

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

PARETO-IMPROVEMENT IN MATCHING PROBLEMS TROUGH GENETIC ALGORITHMS

Marelli, Virginie; Niccolo', Antonio; Carletti, Timoteo

Publication date:
2013

Document Version
Early version, also known as pre-print

[Link to publication](#)

Citation for published version (HARVARD):
Marelli, V, Niccolo', A & Carletti, T 2013, *PARETO-IMPROVEMENT IN MATCHING PROBLEMS TROUGH GENETIC ALGORITHMS*. naXys Technical Report Series, no. 8, vol. 4, vol. 4, 8 edn, Namur center for complex systems, Namur.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

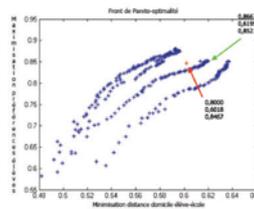
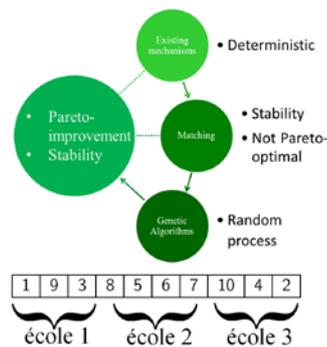
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

PARETO-IMPROVEMENT IN MATCHING
PROBLEMS THROUGH GENETIC ALGORITHMS

By Virginie MARELLI, Antonio NICOLÒ and
Timoteo CARLETTI

Report naXys-08-2013

19 August 2013



Namur Center for Complex Systems

University of Namur
8, rempart de la vierge, B5000 Namur (Belgium)

<http://www.naxys.be>

PARETO-IMPROVEMENT IN MATCHING PROBLEMS THROUGH GENETIC ALGORITHMS

VIRGINIE MARELLI, ANTONIO NICOLÒ, TIMOTEO CARLETTI

Abstract

Often matching problems and problems of coalition formation, are computationally hard to solve and existing algorithms are able to find allocations that are stable but not Pareto-optimal. We show in two specific applications, hedonic games and school choice with constraints, that Genetic Algorithms can be successfully applied to find outcomes that Pareto-improve over allocations obtained by existing algorithms.

1. INTRODUCTION

Very often operational optimisation problems do not fit in the “classical” optimisation framework, that requires smooth functions and the possibility to access, speedily and easily, to their derivatives. Actually the former ones usually lack both requirements, functions are defined on discrete variables, sometimes also on categorical ones, and thus derivatives do not exist; however, even when derivatives are computable in principle, the huge number of involved variables or the definition of function through recursive schemes, make the computation of the latter almost impossible.

Some of the above problems, can be solved or their impact be reduced, using heuristic methods. *Genetic Algorithms*, for short GAs in the following, belong to this class. They are directly inspired by biology and Darwinian evolution: organisms are suited to their habitats (Adaptation); offsprings are similar but not equal to their parents (Inheritance); new, better adapted types of organisms, emerge and those that fail to change adequately are subject to extinction (Natural selection). Thus the fittest individuals have a high chance of having a large number of offsprings and finally their traits can spread across the population, generation by generation.

GAs are aimed at exploiting these key concepts for problem solving. The algorithm first finds partial solutions, that are then filtered and improved to select the “best ones”, that is the ones closer to the optimal solution we

are looking for. By their very first definition, GAs are blind search optimisation methods, exploring the solutions' space without being constrained by derivative or local minima.

They also avoid numerical errors since they only use value of the objective function and thus no gradient or approximate models should be computed.

For all these reasons, they are prone to be applied to a wide range of problems, without too much effort. Of course, the more is possible to adapt a GA to the problem under investigation, the better will be the quality of the solution. The aim of this paper is to apply GAs on two distinct problems: the *hedonic games* [ABH11, ABS13, BJ02] and the *students-school assignment* [GS62, Pat11, RS90, Abd05, AS03]. Usually, in game theory, genetic algorithms are used to reproduce the behaviour that people have in the laboratory, playing games (adaptive behaviour) ¹. However, in this paper, we chose a different approach: using genetic algorithm to look for outcomes that improve over allocations obtained by other algorithms. To the best of our knowledge this is the first time that GAs are applied, in this way, to a matching problem, being the literature very scarce [AC99].

Matching is a well known problem and relevant to many fields of economics research; the available literature provides nice theoretical results, but whose operational implementation is often by far unreachable, because direct application of such results will produce algorithm whose time complexity will be larger than polynomial [ABH11, Bal04], but also very cumbersome whenever we want to add some constraints. Let us finally observe that because of the combinatoric nature of the problems, brute force enumeration of all possible cases becomes rapidly impossible even for small sizes problems.

In a first part of this paper we consider the problem of the formation of coalitions and in particular the renowned *hedonic games*.

A finite set of players must be divided according to their preferences, into disjoint groups, called coalitions, that should partition all the individuals. In the following we will focus on anonymous games with single-peaked preferences, meaning that players only care about the size of their coalition or the amount of public good it produces. A partition is stable if there is no envy among the players, let us observe that stability is a hard condition to be satisfied and hence we must often rely on a relaxed weaker notion, to be able to have positive results. In this setting, Bogomolnaia et al. [BJ02] propose an algorithm which gives a stable and weakly Pareto-optimal² partition under some suitable conditions. Unfortunately, if these assumptions are not met, the algorithm can only find a stable partition. As already stated most of the existing algorithms are not totally satisfying, we thus hereby introduce a new method based on GAs that allows us to Pareto-improve the result by Bogomolnaia et al. [BJ02]. Such a Pareto-improvement is very interesting

¹For more details, see Haruvy et al. [HRU06] and Ünver [Unv05].

²Weakly Pareto-optimal means that there exists no partition that all players strictly prefer.

since some players are better off, while the remaining ones are left with their previous assignment.

The second matching problem we are interested in, is the *matching in school choice with constraints and indifferences*. This problem is more complex than the previous one and moreover, schools and students around the world are currently faced to it: assigning students to schools has never been easy since there is always someone to complain. Giving every student his preferred school is not always possible because of the limited schools' capacity. Moreover, some well appreciated schools are overwhelmed by students proposals and cannot deal with it. That is why it is common to use a central house to avoid congestion and to try to satisfy students as well as schools. One usually uses the Deferred Acceptance Algorithm (DAA) [GS62] which yields a stable matching that is optimal for students. Here, stable means that no-one has incentive to break the deal (the inscription). Furthermore, this direct mechanism is strategy-proof [GS62] for students, i.e. their dominant strategy is to reveal their true preferences.

Since every school chooses the way it ranks students, there exists some inequalities among the latter. To solve this problem, states introduce quotas to fight against social segregation or economic segregation [Abd05, EHYY12]. Every student gets a type, for instance race or social status; quotas are introduced to regulate the number of students of each type that should be present in every school. Let us observe that either quotas can be strictly respected (hard bounds approach) or they should serve more as guidelines (soft bounds approach) and thus students' preferences are considered firstly, see [EHYY12]. Another source of complexity arises when schools can not strictly rank all students and a tie-breaking rule is needed in order to use the DAA [APR09, EE08]. As a consequence of this random sorting, the DAA does not longer find necessarily an optimal matching. This loss of utility raises the following question: can efficiency be restored? Is there a mechanism that Pareto-improves upon the results obtained using the existing algorithms? Our goal is to positively answer to these questions by introducing a new method based on GAs to this problem, more precisely we will show that the efficiency can be partly restored and the solution Pareto-improved.

The paper is organized as follows. In the next section 2 we will provide a short but comprehensive introduction to genetic algorithms; then in section 3 we will introduce the Hedonic Games, some results available in the literature and our new method. Section 4 will be devoted to the presentation of the Student–School matching problem, once again after some definitions and already known results, we will present our method and numerical simulations. Finally Section 5 will be devoted to rise our conclusions and to present some open questions that need some further studies.

2. GENETIC ALGORITHMS: A RUSH COURSE

The goal of this section is to present a short introduction of Genetic Algorithms; the literature is extremely large, we thus prefer let the interested reader to consult Goldberg, [Gol89] and Mitchel, [Mit98]. GAs are optimisation methods inspired by evolutionary genetic, their goal is to find individuals, i.e. solutions, who best fit to their environment, i.e. whose quality is very high in the under scrutiny problem.

Because, evolution is largely based on genetic, the vocabulary of GA is very close to the latter. For instance, a chromosome will denote a partial solution, and it will be composed by genes which determine the quality of the solution, that is the fitness of the chromosome. Each gene can take several possible values, called allele, and the position of the gene in the chromosome is named, the locus. A set of chromosomes forms the population.

The GAs can be applied on a wide range of problems because they don't use derivative neither they assume continuity for the objective function. They are mainly blind search procedures, exploiting only the coding and the objective function to determine plausible solutions for the next steps. In this section, we will limit ourselves to present a basic GA, however as already remarked, the goodness of a GA can greatly increase whenever the researcher adapt it as much as possible to the problem to be solved. We will present more details about this point in the 3.5 and in the 4.3.

2.1. Encoding the problem. The first step is to encode the problem: associate to each chromosome a value for the objective function. Using, for instance, a simple binary value ³ for each gene, a chromosome will be a string of 0's and 1's; the number of genes that should be used will be dictated by the problem. In any case once this length has been fixed, all the population will be composed by chromosomes of the same length, N . The main goal of the coding is to propose a way to compute a numerical value of the objective function starting from a chromosome. Assume for instance the problem is to maximise a function of a real variable, then a possible coding could be to associate to each chromosome a number written in its binary form. Let us observe that often the GA is designed as not to maximise directly the objective function but a related function called *fitness function*, the fitness should give a measure of the level of adaptation of an individual in its environment; in the rest of this section we will use both terms interchangeably. The values of the alleles in a chromosome define its *genotype*, while the value of the objective function on a chromosome is called *phenotype*.

2.2. Creation of the population. Once the encoding has been chosen, we have to create the initial population of chromosomes, that is a set of potential solutions. However, very often the latter are chosen at random in the

³We will see in the next sections that this encoding is not sufficient to solve the hedonic problem or the students-schools matching problem.

solution space, in such a way to avoid any bias. The size of the population, N_p , depends once again on the problem to be solved: a larger population will allow to explore, in principle, a larger portion of the solution space ⁴, however the algorithm speed is inversely proportional to the population size.

2.3. The main GA loop. Starting from the initial population a new one, with the same size N_p , will be constructed using three operators acting directly on the chromosomes: *selection*, *crossover* and *mutation*. This loop will be iterated until a stop criterion will be reached; usually this can be given either in terms of *maximum number of generations* allowed, that is how many times the loop should be performed, either in terms of the objective function, for instance the mean value of the latter on the whole population should be larger than a threshold.

Let us analyse in details each operator.

2.3.1. Selection. Individuals are selected in the current population, G_0 , to form an intermediary population $G_1^{(s)}$, usually this population has the same size of the initial one. Copying natural evolution, we allow best fitted individuals to have a larger chance to survive and thus to pass into $G_1^{(s)}$, in other terms the selection mechanism will be based on the fitness function. There are several possibilities to perform such selection, the simplest and most used one is the roulette wheel; other methods do exist, for instance the tournament, the elitism. In the roulette wheel, the fitness is computed for each individual in G_0 , then the probability to select individual i -th is given by $f_i / \sum f_i$, where f_i is the fitness of the i -th individual.

2.3.2. Crossover. The crossover operator mimics the reproduction of the individuals. Two chromosomes are selected, usually with a uniform random probability, and with probability p_X they are recombined and their offsprings set into an intermediary population $G_1^{(X)}$, otherwise with probability $1 - p_X$ the parents pass directly to $G_1^{(X)}$. The recombination is once again inspired to biology, and in the simplest case it consists in cutting both chromosomes at some locus l and then to exchange the two pieces of chromosomes to form the two offsprings. The rationale of this operator is that good parents can produce better offsprings by passing pieces of genome. Other possibilities do exist, for instance the two-points crossover or more technical ones as the one that will be presented and used in the next sections.

2.3.3. Mutation. The role of the mutation operator is to slightly change the genome of the offspring with respect to the one of the parent and thus to locally explore the solution space in a neighbourhood of the parent chromosome. With probability p_m each gene of each chromosome in the population can mutate, that is change its value; if the chosen coding is a binary allele,

⁴Actually this is true if the diversity of the chromosomes in the population is high, genotype diversity, or if the values of the objective function computed on the whole population are scattered enough, phenotype diversity.

then this means to exchange 0 and 1, if other codings have been chosen, the mutation will be accordingly defined. Because usually p_m is very small, inspect all the genes of all the chromosomes to decide if a mutation should be performed is very time consuming, we can thus use an alternative way consisting of computing the expected number of mutations. We will have in the whole population on average $p_m N N_p$ mutations performed, being the process a Bernoulli one. Then this exact number of chromosomes are selected and the mutation occurs by randomly choosing the loci and changing the value of the corresponding gene. This will produce a third intermediary population $G_1^{(m)}$ whose size is not fixed a priori.

2.3.4. *Next generation.* Finally, starting from the current population G_0 we constructed three populations $G_1^{(s)}$, $G_1^{(X)}$ and $G_1^{(m)}$, the next population G_1 will be obtained by selecting N_p individuals from $G_1^{(s)} \cup G_1^{(X)} \cup G_1^{(m)}$. Once again such individuals are chosen according to their fitness but other methods do exist.

3. HEDONIC GAME

Forming coalitions is an extremely important issue that can be found in several fields ranging from socio-economics to political groups, from students clubs to sport teams, just to mention few of them. Despite these problems occur daily, find a solution has never been a trivial problem. We decided to focus here on hedonic games, where players only care about their own coalition and not about the other ones, in other words, a player's utility only depends on his coalition. When players have preferences over all possible coalitions they could belong to, the problem is quite complicated. In most of the cases, a stable partition, i.e. a set of disjoint coalitions, encompassing all players, cannot be found in polynomial time (for more details see [ABH11, ABS13]). In the literature there exist some interesting results, if preferences are restricted to single-peaked and anonymous ones, that is players only care about the size of their coalition or the public good it produces and not about the individual players belonging to it. Under these particular assumptions, Bogomolnaia et al. [BJ02] propose an algorithm to find a stable partition which may not be necessarily Pareto-optimal. Our goal is thus to propose a method, based on genetic algorithm, to improve such partition. In the rest of this section we will introduce the basic problem and notations, then the results of Bogomolnaia et al. and finally the genetic algorithm adapted to this problem and the results.

3.1. Definitions and notations. Our approach is based on the paper [BJ02] by Bogomolnaia et al. and all definitions and notations below are adapted from this paper.

Consider a finite set of n players, $N = \{1, \dots, n\}$, a **coalition** is a set $S \subseteq \mathcal{P}(N)$, and a **partition** of N is a collection of coalitions $\Pi = \{S_k\}_{k=1}^K$ such that $S_k \subseteq N, \forall k \in \{1, \dots, K\}$; $S_k \cap S_j = \emptyset, \forall k \neq j$ and $\cup_{k=1}^K S_k = N$.

Each player has **preferences** over possible coalitions, that can be represented by an order relation \succeq_i (complete, transitive and reflexive) over the set $\{S \subseteq N : i \in S\}$. Given the partition Π , $S_\Pi(i)$ is the set $S \in \Pi$ such that $i \in S$. A **game** (N, \succeq) is formed by a set of players N and their preferences profile \succeq .

Preferences of a player are **anonymous** if $\forall S_1, S_2 \ni i$,

$$\#S_1 = \#S_2 \text{ implies that } i \text{ is indifferent between } S_1 \text{ and } S_2$$

where $\#S$ denotes the number of players in S .

Let $c(S) \in \mathbb{R}$ denote the level of production of coalition S , that only depends on the number of players belonging to S , for instance it can be the cardinality of S .

A player i 's preferences on some set $\{1, \dots, K\} \subseteq \mathbb{R}$ are **single-peaked** if there exists a number p_i , called i 's peak, such that $\forall S_1, S_2 \ni i$, $c(S_1) = s_1, c(S_2) = s_2$ and $s_1, s_2 \in \{1, \dots, K\}$

$$[s_1 < s_2 \leq p_i \text{ or } s_1 > s_2 \geq p_i] \Rightarrow S_2 \succ_i S_1$$

A partition Π is **Pareto-optimal** if there exists no other partition that Pareto-dominates it:

$$\nexists \Pi' \text{ such that } \forall i \in N, \quad S_{\Pi'}(i) \succeq_i S_\Pi(i) \text{ et } \exists j : S_{\Pi'}(j) \succ_j S_\Pi(j)$$

A partition Π is **weakly Pareto-optimal** if there exists no other partition in which every player is better off:

$$\nexists \Pi' \text{ such that } \forall i \in N, \quad S_{\Pi'}(i) \succ_i S_\Pi(i)$$

Stability can be defined in various ways but we will only focus on two definitions.

A partition Π is **Nash stable** if $\forall i : S_\Pi(i) \succeq_i S_k \cup \{i\}, \quad \forall S_k \in \Pi \cup \{\emptyset\}$

A partition Π is **Individually stable** if $\nexists i \in N$ and $S_k \in \Pi \cup \{\emptyset\}$ such that $S_k \cup \{i\} \succ_i S_\Pi(i)$ and $\forall j \in S_k, S_k \cup \{i\} \succeq_j S_k$

A coalition is said to be **opened** if at least one player can be added to this coalition without making any player worse off. Otherwise, the coalition is said to be closed.

To go further, we also need those two following definitions, introduced by Bogomolnaia et al.

Definition 1 (Ordered characteristics). *Each coalition S is described by a characteristic level of production $c(S)$ that lies between 0 and $\#S$. Players have single-peaked preferences over the public good $c(S)$, with peaks p_i such*

that $S \succeq_i S' \iff [c(S') < c(S) \leq p_i \text{ or } c(S') > c(S) \geq p_i]$.

A hedonic game has ordered characteristics if players' preferences are single-peaked over $c(S)$ and $c(S)$ satisfies the following properties:

- (1) If $c(S) < \sharp S$, then $c(S) = p_i$, with $i \in S$;
- (2) If $i \notin S, j \notin S$, and $p_i \geq p_j$ then $c(S \cup i) \geq c(S \cup j)$. Furthermore if $c(S \cup i) > p_i$ then $c(S \cup i) = c(S \cup j)$

Definition 2 (Consistency). A hedonic game with ordered characteristics is consistent if

$$[\exists i, \exists S \text{ such that } c(S \cup i) = p_i < c(S) < \min_{j \in S} p_j] \implies c(T \cup i) \leq p_i, \forall T$$

After having introduced the preliminary definitions, we are now able to present some results.

3.2. Results. The next theorem is given by Bogomolnaia et al. [BJ02].

Theorem 3. *If a hedonic game has ordered characteristics, then there exists an individually stable coalition partition. If in addition, consistency is satisfied, there exists a weakly Pareto-optimal individually stable coalition partition.*

To prove this theorem authors propose an algorithm that yields an individually stable coalition partition and in case of consistency, a weakly Pareto-optimal and individually stable outcome. We refer the interested reader to [BJ02] to have more details.

3.3. Bogomolnaia et al.'s Algorithm. Initially all players are sorted in increasing number of their peak, that is $i \geq j \implies p_i \geq p_j$. Then the algorithm proceeds as follows:

- (1) The first coalition S_1 is created by adding firstly the player n . Then players $k < n$ are added to S_1 as long as $p_k \geq c(\{k, \dots, n\})$ and $\{k+1, \dots, n\}$ is opened to k (acceptation rule);
- (2) The next coalitions $(S_k)_{k \geq 2}$ are formed in a similar way as the first one, except that they begin by adding the player with the highest peak not yet in a already formed coalition. Then for $j \in \{1, \dots, k-1\}$, if S_j is opened, we check for every player i of S_k , to whom S_j is opened, if i would be strictly better off in S_j ($c(S_j \cup i) \succ_i c(S_k)$ and $c(S_j \cup i) \succeq_f c(S_j), \forall f \in S_j$). If is the case, i is removed from S_k to S_j . If at this stage, there still exists a coalition S_j opened to some players of S_k , then all players, to whom a coalition is opened and who are indifferent to move to this coalition, are moved to S_j . For each player removed, we check if the player with the highest peak, not yet into a coalition, could be added to S_k (respecting the acceptation rule);
- (3) The algorithm ends when every player belongs to a coalition.

This algorithm finds a solution in polynomial time and always yields an individually stable partition. However, if the condition of consistency is not satisfied, the partition obtained by the algorithm may not be Pareto-optimal. Here we develop a general hedonic game that has ordered characteristics but is not consistent.

3.4. Non-consistent game. Let $1 \leq a < b < n$ and players' preferences to be single-peaked, then the public good c is defined as follow:

$$c(S) = \begin{cases} \#S & \text{if } \#S < a \text{ or either } \#S > b \\ \min\{\#S, \min_{i \in S} p_i\} & \text{otherwise.} \end{cases}$$

It is easy to prove that this game has ordered characteristics and it is not consistent.

Let us show this second point. By definition of consistency we should find i and S such that $c(S \cup i) = p_i < c(S) < \min_{j \in S} p_j$ and then it will follow that for all T : $c(T \cup i) \leq p_i$. Let us proceed by showing a contradiction; let us take S such that $\#S < \min_{j \in S} p_j$ and $a \leq \#S < b$ and choose a player i such that $p_i < \#S + 1$, thus we have $c(S \cup i) = p_i < \#S + 1 < \#S < \min_{j \in S} p_j$. If we take T such that $\#T > b$ we get $c(T \cup i) = \#T + 1 > b + 1 > \#S + 1 > p_i$, which proves the non-consistency.

3.5. Numerical simulations. A theoretical characterisation of a GA is almost impossible to be done, we thus resort to numerical simulations to check their efficiency.

In order to apply GA, we need first to model our problem. A solution or chromosome is a partition, modelled by a vector whose components represent players and the value of each component is the partition to which the player belongs. We also choose the following arbitrary representation of utility for each player i :

$$u_i(S) = \begin{cases} 1 - \frac{r_i(\#S)-1}{n} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} ,$$

where $r_i(\#S)$ is the position of $\#S$ in the preferences of i . For example, if i gets his peak, his utility will be equal to 1. To ensure the partitions we are dealing with are stable, the fitness of a partition which is not individually stable is set to zero, otherwise the fitness is defined as the normalised sum of single utility functions. Thereby, the fitness lies between 0 and 1.

This fitness can be used as objective function to be maximised by a GA. In this way we get good results since very often the GA finds a partition with a higher fitness than the one obtained via the algorithm of Bogomolnaia et al. [BJ02], see Figure 1. However, this approach is somehow subjective since it is based on a given fitness function, which after all has been fixed arbitrarily. A second approach can be thus considered that consists in Pareto-improving the partition found by the algorithm of Bogomolnaia et al. [BJ02]. Let us call this partition, Π ; a Pareto-improvement is defined

as a partition where all players are at least as well as in the previous assignment and some are better off. We can thus define a GA whose fitness function is computed as follows: we check whether the partition, we evaluate, is a Pareto-improvement of Π and if not, its fitness is penalised and made smaller than the fitness of Π , for instance by multiplying it by a small number, in this way we can ensure that partitions with a higher fitness than Π , are Pareto-improvements.

We can now define the various operators used by our genetic algorithm. The crossover operator is the basic one previously described, that is the 1 point crossover. On the other hand, we cannot use the simple mutation operator because it would only change one player from his coalition to another. This would not lead to an improvement since the algorithm of Bogomolnaia has already checked that no-one could be better off by changing of coalition, we thus realise a 2 steps mutation. The first step consists of randomly picking a chromosome in the population, choosing randomly a coalition and moving every players from that coalition to another existing one, by simply changing the number of their coalition. During the second step, a partition is randomly picked up and from this partition, one player is randomly chosen and moved to one another coalition. Those two operations are performed a certain number of times (see 2.3.3). Finally, the selection operator is the standard one, where higher probability of selection is given to best fitted partitions.

Here are the results for one simulation, with 100 players, the aim being the maximisation of the utilitarian function:

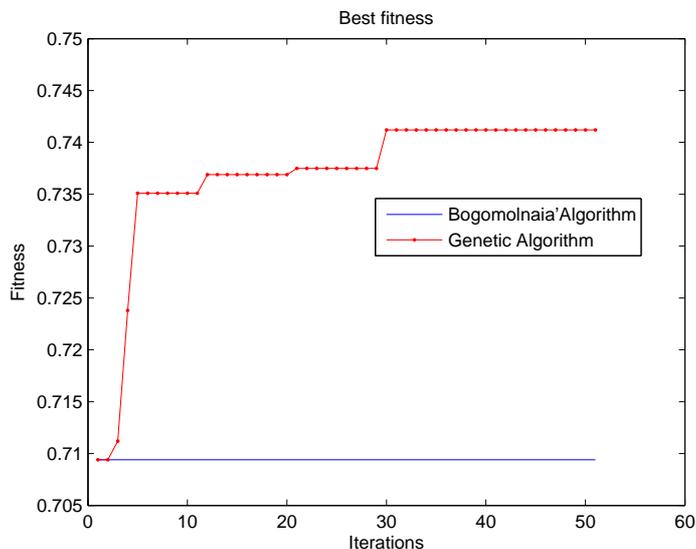


FIGURE 1. Utilitarian function (fitness), for $n = 100$ and the parameters of the non-consistent game being $a = 25$, $b = 30$. GA's parameters are the following ones: 700 initial chromosomes (within 100 are the partition found by Bogomolnaia et al.), probability of mutation =0.2, probability of crossover =0.4, the GA runs for 50 generations. In blue, the utility of the partition obtained by the algorithm of Bogomolnaia et al. and in red the utility at every step of the genetic algorithm.

In this simulation, we see that the utility of the partition of Bogomolnaia et al. is not maximum and that genetic algorithm, thanks to mutation and reproduction, can find partitions with higher utility. However, this optimisation is arbitrary (depending of the choice of the fitness) and we prefer to focus on Pareto-improving the partition by Bogomolnaia et al., see the following.

The following simulation is done using 20 players, a varies from 1 to 18 and b from $a + 1$ to 19. Single-peaked preferences are created randomly as well as the initial population of partitions. The remaining parameters used by the genetic algorithm are given in the following table:

- 500 initial partitions, among which 100 are the partition II;
- probability of mutation 0.1;
- probability of crossover 0.4;
- the GA runs for 20 generations.

For every set of parameters, the algorithm is executed 100 times and the percentage of Pareto-improvement⁵ is reported in Table 1.

$a \backslash b$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	11	3	8	6	5	2	2	3	2	0	1	0	0	0	0	0	0
2	-	9	2	6	3	4	6	4	1	2	1	0	0	0	0	0	0	0
3	-	-	4	4	7	7	3	4	3	1	2	0	2	3	2	1	0	2
4	-	-	-	2	4	4	1	2	4	3	3	4	1	2	2	1	3	2
5	-	-	-	-	0	7	3	2	1	3	1	0	1	3	0	1	4	3
6	-	-	-	-	-	2	2	3	2	3	4	3	3	2	0	0	1	2
7	-	-	-	-	-	-	2	1	5	1	2	3	1	0	0	1	1	4
8	-	-	-	-	-	-	-	4	1	2	3	2	2	0	0	1	1	1
9	-	-	-	-	-	-	-	-	1	2	1	1	1	0	1	1	0	2
10	-	-	-	-	-	-	-	-	-	3	3	2	2	1	2	3	3	0
11	-	-	-	-	-	-	-	-	-	-	4	4	0	4	2	4	1	6
12	-	-	-	-	-	-	-	-	-	-	-	3	2	3	1	2	0	2
13	-	-	-	-	-	-	-	-	-	-	-	-	3	4	1	2	2	3
14	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3	2	2	2
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	3	3	6
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	4	7
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3

TABLE 1. Percentage of Pareto-improvement found by GA, out of 100 simulations

We can clearly see from the previous table that for almost every set of parameters a and b , GA finds a Pareto-improvement of the partition obtained by the algorithm [BJ02]. This improvement can be quantified as varying from 1 up to 11 percent. We found on average of 4.08 players who are better off (when there is a Pareto-improvement) that is close to 20 percent of players that are better off. Unfortunately, we don't know if the partition we get from the genetic algorithm is Pareto-optimal, because this algorithm is not deterministic. Furthermore, according to Aziz et al. [ABH11] verifying if a partition is Pareto-optimal (under anonymous preferences) has a non-polynomial time complexity. So, the only thing we can say is that this new partition dominates the one found by the algorithm of Bogomolnaia et al. [BJ02]. We will not report here the simulations done using 50 or 100 players being the results similar to those reported in Table 1.

⁵For each simulation, we only consider the best partition (stable and with the highest fitness) and whether it is a Pareto-improvement of the partition by Bogomolnaia et al.

4. CONSTRAINED SCHOOL CHOICE WITH INDIFFERENCES

In their celebrated 1962 paper [GS62], Gale and Shapley developed in the framework of the marriage problem, the algorithm called *Deferred Acceptance Algorithm* (DAA). The aim of the latter is to match men and women such that everyone is matched with its feasible preferred choice (efficiency) and no-one has incentive to get divorced and exchange partner (stability). This algorithm has been revisited, exploited and used in many different settings. In this paper, we will be interested of its application to the case of many-to-one matching in school choice.

The problem is defined by a set of students, a set of schools and a complete set of preferences of each group over the agents of the opposite set. Furthermore, schools have a limited number of seats available and it is not always possible to give every student his first choice. Three algorithms are usually used to solve this problem [EE08], however for a sake of simplicity we will only consider in the following the DAA. We advice the interested reader to have a look at [Rot08], for more details about the history of the DAA.

To avoid social segregation, often schools have to respect some quotas, that is to reserve a certain number of seats to students of each type. Students' type is usually defined according to socio-economic characteristics of the students, like income, ethnicity, gender, etc. Quotas are strong constraints (hard bound approach) of the matching problem, when is the legislator imposes that quotas have to be respected irrespectively of students' preferences. On the other hand, quotas can also be seen as guidelines, a soft bounds approach that will be explained in details later. Beside these facts, let us stress that another problem appears when schools cannot strictly rank every student and these indifferences lead to a loss of efficiency.

The goal of this section is to present a new method, based on GA, aimed at increasing the efficiency of the matching algorithm.. Let us start by introducing some definitions and notations.

4.1. Definitions and notations. A constrained school choice where schools have weak preferences over students can be modelled as follows:

- (1) A set of n students : $S = \{s_1, s_2, \dots, s_n\}$;
- (2) A set of m schools : $C = \{c_1, c_2, \dots, c_m\}$;
- (3) A capacity vector : $q = \{q_{c_1}, q_{c_2}, \dots, q_{c_m}\}$, where q_c is the number of seats available in the school c ;
- (4) Students' strict preferences (complete, transitive and reflexive relation) over schools: $P = \{P_{s_1}, P_{s_2}, \dots, P_{s_n}\}$, where $c_i P_{s_j} c_k$ means that student s_j prefers school c_i to school c_k . Let \mathcal{P}_s denote the set of strict preferences of s over $C \cup \{s\}$, $\mathcal{Q} \equiv \prod_{s \in S} \mathcal{P}_s$ and $\mathcal{Q}_{-s} \equiv \prod_{s' \in S \setminus \{s\}} \mathcal{P}_{s'}$;
- (5) A set of k types: $T = \{\tau_1, \tau_2, \dots, \tau_k\}$ and a type function; $f : S \rightarrow T$, $f(s) = \tau_j$, where τ_j is the type of student s

- (6) Schools' **weak preferences** (complete, transitive and reflexive relation) over students : $\succeq = \{\succeq_{c_1}, \succeq_{c_2}, \dots, \succeq_{c_m}\}$, where $s_i \succ_{c_j} s_k$ means that c_j strictly prefers students s_i to student s_k and $s_i \sim_{c_j} s_k$ means that school c_j is indifferent between both students. Let \mathcal{C}_c be the set of all weak preferences of school c over the set of students S and let $\mathcal{C} \equiv \prod_{c \in C} \mathcal{C}_c$;
- (7) Each school has **constraints over types**: $\underline{q}_c^T = (\underline{q}_c^{\tau_1}, \underline{q}_c^{\tau_2}, \dots, \underline{q}_c^{\tau_2})$ minimal quotas and $\overline{q}_c^T = (\overline{q}_c^{\tau_1}, \overline{q}_c^{\tau_2}, \dots, \overline{q}_c^{\tau_2})$ maximal quotas, where $\underline{q}_c^{\tau_j}$ is the minimum number of seats that must be available in school c for students of type τ_j and $\overline{q}_c^{\tau_j}$ is the maximum number of seats assigned in school c for students of type τ_j . With $\underline{q}_c^{\tau_i} \leq \overline{q}_c^{\tau_i} \leq q_c, \forall i \in \{1, \dots, k\}$, $Q = (\underline{q}_{c_i}^T, \overline{q}_{c_i}^T), i \in \{1, \dots, m\}$.

We denote a problem of matching with indifferences and constraints by

$$(S, C, q, T, f, P, \succeq, Q).$$

We will assume that schools' preferences are **responsive**: $\forall s, s' \in S, \forall R \subseteq S$ such that $s, s' \notin R, R \cup \{s\} \succeq_c R \cup \{s'\} \iff s \succeq_c s'$.

Remark 4. *Let us emphasise two basic points:*

- (1) *To avoid impossibility cases, we assume that the number of students of each type is higher than the sum of minimum quotas of that type, in every school:*

$$|S^{\tau_j}| \geq \sum_{c \in C} \underline{q}_c^{\tau_j}, \quad \forall j \in \{1 \dots k\},$$

where $S^{\tau_j} = \{s \in S : f(s) = \tau_j\}$;

- (2) *We will also assume that there are enough seats available for students, in other words, that the number of students of each type is not higher than the sum of maximum quotas of this type, in every school:*

$$|S^{\tau_j}| \leq \sum_{c \in C} \overline{q}_c^{\tau_j}, \quad \forall j \in \{1, \dots, k\}.$$

We now define a matching as an allocation of students to schools such that each student is assigned to a school and the number of students matched with a school is less or equal than its capacity; to handle the case where a student will not be assigned to any school, we will match the student with himself. More formally a **matching** μ is an allocation of schools and students such that

- $|\mu(s)| = 1, \forall s \in S$, such as if $\mu(s) \neq s$, then $\mu(s) \in C$;
- $\mu(c) \subset S$ and $|\mu(c)| \leq q_c, \forall c \in C$;
- $\mu(s) = c \iff s \in \mu(c)$

A student s is matched to $\mu(s)$ and a school c is assigned the set of students $\mu(c)$. Let $\mu^{\tau_j}(c)$ denote the set of student of type τ_j assigned by μ to

school c .

A **matching rule** yields a matching for every preference profile and is defined as a mapping φ from $\mathcal{D} = (P, \succeq)$ to \mathcal{M} (\mathcal{M} is the set of all feasible matchings), $\varphi : \mathcal{D} \rightarrow \mathcal{M}$.

Matching should have some desirable properties like strategy-proofness, stability and optimality.

A matching rule φ is **strategy-proof** if it is a dominant strategy for each student to reveal his true preferences ⁶:

$$\forall s \in S, \forall c \in C, \forall P_{-s} \in \mathcal{Q}_{-s}, \forall \succeq \in \mathcal{C}, \varphi(P_{-s}, P_s, \succeq) P_s \varphi(P_{-s}, P'_s, \succeq), \forall P'_s \in \mathcal{Q}_s.$$

Where P_s are the true preferences of agent s and P_{-s} are the preferences of all agents in S but s .

In this paper, we will focus on the **soft bounds** case ⁷ and we follow the notation introduced by Elhers et al. [EHYY12]. Under this approach, schools do not have to strictly respect the quotas, those are seen as guidelines and serve to define dynamic priorities among students. Given a school c and a student of type τ whose type's minimum level (\underline{q}_c^τ) is not filled, then we assign to the latter a high priority over other applying students; to a student of type τ' whose type's minimum level ($\underline{q}_c^{\tau'}$) is filled but not the maximum one ($\bar{q}_c^{\tau'}$), we assigned a medium priority and finally to a student of type τ'' whose type's maximum level ($\bar{q}_c^{\tau''}$) is filled it is assigned a low priority. According to the soft bounds approach, schools do not have to strictly fill the quotas but can admit less students of a type than their floors if no students with a high priority apply. This approach gives more importance to students' preferences than to reduce segregation.

A matching is **non-wasteful under soft bounds** (NWSB) if for any school c and student s , $cP_s\mu(s)$ implies that $|\mu(c)| = q_c$. A matching rule is NWSB if for any preference profile, it yields a matching NWSB.

A matching is **stable under soft bounds**⁸ (SUSB) if $\forall s \in S, \forall c \in C : cP_s\mu(s)$, with $f(s) = \tau_i$, we have $|\mu^{\tau_i}(c)| \geq \underline{q}_c^{\tau_i}$ and $s' \succ_c s, \forall s' \in \mu^{\tau_i}(c)$ and either:

- (1) $|\mu^{\tau_i}(c)| \geq \bar{q}_c^{\tau_i}$ et $s' \succ_c s, \forall s' \in \mu(c)$ such that $|\mu^{f(s')}(c)| \geq \bar{q}_c^{f(s')}$;
or
- (2) $\bar{q}_c^{\tau_i} > |\mu^{\tau_i}(c)| \geq \underline{q}_c^{\tau_i}$, and

⁶Normally, strategy-proofness should encompass all agents however we here only consider the students case because in the college problem, there is no stable matching rule that is strategy-proof for all agents but there can be matching rules that are strategy-proof for students [Rot85].

⁷The interested reader can consult [EHYY12] for the hard bounds approach.

⁸In [EHYY12], L. Elhers defines a matching fair under soft bounds, that removes justifiably envy under soft bounds but this is just an extension of the notion of stability previously defined by Gale and Shapley.

- $|\mu^{\tau_j}(c)| \leq \bar{q}_c^{\tau_j}, \forall j \in \{1, \dots, k\} \setminus \{i\}$; and
- $s' \succ_c s, \forall s' \in \mu(c)$, such that $\bar{q}_c^{f(s')} > |\mu^{f(s')}(c)| \geq \underline{q}_c^{\tau_i}$.

A matching rule is SUSB if for any preference profile it yields a matching SUSB.

A matching μ is Pareto-optimal⁹ if there exists no other matching ν that Pareto-dominates it:

$$\exists s' \in S : \nu(s') P_{s'} \mu(s') \text{ and } \forall s \in S \setminus \{s'\}, \mu(s) = \nu(s)$$

Finally, a matching rule is **student-optimal** (S-optimal) if it is stable under soft bounds and for any preference profile, yields a matching Pareto-optimal among the set of stable matchings.

4.2. Results. In the paper by Elhers et al. [EHYY12], authors present an algorithm, called Deferred Acceptance Algorithm with soft bounds (DAA-SB), adapted from the Deferred Acceptance Algorithm by Gale and Shapley [GS62], with the introduction of dynamic priorities, defined according to the type of students and quotas. Note that schools have strict preferences over students, we will introduce the indifferences later on.

4.2.1. Deferred Acceptance Algorithm with soft bounds (DAA-SB). Let us start by briefly introduce this algorithm.

- (1) At step 1, every student applies to his favourite school. Schools give first priorities to their applying students (depending on the number of applicants of each type): given a school c , for every type τ , if the number of applying students of type τ is lower than the minimum quota (\underline{q}_c^τ), they all get a high priority; if this number is higher than the minimum quota but lower than the maximum one (\bar{q}_c^τ), students receive a medium priority and finally, if the number of applying students of type τ is higher than the maximum quota, they get a low priority. After this phase, every school temporarily assigns students by filling minimum quotas (if possible), by choosing for each type τ its favourite \underline{q}_c^τ students among the applicants. At that stage, all students with a high priority are assigned to their preferred school. Then, every school tries to fill its maximum quotas, selecting its favourite students among the remaining ones, as long as seats are available. Finally, either the capacity of the school is reached and all non-assigned students are rejected or every student with a medium priority has been assigned and school fills the remaining seats with left low-priority students;
- (2) At step $k \geq 2$, every student who has been rejected at the previous step applies to his preferred school c' among the remaining ones on his preference list. Then, priorities are calculated again for the set of new applicants and the temporarily assigned students to c' .

⁹The optimality is defined according students' preferences

Schools assign students following the mechanism described in step 1 and rejected students go to the next step;

- (3) The algorithm terminates when every student is assigned to a school or has been rejected by all schools on his preference list.

Through this algorithm, each student is assigned a dynamic priority at every step, nevertheless the algorithm ends in finite time because the number of students and their preferences lists are finite objects.

The first nice property of the $DAA - SB$ is that it is strategy-proof. If we introduce indifferences among schools' preferences, in order to apply the algorithm previously described, we need a tie-breaking rule. Given a tie-breaking rule t , we denote by $DAA - SB^t$, the $DAA - SB$ using this tie-breaking rule to create strict preferences. Since the $DAA - SB$ is strategy-proof for any preference profile, then, it is also strategy-proof for any tie-breaking rule. In a similar framework, Abdulkadiroğlu et al. [APR09] analyse the properties of efficiency and strategy-proofness when preferences are not strict. Let us observe that their study is based on the assumption of absence of constraints, in the following theorem we will extend the result of Abdulkadiroğlu et al. about strategy-proofness as to include the constrained case

Theorem 5. *For any tie-breaking rule t , there is no matching rule that is strategy-proof and Pareto-dominates the $DAA - SB^t$.*

The proof of this Theorem follows closely the one by Abdulkadiroğlu [APR09] once we have the following lemma

Lemma 6. *Suppose that ν dominates μ , the matching obtained by the $DAA - SB^t$, for any tie-breaking rule t . Then, the same set of students are matched in both ν and μ .*

Proof. If there exists a student s unassigned under ν but assigned under μ , we have that $\nu(s) = s$ and $sP_s\mu(s)$, thus $\mu(s)$ must be unacceptable for s which is impossible (building of the $DAA - SB^t$). So we have that $|\nu(c)| \geq |\mu(c)|, \forall c \in C$. Suppose there exists $c \in C$ such that $|\nu(c)| > |\mu(c)|$, it means that there is a vacancy at c under μ and that there exists a student $s \in S$ such that $\nu(s) = c \neq \mu(s)$. Furthermore, $cP_s\mu(s)$ since ν Pareto-dominates μ . And since we are only considering the soft bounds approach, if there is a seat available at c and even if the maximal quota of the type of student s is exceeded, then, s could be assigned to c . This means that μ violates stability under soft bounds (which is impossible because μ is the outcome of the $DAA - SB^t$). So we have that $|\nu(c)| = |\mu(c)|$ and every student assigned under μ is also assigned under ν . \square

Elhers et al. [EHYY12] shown another relevant property of the algorithm previously described.

Theorem 7. *For any problem of matching with indifferences and constraints, $DAA-SB$ yields a matching that is stable under soft bounds and*

non-wasteful under soft bounds. Moreover, this matching is S-optimal among the set of matchings that satisfy the above properties.

This theorem holds only if schools have strict preferences over students. In their study¹⁰, Abdulkadiroğlu et al. [APR09] concluded that any tie-breaking rule leads to a loss of efficiency since the outcome of the DAA is not more necessarily a S-optimal matching. They also proved that, considering indifferences, a matching that can be produced by the DAA with some multiple tie-breaking rule but cannot be produced by the DAA with any single tie-breaking rule, is not S-optimal. For this reason, we will use a single tie-breaking rule: a single tie-breaking to sort all students, the same for every school; before applying the $DAA-SB$, for short $DAA-STB-SB$ in the following.

We are now able to present our extension of the results by Abdulkadiroğlu et al. [APR09] about efficiency. Let us first observe that in our simulations we decide to sort randomly all the students (calling this random sorting, random tie-breaking rule) and thus notice that

Remark 8. *With the random tie-breaking rule, the $DAA-STB-SB$ is not S-optimal.*

The above remark can easily be illustrated by the following example.

4.2.2. *Not S-optimal example.* Consider a matching problem with 8 students, numbered from 1 up to 8, and 4 schools, A, B, C and D , such that students and schools have the preferences reported in the Tables 2 and 3:

1	2	3	4	5	6	7	8
A	A	B	A	B	C	C	C
D	B	A	D	C	B	A	D

TABLE 2. Students' preferences

A	B	C	D							
3 7	6	4	1 2 3 4 5 6 7 8							
1 2	2 5	5 7	-							
4 5 6 8	3 1 4 7 8	8	-							
-	-	6 1 2 3	-							

TABLE 3. Schools' preferences

Assume that there exist two types of students, normal and disadvantaged and assume moreover that students 1, 2, 5, 7 belong to type normal while the remaining ones to type disadvantaged.

Finally, every school has a capacity of 2 and schools A, B and C must admit at least one disadvantaged student.

¹⁰Note that authors in [APR09] only consider the unconstrained case.

A random sort of the students can provide the following results: $3 \succ 4 \succ 1 \succ 2 \succ 5 \succ 7 \succ 6 \succ 8$. Then applying the $DAA - SB$ we get the final stable matching μ :

$$\mu = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ D & B & A & D & C & B & A & C \end{pmatrix}$$

Let us observe that the former is Pareto-dominated by the following stable matching:

$$\mu^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ D & A & A & D & B & B & C & C \end{pmatrix}$$

We then have proved that the $DAA - SB$ with the random single tie-breaking rule is not S-optimal.

The following question naturally rises: does there exist a matching rule which is Pareto-optimal? And if not, is it possible to find a matching rule that Pareto-improves upon the $DAA - STB - SB$?

For a sake of completeness we hereby report the summary of properties of the different algorithms, where X is the unknown matching rule.

Algorithm	DAA	DAA-STB + SICA ^a	DAA-SB	DAA-STB-SB + X
Properties				
Preferences ^b	Strict	Non strict	Strict	Non strict
Quotas	No	No	Yes	Yes
Stability ^c	Stable	Stable	Stable	Stable
S-optimal	Yes	Yes	Yes	unknown
Strategy-proofness	Yes	No	Yes	No

TABLE 4. Properties of the different algorithms.

^aSICA is a mechanism used by Erdil and Ergin to Pareto-improve over the DAA. For more details, see [EE08].

^bPreferences are the ones of the schools

^cThe definition of stability depends on whether there are quotas or not. See Gale and Shapley, ([GS62]) for the definition of stability without quotas.

In the next part, we will apply genetic algorithms to our problem of controlled school choice.

4.3. Numerical simulations. In this section, we develop a GA to study whether it is possible to get a Pareto-improvement upon the matching obtained through the $DAA - STB - SB$. Our numerical simulation is directly inspired by the Belgian school system even if we introduce some simplifications, in particular we only consider 15 schools and 1750 students. According to the Belgian law, we define two types of students: ISEF students¹¹ and the 'normal' students, and we assume that 20% of students population belong to the ISEF type. Every student strictly ranks 5 schools and schools

¹¹ISEF will be the abbreviation for socially disadvantaged students.

are indifferent among students. We took exactly the same number of seats available as the total amount of students and schools' capacities vary from 50 to 200. Each school should at least have 20 % of ISEF students (minimal quota) ¹².

Students' preferences are created using a random scheme: for each student, we pick up and sort, with uniform probability, 5 schools among the 15 available ones. Then, 400 matchings are also randomly drawn, to form the initial population of chromosomes, i.e. possible solutions. In order to evaluate a matching, we need to define a fitness, whose optimisation should produce solutions meeting our objectives. We choose the following arbitrary utility function, for each student s :

$$u_s(\mu(s)) = \begin{cases} 1 - \frac{r_s(\mu(s)) - 1}{|P_s|} & \text{if } \mu(s) \in P_s \\ 0 & \text{otherwise.} \end{cases} \quad \forall s \in S,$$

where $r_s(\mu(s))$ is the ranking of school $\mu(s)$ in student- s ' preferences and $|P_s|$ the length of student- s ' preferences, that is 5 in our case. If the matching is not stable under soft bounds, the fitness is set to zero, otherwise the fitness is defined as the normalised sum of utilities for all students such that the fitness lies between 0 and 1. Thereby, we ensure that if the fitness is positive then our matching is stable.

Since we want to improve upon the *DAA – STB – SB*, we first execute this algorithm and then we use the obtained matching, let us call it μ , to form the initial population for the GA, more precisely in a population of size 800, 400 chromosomes will be exactly equal to μ . When calculating the fitness of a matching, we also check whether it is a Pareto-improvement of μ and if not, the fitness is penalised to make it smaller than the fitness of μ , for instance by multiplying the former by a small enough number. Thus, the only matchings which have a fitness higher than μ are Pareto-improving upon it.

Because of the constraints present in the model, for instance each student can appear in a chromosome only once and the same holds for each school, the crossover and mutation operators cannot be the basic ones that would not respect such constraints. The crossover we decided to use is called in the literature PMX and has been introduced by Olivier et al. [OSH87] in the Traveller Salesman Problem. We do not describe this operator here, we only fix the probability of crossover to 0.5. The mutation is a permutation of students, respecting the quotas, such that students get a school they prefer whenever it is possible. We fix the probability of mutation to 0.4.

Finally, the selection operator acts as usual by giving more chances to be selected to chromosomes whose fitness is larger, in this way we are sure

¹²The choice of the number of students, of schools, of preferences that students may express, the proportion of disadvantaged students, the capacities of the schools and their quotas are chosen to reproduce the real data's of Namur district, in Belgium.

that matchings who are a Pareto-improvement of μ will survive and matchings that are not stable will never be selected. Our GA will stop after 20 generations.

Because of the inherently random structure of the GA, we need to produce several replicas to draw our conclusions. So we ran the *DAA – STB – SB* followed by the GA, for 100 times. For each simulation, we only consider the best matching found by our GA and report the average improvement in Table 5. The results are very positive since we actually get Pareto-improvements upon the matching of the *DAA – STB – SB* (some students are better off and none is worse off) and those matchings are also stable. Unfortunately, there is no way to know whether further improvements are possible because genetic algorithms are not deterministic and convergence can not be proved theoretically. By the Theorem 5, we also know that this mechanism is not strategy-proof, even though it is less manipulable since it is a random process. It is difficult to cheat on the own preferences when you don't know what the final outcome will be.

Let us observe that Abdulkadiroğlu et al. [APR09] were able to improve the DAA, without quotas, to find a S-optimal matching where on average 1.9% of students are better off. Considering that our problem is far more complicated, because of the introduction of quotas, we believe that an average of 0.8% students who are better off, remains a relevant result. Even if our final matching may not be S-optimal, we strengthen our previous result that is, when preferences are not strict, the DAA-STB-SB fails to obtain a S-optimal matching and a Pareto-improvement can be found. We conclude saying that genetic algorithm are an alternative to Pareto-improve upon the DAA-STB-SB, they allowed us to improve the students' welfare.

Choice \ Algorithm	DAA-STB-SB	DAA-STB-SB + GA	Improvement	# students
1	1245.3 (27.4963)	1254.3 (27.4084)	+1	7.30 (1.8007)
2	222.55 (16.8369)	221.85 (16.5642)	+2	3.80 (1.9746)
3	130.20 (11.7748)	127.55 (11.4877)	+3	2.45 (1.4381)
4	69.40 (7.9671)	65.85 (8.5474)	+4	0.75 (0.7017)
5	37.25(5.2211)	35.10(4.9329)	+5	0 (0)
Unassigned	45.35 (6.3347)	45.35 (6.3347)	-	0 (0)
Total	1750	1750		14.30 (2.6034)

TABLE 5. Pareto-Improvement, using GA, average of 100 simulations. The first column reports the ranking of the school received by students; the second column reports the average number of students receiving their first choice, second, etc from the DAA-STB-SB; the third column reports the average number of students receiving their first choice, second, etc from the DAA-STB-SB+GA; the fourth column reports the number of places that students improve, according to their rank order list; the fifth column reports the average number of students improving from the GA. The standard deviation is reported in brackets.

5. DISCUSSION AND OPEN QUESTIONS

Our main point with the present work, was to show that genetic algorithms can be useful in economic problems that are computationally hard, such as hedonic games and school choice with indifferences and quotas. The difficulty of the latter problems is mainly in the constraints that should hold. Based on the previous results, we can safely state that in both cases, genetic algorithms allowed us to Pareto-improve upon the existing algorithms. Genetic algorithms seems more general than other existing algorithms of matching and they can easily be adapted to other problems. There still remains open questions about the theoretical characterisation of genetic algorithms, however such questions go beyond the scope of the present work and will deserve further investigations.

REFERENCES

- [Abd05] Abdulkadiroğlu, A., College admission with affirmative action, *International Journal of Game Theory*, Vol. 33, No. 4, 2005, pp. 535-549

- [ABH11] Aziz, H.; Brandt, F.; Harrenstein, Paul, Pareto Optimality in Coalition Formation, *Algorithmic Game Theory, 4th International Symposium, SAGT 2011, Amalfi, Italy, October 17-19, 2011. Proceedings*, Springer-Verlag 2011, pp. 93-104
- [ABS13] Aziz, H.; Brandt, F.; Seedig, H. G., Computing partitions in additively separable hedonic games, *Artificial intelligence*, 195, 2013, pp. 316-334
- [AC99] Aldershof, B.; Carducci, O. M., Stable Marriage and Genetic Algorithms: A Fertile Union, *Journal of Heuristic*, 5, 1999, pp. 26-46
- [APR05] Abdulkadiroğlu, A.; Pathak, P. A.; Roth, A. E., The New York City High School Match, *American Economic Review*, Vol. 95, No.2, 2005, pp. 364-367
- [APR09] Abdulkadiroğlu, A.; Pathak, P. A.; Roth, A. E., Strategy-proofness versus efficiency in matching with indifference: redesigning the NYC High School match, *American Economic Review*, Vol. 99, No. 5, 2009, pp. 1954-1978
- [AS03] Abdulkadiroğlu, A.; Sönmez, T., School Choice: A Mechanism Design Approach, *American Economic Review*, Vol. 93, No.3, 2003, pp. 729-747
- [Bal04] Ballester, C. NP-completeness in hedonic games, *Games and Economic Behaviour*, 49 (1), (2004), pp. 1-30
- [BJ02] Bogomolnaia, A.; Jackson, M. O.; The stability of Hedonic Coalition Structures, *Games and Economic Behaviour*, 38, 2002, pp. 201-230
- [CG09] Cantillon, E.; Gothelf, N., Quel enfant dans quelle cole? Rflexion sur la rgulation des inscriptions scolaires en Belgique, *Proceedings of the 18th Congress of Belgian French-speaking Economists*, Nov. 2009, CIFO Editions
- [EE08] Erdil, A.; Ergin, H., What's the Matter with Tie-Breaking? Improving Efficiency in School Choice, *American Economic Review*, American Economic Association, Vol. 98(3), Juin 2008, pp 669-689
- [Ehl10] Ehlers, L., School Choice with Control, Working Paper (CIREQ Cahier 13-2010), 2010, University of Montreal
- [EHYY12] Elhers, L.; Hafaldir, I. E.; Yenmez, M. B.; Yildirim, M. A., School Choice with Controlled Choice Constraints: Hard Bounds versus Soft Bounds, Working Paper Series, 2012, University of Montreal
- [GS62] Gale, D.; Shapley, L. S., College Admissions and the Stability of Marriage, *The American Mathematical Monthly*, Vol. 69, No. 1, 1962, pp. 9-15
- [Gol89] Goldberg, David E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, Inc, United States of America, 1989
- [HRU06] Haruvy, E.; Roth, A.; Ünver, M. U., The dynamics of law clerk matching: An experimental and computational investigation of proposals for reform the market, *Journal of Economics Dynamics & Control*, 30, 2006, pp. 457-486
- [Mit98] Mitchel, M., *An introduction to genetic algorithms*, No 1, Massachusetts Institute of Technology Press paperback edition, Cambridge, 1998
- [OSH87] Olivier, I. M.; Smith, D. J.; Holland, J. R. C., A study of permutation crossover operators on the travelling salesman problem, *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 224-230
- [Pat11] Pathak, Parag A., The Mechanism Design Approach to Student Assignment, *Annual Review of Economics*, Vol. 3 (c), Sept. 2011, pp. 513-536
- [Rot85] Roth, A. E., The College Admission Problem Is Not Equivalent to the Marriage Problem, *Journal of economic theory*, 36, 1985, pp. 277-288
- [Rot08] Roth, A. E., Deferred Acceptance Algorithm: history, theory, practice and open questions, *International Journal of Game Theory*, 36, 2008, pp. 537-569
- [RS90] Roth, A. E.; Sotomayor, Marilda A. Oliveira, *Two-Sided Matching, A study in game-theoretic modeling and analysis*, No. 18, Cambridge University Press, Cambridge, 1990
- [Unv05] Ünver, M. U., On the survival of some unstable two-sided matching mechanisms, *International Journal of Game Theory*, 33, 2005, pp. 239-254

(Marelli Virginie) DEPARTMENT OF MATHEMATICS, UNIVERSITY OF NAMUR, REMPART DE LA VIERGE 8, B 5000 NAMUR, BELGIUM

(Antonio Nicolò) DEPARTMENT OF ECONOMICS, UNIVERSITY OF PADUA, VIA DEL SANTO 33, 35123 PADUA , ITALY

(Timoteo Carletti) NAMUR CENTER FOR COMPLEX SYSTEMS, nAXys, UNIVERSITY OF NAMUR, REMPART DE LA VIERGE 8, B 5000 NAMUR, BELGIUM