

CA

Artificial Intelligence  
And  
Data Visualisation &   
Communication

Studant: Charles Franklin Jahn   
Student number: 2020315  
Lecturers: David McQuaid & Sam Weiss

04/12/2023 - 05/01/2024

BSc (Hons) in Computing in IT - 3rd Year  
Module: AI and Data Visualization & Communication

[**AI: Constraint Satisfaction Problem 2**](#_w1hcouxh714j)

[Code 2](#_xanhbyabpv6z)

[Scenario 1 3](#_pmk9qyfselvs)

[Case 1: 3](#_z27effcam8k7)

[Case 2: 4](#_856uumgmlwrz)

[Scenario 2 5](#_iduhk2wnvvtq)

[Case 1: 5](#_moifcpy60gu2)

[Case 2: 5](#_rwxqfgaere8t)

[**Constraint Satisfaction vs Standard Algorithm 6**](#_t4txo2ux7nj7)

[**Constraint Satisfaction vs Depth-First Search 7**](#_5iif45naebe9)

[DFS (Depth-First Search): 7](#_kriox3k8u6vk)

[Constraint Satisfaction Problem (CSP): 8](#_aixpkhxx7ee2)

[**Constraint Satisfaction Problem Visualization 8**](#_mo8t7da5trun)

[Pie charts 8](#_m4gq1qpxba0d)

[Bar charts 9](#_n9muro2wr4xm)

[Line charts 10](#_j1ki21z5b9rj)

[**Github repository -> https://github.com/CharlesCCT2020315/CA3 11**](#_9ci3ueutqom9)

# 

# 

# 

# 

# 

# AI: Constraint Satisfaction Problem

In this project, I will use CSP (Constraint Satisfaction Problem) to offer a solution to the proposed problem and Tkinter for the graphical presentation of the application (GUI). The given problem has two scenarios, which are:

Scenario 1: "Ciara has determined that she needs:

* 2 Python programmers,
* 2 AI engineers,
* 1 Web Designer,
* 1 database administrator, and
* 1 systems engineer.

Suppose that if a person has two skills, he can take on two roles in the company. Suppose Ciara knows Python and only has the funds to hire three more people."

Scenario 2: "Suppose Ciara and Juan become partners. With the additional funds, they can now employ four more people, but they must hire another AI engineer. So they need:

* 2 Python programmers,
* 3 AI engineers,
* 1 Web Designer,
* 1 database administrator, and
* 1 systems engineer."

For each scenario, I used data from a different employee. In scenario 1, I will use the data provided by the lectures. In scenario 2, I will use the same data, but excluding the name "Juan", as Ciara and he are already included in the project.  
 I had a doubt that I couldn't resolve, so I will present two answers for each scenario. The doubt at hand was whether Ciara and Juan would cover some of the requested roles.

## Code

The code uses a CSP-based algorithm to try to "intelligently" find matches for hiring new employees based on their skills.

Definition of Candidates and Their Skills:  
candidatesScene1 and candidatesScene2 are dictionaries that represent employees and their skills for two different scenarios.

find\_solutions function:  
This function will be called when pressing the "Find Solutions" button.

Gets GUI input values for hiring requirements (maximum number of employees, Python programmers, AI engineers, etc.).

Defines a problem using the constraints library (python-constraint) to find optimal hiring combinations based on defined criteria.

Get all possible solutions to the problem.

Updates the GUI with the solutions found or displays a message if no solutions were found.

Graphical Interface (GUI):  
Creates a simple interface using the tkinter library with input fields for hiring requirements and a button to find solutions.  
Shows the results of the solutions found or a message indicating that no solutions were found.

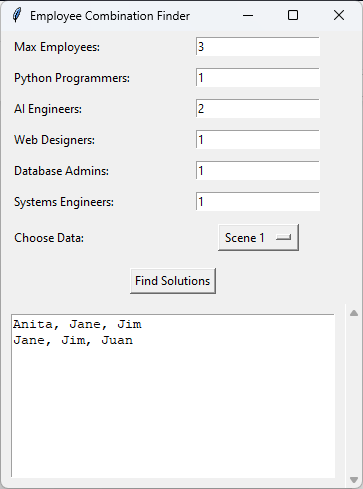
Additional Features:  
There is a menu to choose between the two datasets (Scene 1 and Scene 2).

This code creates an interface where you can enter hiring criteria and choose between two sets of candidate data. It then uses constraints to find possible hiring combinations based on the criteria you enter and displays the solutions found in the graphical interface.

## Scenario 1

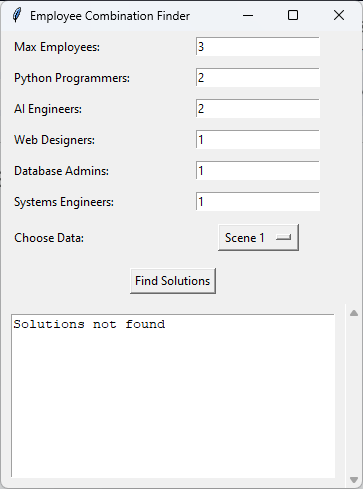
### *Case 1*:

In case 1, considering that Ciara knows Python and will fill one of the Python Programmers role and there is funds to hire 3 more employees, I considered removing one of the Python Programmers role, transforming the search into '1 Python Programmer, 2 AI Engineers, 1 Web Designer, 1 Database Admin and 1 Systems Engineer'.  
In this case, we can see, in the image below, that the app found 3 possible combinations of contracts, which meet Ciara's needs.



### *Case 2*:

In case 2, although Ciara knew Python, I considered that 2 more Python programmers would still be needed to complete the team. Considering the original search for '2 Python programmers, 2 AI engineers, 1 web designer, 1 database administrator and 1 systems engineer', we can see in the image below that this search cannot find a solution.



The algorithm may not find a solution in some scenarios, as everything will depend on the number of restrictions, variables or lack of employees available in the data.

In this case, 2 can be applied to the simple mathematics of why they can't find a solution: they have a total of 7 vacancies, considering that each new employee can occupy 2 positions, based on their skills and Ciara only has the budget for 3 more , this would give a total of 2x3 = 6 occupied roles, there would still be 1 vacancy to be filled, and this would end up making the solution impossible.

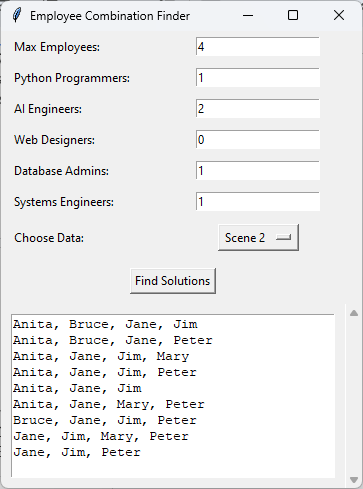
The algorithm may not find a solution in some scenarios, as everything will depend on the number of restrictions, variables or the lack of employees available in the data. In case 2, even simple mathematics can be applied to explain why a solution may not be found: we have a total of 7 places available. Considering that each new employee can occupy 2 positions, based on their skills, and that Ciara only has funding for 3 more, this would result in a total of 2x3 = 6 positions occupied. There would still be 1 vacancy to be filled, which would make the solution impossible.

## Scenario 2

### *Case 1*:

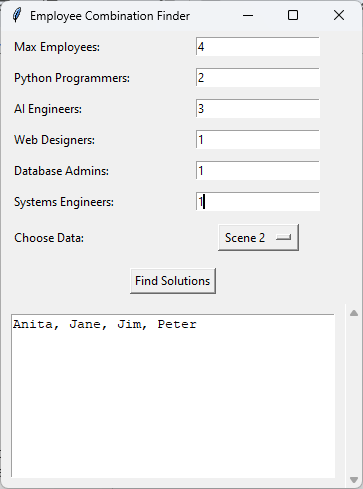
In case 1, considering that Ciara knows Python and will occupy one of the Python Programmer roles, and Juan knows A.I. and Web, he will occupy 2 vacancies: 1 AI Engineer and 1 Web Designer. With resources available to hire 4 more employees, I considered removing these 3 vacancies from the already filled list, transforming the search into '1 Python Programmer, 2 AI Engineers, 0 Web Designer, 1 Database Administrator and 1 Systems Engineer'.

In this case, we can see, in the image below, that the application found 9 possible combinations of contracts, which meet Ciara and Juan's needs.



### *Case 2*:

In case 2, although Ciara and Juan have knowledge in some technologies, I considered that, in this case, they only joined to obtain more funding and they still need to fill all the available vacancies, which are: '2 Python programmers, 3 AI engineers, 1 web designer, 1 database administrator and 1 systems engineer'. So we can see in the image below that the application found only one possible solution.



In this code, the Constraint class is used to implement restrictions to the employee allocation problem based on their skills. Constraint is used to import rules about possible solutions to the employee choice problem. In the code there are several restrictions being applied to ensure that hiring meets specific criteria, such as maximum number of employees and needed skills. During the search for solutions (solutions = problem.getSolutions()), the algorithm considers all the restrictions provided to try to find possible hires that meet the necessary criteria.

# Constraint Satisfaction vs Standard Algorithm

Constraint Satisfaction (CS) is an Artificial Intelligence approach that uses the assignment of values to a set of variables respecting certain constraints to deal with problems.  
In the employee allocation code, CS is used to solve the problem of choosing new employees for specific positions based on personal skills and restrictions such as maximum number of hires.

Variables and Domains:

The variables are the employees available for hiring. The domains are the possible value assignments for these variables (0 for not hiring, 1 for hiring).

Restrictions:

Constraints are the rules that value assignments must follow. In the code, restrictions are set to ensure that the maximum number of employees is not exceeded and that there is a minimum number of employees with specific skills.

Search for solutions:

The CS algorithm analyzes different value assignments to the variables (employees) while trying to satisfy all constraints.

Finding a solution:

If a combination of variables meets all constraints, a solution has been found. Otherwise, the program informs you that it is impossible to solve the problem.

Differences between Constraint Satisfaction and Standard Algorithms:

Algorithms like Dijkstra are used to find the shortest path in a weighted graph. Dijkstra's algorithm is excellent for finding the shortest path in a weighted graph. It is very efficient for problems such as finding the fastest route between two points on a map, calculating the shortest route in a transport network, among others. They (standard algorithmic) follow a determined approach, finding an optimal solution based on specific conditions in which they work with real values or floats, such as minimum distances.

Constraint Satisfaction:

CS deals with problems where it is necessary to find an assignment of values to variables that respects several constraints. CS does not necessarily seek an optimal solution, but rather an assignment of values that satisfies all restrictions. It is more flexible and can deal with a wider range of problems, from resource allocation to scheduling (consultation or meetings), as it uses a set of variables with restrictions.

In other words, CS differs from standard algorithms by solving complex problems that involve assigning values that meet several simultaneous constraints, as opposed to finding a specific path or an optimal solution in a specific scenario, as in the case of algorithms like Dijkstra.

# Constraint Satisfaction vs Depth-First Search

The Problem presented involving Ciara's hiring needs for her company can in fact be solved using several algorithms, in addition to CSP, including algorithms based on heuristics such as A\* or traditional graph algorithms such as Dijkstra's algorithm or search algorithms such as Depth - First Search (DFS) and Breadth First Search (BFS).

I will consider using a graph search algorithm, such as Depth-First Search (DFS), to find a solution for Scenario 1. We can represent the search problem where each state in the search space represents a possible combination of hired employees. The goal would be to find a match that meets Ciara's requirements.The main difference between code using Depth-First Search (DFS) and a solution based on Constraint Satisfaction Problem (CSP) is in the approach to solving the problem.

## DFS (Depth-First Search):

DFS is a search algorithm that exhaustively evaluates all possibilities. It uses a depth-first search approach, where it explores a search branch to the end before backtracking and exploring other branches. The DFS code tests all possible contracting combinations, cycling through all contracting options through recursion. It does not have a built-in mechanism to check and apply more complex restrictions or specific rules. There are no sophisticated heuristics or built-in constraint propagation; rather, it is a straightforward search for possible solutions.

## Constraint Satisfaction Problem (CSP):

CSP is problem solving that focuses on exploring variables, domains and constraints. It offers a structured way of representing relationships between variables and imposes explicit constraints. In a CSP, constraints (such as people skill limitations or hiring requirements) are defined and the resolution algorithm tries to find a solution that meets all of these constraints. Built-in constraint propagation mechanisms are used to reduce search time by eliminating futile values of variables, which can help reduce computational complexity. While DFS is a straightforward, exhaustive search approach that does not explicitly consider problem constraints, CSP is a more constructed framework that specifically exploits constraints and focuses on techniques to reduce search time, making it more efficient to solve. problems with complex constraints.

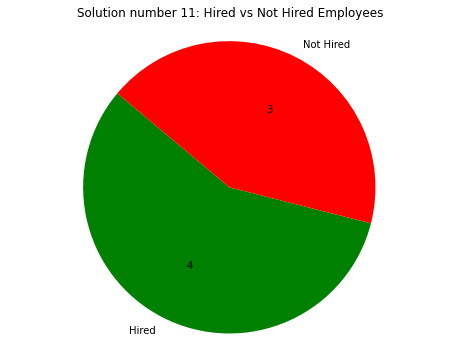
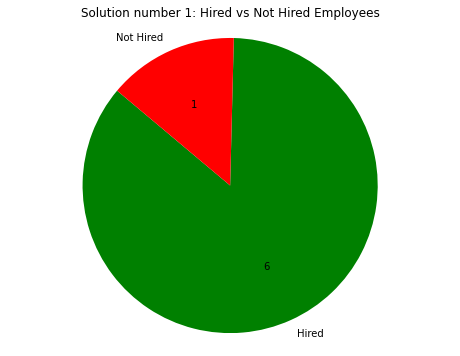
# 

# Constraint Satisfaction Problem Visualization

The implemented code uses graphical visualizations to represent the Constraint Satisfaction Problem (CSP) scenario and its solutions. I decided to use 3 different views:

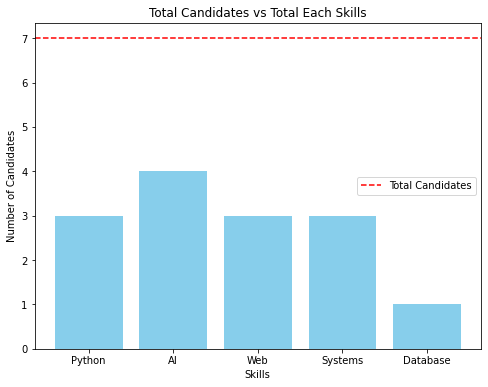
## Pie charts

Pie charts for hired and non-hired employees: these charts show the number of hired and non-hired employees for each solution found. Using colors ('green' for hired and 'red' for non-hired) provides an immediate understanding of hiring results. Pie charts effectively illustrate the outcome of each solution, especially on a large scale. The use of pie charts allows a clear visual comparison between the numbers of hired and non-hired employees.



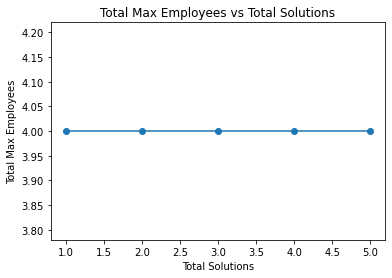
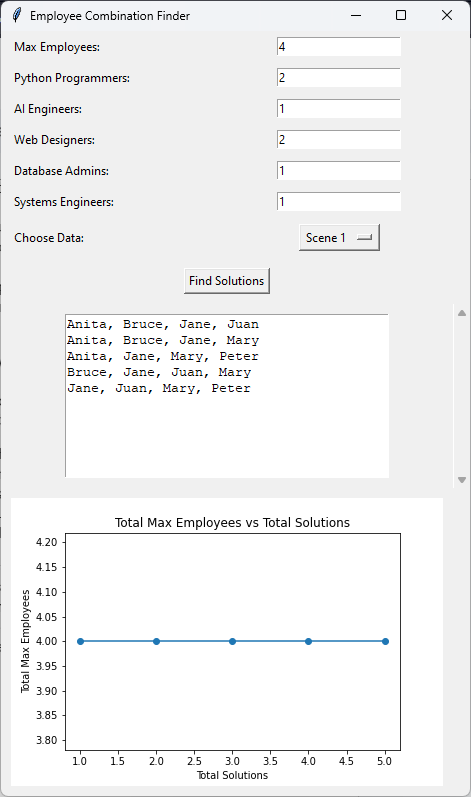
## Bar charts

Bar chart for skill distribution: This chart shows the total count of each specific skill across all candidates. By plotting the number of candidates against each skill, Ciara can visualize the availability of skill sets among potential employees. Bar charts are effective for showing the distribution of skill sets among candidates. This visualization method simplifies comparisons between different skills and also helps with a future increase in employees, as Ciara will have in mind a possible number of candidates for certain skills.

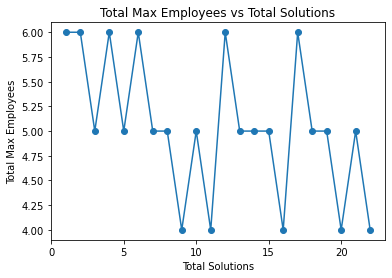
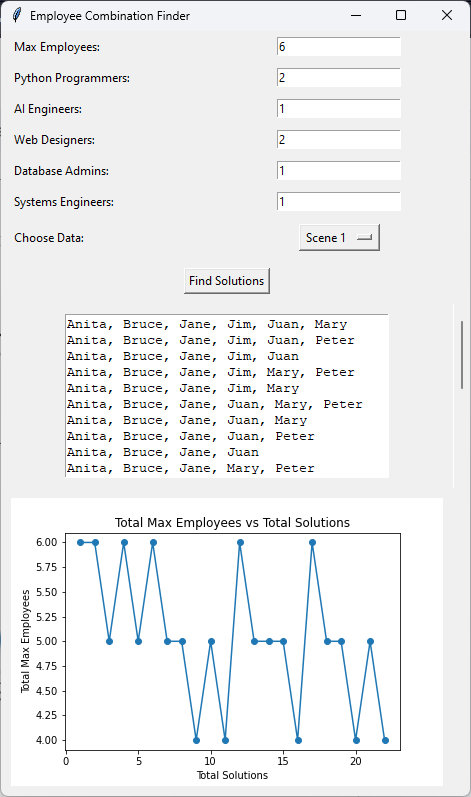
  
As we can see in the image, Ciara has 7 candidates, 4 of them have AI skills and only 1 has Database skills.

## Line charts

Line chart for evaluating solutions: This chart displays the relationship between the total number of solutions found and the maximum number of employees hired in each solution. Providing an overview of solutions in relation to the number of employees. The line chart provides a comprehensive view of how increasing the number of solutions affects the maximum number of employees hired. This visualization helps in deciding which team can be hired, as Ciara will receive information about min and max employees, we can make a decision to work with a smaller or larger team.



In the previous simulation, I used 4 employees as a maximum. We can clearly see in the line graph that 5 solutions were found, each with 4 employees. However, when you expand the limit to allow for more employees, the chart becomes more dynamic but still clearly shows the available options, as you will see in the image below.



# Github repository -> <https://github.com/CharlesCCT2020315/CA3>