Constraint Satisfaction Problems

a special subset of search problems

▪ We care about the goal itself, not the path

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu▪ CSPs are specialized for identification problems

States defined by variables

Factored representation

* Variables – value atayacak olduklarımız
* Domain (possible values)
* Constraints (specify combinations)

solution → a consistent assignment

▪ Unary constraints: constraint on a single variable, e.g., 𝑺𝑨 ≠ 𝒈𝒓𝒆𝒆𝒏

▪ Binary constraints: constraints between pairs of variables, e.g., 𝑺𝑨 ≠ 𝑾𝑨

▪ Global constraints (n-ary): involve 3 or more variables, e.g.,

Alldiff constraint specifies that all variables must have different values (e.g., of this kind of constraint involves, crypt-arithmetic puzzles, Sudoku)

▪ Preferences (soft constraints)

Solving CSPs

▪ BFS: Develop the complete tree ▪ DFS: time consuming

▪ **BTS: Backtracking search** is the basic uninformed search for CSPs.

It’s a DFS with these two improvements:

1. Assign one variable at a time. Variable assignments are commutative, so fix ordering

2. Check constraints on the go. consider values that do not conflict with previous assignments

CSPs = DFS with some backtracking search, which means that we're going

to step back whenever we make an assignment that violates the previous assignments

▪ Initial state: empty assignment {} - we did not assign any value to variables

▪ States: defined by the values assigned so far (partial assignments)

E.g., providing some variables with digits in the crypt-arithmetic puzzle

▪ Successor function: assign a value to an unassigned variable that not conflict with current

▪ Goal test: current assignment satisfies all constraints – complete and consistent

How can we improve BTS? -with heuristic

1. Which variable should be assigned next?

**Minimum Remaining Values (MRV)**

Choose the variable with the fewest legal values in its domain (seçeneği en az olan)

2. In what order should its values be tried?

**Least Constraining Values (LCV)**

▪ The one that rules out the fewest values in the remaining variables – likely to work

3. Can we detect inevitable/unavoidable failure early?

**Forward Checking (FC)**

Keep track of remaining legal values for unassigned variables (using immediate constraints)

Terminate when any variable has no legal values \*detects failure early

**Constraint Propagation**

▪ Forward checking propagates information from assigned to adjacent unassigned

variables, but doesn't check interaction between unassigned variables!