Constraint Satisfaction Problem of Classroom Assignment

Project Report

Meetika Sharma

mxs173530

Abstract

This project solves a Constraint Satisfaction Problem on classroom assignment in a school or university. Different subjects are the variable of CSP and value for the variable is the possible time and day of the week and room where that class can take place. The problem of classroom assignment given all the constraints is an NP-complete problem. This project solves this problem using two standard algorithms which are **Backtracking** and **Minimum Remaining Values** using **Least Constraint Value.**

Problem Description

A constraint satisfaction problem is solved when each variable has a value that satisfies all the constraints on the variable. A problem described this way is called a constraint satisfaction problem, or CSP.

Classroom assignment is a constraint satisfaction problem because it has variable which are different subjects and the values those variables can take are day and time during which the class can happen and room in which the class will happen satisfying all the constraints on the variables.

Constraints on courses are:

1) The timing of core courses should not clash with other courses. i.e. only one core course can happen at a specific time in a day.
2) Elective courses can be assigned the same day and time but different classrooms.

Previous Work

Artificial Intelligence - A Modern Approach third edition by Stuart J. Russell and Peter Norvig has been used to gain theoretical knowledge used in this project. Different constraint satisfaction problem such as N-Queen, 8 puzzle, map coloring have been studied to understand the problem and various approaches and heuristics to solve the classroom assignment problem.

I have read various papers to understand the tradeoff among different algorithms such as backtracking and minimum remaining values.

Formalism of Constraint Satisfaction Problem:

A constraint satisfaction problem consists of three components, X, D, and C: X is a set of variables, {X1, . . ., Xn}. D is a set of domains, {D1, . . ., Dn}, one for each variable. C is a set of constraints that specify allowable combinations of values. Each domain Di consists of a set of allowable values, {v1, . . ., vk} for variable Xi.

Approach

Two algorithms have been used to solve the constraint satisfaction problem of classroom assignment.

1. **Backtracking Search**: The term backtracking search is used for a depth-first search that chooses values for one variable at a time and backtracks when a variable has no legal values left to assign. It repeatedly chooses an unassigned variable, and then tries all values in the domain of that variable in turn, trying to find a solution. If an inconsistency is detected, then backtracking search returns failure, causing the previous call to try another value.
2. **Minimum remaining values**: It is a heuristic for choosing some ordering for the assignment of variables and it chooses a variable which has fewest legal values.
   - **Degree Heuristic**: It is used as a tie breaker for minimum remaining values heuristic. It selects the variable that is involved in the largest number of constraints on other unassigned variables.
   - **Least constraining value**: It is a heuristic that is used to order the assignment of a value to a variable after a variable has been chosen. It prefers the value that rules out the fewest choices for the neighboring variables in the constraint graph.

I have created separate functions to perform the above-mentioned algorithms.

Functions used in the program are as follows:

i) backtracking – This function tries all possible assignments to a variable until it finds a solution that satisfies all the constraints. If at any point of time it fails to find a solution, then it undoes its action and searches for a different assignment.

ii) degreeCalculate – It is a helper function to the degreeHeuristic function and is used to calculate the degree of a variable.

iii) degreeHeuristic – This function is called by minimumRemainingValues function and is used to find the variable with maximum degree in case there are multiple variable having equal degree to choose from for the next assignment.

iv) minimumRemainingValues – This function is used to pick the variable which has minimum number of remaining values to assign from. I am keeping a dictionary of variables and a list of their possible values. To choose the next variable, I select a list with minimum length.

v) leastConstrainingValue – This function is used to pick the next value to assign to a variable and it chooses a value which leaves maximum possible options for other variables.

Data and experiments

Input: Input is given in a CSV (Comma separated values) file.

Following values are given in input:

1. Subject name
2. If the subject is core or elective as both have different constraints
3. Day and time during which class can happen. These are possible values of the variables.

   e.g. AI, core, Monday4, Wednesday4

Output:

Output is given in a CSV file as well as printed on the console.

Output is of the form:

Subject name, day and time assigned, room assigned

e.g. NLP, Monday1, Room1

I am also printing the time taken by backtracking and Minimum Remaining Values to find a solution to the constraint satisfaction problem on the same input.

Input CSV file format:

	A	B	C	D
1	Algo	core	Monday1	Monday2
2	NLP	core	Monday1	
3	WPL	core	Wednesday2	
4	Database	elective	Monday1	Wednesday3
5	BigData	elective	Monday2	Wednesday3
6	AI	elective	Wednesday3	
7	Stats	core	Monday3	Monday5
8	ML	core	Wednesday4	
9	CV	elective	Wednesday3	
10	OOAD	elective	Monday5	
11	HCI	elective	Monday5	Thursday7
12	CV	elective	Thursday7	
13	Room1	Room2	Room3	Room4

Output of backtracking algorithm on the console:

```
(base) C:\Users\a702325>cd .spyder-py3

(base) C:\Users\a702325\.spyder-py3>python AIProject.py
Enter the path to input(CSV) file : input.csv
Enter the algorithm to use for 'Classroom assignment' CSP :
Enter 1 for backtracking
Enter 2 for Minimum Remaining Value
1

Time taken by backtracking is  0.0

Algo Monday2 Room1

NLP Monday1 Room1

WPL Wednesday2 Room1

Database Wednesday3 Room1

BigData Wednesday3 Room2

AI Wednesday3 Room3

Stats Monday3 Room1

ML Wednesday4 Room1

CV Wednesday3 Room4

OOAD Monday5 Room1

HCI Monday5 Room2

CV Thursday7 Room1
```

Output of backtracking algorithm in CSV format:

	A	B	C
1	Algo	Monday2	Room1
2	NLP	Monday1	Room1
3	WPL	Wednesday2	Room1
4	Database	Wednesday3	Room1
5	BigData	Wednesday3	Room2
6	AI	Wednesday3	Room3
7	Stats	Monday3	Room1
8	ML	Wednesday4	Room1
9	CV	Wednesday3	Room4
10	OOAD	Monday5	Room1
11	HCI	Monday5	Room2
12	CV	Thursday7	Room1

Output of Minimum Remaining Values algorithm on the console:

```
(base) C:\Users\a702325\.spyder-py3>python AIProject.py
Enter the path to input(CSV) file : input.csv
Enter the algorithm to use for 'Classroom assignment' CSP :
Enter 1 for backtracking
Enter 2 for Minimum Remaining Value
2

Time taken by Minimum Remaining Value is  0.0

NLP Monday1 Room1

Algo Monday2 Room1

WPL Wednesday2 Room1

ML Wednesday4 Room1

Stats Monday3 Room1

Database Wednesday3 Room1

BigData Wednesday3 Room2

AI Wednesday3 Room3

CV Thursday7 Room1

OOAD Monday5 Room1

HCI Monday5 Room2

(base) C:\Users\a702325\.spyder-py3>
```

Output of Minimum Remaining Values algorithm in CSV format:

	A	B	C
1	NLP	Monday1	Room1
2	Algo	Monday2	Room1
3	WPL	Wednesday2	Room1
4	ML	Wednesday4	Room1
5	Stats	Monday3	Room1
6	Database	Wednesday3	Room1
7	BigData	Wednesday3	Room2
8	AI	Wednesday3	Room3
9	CV	Thursday7	Room1
10	OOAD	Monday5	Room1
11	HCI	Monday5	Room2

Conclusion

From different experiments and readings, it can be concluded that search space for constraint satisfaction problem can be significantly reduced by using some heuristic instead of searching using backtracking which searches all the assignment possible for a variable. By using various heuristics to find the next variable to assign and to select the next value to try, we are trying to have the failure earlier in our assignment so that we do not have to do extra computation which is eventually going to lead to a failure in assignment.

References:

1. Mouelhi, Achref & Jégou, Philippe & Terrioux, Cyril & Zanuttini, Bruno. (2012). On the Efficiency of Backtracking Algorithms for Binary Constraint Satisfaction Problems. International Symposium on Artificial Intelligence and Mathematics, ISAIM 2012.
2. Artificial Intelligence – A modern approach, 3rd edition by Stuart J. Russell and Peter Norvig