**EXPERIMENT 11 - Constraint Satisfaction Problems**

**Aim:**

To solve constraint satisfaction problems.

**Theoretical Background:**

Constraint Satisfaction Problem (CSP) is a fundamental topic in artificial intelligence (AI) that deals with solving problems by identifying constraints and finding solutions that satisfy those constraints. CSP has a wide range of applications, including scheduling, resource allocation, and automated reasoning.

There are mainly three basic components in the constraint satisfaction problem: Variables: The things that need to be determined are variables. Variables in a CSP are the objects that must have values assigned to them in order to satisfy a particular set of constraints.

Domains: The range of potential values that a variable can have is represented by domains.

Constraints:The guidelines that control how variables relate to one another are known as constraints.

A CSP is defined as a triple (X,D,C), where:

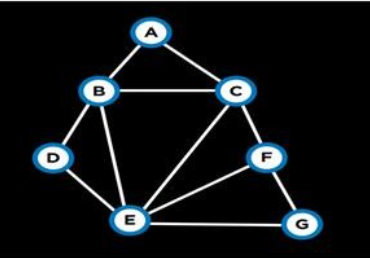
X is a set of variables { X1, X2, X3... Xn }.

D is a set of domains {D1, D2, ..., Dn}, where each Di is the set of possible values for Xi.

C is a set of constraints {C1, C2, ..., Cm}, where each Ci is a constraint that restricts the values that can be assigned to a subset of the variables.

Let us take an example, each student in a class is taking three courses from A, B, C, D, E, F and G. Each course needs to have a slot in a day, and the possible days are Monday, Tuesday, and Wednesday. However, the same student can't have two courses on the same day. In this case, the variables are the courses, the domain is the days, and the constraints are which courses can't be scheduled to have a session the same day because the same student is taking them.

This problem can be solved using constraints that are represented as a graph. Each node on the graph is a course, and an edge is drawn between two courses if they can't be scheduled on the same day. In this case, the graph will look this:



**Algorithm:**

1. The code starts by defining some variables.

2. The first variable, assignment, is a list of strings.

3. The second variable, VARIABLES, is a list of five strings.

4. Next, the code defines some constraints.

5. The first constraint is that the two lists have the same length (len(assignment) == len(VARIABLES)).

6. The next constraint is that each string in VARIABLES must be in one of the three domains (Monday, Tuesday, and Wednesday).

7. Finally, the last constraint is that each string in DOMAIN must be associated with at least one variable in VARIABLES.

8. The code then starts to search for an assignment that meets all the constraints.

9. First it checks if assignment is complete (if len (assignment) == len (VARIABLES)) and then it selects a variable from VARIABLES based on its domain and value.

10. If there are no more variables to select, then the code returns none.

11. Otherwise it backtracks through the assignment looking for a consistent combination of values for each selected variable.

12. If successful, it assigns those values to corresponding variables in result and returns this as solution.

13. The code first looks for a variable not yet assigned in the VARIABLES list.

14. If no variable is found, it returns none.

15. If an assignment is found, the code checks to see if the variables are consistent with each other.

16. If they are not, the code backtracks and tries again with a different variable.

17. If all variables are consistent with each other, then the code assigns the value of day to var3 and returns False.

**Conclusion:**

Python program to implement constraint satifaction problem is implemented successfully.