Pedro Henrique Simoes Marcal

2020300

Artificial Intelligence / Data Visualization and Communication CA

Contents

[Introduction 2](#_Toc155648131)

[Report 3](#_Toc155648132)

[**Artificial Intelligence** 3](#_Toc155648133)

[**Data Visualization and Communication** 6](#_Toc155648136)

[**Rationale of the code** 10](#_Toc155648142)

[**Conclusion** 14](#_Toc155648143)

[**References** 15](#_Toc155648144)

# **Introduction**

**Tasks for Artificial Intelligence**

Ciara is looking for employees for her new company, which develops and provides AI based logistic software for retailers. Ciara has determined that she needs: 2 Python Programmers, 2 AI Engineers, 1 Web Designer, 1 Database Admin, and 1 Systems Engineer. Assume that if a person has two abilities, he or she can take on two roles in the company.

Using any CSP (Constraint Satisfaction Problem) framework (using variables, value domains, and constraints), discover if the above problems can be solved and if so, detail who would be in hired**.**

Discuss in detail how using Constraint Satisfaction finds an answer or finds no solution to the problems in Tasks for Artificial Intelligence part 1. How does this differ from standard algorithmic solutions?

These problems be solved using several other algorithm’s we have studied in the module. Choose one of these algorithms and discuss your answer in detail including a proof of your hypothesis in code**.**

**Tasks for Data Visualisation**

Use appropriate visualisations to help communicate the CSP scenario and the corresponding solutions, if any, to the appropriate stakeholders.

Create interactive visualisation(s) to allow a user to explore alternate constraint scenarios.

Create GUI(s) to allow a user to explore alternate constraint scenarios.

Include in your report a section for a theoretical AI “team” you are part of, explaining the visualisation processes and rationalising your visualisation decisions (e.g., chart choice, colour, layout etc).

# **Report**

# **Artificial Intelligence**

*Scenario 1*

*Variables* The variables for this scenario are all the names involved. {Peter, Juan, Jim, Jane, Mary, Bruce, Anita}.

*Value Domain*

Value Domain for this scenario are the abilities each person has. {Python Programmer, AI Engineer, Web Designer, Database Admin and Systems Engineer}

*Constraints*

Constraints for this scenario are the numbers of hires they can have, 3, and the number of hires needed for each department, 2 Python Programmers, 2 AI Engineers, 1 Web Designer, 1 Database Admin, 1 Systems Engineer and that Ciara knows Python.

*Result* Jane, Juan and Jim. Jane and Ciara will act as a Python Programmers, Jane will take the other position as a Database Admin since she is the only one with this skill, Juan will be working as the only Web Designer and as one of the AI Engineers and Jim will take the second position as an AI Engineer and the only position as Systems Engineer.

Scenario 2

*Variables* The variables for this scenario are all the names involved. {Peter, Juan, Jim, Jane, Mary, Bruce, Anita}.

*Domain*

Value Domain for this scenario are the abilities each person has. {Python Programmer, AI Engineer, Web Designer, Database Admin and Systems Engineer}

*Constraints* Constraints for this scenario are the numbers of hires they can have, 2 Python Programmers, 3 AI Engineers, 1 Web Designer, 1 Database Admin, 1 Systems Engineer and that now they can hire 4 people with one extra person needed for AI Engineer.

# *Result*

Peter, Jane, Jim and Anita. Peter will act as one of the two Python Programmers and as a one of the three AI Engineers, Jane will act as a second Python Programmer and as the only Database Admin, Jim will act as the second of the three AI Engineers and as the only Systems Engineer and Anita will act as the third AI Engineer and as the only Web Designer.

# *CSP Framework for Solving Problems*

Constraint Satisfaction Problem is the method used to find a solution to a one or more constraints within a problem, finding values for a group of variables that will satisfy the requirements. CSP has three components: Variables, Domain and Constraints.

Variables: Values that need to be determined are known as variable, they need to be assigned to satisfy a set of constraints. In our case the variables were the name of each person.

Domain: The value that each variable can have or hold is known as value domain. In our case, each variable, name, hold one or two values, the abilities and those abilities were the Value Domain.

Constraints: Constraints are the specific requirements or rules that need to be satisfied in order to get a result. In our cases constraints where the number of hires they needed in each department and the number of hires they could do, per example.

After defining those three components within our problem, we can code, using python in our case per example, using the constraints library, and to come to an answer by using backtracking or forwardtracking algorithms to return our answer.

*How CSP finds an answer for both scenarios*

CSP uses a backtracking or forwardtracking algorithm to return the answer to our set of constraints. In our case, it checks possibilities of assigning combinations of variables and values, so it combines three different people and their respective roles to see if all the constraints are satisfied, if not, it deletes and will not try that combination again, if all constraints are satisfied then it returns that outcome. In the second case, it takes set of combinations between four people and their abilities until it finds a combination that satisfies all constraints. One of the possible solutions for scenario 1 is: Jane {Python and Database Admin}, Juan {Web Designer, AI Engineer} and Jim {Systems Manager and AI Engineer}. Ciara will take the second Python position available for this scenario, filling the requirements of 2 Python Programmers, 2 AI Engineers, 1 Web Designer, 1 Database Admin, and 1 Systems Engineer with 3 hirings. Scenario 2: Peter {Python and AI Engineer, Jane {Python and Database Admin}, Jim {Systems Manager and AI Engineer} and Anita {AI Engineer and Web Designer} filling the requirements of 2 Python Programmers, 3 AI Engineers, 1 Web Designer, 1 Database Admin, and 1 Systems Engineer with 4 hirings.

*CSP Vs Standard Algorithms*

CSP stands for Constraints Satisfaction Problem and those constraints have requirements that need to be met to be completed. Is very used to solve problems where it is needed to assign values to variables in a way where all restrictions are satisfied. To solve CSP all is needed is to define what are the variables, domain value and constraints of our problem, apply the algorithm and look for the optimal solution that satisfies all constraints. One of the algorithms used to solve it is known as backtracking algorithm used for search and optimization, it works by “building” candidates and checking is that is a viable and accept solution, if it is not the algorithm will not consider those candidates again, will move back and try with another set of candidates this way will check and get to all the possible solutions. In standard algorithm the approach is different as well as the application field, while CSP represents the term by variables, domain values and constraints meanwhile in the standard algorithm require a more step by step explanation of each phase of the process and involves a more logic and detailed code for problem-solving.

*Alternative Framework for Solving Problems*

Genetic algorithm is an algorithm based on the natural selection, our biological evolution. The code selects individuals from our current population to be “parents” to produce “kids” for the next generation. When successful, the population “evolves” closer to the best solution. This algorithm belongs to the class of evolutionary algorithms. It works following the process of natural selection, the species that can adapt will survive. Each individual, variable, compete in population with other individuals. Individuals that survive, known as “fittest”, are put with another individual that survived to spread the “genes”, the values, and following this until it finds the best solution. Through each step of the process, means you are closer to the solution. Here is the step by step:

Chromosome Representation: In this algorithm, each chromosome represents a possible solution for our problem. Per example; chromosome = {“Jim”: [“Python”], “Jane”: [“Database”], “Anita”: [“AI Engineer”]} would be one of the possible solutions for our issue.

Fitness Function: In this phase, will be evaluated how much the chromosome is fit for the answer, remember the more “generation” the closer you are to the best solution.

Initializing: Here the algorithm starts by defining a variable that will be used as an initial population and each of the solutions are the Chromosomes used in the first phase. Per example; initial\_pop = {“Jim”: [“Python”], “Jane”: [“Database”], “Anita”: [“AI Engineer”]}; {“Ciara”: [“Python”], “Jane”: [“Database”], “Anita”: [“AI Engineer”]} etc.

Evaluation: Here is when the defined Fitness are used. Is evaluated, by checking each chromosome within the “initial\_pop” variable and seeing how much that chromosome fits in to our Fitness function. The outcome is a fitness score for each of the chromosomes.

Selection/Cross Over/Replacement/Mutation: Selection is used to select for the reproduction of each gene, like mentioned above, based on their fitness score, also known as “survivors”. The higher the fitness score the higher your chance to be selected for the next “generation”. Cross Over is when the pairs of the “survivors” selection are exchanging “genes”, or the roles, to create new combinations. Mutations are random changes that are made to create new combinations. Replacement is when the old chromosome defined first on the initial population variable is replaced with a new set of chromosomes, that are new different combinations of answer with a higher fitness score. The further you go with the “generations” or new set of chromosomes, the closes you are to first the best solution.

Termination: Here is used to define how many generations we want go to find our best solution or until the termination criteria is met.

# **Data Visualization and Communication**

**Brief introduction to CSP and solutions to our case**

# Constraint Satisfaction Problem is used when there are variables that need to be assigned to a value and must satisfy a set of rules, or constraints. It has three components:

Variables: Variables are the values that need to satisfy a set of constraints.

Domain: Domains are the values that are assigned to the variables.

Constraints: Constraints are the set of rules that need to be satisfied to have an outcome.

CSP Solutions for Scenarios 1 and 2

For scenario 1, they can hire 3 more people and they are: Jane, Juan and Jim. Jane and Ciara will act as a Python Programmers, Jane will take the other position as a Database Admin since she is the only one with this skill, Juan will be working as the only Web Designer and as one of the AI Engineers and Jim will take the second position as an AI Engineer and the only position as Systems Engineer.

For scenario 2, they can hire 4 more people but they need a extra AI Engineer and they are: Peter, Jane, Jim and Anita. Peter will act as one of the two Python Programmers and as a one of the three AI Engineers, Jane will act as a second Python Programmer and as the only Database Admin, Jim will act as the second of the three AI Engineers and as the only Systems Engineer and Anita will act as the third AI Engineer and as the only Web Designer.

**Code Demonstration**

For information about rationale of the code, check page 8.

Here is the interface of our project, with a scroll down menu to alternate between the scenarios and the solutions and the name and student number of the author:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# Once that you choose which scenario you wish to know the solution for, just click on the dropdown menu, select it and click the “View Solution to Scenario” to see the answer.

If selected scenario 1:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

If selected scenario 2:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# **Graphs**

Here is the graph explanation for a better visualization of the problem. Ciara has a group of people listed with their respective abilities to choose from, in order to fill all the positions that the company needs. This graph below shows us each position and their respective number of hires. A bar chat was chosen for both scenarios since it is more readable when talking about variables and number, we can see that at the “Count” sector we have a range of number going from 0 until the maximum value within our data, in this case the number 2 and the colours helps to distinguish the roles that are mapped with the name of each role. We can see that for the Python and AI Engineer positions, 2 people are needed and for Web Designer, Database Admin and Systems Engineer, just one person for each role.

Scenario 1:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

# Scenario 2:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

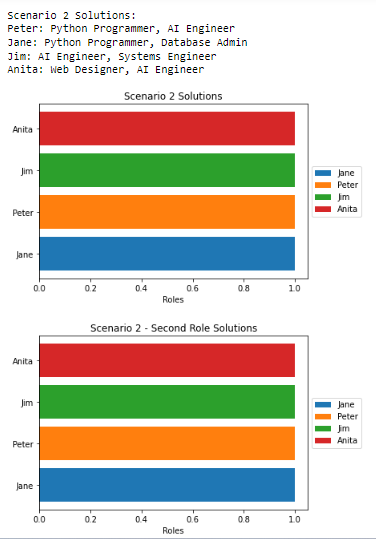
# Here is the graph referent to the solution, showing the group of people that satisfied the set of constrains.

Scenario 1:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Scenario 2:



# **Rationale of the code**

**Artificial Intelligence**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

On this part of the code, first we create a method to be the problem by using “def create\_problem()” and then creating a variable to call the “Problem()” function. After, we create two variables, “names” to store the names of each candidate and “roles” to be the roles available to be filled. Then create a for loop to itinerate through the names and write the “addVariable” to store name and roles into the problem variable. After, return function to return the name.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Here is the second part of the code. We started by creating another def function to solve the constraints named “def solve\_problem” with the problem created before assigned as a parameter and after we returns the solutions, if any are found. Next step was to create a variable for each scenario, “problem1” and “problem2” and assign the def function to it, “def create\_problem()”, containing the variables and domain needed. After, we create the constraints required for each scenario with the “addConstraint” method, for first scenario using lambda, that allows us to create functions without having to create it before hand, defines the constraints needed to satisfy, in the first scenario, is programmed to fill all the positions with no more than three hires and defining the roles that are required more than one person and for the second scenario, also using lambda, it is programmed to fill all the roles, defining which role requires more than one person and the maximum number of hires, that are four. Last step is to assign our constraints by assigning it to the def function to solve the problem with the scenario used as a parameter and print it, “solution1 = solve\_problem(scenario1)” and “solution2 = solve\_problem(scenario2)”.

**Data Visualization and Communication**

**Texto

Descrição gerada automaticamente**

On this part of the code, I am creating graphs to show the relationship between the candidates and the roles they are filling. First, the def function, “def plot\_solutions\_graph”, is used to extract each role and employees. After, a for loop is created to run until all the employees are met and fill up. Then using plt library, we create your graph, design it by including titles, labels and legends and display it. Last step is to use print functions to have a better understanding working together with the graph and plot each scenario by assigning it as a parameter.

Texto

Descrição gerada automaticamente

Here using def function to create the bar chart graph with the number of roles needed for each scenario as a parameter and assigning roles to the respective roles and counts, as the number that was defined by each scenario. After, define the colours, labels and titles needed using plt library and rotate the labels so it will not conflict with the other labels and gets unreadable, “plt.xticks(rotation=45, ha=’right’). Last step is to define the roles and how many people are needed to fill them for each scenario with two different variables, “hires\_scenario1” and “hires\_scenario2” and apply it as a parameter on the first def created for the bar chart, “def create\_bar\_chart”.

Texto

Descrição gerada automaticamente

Here we are creating a def function, “def on\_scenario\_selected” to clean previous outputs and display the solution wished as the button is clicked. After, if else if and else loop is used to display, breaking lines, each solution for each scenario as the button is clicked, “if selected\_scenario == “Scenario 1” then display solutions for scenario 1 and “if selected\_scenario == “Scenario 2” then will display solutions for scenario 2, “else” function is used to prevent errors only since we do not have another button to click on. Next step is to create the GUI using tkinker library, “app = tk.Tk()” and define title of the application, “app.title”, create a dropdown menu to navigate through the solutions, name button as required, in this case I have also included my student name and number as a button but it does not do any action when clicked on, set up the size, height and width for the app and run it by using “app.mainloop()”.

# **Conclusion**

# At the end of this project, I acquired a deeper understanding of what are Constraints Satisfaction Problems, a use- case of how it works in the real world and how it is important for machine learning. In the first scenario, after writing and defining which one are variables, which one are the domains and what are the constraints that need to be satisfied to find a better solution. After this, before coding, I just had to apply the constraints in each scenario to find the answer. For the first scenario, with a number of three hires to fill the positions, I have found Jane, Juan and Jim to be the best solutions with the additional of Ciara working as a Python Programmer, for the second scenario, with a number of four hires available but one extra AI Engineer, the best solution that I found was Peter, Jane, Jim and Anita. The difference between two very high-quality codes that are CSP, that uses backtracking or forwardtracking algorithm to find the answer, and Genetic algorithm, that is an algorithm based on the natural selection, very interesting because it is the first time I have read about it, which both algorithms being used for different purposes, Genetic algorithm is used more for optimization problems and CSP is used when we have our explicit set of constraints. For Data Visualization and Exploration, was very important to write down the report since it helped myself to understand even more of the subject as I was explaining it on to the report and this has been the first time I have ever created a GUI Interface using Python language before. By using google I was able to choose the library needed for a GUI, mine was tkinker, which is a very easy, simple and efficient tool to use for graphic user interface, all is needed is to create the function and create buttons, set titles and layouts.

# **References**

GeeksforGeeks (2023) *Constraint satisfaction problems (CSP) in Artificial Intelligence*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/constraint-satisfaction-problems-csp-in-artificial-intelligence/ (Accessed: 12 December 2023).

Upadhyay, S. (2023) *Backtracking algorithm [comprehensive guide]*, *Simplilearn.com*. Available at: https://www.simplilearn.com/tutorials/data-structure-tutorial/backtracking-algorithm#:~:text=Backtracking%20is%20a%20general%20algorithm,completed%20to%20a%20reasonable%20solution. (Accessed: 04 January 2024).

Wikipedia (2024) *Genetic algorithm*, *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Genetic\_algorithm (Accessed: 06 January 2024).

*What Is the Genetic Algorithm?* (2023) *What is the genetic algorithm? - MATLAB & simulink - mathworks United Kingdom*. Available at: https://uk.mathworks.com/help/gads/what-is-the-genetic-algorithm.html (Accessed: 08 January 2024).

GeeksforGeeks (2023b) *Genetic algorithms*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/genetic-algorithms/ (Accessed: 08 January 2024).