that  $\mathbf{v}^* \in T_K^\downarrow(\mathbf{x}^*)$ . In particular,  $\mathbf{x}_{\text{IN}_{\mathbf{v}^*}}$  (resp.  $\mathbf{x}_{P \setminus \text{OUT}_{\mathbf{v}^*}}$ ) is a best possible stable marriage for men (resp. women) where sets  $\text{IN}_{\mathbf{v}^*}$  and  $\text{OUT}_{\mathbf{v}^*}$  are generated in the same fashion as in Algorithm 2 (Lines 1–9).

---

### Algorithm 1: Enumerate

---

**input** : the GGISM instance and the value of  $K$   
**output**:  $T_K^\downarrow$

- 1  $T_0^\downarrow \leftarrow \{()\}$
- 2 **for**  $k = 1, \dots, K$
- 3      $T_k^\downarrow \leftarrow \emptyset$
- 4     **for**  $\mathbf{v} \in T_{k-1}^\downarrow$
- 5          $T \leftarrow \text{NextTriples}(\mathbf{v}, k)$
- 6         **for**  $t \in T$
- 7              $T_k^\downarrow \leftarrow T_k^\downarrow \cup \{\mathbf{v} \circ t\}$
- 8 **return**  $T_K^\downarrow$

---



---

### Algorithm 2: NextTriples

---

**input** : vector  $\mathbf{v}$  of imposed triples, index  $k$  of the next triple  
**output**: set  $T$  of possible next triples

- 1  $\text{IN}_{\mathbf{v}} \leftarrow \emptyset$ ;  $\text{OUT}_{\mathbf{v}} \leftarrow \emptyset$ ;  $\mathcal{A}_{\overline{\mathbf{v}}} \leftarrow \mathcal{M} \cup \mathcal{W}$
- 2 **for**  $i = 1, \dots, k - 1$
- 3      $(a, a', d) = v_i$ ,  $\mathcal{A}_{\overline{\mathbf{v}}} \leftarrow \mathcal{A}_{\overline{\mathbf{v}}} \setminus \{a\}$
- 4      $\text{IN}_{\mathbf{v}} \leftarrow \text{IN}_{\mathbf{v}} \cup \{\rho_{\text{get}}(a, a')\} \cup \text{Anc}(\rho_{\text{get}}(a, a'))$
- 5      $\text{OUT}_{\mathbf{v}} \leftarrow \text{OUT}_{\mathbf{v}} \cup \{\rho_{\text{break}}(a, a')\} \cup \text{Desc}(\rho_{\text{break}}(a, a'))$
- 6 **for**  $\rho \in P \setminus \text{OUT}_{\mathbf{v}}$  s.t.  $d_{\text{max}}^w(\rho) > d_{\text{min}}(v)$
- 7      $\text{IN}_{\mathbf{v}} \leftarrow \text{IN}_{\mathbf{v}} \cup \{\rho\} \cup \text{Anc}(\rho)$
- 8 **for**  $\rho \in P \setminus \text{IN}_{\mathbf{v}}$  s.t.  $d_{\text{max}}^m(\rho) > d_{\text{min}}(v)$
- 9      $\text{OUT}_{\mathbf{v}} \leftarrow \text{OUT}_{\mathbf{v}} \cup \{\rho\} \cup \text{Desc}(\rho)$
- 10 **return**  $\text{NextWomen}(\text{IN}_{\mathbf{v}}, \text{OUT}_{\mathbf{v}}, k, \mathcal{A}_{\overline{\mathbf{v}}}) \cup \text{NextMen}(\text{IN}_{\mathbf{v}}, \text{OUT}_{\mathbf{v}}, k, \mathcal{A}_{\overline{\mathbf{v}}})$

---

**Algorithm 3:** NextWomen

---

**input** : set  $\text{IN}_v$  and  $\text{OUT}_v$  of mandatory and forbidden rotations, index  $k$  of the next triple, set  $\mathcal{A}_v$   
**output**:  $T_W(v)$

- 1 Compute  $\mathbf{x}_{\text{IN}_v}$  and  $\mathbf{x}_{P \setminus \text{OUT}_v}$
- 2  $T \leftarrow \emptyset$ ;  $R \leftarrow \text{IN}_v$ ;  $i \leftarrow 0$ ;  $\mathbf{x}_i \leftarrow \mathbf{x}_R$
- 3  $W_i \leftarrow \{w \in \mathcal{A}_v \cap \mathcal{W} : d(w, \mathbf{x}_i) = d_k^{\downarrow}(\mathbf{x}_i)\}$
- 4 Flag  $\leftarrow$  False
- 5 **while**  $W_i \neq \emptyset$  **do**
- 6     **for**  $w \in W_i$
- 7         let  $m$  be the match of  $w$  in  $\mathbf{x}_i$
- 8          $T \leftarrow T \cup \{(d(w, m), w, m)\}$
- 9         **if**  $(m, w) \in \mathbf{x}_{P \setminus \text{OUT}_v}$  **then** Flag  $\leftarrow$  True
- 10        **else**  $R \leftarrow R \cup \rho_{\text{break}}(m, w) \cup \text{Anc}(\rho_{\text{break}}(m, w))$
- 11     **if** Flag **then return**  $T$
- 12      $i \leftarrow i + 1$ ;  $\mathbf{x}_i \leftarrow \mathbf{x}_R$
- 13      $W_i \leftarrow \{w \in \mathcal{A}_v \cap \mathcal{W} : d(w, \mathbf{x}_i) = d_k^{\downarrow}(\mathbf{x}_i)\}$
- 14 **return**  $T$

---

**Algorithm 4:** NextMen

---

**input** : set  $\text{IN}_v$  and  $\text{OUT}_v$  of mandatory and forbidden rotations, index  $k$  of the next triple, set  $\mathcal{A}_v$   
**output**:  $T_M(v)$

- 1 Compute  $\mathbf{x}_{\text{IN}_v}$  and  $\mathbf{x}_{P \setminus \text{OUT}_v}$
- 2  $T \leftarrow \emptyset$ ;  $R \leftarrow P \setminus \text{OUT}_v$ ;  $i \leftarrow 0$ ;  $\mathbf{x}_i \leftarrow \mathbf{x}_R$
- 3  $M_i \leftarrow \{m \in \mathcal{A}_v \cap \mathcal{M} : d(m, \mathbf{x}_i) = d_k^{\downarrow}(\mathbf{x}_i)\}$
- 4 Flag  $\leftarrow$  False
- 5 **while**  $M_i \neq \emptyset$  **do**
- 6     **for**  $m \in M_i$
- 7         let  $w$  be the match of  $m$  in  $\mathbf{x}_i$
- 8          $T \leftarrow T \cup \{(d(m, w), m, w)\}$
- 9         **if**  $(m, w) \in \mathbf{x}_{\text{IN}_v}$  **then** Flag  $\leftarrow$  True
- 10        **else**  $R \leftarrow R \setminus (\rho_{\text{get}}(m, w) \cup \text{Des}(\rho_{\text{get}}(m, w)))$
- 11     **if** Flag **then return**  $T$
- 12      $i \leftarrow i + 1$ ;  $\mathbf{x}_i \leftarrow \mathbf{x}_R$
- 13      $M_i \leftarrow \{m \in \mathcal{A}_v \cap \mathcal{M} : d(m, \mathbf{x}_i) = d_k^{\downarrow}(\mathbf{x}_i)\}$
- 14 **return**  $T$

---

**6 Conclusion**

In this paper, we have shown that the minimization of a Generalized Gini Index (GGI) of the dissatisfactions of men and women in a stable marriage problem is an NP-hard problem. Then, we have proposed a polynomial time 2-approximation algorithm for the problem, based on a rounding of the optimal solution to the linear programming relaxation of the problem. Lastly, we have shown that minimizing a GGI of the dissatisfactions of men and women in a stable marriage is in the class XP with respect to the number of strictly positive weights in the GGI operator.

For future works, following Aziz and Klaus [2017], it could be worth investigating the randomized version of the GGI stable marriage problem. By *randomized*, we mean that we consider mixed stable marriages, and not only deterministic stable marriages. A mixed stable marriage is a probability distribution over stable marriages. This enlargement of the



set of feasible solutions could make it possible to enhance the optimal GGI value (where the GGI operator is applied to the vector of expected dissatisfactions of the agents). Note that the relaxed solution we compute in the first step of the 2-approximation algorithm proposed in Section 4 can be converted into a mixed stable marriage by using a trick proposed by Teo and Sethuraman [1998]. It turns out that the obtained approach returns an optimal marriage for the randomized variant of the GGI stable marriage problem. A more thorough investigation of the randomized GGI stable marriage problem is underway.

## References

- Haris Aziz and Bettina Klaus. Random matching under priorities: Stability and no envy concepts. *arXiv preprint arXiv:1707.01231*, 2017.
- Péter Biró. Applications of matching models under preferences. *Trends in Computational Social Choice*, page 345, 2017.
- Christine T Cheng. Understanding the generalized median stable matchings. *Algorithmica*, 58(1):34–51, 2010.
- Edith Elkind and Anisse Ismaili. OWA-based extensions of the chamberlin–courant rule. In *International Conference on Algorithmic Decision Theory*, pages 486–502. Springer, 2015.
- Tomás Feder. Network flow and 2-satisfiability. *Algorithmica*, 11(3):291–319, Mar 1994.
- Janos Fodor, Jean-Luc Marichal, and Marc Roubens. Characterization of the ordered weighted averaging operators. *IEEE Trans. on Fuzzy Systems*, 3(2):236–240, 1995.
- David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- Judy Goldsmith, Jérôme Lang, Nicholas Mattei, and Patrice Perny. Voting with rank dependent scoring rules. In *AAAI*, pages 698–704, 2014.
- Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.
- Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.
- Tobias Heinen, Nhan-Tam Nguyen, and Jörg Rothe. Fairness and rank-weighted utilitarianism in resource allocation. In *International Conference on Algorithmic Decision Theory*, pages 521–536. Springer, 2015.
- Robert W Irving and Paul Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, 1986.
- Robert W Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the “optimal” stable marriage. *Journal of the ACM (JACM)*, 34(3):532–543, 1987.
- Adam Kasperski and Paweł Zieliński. Combinatorial optimization problems with uncertain costs and the OWA criterion. *Theoretical Computer Science*, 565:102–112, 2015.
- Akiko Kato. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial and Applied Mathematics*, 10(1):1, 1993.
- Rajeev Kohli, Ramesh Krishnamurti, and Prakash Mirchandani. The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7(2):275–283, 1994.
- Julien Lesca, Michel Minoux, and Patrice Perny. The fair OWA one-to-one assignment problem: NP-hardness and polynomial time special cases. *Algorithmica*, pages 1–26, 2018.
- David F. Manlove. *Algorithmics of Matching Under Preferences*. Series on theoretical computer science. World Scientific, 2013.

- Eric McDermid and Robert W Irving. Sex-equal stable matchings: Complexity and exact algorithms. *Algorithmica*, 68(3):545–570, 2014.
- Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2006.
- Włodzimierz Ogryczak and Tomasz Śliwiński. On solving linear programs with the ordered weighted averaging objective. *European J. of Operational Research*, 148(1):80–91, 2003.
- Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.
- Chung-Piaw Teo and Jay Sethuraman. The geometry of fractional stable matchings and its applications. *Mathematics of Operations Research*, 23(4):874–891, 1998.
- John H. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153, 1989.
- John A. Weymark. Generalized gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981.
- Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. on Sys., Man, and Cybernetics*, 18(1):183–190, 1988.