André Luiz Netto Casotti

A multi-objective approach for the team formation problem considering criteria of cohesion and disagreement

Vitoria

2022

André Luiz Netto Casotti

Uma abordagem multi-objetivo para o problema de formação de equipes considerando critérios de coesão e discordância

Vitoria

2022

André Luiz Netto Casotti

A multi-objective approach for the team formation problem considering criteria of cohesion and disagreement

Monography presented to the Undergraduate Program in Production Engineering of the Federal University of Espirito Santo as a partial requirement to obtain the degree of Production Engineering

Federal University of Espirito Santo

Undergraduate Program in Production Engineering

Supervisor Prof. Dr-Ing. Renato Antônio Krohling

Vitoria

2022

André Luiz Netto Casotti

Uma abordagem multi-objetivo para o problema de formação de equipes considerando critérios de coesão e discordância

Trabalho de Conclusão de Curso apresentado ao Departamento do curso de Engenharia de Produção, do Centro Tecnológico, da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Produção

Federal University of Espirito Santo

Undergraduate Program in Production Engineering

Supervisor Prof. Dr-Ing. Renato Antônio Krohling

Vitoria

2022

André Luiz Netto Casotti

Uma abordagem multi-objetivo para o problema de formação de equipes considerando critérios de coesão e discordância.

Trabalho de Conclusão de Curso apresentado ao Departamento do curso de Engenharia de Produção, do Centro Tecnológico, da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Produção

Trabalho aprovado em de março de 2022 :

Prof. Dr-Ing. Renato Antônio Krohling
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Renato Elias Nunes de Moraes
Universidade Federal do Espírito Santo
Membro Interno

Prof. Dr. Rodrigo de Alvarenga Rosa
Universidade Federal do Espírito Santo
Membro Interno

Vitoria
2022

Abstract

Organizations are made up of teams, and team formation is a crucial attribution for decision-makers. Several studies indicate the increased correlation between team arrangement and performance. Moreover, some social aspects of team members' interaction, such as cohesion and disagreement have a strong relationship with team performance and innovative practices. To measure and then optimize those attributes, our work has employed sociometric evaluation to measure cohesion, and proposes to compute intra-group disagreement by means of Krippendorff's alpha, a metric that generalizes disagreement evaluations. A multi-objective Genetic Algorithm was applied to perform optimization, maximizing objectives of cohesion and disagreement and generating an approximate Pareto Front of the solutions. The NSGA-II algorithm, a standard algorithm for multi-objective optimization was employed and the method was applied to a benchmark dataset, consisting of 7 team formation problems with the same structure but different numbers of individuals and groups. The algorithm was capable of generating sufficient good Pareto approximations and demonstrated to be a suitable approach for this problem.

Resumo

As organizações são formadas por equipes, e a formação de equipes é uma atribuição crucial para os tomadores de decisão. Vários estudos indicam a correlação entre a disposição das equipes e desempenho. Alguns aspectos sociais relativos à interação dos membros das equipes, tais como a coesão e discordância, têm demonstrado uma forte relação com desempenho e práticas inovativas. Este trabalho aplicou um Algoritmo Genético Multi-Objetivo para resolver o problema de formação de times, maximizando objetivos de coesão e discordância e gerando uma fronteira aproximada de Pareto das soluções. Além disso, uma maneira quantitativa de computar a discordância utilizando o alpha de Krippendorff, um padrão na medição de confiabilidade, foi proposta O algoritmo NSGA-II, um algoritmo para otimização multiobjetivo foi empregado, e o método foi aplicado para um conjunto de dados de referência, constituído por 7 problemas de formação de times, com a mesma estrutura mas um número diferente de indivíduos e grupos. O algoritmo foi capaz de gerar aproximações de Pareto suficientes boas, obtendo resultados satisfatórios para o problema.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Team Performance

Groups and Teams are the backbones of organizations. Due to the systems effect, the performance of one team is affected at least by one other team, and each group affects the organization's performance (LUSSIER, 2017). Therefore, an organization is made up of many sub-systems, enhanced by collective intelligence, to generate positive outcomes, and make smarter decisions with collaboration among the members. So, group formation is an important step to achieve success and assure an effective working of these systems.

An essential capability for creating efficient teams is Cohesion between the group members. Cohesion refers to the commitment of team members to their work team and their desire to maintain group membership (LOTT; LOTT, 1965).

Meta-analysis reveals a strong correlation between cohesion and performance (EVANS; DION, 1991). Sanders e Nauta (2004) pointed out how increased cohesion reduces employee absenteeism while increasing job satisfaction and motivation. Moreover, cohesion can also lead to Innovative teams, as concluded by Hülsheger, Anderson e Salgado (2009) in a study containing three decades of research.

But, despite the potential of cohesive teams to generate positive outcomes, it has also disadvantages regarding information sharing. Mesmer-Magnus e DeChurch (2009) discovered that cohesive teams tends to share openness information instead uniqueness information. Uniqueness information refers to the number of group members with knowledge about a piece of information. Openness information regards to aspects of communications related to work, goals, and reports of a task.

Some studies indicate a correlation between diversity and performance. A technical report by Hunt, Layton e Prince (2015), has shown diverse teams lead to better financial performance. However, recent analysis has explored this relationship, comparing surface-level diversity and deep-level diversity.

Surface-level diversity refers to easily discernible demographic characteristics that clearly distinguish social group membership. Deep-level diversity involves disparities in attitudes, beliefs, and values among members. These components' information is transmitted through verbal and nonverbal behavior patterns, and it can only be discovered through considerable, individual contact and information collecting (HARRISON; PRICE; BELL, 1998).

The exact type of diversity that increases a team's performance is deep-level diversity, where perspectives and information differ. While surface-level demographic features doesn't increase team performance (WANG et al., 2019).

Both surface and deep-level diversity in culturally diverse teams incurs higher social costs. However, in surface-level it becomes higher than informational benefits (BUSSE; MAHLENDORF; BODE, 2016). While for deep-level, informational benefits for team performance are relatively higher than the social costs (WANG et al., 2019).

Veen, Kudesia e Heinimann (2020) have studied how information sharing strategies influence the accuracy and speed of decision making, comparing strategies of advocacy, agreement, random, and disagreement. In terms of accuracy, both random and disagreement, have outperformed among other strategies, and random strategy has presented a slightly superior value. But, regarding decision speed, the disagreement strategy was quite better, in comparison to the random strategy, demonstrating that it is the most indicated strategy for effective collaborative decisions. So, they conclude that deep-level diversity is important, however, combined with a bad information sharing strategy, it can not produce the desired outcomes.

1.2 Optimization of Team Formation

Metaheuristics are considered the most effective methods to solve human resource allocation problem (HRAP), providing good solutions, at a reasonable computational time. The table 1 provides information regarding HRAP solving techniques, in terms of numbers of publications (BOUAJAJA; DRIDI, 2017).

Method	Exact	Heuristics	Metaheurístics	Hybrid
Number of publications	20	34	41	12
Percentage (%)	19	32	38	11

Table 1 – HRAP solving techniques

Genetic algorithm (GA), tabu search (TS), ant colony optimization (ACO), particle swarm optimization (PSO) and simulated annealing (SA) are widely used algorithms in HRAP, and table 2 indicates that GA was the most used algorithm to solve the combinatorial optimation problem (BOUAJAJA; DRIDI, 2017).

Metaheuristic	ACO	TS	GA	SA	PSO	HA
Number of publications	8	9	16	3	5	12
Percentage (%)	15	17	30	6	9	23

Table 2 – Most used algorithms in HRAP

A comprehensive review indicates that NSGA-II (Non-Dominated Genetic Algorithm-II) is the most used algorithm to the multi-objective combinatorial optimization problems being a off-the-shelf method for Allocation problems (VERMA; PANT; SNASEL, 2021).

1.3 Related Works

Assembling teams is a challenge for decision makers, and with the revealing of factors that drive teams to better performance, some computational methods could be useful tools to help managers to choose and allocate people to form teams, in order to achieve those desired metrics. Ballesteros-Perez, González-Cruz e Fernández-Diego (2012) has developed an approach to measure the cohesion among group members with sociometric techniques, resulting in the multiple team formation problem. Manual and computational exhaustive methods were suggested to find the allocation that maximizes cohesion. Indicating that the method is not feasible to apply in case of large allocations.

Esgario, Silva e Krohling (2019) extended the work proposed by Ballesteros-Perez, González-Cruz e Fernández-Diego (2012) by optimizing the group formation in terms of cohesion using a Genetic Algorithm (GA). In addition, they proposed a benchmark dataset, with seven instances of the problem, in which GA was compared to exhaustive method. The study revealed that evolutionary approach is effective when it is not feasible to compute the optimal solution in an acceptable computational time due the required effort. So, the approach turns out to be quite promising, since the NP-hard nature of the problem.

1.4 Objectives

The objective of this work is to solve the Team Formation Problem for the Benchmark proposed by Esgario, Silva e Krohling (2019), considering objectives of intra-group Cohesion and Disagreement. Firstly, a definition of Disagreement measure must be defined, and data for measure the indicator should be generated. Hence, to perform the optimization analysis the following steps should be taken. First, a single-objective optimization for each criteria will be carried out. Second, to handle 2 objectives simultaneously, a multi-objective Optimization will be performed in order to find a Pareto set for the problem, which means the set of solutions (Allocations) that maximizes the values for both objectives.

- Specific Objectives:
  - Perform Optimization with single-objective Genetic Algorithm for Cohesion and compare to Benchmark proposed in literature.

– To attempt measuring Disagreement and randomly simulate data to compute the metric.
– Optimize the allocation for Disagreement in the proposed Benchmark, adding the simulated data.
– Perform a Multi-objective Optimization considering both criteria at the same time.

1.5 Organization of the work

The work has been divided into chapters of Problem Definition Methodology, Experimental Results and Conclusion. The Problem Definition describes the approach used for the human resource allocation problem. In the methodology, a brief intuition about the Genetic Algorithm is provided, explaining it in terms of its operators, and applying it in a simple example. Next, important concepts for understanding the multi-objective approach are described and then the NSGA-II algorithm is explained.

In the experimental results chapter, an example of application of the methodology is detailed in the first problem of the benchmark dataset. Then, the benchmark dataset is presented in terms of the number of individuals and groups, and the parameters of the algorithms for each case. Then the results are presented.

At the end, a conclusion on the development of the work is presented, also indicating possible directions for future research.

2 Problem Formulation

2.1 Problem Definition

The problem of allocation is defined as an optimization problem, with two objectives, cohesion and disagreement, and two constraints, one to ensure that each individual belongs to only one group, and the second is regarding group requirements. In this section, we describe how to compute each objective and also the constraints.

Based on Ballesteros-Perez, González-Cruz e Fernández-Diego (2012), some definitions are provided. Next, the total number of individuals that should be allocated is expressed by n_i and the number of groups as n_k.The matrix A represents the indication of which individual belongs to each group, and it is used to compute the cohesion and disagreement for a solution. An element a_{ij} from matrix A indicates that individual i belongs to group j.

$$A = \begin{array}{c} \\ I_1 \\ \vdots \\ I_{n_i} \end{array} \begin{array}{c} \begin{array}{ccc} G_1 & \dots & G_{nk} \end{array} \\ \begin{pmatrix} a_{1_1} & \dots & a_{1_k} \\ \vdots & \ddots & \vdots \\ a_{ni_1} & \dots & a_{ni_k} \end{pmatrix} \end{array}$$

In this approach, each worker is allocated full-time in a group, which means that the sum of each row must be one.

$$\sum_{j=1}^{k} a_{ij} = 1 \qquad \forall i \in 1, 2, ..., n_i \tag{2.1}$$

The individuals are allocated constrained by a requirement matrix. The individuals have different skills, and they are divided into Departments. Each group requires a specific number of resources from a given Department, and this Requirement is described by a matrix R.

$$R = \begin{array}{c} \\ D_1 \\ \vdots \\ D_j \end{array} \begin{array}{c} \begin{array}{ccc} G_1 & \dots & G_{nk} \end{array} \\ \begin{pmatrix} y_{1_1} & \dots & y_{1_k} \\ \vdots & \ddots & \vdots \\ y_{j_1} & \dots & y_{j_k} \end{pmatrix} \end{array}$$

where Di and Gj represent the department i and group j, respectively, and element y_{ij} indicates the demand from group j for workers of department i.

2.2 Cohesion Measure

There are many forms of compute cohesion. Although, a simple and effective way to measure the indicator is using a Sociometric Test. To compute intra-group cohesion, a Sociometric Matrix, of dimensions n_i x n_i, must be defined, whereas each individual Ii assigns a value s to each member that can be a possible teammate. In this approach, a discrete definition of s values was employed, and it can be: -1, 0, or 1. Depending on the answers given to sociometric test questions.

$$S = \begin{array}{c} \\ I_1 \\ \vdots \\ I_{ni} \end{array} \begin{array}{c} \begin{array}{ccc} I_1 & \dots & I_{ni} \end{array} \\ \begin{pmatrix} s_{1_1} & \dots & s_{1_{ni}} \\ \vdots & \ddots & \vdots \\ s_{ni_1} & \dots & s_{ni_{ni}} \end{pmatrix} \end{array}$$

To calculate the general cohesion, E_g, of a possible solution, the following equation was used.

$$E_k = \frac{\sum_{j=1}^{n_i} a_{ik} a_{ij} s_{ij}}{n_{ik}} \tag{2.2}$$

where E_k represents the cohesion for the E_{kth} group. The cohesion must be multiplied by a weight to deal with values in the same scale.

$$E_g = \sum_{k=1}^{n_k} W_k E_k \tag{2.3}$$

where W_k is the weight of each group, calculated according to

$$W_k = \frac{n_{ik}}{n_i} \tag{2.4}$$

where n_{ik} is the number of individuals of a group k, and n_i is the total numbers of individuals of the problem.

2.3 Disagreement Measure

This work proposes to compute intra-group disagreement by means of Krippendorff's Alpha (α) metric (KRIPPENDORFF, 2011), because of its versatility, due to the possibility of using many types of evaluation, nominal, ordinal, cardinal, and so on. Moreover, it presents no limitations regarding missing data and number of evaluations or evaluators. The input for the disagreement calculation is a Reliability Matrix, formed by the evaluations made from each individual. In the Reliability Matrix C, each row represents one individual and each column an evaluation.

$$C = \begin{array}{c} \\ \mathrm{I}_1 \\ \vdots \\ \mathrm{I}_{n_i} \end{array} \begin{array}{c} \begin{array}{ccc} \mathrm{Ev1} & \dots & \mathrm{Ev}_n \end{array} \\ \begin{pmatrix} \mathrm{r}_{1_1} & \dots & \mathrm{r}_{1_n} \\ \vdots & \ddots & \vdots \\ r_{ni_1} & \dots & r_{ni_n} \end{pmatrix} \end{array}$$

With this matrix in hand, a coincidence matrix is obtained, tabulating coincidence of evaluations in C. A binary example illustrates the calculation.

$$C_e = \begin{array}{c} \\ J_1 \\ J_2 \end{array} \begin{array}{c} \begin{array}{cccc} Av_1 & Av_2 & Av_3 & Av_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Where, C_e is an example of a Reliability matrix.

The coincidence matrix M in binary evaluations is described by:

$$M = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} o_{00} & o_{01} \\ o_{10} & o_{11} \end{pmatrix} \end{array}$$

The coincidences are tabulated twice, one for each row evaluation. The coincidence matrix of C_e is described by:

$$M_e = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 4 & 2 \\ 2 & 0 \end{pmatrix} \end{array}$$

Then, Disagreement is calculated by:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.5}$$

where, D_o is the Observed Disagreement and D_e is the Expected Disagreement.

For the example of C_e, the calculation procedure is:

$$\alpha = 1 - (n-1)\frac{o_{01}}{n_0 * n_1} \tag{2.6}$$

where,

- n_0 is the sum of 0 column in coincidence matrix
- n_1 is the sum of 1 column in coincidence matrix
- n is the sum of n_0 and n_1

So,

$$\alpha = 1 - (8 - 1)\frac{2}{6 * 2} = -0.16 \tag{2.7}$$

The lower the value the higher is the disagreement. Krippendorff (2011) indicates that 1 means perfect agreement among members, and values below of 0 indicates systematic disagreement. Since we are aiming to find the highest possible Disagreement, most of values found were below of zero.

3 Methodology

3.1 Genetic Algorithm

Genetic Algorithm (GA) (HOLLAND et al., 1975) is a nature-inspired algorithm based on Charles Darwin's evolution theory. It is one of the most used optimization algorithms, due to its effectiveness to deal with NP-hard problems. To perform optimization, GA works only with information of the fitness of an individual. It does not need information of the first and of the second derivatives. Therefore, it is an alternative to deal with non-continuous functions, in which it's not possible to calculate derivatives.

Although it is not possible to ensure that GA converges to an optimal solution, due to its stochastic nature, the algorithm presents fast convergence and is a promising algorithm for non-linear optimization problems achieving good results within acceptable computational time.

GA main procedure is illustrated in the figure 1. It consists in combining operators for exploration and exploitation of the search space, by Selection, Crossover and Mutation

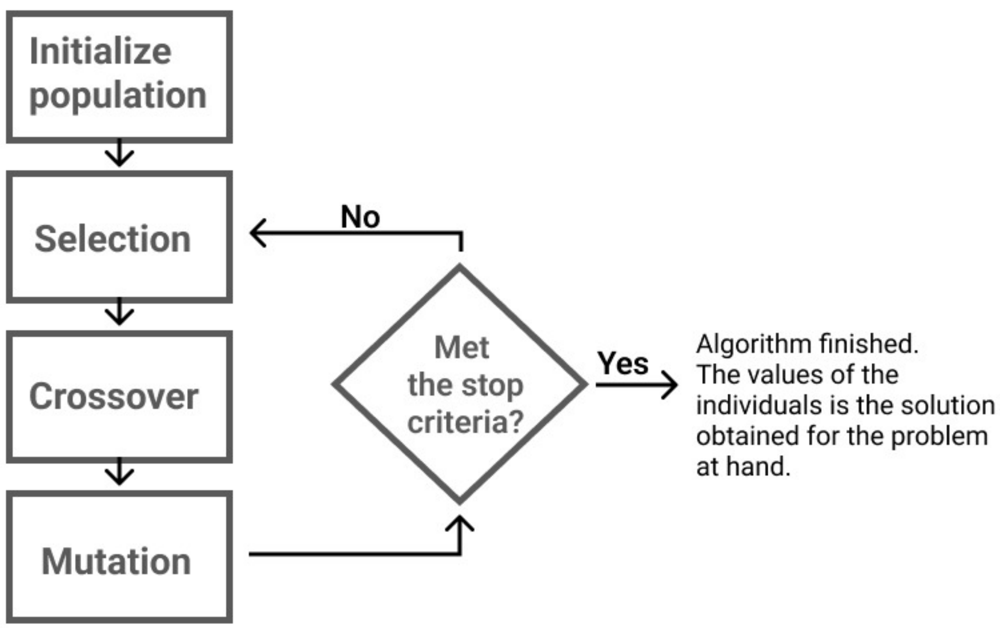

Figure 1 – GA Workflow

To illustrate the approach, Problem 1, a simple maximization problem will be solved with GA (LACERDA; CARVALHO, 1999).

Problem 1:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) = x^2 \\ \text{subject to} \quad & 0 \leq x \leq 31, \\ & \text{x} \in \mathbb{Z} \end{aligned} \tag{3.1}$$

3.1.1 Individual

An individual in GA is a possible solution. A genetic representation of this solution is called chromosome. Each gene of the chromosome stores the information about a variable of the optimization problem. Originally, GA was designed to deal with binary encoded chromosomes, however, it has been developed operators that handle Integer, Real and many types of encoding, expanding the possibilities of applications for GA. The set comprising all the individual is called population. Figure 2 shows an example population for a two variable problem.

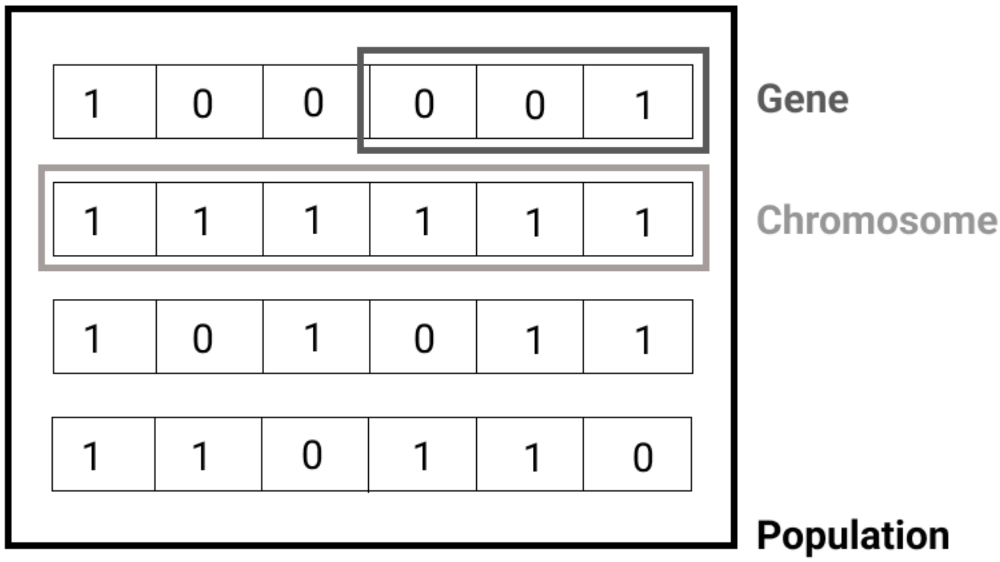

Figure 2 – GA Population

3.1.2 Fitness Evaluation

The second step is to evaluate each chromosome using the objective function, with the result of this evaluation the fitness can be computed (typically the function is the objective function) and each chromosome storage a fitness value. In this case, the fitness of the function is just the value of the function in the given point.

3.1.3 Generation of the initial population

In order to run a GA, it is necessary first to initialize the population of possible solutions. The most common strategy is generating random possible solutions. Population Size is the number of individuals in the population. Illustrating with the Problem 1. The first generation was generated with the following attributes

- 5 bits binary encoding; 01101 = 13
- Population size = 4
- Fitness is the objective function

Choromosome	**x**	**f(x)**
$A_1 = \mathbf{11001}$	25	625
$A_2 = \mathbf{01111}$	15	225
$A_3 = \mathbf{01110}$	14	196
$A_4 = \mathbf{01010}$	10	100

3.1.4 Selection

Inspired by natural selection, and Charles Darwin's premise of "Survival of the fittest", selection operator focus on choosing the best individual of the population. The fittest individuals are selected to generate new individuals using Crossover and Mutation operators.The main types of Selection are Roulette and Tournament. In Roulette, individuals with highest fitness have higher chance to be selected. Tournament selection splits the Population into groups of n individuals, and then, the best individual of the group survives. The size of n is called the selection pressure, and values are typically equal to 2. The Tournament operator is applied to the Population generated for Problem 1:

Individuals	**Fitness**	**Tournaments**	**Selected**
A_1	625	A_1 **vs** A_4	A_1
A_2	225	A_2 **vs** A_3	A_2
A_3	196	A_3 **vs** A_3	A_3
A_4	100	A_4 **vs** A_2	A_2

3.1.5 Crossover and Mutation

Crossover and mutation are operators for Exploring the search space, which means finding regions still not explored in the feasible space. To perform the diversification, crossover operator is used to combine a percentage of the selected individuals to produce an offspring to fill the next population. The percentage is an user-defined parameter, named crossover rate, and values are normally between 0.6 to 0.8. Individuals are randomly selected to be combined with a probability of crossover rate.

The figure 3 illustrates the binary Crossover

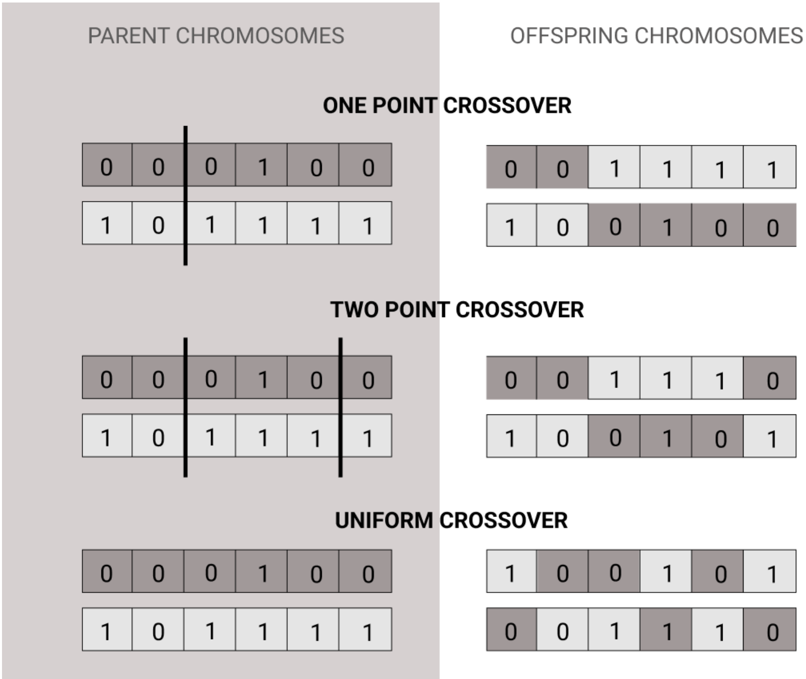

Figure 3 – Crossover Operators

After Crossover, Mutation is the next operator. Mutation adds a random noise to offspring chromosomes. It also occurs under a given probability, called Mutation rate, and typical values are in the range of 1-2%. The operator is a strategy to escape from local minima. Binary forms are shown in figure 4.

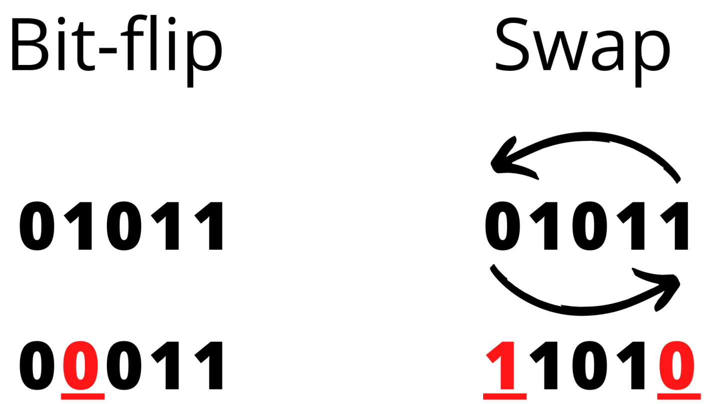

Figure 4 – Mutation Operators

For the Problem 1, the one-point crossover and bit-flip mutation were applied, and the working principle are illustrated in figure 5.

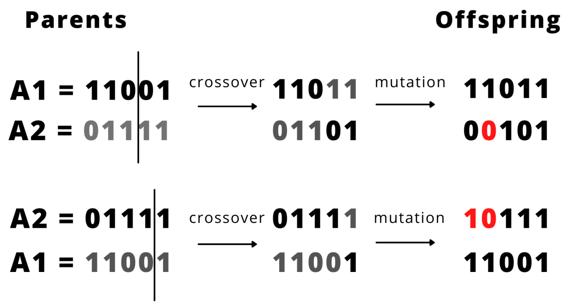

Figure 5 – Crossover and Mutation operators in Problem 1

The loop persists until a stop criterium is met. For the Problem 1, five generations were produced, generating the following Populations. The algorithm has found the optimal in the 4th Generation. The generated populations are presented in table 3.

		x	$f(x)$
Generation 1	**11011**	27	729
	11001	25	625
	01110	25	625
	01010	23	529
Generation 2	**11011**	27	729
	11000	24	576
	10111	23	529
	10101	21	441
Generation 3	**11011**	27	729
	11000	23	529
	10111	15	225
	10101	7	49
Generation 4	**11111**	31	961
	11000	27	729
	10111	23	529
	10101	23	529
Generation 5	**11011**	31	961
	11000	31	961
	10111	31	961
	10101	23	529

Table 3 – GA generations

3.1.6 Considerations about Genetic Algorithms

GA originally has been developed to deal with single-objective problems with no constraints. Approaches to constraint-handling have been proposed in two main directions. Add a penalty function to the fitness of individuals, where the penalty increases with the degree of violation of the constraint. And when comparing feasible and unfeasible individuals in selection, the feasible always wins, and this will be the principle used in in this work.

Regarding the number of objectives, multi-objective Genetic Algorithms (MOGA) approaches have been developed to handle many objectives. A hybrid technique, combining multi-criteria decision making and GA, have been tested in the work, but indeed, a state-of-the-art algorithm, NSGA-II (DEB, 2002b) outperformed it, in every tested case. Therefore, NSGA-II will be employed and explained in the following section.

3.2 Multi-Objective Optimization

Deb (2011) defined some important concepts for defining a multi-objective optimization problem, and this section highlights those definitions.

A multi-objective optimization problem comprises minimization or maximization of two or more of objective functions. Optimization may be performed either to maximize or minimize functions. The multi-objective optimization problem, like any other optimization problem, may include a number of constraints that any feasible solution must meet. The solutions satisfying the constraints and variable bounds constitute a feasible decision variable space (DEB, 2014).

3.2.1 Dominance

The main difference between single and multi-objective problem is that objective functions for more than one objective constitute a multi-dimensional space. The optimal solutions, in this case, are defined guided by the concept of Dominance (DEB, 2014).

A solution $x^{(1)}$ dominates other solution $x^{(2)}$ if and only if:

- A solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives
- $x^{(1)}$ is better than $x^{(2)}$ in at least one objective,

The notation to indicate the dominance is: $x^{(1)} \succeq \mathrm{x}^{(2)}$

3.2.2 Non-Dominated Set

Since the objective space is multi-dimensional, the approach to find optimal solutions does not rely on finding a single point, but a set of non-dominated points. Figure 6 shows a non-dominated front.

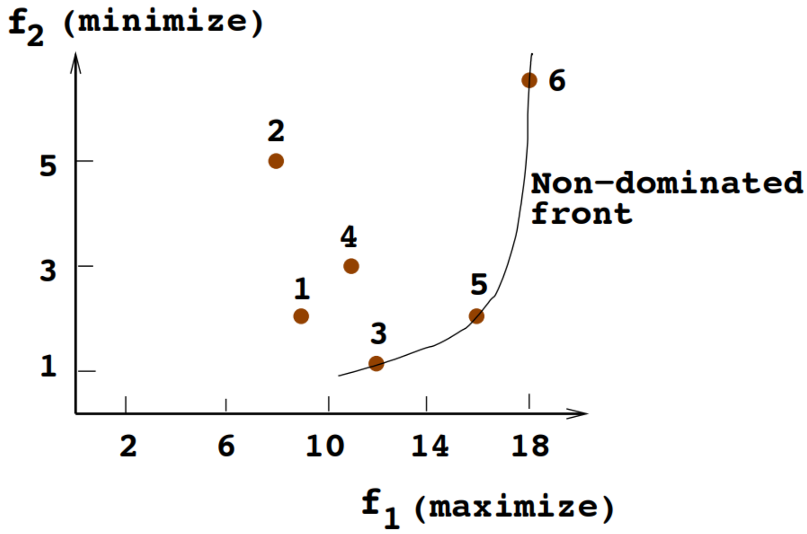

Figure 6 – Non-Dominated front (DEB, 2014)

A property of this set of points is that is not possible to improve one objective without worsening another one. Because of this trade-off, finding a diverse set of points is crucial to make an accurate final decision (DEB, 2014).

3.2.3 Pareto Front

Pareto globally optimal solutions are all non-dominated solutions within the search space. A solution is the Pareto optimal set if there is no feasible vector that dominates any solution of the set (COELLO et al., 2007). Figure 7 shows Pareto fronts in different optimization directions.

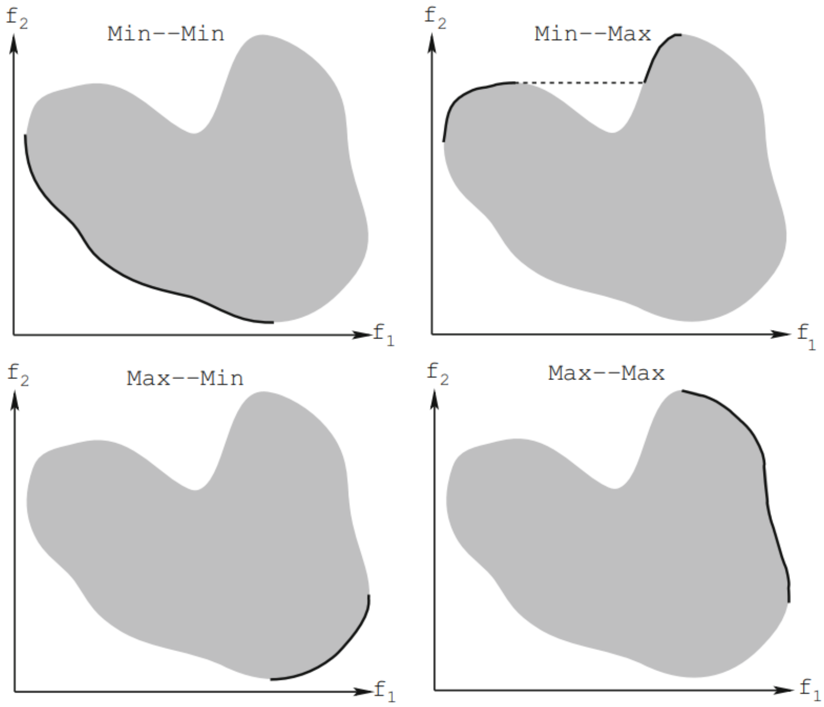

Figure 7 – Pareto Front examples in different types of problems (DEB, 2014)

Finding the true Pareto Front of a problem is usually quite difficult. Therefore, reasonably good approximations of are generally acceptable within limited computational time (COELLO et al., 2007).

3.2.4 NSGA-II

The evolutionary algorithm NSGA-II (DEB, 2002a) is the most used algorithm for multi-objective (VERMA; PANT; SNASEL, 2021). Based on an elitism mechanism, the genetic inspired algorithm main loop can be divided in three stages.

- Combination
- Non-dominated sorting
- Compute crowding distance

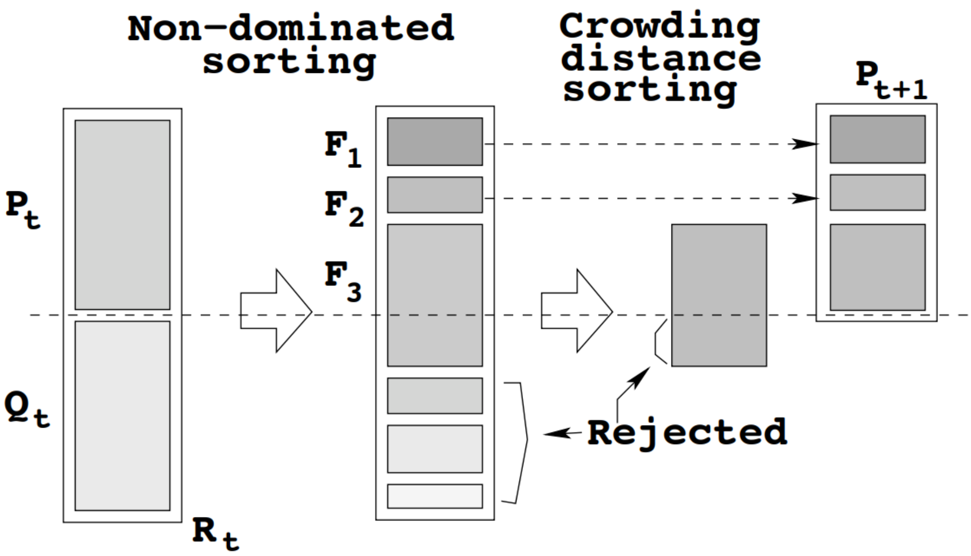

Figure 8 – NSGA-II main loop (DEB, 2002a)

Figure 8 illustrates NSGA-II procedure, and given an initial Population P_t, the first step is to perform crossover and mutation to generate the offspring Q_t. Combined, these two populations form R_t. The second step is to split R_t into frontiers by non-dominated sorting, in which, the first front F_1 is composed by the non-dominated solutions. To find the second, the elements of F_1 are not considered, and again, the non-dominated solutions must be selected to F_2, and so on, until fit the maximum possible solutions in a front. Figure 9 shows the solutions divided into frontiers.

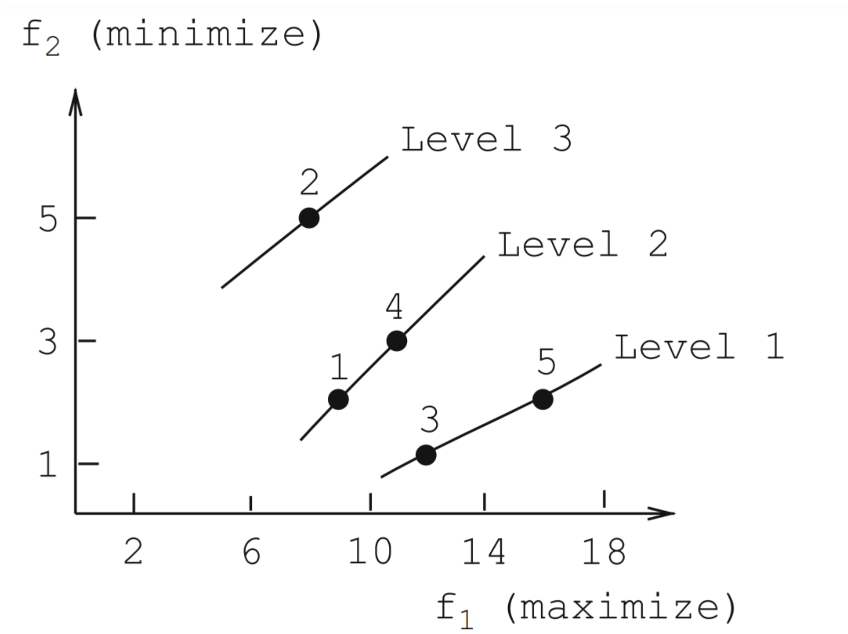

Figure 9 – Set of solutions and corresponding fronts (DEB, 2014)

Since the overall population size of R_t is bigger than the expected population size, some front could not accommodate in P_{t+1}. All fronts which could not be accommodated are simply deleted. When the last allowed front is being considered, there may exist more solutions in the last front than the remaining slots in P_{t+1}.

To select which solutions will survive, a crowding distance operator must be applied. The selection aims to preserve the diversity of the population by measuring the density of solutions surrounding a particular solution.

Let us consider the front F formed by the points 0, i-1, i, i+1, 1, in figure 10.

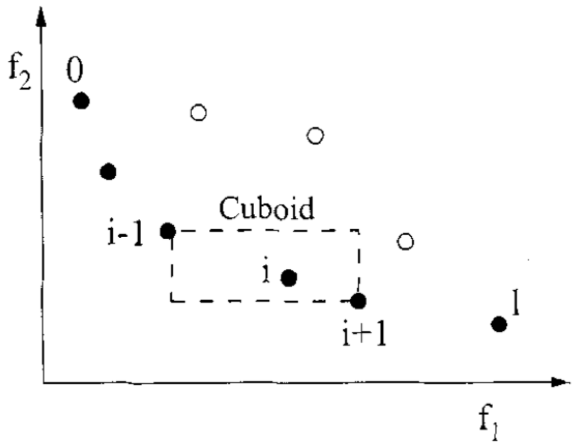

Figure 10 – Front F (DEB, 2014)

The crowding distance for point 0 and point 1, the extreme values of the front, tends to infinite. For each objective, a distance between a point and the extreme values must be computed. The sum of distances for each objective is the crowding distance. The pseudo code in Algorithm 1 describes the procedure to calculate the crowding distance for each solution.

Algorithm 1 Crowding Distance Calculation

r = F
for each i in r set $D_i = 0$
for each objective m **do**
    $F = \text{sort}(F, m)$
    $d_0 = d_1 = \infty$
    for $i = 2$ to $(r\text{-}1)$ **do**
        $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{max} - f_m^{min}}$
    end for
end for

Next, the solutions with highest crowding distance value survive in the population, and the main loop run until NSGA-II meet the stop criterium.

4 Experimental Results

4.1 Instances of the Problem

The problems presented in the benchmark dataset are detailed in table 4.

Dataset	Pessoas	Grupos	Departamentos
1	10	3	4
2	15	3	3
3	20	2	4
4	21	3	3
5	50	4	4
6	100	5	4
7	200	6	5

Table 4 – Benchmark Datasets (ESGARIO; SILVA; KROHLING, 2019)

Esgario, Silva e Krohling (2019) provided the sociometric and the requirement matrices for the given problems. So, to estimate disagreement among members the evaluations should be generated. Four evaluations for each individual were generated under Uniform Distribuition, ranging from 0 to 8, and the values were adopted in nominal scale to compute disagreement.

4.2 Ilustrative Example

To illustrate the approach, is presented the fitness calculation for a solution of the first instance of the proposed problems in the Benchmark Dataset. The sociometric matrix, S_1, of the problem with 10 Individuals given by:

$$S_1 = \begin{array}{c c} & \begin{array}{cccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} \end{array} \\ \begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \end{pmatrix} \end{array}$$

The proposed calculation of disagreement consists in computing the opinion of the members about recorded meetings, where 4 questions were made and evaluations are in nominal scale. The answers were simulated under Uniform distribution, values ranging from 0 to 8. This method of evaluation is an example, but Krippendorff's alpha is an indicator that can be used with almost every kind of evaluation, by computing the difference of personality, values, and opinions, with the possibility to combine evaluations.

The reliability matrix, C_1, for the problem is described by:

$$C_1 = \begin{array}{c c} & \begin{array}{cccc} Ev_1 & Ev_2 & Ev_3 & Ev_4 \end{array} \\ \begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{array} & \begin{pmatrix} 3 & 6 & 3 & 4 \\ 0 & 7 & 5 & 5 \\ 3 & 4 & 1 & 0 \\ 7 & 1 & 6 & 5 \\ 1 & 6 & 0 & 7 \\ 6 & 0 & 5 & 4 \\ 2 & 4 & 0 & 4 \\ 1 & 6 & 4 & 6 \\ 6 & 6 & 0 & 6 \\ 2 & 4 & 7 & 6 \end{pmatrix} \end{array}$$

The Requirement matrix R_1 is described by:

$$R_1 = \begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{ccc} G_1 & G_2 & G_3 \end{array} \\ \begin{pmatrix} 2 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

So, a possible solution A_1 for the problem is given by:

$$A_1 = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{array} \begin{array}{c} \begin{array}{ccc} G_1 & G_2 & G_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The Matrix notation is used to calculate the fitness of a possible solution, however, the chromosome of GA must be converted to the following vector notation:

$$A_1 = \begin{array}{c} \begin{array}{cccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} \end{array} \\ \begin{pmatrix} 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 \end{pmatrix} \end{array}$$

where the value stored on column I_i indicates the number of the group of element i.

4.2.1 Cohesion Measure

General cohesion for A_1 is calculated according to the procedure explained in section 2.2, and the result was 0.8. For GA, the fitness of a solution regarding cohesion is calculated by -1*E_g, in order to use both objective as minimization problems.

$$S_1 = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{array} \begin{array}{c} \begin{array}{cccccccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 & I_7 & I_8 & I_9 & I_{10} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & -1 & 1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \end{pmatrix} \end{array}$$

Each group's sociometric matrix was selected, according to the procedure of the sum of all values divided by the numbers of individuals of the group, and multiplied by a weight, that is the division of $n_i k$ per n_i. So, the value for cohesion is obtained for each group, in turn, the general cohesion, E_g, is the sum of all groups cohesion.

$$S_{g1} = \begin{array}{c} \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{array} \begin{array}{c} \begin{array}{cccc} I_3 & I_4 & I_5 & I_6 \end{array} \\ \begin{pmatrix} 0 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 \\ -1 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array} = E_1 * W_1 = 0.2$$

$$S_{g2} = \begin{array}{c} \\ I_1 \\ I_2 \\ I_7 \\ I_8 \end{array} \begin{array}{c} \begin{array}{cccc} I_1 & I_2 & I_7 & I_8 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array} = E_2 * W_2 = 0.7$$

$$S_{g3} = \begin{array}{c} \\ I_9 \\ I_{10} \end{array} \begin{array}{c} \begin{array}{cc} I_9 & I_{10} \end{array} \\ \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} \end{array} = E_3 * W_3 = -0.1$$

So, the value of general cohesion $E_g = 0.8$ for the solution A_1.

4.2.2 Disagreement Measure

The disagreement is calculated by means of the metric α for each group.

- $$C_{G_1} = \begin{array}{c} \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{array} \begin{array}{c} \begin{array}{cccc} Ev_1 & Ev_2 & Ev_3 & Ev_4 \end{array} \\ \begin{pmatrix} 3 & 4 & 1 & 0 \\ 7 & 1 & 6 & 5 \\ 1 & 6 & 0 & 7 \\ 6 & 0 & 5 & 4 \end{pmatrix} \end{array} \implies \alpha = -0.11$$

- $$C_{G_2} = \begin{array}{c} \\ I_1 \\ I_2 \\ I_7 \\ I_8 \end{array} \begin{array}{c} \begin{array}{cccc} Ev_1 & Ev_2 & Ev_3 & Ev_4 \end{array} \\ \begin{pmatrix} 3 & 6 & 3 & 4 \\ 0 & 7 & 5 & 5 \\ 2 & 4 & 0 & 4 \\ 1 & 6 & 4 & 6 \end{pmatrix} \end{array} \implies \alpha = \text{-0.018}$$

- $$C_{G_3} = \begin{array}{c} \\ I_9 \\ I_{10} \end{array} \begin{array}{c} \begin{array}{cccc} Ev_1 & Ev_2 & Ev_3 & Ev_4 \end{array} \\ \begin{pmatrix} 6 & 6 & 0 & 6 \\ 2 & 4 & 7 & 6 \end{pmatrix} \end{array} \implies \alpha = 0.045$$

The value α for A_1 is the sum of all groups α. In this case, the value is -0.083.

4.2.3 Constraint Violation

In order to compute the degree of constraint violation, a real allocation matrix must be created, indicating the numbers of members from a given department belonging to a given group. For example, the real allocation matrix of the problem, R_a is described by:

$$R_a = \begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{ccc} G_1 & G_2 & G_3 \end{array} \\ \begin{pmatrix} 2 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The degree of violation can be computed by means of the absolute value of the difference between R_1 and R_a. In this case, R_a is equal to R_1, which means that A_1 is a feasible solution.

4.2.4 Results interpretation

For α measure for disagreement, lower values indicates higher disagreement, the opposite for cohesion, where higher values indicates higher cohesion. So, values for cohesion in GA and NSGA-II is multiplied by -1, in order to transform to a minimization problem for both objectives because of the algorithm implementation used. In order to analyze the results, visualizing values in positive ranges are better to decision-makers understand the result and make an accurate choice.Therefore, both values for cohesion and disagreement are multiplied again by -1. So, to interpret results, the higher the value, the better is the solution.

Next, the proposed methodology is applied, and the experimental setup will be discussed in the following section.

4.3 Experimental Setup

To perform simulations, python (ROSSUM; JR, 1995) was used in google colab educational environment. Pymoo, a python based framework for Single and multi-objective optimization (Blank; Deb, 2020), was employed to run GA and NSGA-II.

An implentation of Krippendorff's alpha calculation was imported from a github repository. Pandas (MCKINNEY et al., 2010), Numpy (HARRIS et al., 2020) and Numba (LAM; PITROU; SEIBERT, 2015) were other python libraries used to support the algorithm implementation.

4.3.1 Parameters Setup

For both GA and NSGA-II, some parameters are set up for the experiments: Population size, selection operator, crossover operator and rate, mutation operator and rate and the stop criteria.

Defining hyperparameters is an important decision when using evolutionary algorithms. Good practices for defining these rates were discussed in literature (JONG, 1975) (SCHAFFER et al., 1989), and parameters for Crossover and mutation rates have been further explored (CZARN et al., 2004). Suggested values, typically, are in the range of 0.6 to 0.8. For mutation, typical values are not higher than 0.1. However, each problem needs an optimal configuration (CZARN et al., 2004), and to find the most effective one, a grid search was performed to support the decision about values for crossover and mutation rate. For the single-objective problem, quality assessment is direct, comparing the result of optimization. While for multi-objective, a criterium of Hypervolume(ZITZLER et al., 2003) was introduced to represent the quality of a Pareto set, and compare the obtained Pareto from different parameters setup.

For selection operator, Tournament is standard. Regarding, crossover and mutation operators, simulated binary crossover and permutations were defined empirically, because those method have outperformed other tested operators.

For population size, the value adopted by Esgario, Silva e Krohling (2019) is used in the initial problems, since the variable space is the same and good results were obtained. Therefore, population size equals to 50. However, for more challenging problems, this number is increased to 80, and better results were found.

The stop criterium adopted was the number of generations. Empirical tests were performed, and due to convergence to local minima, criteria of stopping after many iterations without improvement was not suitable for this problem, since it was possible to obtain improvement even after many generations stuck into local minima. So, the direct approach of number of generations (Number of functions evaluated) was employed.

The number must depend on the instance of the Benchmark Dataset, and empirical tests have found acceptable values, considering both efficacy and computational time. The table 5 presents the values for both problems. For simple GA, the effort needed is lower, because the target is to find only one solution, while for NSGA-II, a multi-dimensional space is searched, and then, more rounds must be performed.

	Multi-Objective		Single-Objective
Instance	**Generations**	**Population Size**	**Generations**
1	500	50	100
2	800	50	200
3	1250	50	400
4	2500	50	800
5	3500	80	1200
6	5000	80	1600
7	10000	80	2500

Table 5 – Population and Generations setup

4.3.2 Grid Search

Possible values for crossover and mutation rates are listed in the table 6 and 7, respectively. Every possible combination of these values will be tested for NSGA-II. Each combination is a optimization round, and a higher amount of computational effort is required. For this reason, one instance of the Benchmark Dataset was selected to be analyzed, and the optimal configuration for this problem will be adopted for other instances. The selected Dataset was the 5th.

Crossover rate
20%
40%
60%
80%

Table 6 – Crossover rates to grid search

Mutation rate
$1/n_i\%$
1%
5%
10%
25%

Table 7 – Mutation rates to grid search

The configuration which presented the best value regarding Hypervolume was crossover rate equal to 20% and mutation rate equal to 25%, in contrast with earlier recommendations in literature (JONG, 1975), but in consonance with more recent analysis (CZARN et al., 2004). For the single-objective problem, the same values adopted by

(ESGARIO; SILVA; KROHLING, 2019) are used, because of the accuracy achieved in many rounds of iterations.

The final parameters setup used in the experiments is presented in table 8:

Problem	Crossover Rate	Mutation Rate
Single-Objective	20%	$1/n_i$
Multi-Objective	20%	25%

Table 8 – Final parameter setup

4.4 Single-Objective Results

The setup obtained for single-objective problems, first for cohesion maximization and second for disagreement maximization, are detailed in the table 9. It is worth to mention that different solutions have been found for maximizing each objective. Each column indicates the fitness value of a solution.

Instance	Best Cohesion	Func. Evaluated	Best Disagreement
1	1.60	25000	0.08
2	2.33	40000	0.28
3	3.5	62500	0.08
4	2.66	125000	0.22
5	3.14	280000	0.15
6	4.18	400000	0.11
7	5.56	800000	0.10

Table 9 – Single-objective results

4.5 Multi-Objective Results

The Pareto fronts generated by NSGA-II are presented in this section. Both objectives are in maximization direction. In the following pages, the obtained Pareto front are presented in figures 11 to 17.

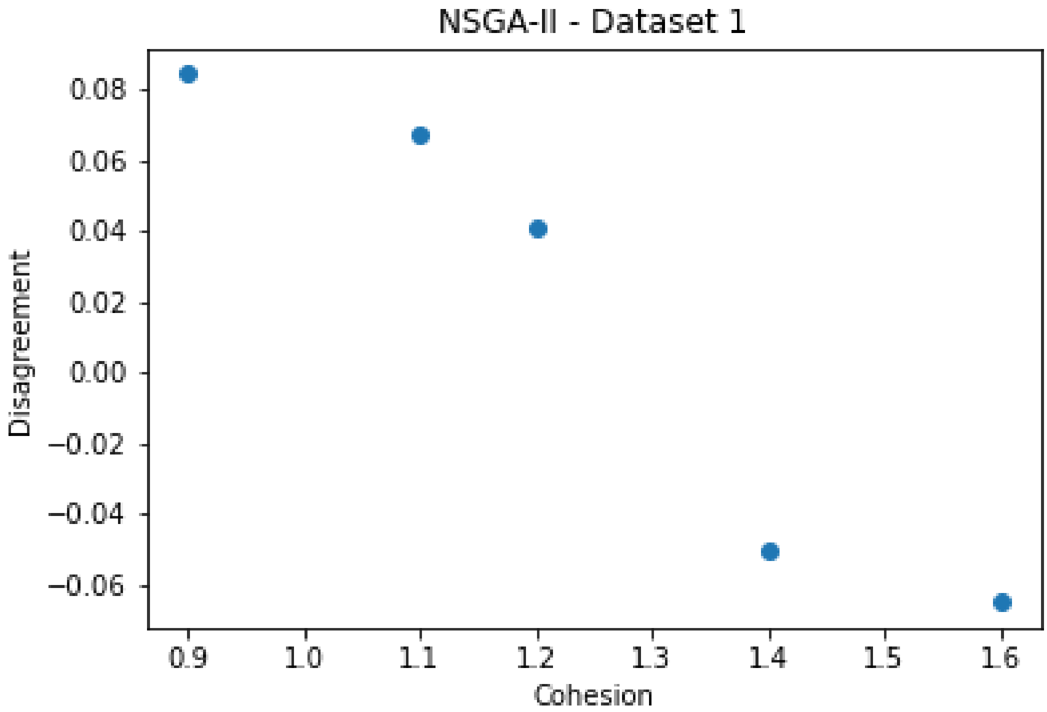

Figure 11 – Pareto Front - 1

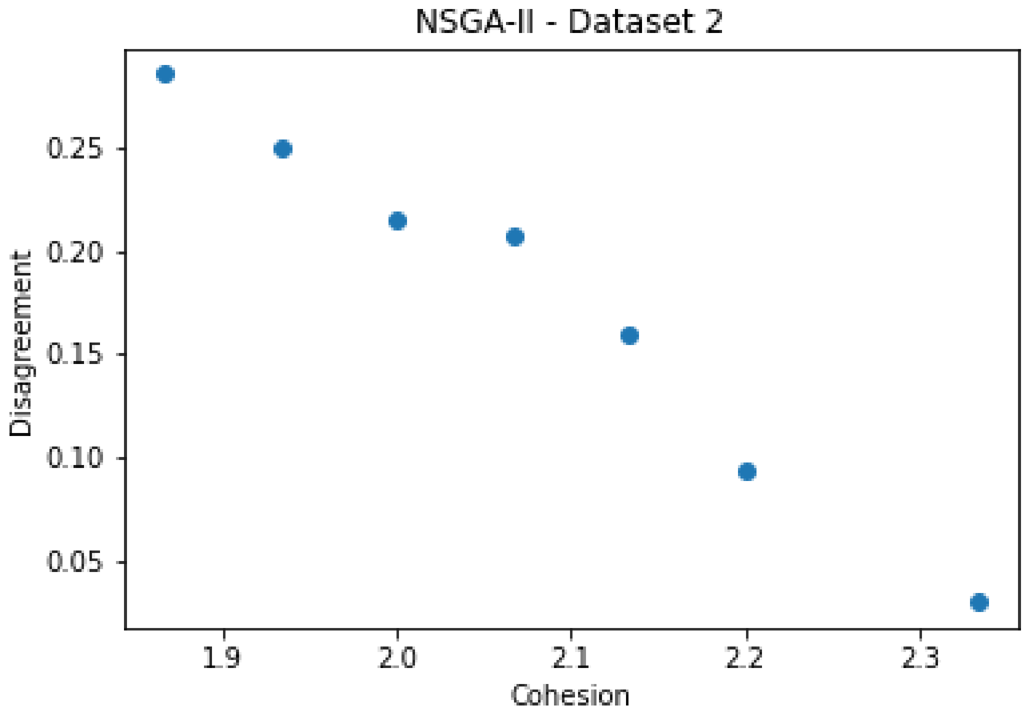

Figure 12 – Pareto Front - 2

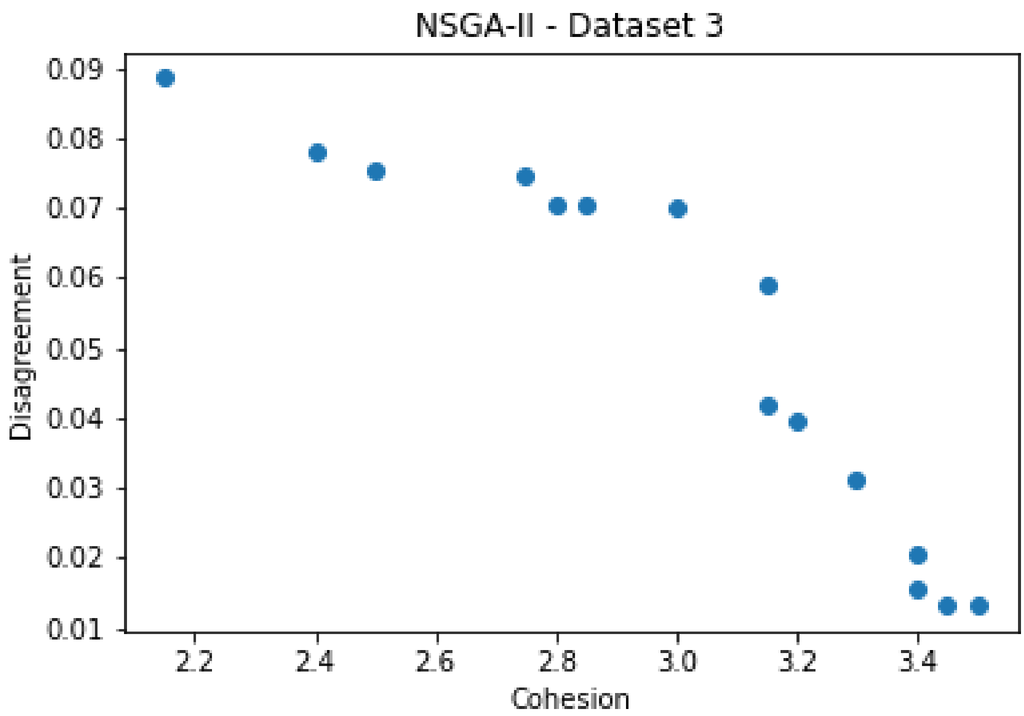

Figure 13 – Pareto Front - 3

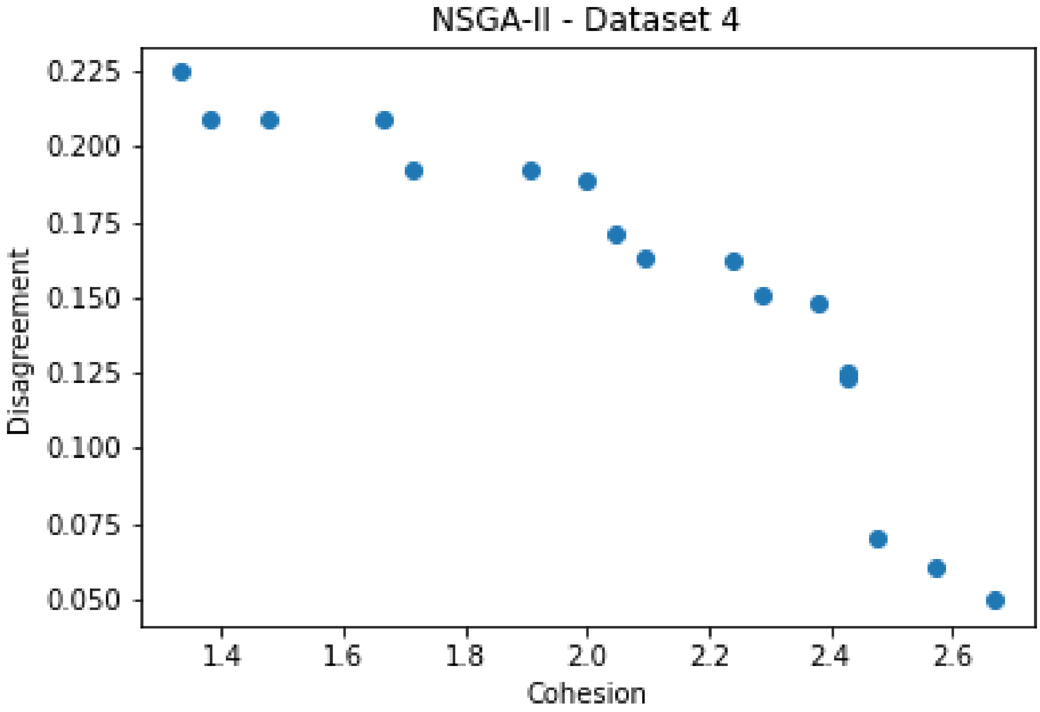

Figure 14 – Pareto Front - 4

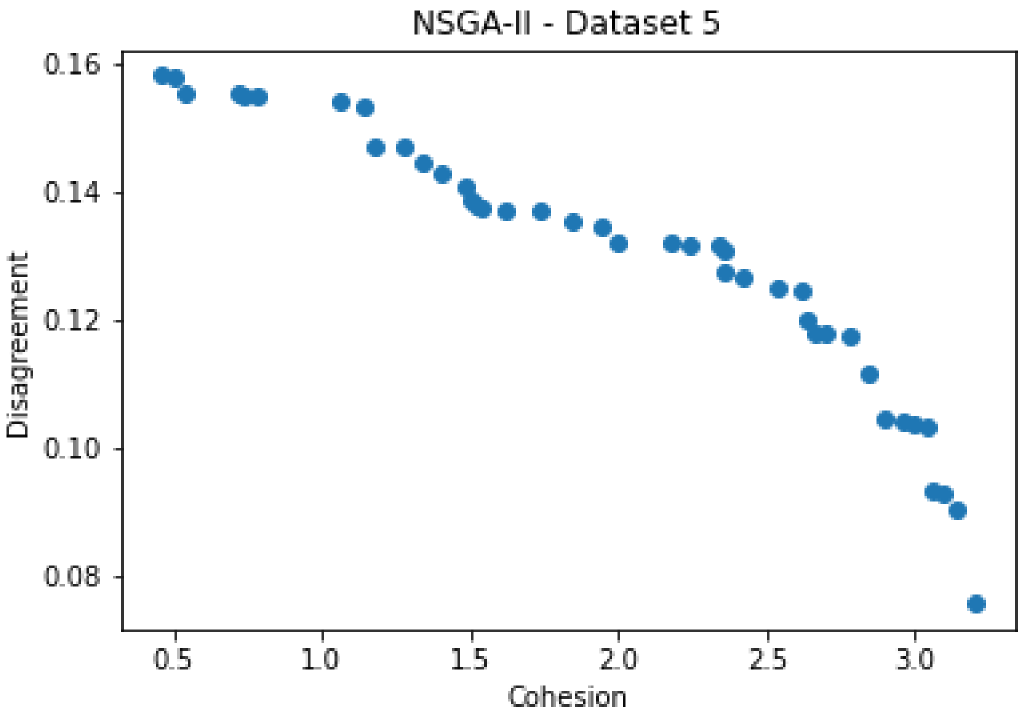

Figure 15 – Pareto Front - 5

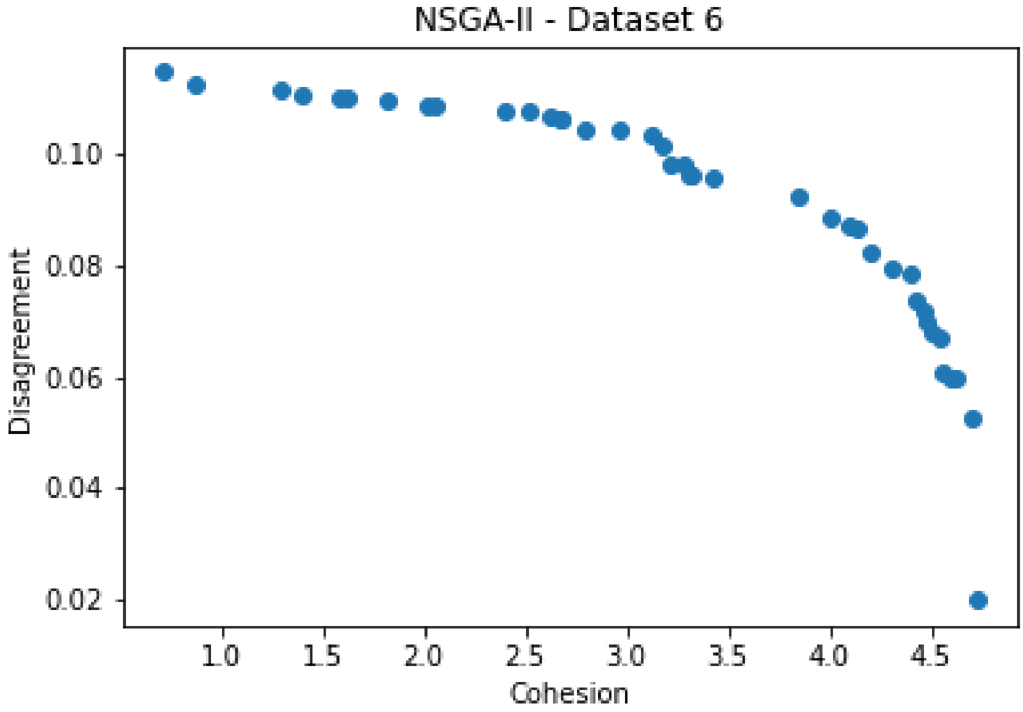

Figure 16 – Pareto Front - 6

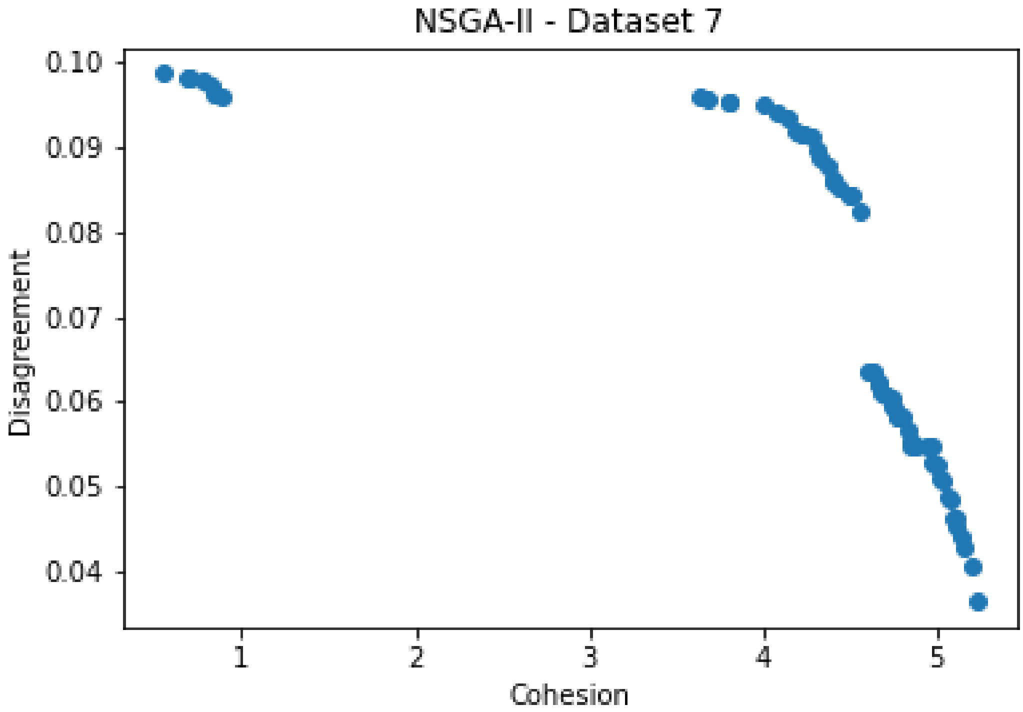

Figure 17 – Pareto Front - 7

NSGA-II was able to generate many and diversified solutions, being a promising choice to tackle the problems. However, challenging problems, with more individuals, and consequently larger search space, presents also more complexity regarding decision-making to select a single solution from optimal set, due to the increased number of solutions in each set.

5 Conclusion

This study has presented a method for Human Resource Allocation considering criteria of cohesion and disagreement. Regarding single-objective optimization, firstly, for cohesion, results are in concordance with that previously published. In addition, it is proposed the application of a quantitative metric, the Krippendorff's alpha, to disagreement measurement, because of its many possibilities of applications. A simulated reliability matrix was generated for each problem, considering four individual evaluations, in which the disagreement among those evaluations was computed. So, the proposed methodology could account intra-group disagreement, and it was possible to compute the fitness of an allocation through the metric and then optimize the allocation.

The single-objective approaches could find the optimal allocations in respect of each objective. Though combining both objectives is a strategy to better satisfy decision-maker needs, finding solutions that best balance both objectives, a multi-objective optimization was performed, using NSGA-II, a standard algorithm for human resource allocation problems. NSGA-II found a reasonable number of satisfactory and diversified solutions in the search space, demonstrating how computational intelligence can support the decision-making process in the problem of human resource allocation. However, in more challenging problems, the obtained Pareto fronts also present more solutions, increasing the complexity of the final decision. So, future works should explore post-Pareto evaluations. Moreover, analysis of disagreement metric behavior with different evaluations and scales should be investigated in future works.

References

BALLESTEROS-PEREZ, P.; GONZÁLEZ-CRUZ, M. C.; FERNÁNDEZ-DIEGO, M. Human resource allocation management in multiple projects using sociometric techniques. *International Journal of Project Management*, v. 30, n. 8, p. 901–913, 2012.

Blank, J.; Deb, K. pymoo: Multi-objective optimization in python. *IEEE Access*, v. 8, p. 89497–89509, 2020.

BOUAJAJA, S.; DRIDI, N. A survey on human resource allocation problem and its applications. *Operational Research*, v. 17, n. 2, p. 339–369, 2017.

BUSSE, C.; MAHLENDORF, M. D.; BODE, C. The abc for studying the too-much-of-a-good-thing effect: A competitive mediation framework linking antecedents, benefits, and costs. *Organizational Research Methods*, v. 19, n. 1, p. 131–153, 2016.

COELLO, C. A. C. et al. *Evolutionary algorithms for solving multi-objective problems.* [S.l.]: Springer, 2007. v. 5.

CZARN, A. et al. Statistical exploratory analysis of genetic algorithms. *IEEE Transactions on Evolutionary Computation*, v. 8, n. 4, p. 405–421, 2004.

DEB, K. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-2. *IEEE Trans. Evol. Comput.*, v. 6, n. 2, p. 182–197, 2002.

DEB, K. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-2. *IEEE Trans. Evol. Comput.*, v. 6, n. 2, p. 182–197, 2002.

DEB, K. Multi-objective optimisation using evolutionary algorithms: an introduction. In: *Multi-objective evolutionary optimisation for product design and manufacturing.* [S.l.]: Springer, 2011. p. 3–34.

DEB, K. Multi-objective optimization. In: *Search methodologies.* [S.l.]: Springer, 2014. p. 403–449.

ESGARIO, J. G.; SILVA, I. E. da; KROHLING, R. A. Application of genetic algorithms to the multiple team formation problem. *arXiv preprint arXiv:1903.03523*, 2019.

EVANS, C. R.; DION, K. L. Group cohesion and performance: A meta-analysis. *Small group research*, SAGE PUBLICATIONS, INC. 2455 Teller Road, Newbury Park, CA 91320, v. 22, n. 2, p. 175–186, 1991.

HARRIS, C. R. et al. Array programming with NumPy. *Nature*, v. 585, n. 7825, p. 357–362, set. 2020.

HARRISON, D. A.; PRICE, K. H.; BELL, M. P. Beyond relational demography: Time and the effects of surface-and deep-level diversity on work group cohesion. *Academy of management journal*, v. 41, n. 1, p. 96–107, 1998.

HOLLAND, J. H. et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* [S.l.]: University of Michigan press, 1975.

HÜLSHEGER, U. R.; ANDERSON, N.; SALGADO, J. F. Team-level predictors of innovation at work: a comprehensive meta-analysis spanning three decades of research. *Journal of Applied psychology*, v. 94, n. 5, p. 1128, 2009.

HUNT, V.; LAYTON, D.; PRINCE, S. Diversity matters. *McKinsey & Company*, v. 1, n. 1, p. 15–29, 2015.

JONG, K. A. D. *An analysis of the behavior of a class of genetic adaptive systems.* [S.l.]: University of Michigan, 1975.

KRIPPENDORFF, K. Computing krippendorff's alpha-reliability. 2011.

LACERDA, E. G. de; CARVALHO, A. D. Introdução aos algoritmos genéticos. *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*, v. 1, p. 99–148, 1999.

LAM, S. K.; PITROU, A.; SEIBERT, S. Numba: A llvm-based python jit compiler. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC.* [S.l.: s.n.], 2015. p. 1–6.

LOTT, A. J.; LOTT, B. E. Group cohesiveness as interpersonal attraction: A review of relationships with antecedent and consequent variables. *Psychological bulletin*, v. 64, n. 4, p. 259, 1965.

LUSSIER, R. N. *Management fundamentals: Concepts, applications, and skill development.* [S.l.]: Sage Publications, 2017.

MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference.* [S.l.], 2010. v. 445, p. 51–56.

MESMER-MAGNUS, J. R.; DECHURCH, L. A. Information sharing and team performance: a meta-analysis. *Journal of applied psychology*, v. 94, n. 2, p. 535, 2009.

ROSSUM, G. V.; JR, F. L. D. *Python reference manual.* [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

SANDERS, K.; NAUTA, A. Social cohesiveness and absenteeism: The relationship between characteristics of employees and short-term absenteeism within an organization. *Small Group Research*, v. 35, n. 6, p. 724–741, 2004.

SCHAFFER, J. D. et al. A study of control parameters affecting online performance of genetic algorithms for function optimization. In: *Proceedings of the 3rd international conference on genetic algorithms.* [S.l.: s.n.], 1989. p. 51–60.

VEEN, D.-J. van; KUDESIA, R. S.; HEINIMANN, H. R. An agent-based model of collective decision-making: How information sharing strategies scale with information overload. *IEEE Transactions on Computational Social Systems*, IEEE, v. 7, n. 3, p. 751–767, 2020.

VERMA, S.; PANT, M.; SNASEL, V. A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*, v. 9, p. 57757–57791, 2021.

WANG, J. et al. Team creativity/innovation in culturally diverse teams: A meta-analysis. *Journal of Organizational Behavior*, v. 40, n. 6, p. 693–708, 2019.

ZITZLER, E. et al. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, v. 7, n. 2, p. 117–132, 2003.