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** 12](#_Toc155648143)

[**References** 13](#_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:

Initializing:

Evaluation:

Selection/Cross Over/Replacement/Mutation:

Termination:

# **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

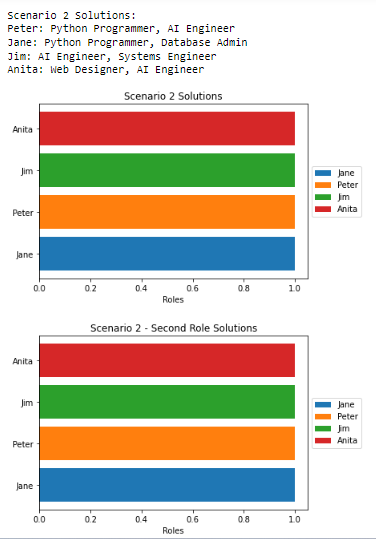
# Here is the graph referent to the solution, showing the group of people that satisfied the set of constrains.

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

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

# **Conclusion**

# **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).