Standard Search Problem vs Constraint Search Problem

Similarities:

1. CSPs and SSPs can both represented as graph problems.
2. CSPs and SSPs can both be defined with search problem formulation/framework. They both have state, action, successor or next function to move to next state given result(action), and goal-test function.

Differences:

1. CSPs are concerned with the final assignment, order of assignment (path) does not matter. Thus, its solution always has same depth N. SSPs are concerned with the optimal path or optimal sequence of actions, each solution may have different depth (hence different cost).
2. States in CSPs are more flexibles because they are formed by variables x taking values from domain D. This implies that x value has been assigned.
3. Goal test of CSPs is in the form of constraints applied on the individual variables in the current state. SSPs goal test checks the if some goal state has been reached.

Cryptarithmetic Problem
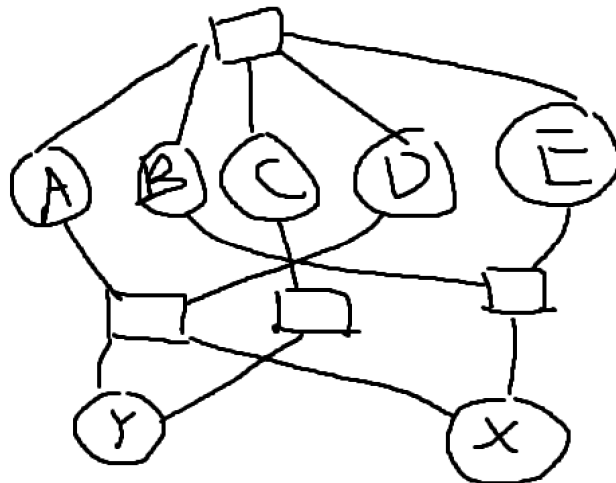
Variables: $\{A, B, C, D, E, X, Y\}$; Domains: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Constraints:

- $all\_diff(A, B, C, D, E)$
- $B + B = E + 10X$
- $A + A + X = D + 10Y$
- $C = Y$

Colouring Constraint Problem

{R, G, B}   {G, B}
V1 ———— V2

V3 ———— V4
{R, G}   {G}

Pure Backtracking

1. V1 R
2. V2 G
3. V3 R => inconsistent, clash with V1 R
4. V3 G => V4 has no value, backtrack
5. V2 B
6. V3 R => inconsistent, clash with V1 R
7. V3 G => V4 has no value, backtrack
8. V1 G
9. V2 G => inconsistent, clash with V1 G
10. V2 B
11. V3 R
12. V4 G => solution: V1 = G, V2 = B, V3 = R, V4 = G

Backtracking with Forward Checking

1. V1 R => V3 R is gone
2. V2 G => V1 G, V4 G are gone => V4 no value, backtrack
3. V2 B => V1 B is gone
4. V3 G => V4 G is gone => V4 no value, backtrack
5. V1 G => V2 G, V3 G are gone
6. V2 B => V1 B is gone
7. V3 R => V1 R is gone
8. V4 G => solution: V1 = G, V2 = B, V3 = R, V4 = G

Arc Consistency

1. V2 -> V4: D2=B, D4=G (Remove G from D2, V2 has changed)
2. V4 -> V2: D4=G, D2=B
3. V1 -> V2: D1=RG, D2=B (Remove B from V1, V1 has changed)
4. V2 -> V1: D2=B, D1=RG
5. V3 -> V1: D3=RG, D1=RG
6. V3 -> V4: D3=R, V4=G (Remove G from V3, V3 has changed)
7. V4 -> V3: V4=G, D3=R
8. V1 -> V3: D1=G, D3=R (Remove R from V1, V1 has changed)
9. V3 -> V1: D3=R, D1=G
10. V2 -> V1: D2=B, D1=G
11. Arc consistent. Got solution immediately because all nodes 1 assignment only: V1 = G, V2 = B, V3 = R, V4 = G. Checking order doesn't matter, so I just chose this V2 -> V4 first.