Hw2

CS23 108062119 鄭幃謙

1. The title, authors, and their affiliations of the paper

Title: Solution Reuse in Dynamic Constraint Satisfaction Problems
Author: Gkrard Verfaillie, Thomas Schiex
Affiliations: ONERA-CERT

2. The source of the paper

Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 – August 4, 1994, volume 1
(https://www.aaai.org/Papers/AAAI/1994/AAAI94-302.pdf)

3. Point out what concepts are different in contrast to the traditional CSP

Traditional CSP is static, the set of constraints is always fixed. So that each solution must satisfy all the constraints.
Dynamic CSP is not static, the set of constraints evolves according to the environment, the user, and other agents in the framework of a distributed system.
In other words, we can view dynamic CSP as a sequence of CSPs, where each one differs from the previous one by the addition or removal of some constraints.

4. What methods and heuristics are used in the solving such a new CSP in order to find a solution fast

The method used in this paper is an algorithm called Local Change. As below.

```
lc-variables(V1, V2, V3)
; V1 is a set of assigned and fixed variables
; V2 is a set of assigned and not fixed variables
; V3 is a set of unassigned variables
    if V3 = ∅
    then return success
    else let v be a variable chosen in V3
         let d be its domain
         if lc-variable(V1, V2, v, d) = failure
         then return failure
         else return lc-variables(V1, V2 ∪ {v}, V3 − {v})

lc-variable(V1, V2, v, d)
    if d = ∅
    then return failure
    else let val be a value chosen in d
         save-assignments(V2)
         assign-variable(v, val)
         if lc-value(V1, V2, v, val) = success
         then return succcss
         else unassign-variable(v)
              restore-assignments(V2)
              return lc-variable(V1, V2, v, d − {val})
lc-value(V1, V2, v, val)
    let be A1 = assignment(V1)
    let be A12 = assignment(V1 ∪ V2)
    if A1 ∪ {(v, val)} is inconsistent
    then return failure
    else if A12 ∪ {(v, val)} is consistent
         then return success
         else let V3 a non empty subset of V2 such that
                   let A123 = assignment(V1 ∪ V2 − V3)
                   A123 ∪ {(v, val)} is consistent
              unassign-variables(V3)
              return lc-variables(V1 ∪ {v}, V2 − V3, V3)
```

And there are two heuristics:

(1) choice of variable to be assigned, unassigned or reassigned

Choose the variable whose domain is the smallest, which is the variable with the most constraints. This reduces the possibility of backtracking.

(2) choice of the value for a variable

Choose the value which minimize the numbers of unsatisfied constraints to increase the choices of the other unassigned variables.