**Problem 3 Report**

● Gives a brief description of the strategy you used to solve the CSP

In my program, I used the idea of Variable Elimination and Variable Assignment Ordering. First, I remove the variable by replacing it with another function. With the new constraint, I can reduce the complexity of the search. After that, I used the idea of variable and Variable Assignment Ordering to solve the problem.

● Provides Pseudo Code of your CSP solver

This program finds solution to the following 3 hierarchically organized constraint satisfaction problems, involving 15 variables {A,B,C,…,N,O} which can take integer values in {1,…,50}.

# Compute the Problem A

set nva = 0

for e = 1 and e <= 28

Add value to nva

for f = 1 and f <= 28

Add value to nva

set d = e + f + 21

set a = (d \* d - 417) / (e \* e)

set mod = (d \* d - 417) % (e \* e)

if a > e + f; mod === 0; a and d >= 1 and <= 50

for b =1 and b <= 47

Add value to nva

set c = a - b - e - f;

if (c >= 1 && c <= 50) then

Get value to a, b, c, d, e, f]

End if

End for

End if

End for

End for

#Compute the Problem B

for idx = 0 and idx < inputs.length

set [a, b, c, d, e, f] = inputs[idx]

set g = f - sqrt(a + d + 1)

set f2 = pow(g - c, 2) - 1) / (c \* c)

if f2 === f

set j = (g \* g + 39) / 4

for i = 1 and i <= 50

Add value to nva

set h = pow(g + i, 2) - e \* 12) / j

set c2 = pow(i - g, 9) = pow(f - h, 3);

if (c2)

Get value to a, b, c, d, e, f, g, h, i, j

End if

End for

End if

End for

#Compute the Problem C

for idx = 0 and idx < inputs.length

set [a, b, c, d, e, f, g, h, i, j] = inputs[idx]

set o = sqrt(g \* h \* i \* b + 133)

for n = 1 and n <= 50

Add value to nva

set isGoodN = pow(n, 3) - 3 \* n \* o \* (n - o) - (f - i) \* n - pow(o, 3) + 7 = 0

if (isGoodN)

set m = sqrt(n \* n - 291)

set k = Math.sqrt(2 \* m + 6)

for l = 0 and l <= 50

Add value to nva

set isGoodL = pow(l, 3) - (l + b) \* k + i = 0;

if (isGoodL)

Get value to a, b, c, d, e, f, g, h, i, j, k, l, m, n, o

End if

End for

End if

End for

End for

# Print Function

if (results.length)

for idx = 0 and idx < results.length

set result = results[idx];

Print the value

End for

End if

else

Print "No solution exists"

End else

Print "nwa: " and nva

Set resultA = Problem A

Print the value of Problem A

Set resultB = Problem B

Print the value of Problem B

Set resultC = Problem C

Print the value of Problem C

● Explains the Pseudo Code in a paragraph

First, I start with problem A. With the for loops for e and f, I can determine a set of e and f. After that, I set the constraints of d and a. Moreover, I also set a mode equal to the divisible of a function, which is an integer. I set the if loops and another for loops to get the value of a, d, and b. After, I set the new constraints of c and if loops to get the value of c. So, we got the value of a, b, c, d, e, and f. Move to problem B, we will use the result in problem A to calculate. After transfer the result from function A to function B, I set the constraints to g, f, and j. After that, I set the for loops and if loops with some new constraints for the j, h, and I to calculate the value of g, h, I, j. Similar to Problem A and Problem B, I transfer the value of Problem B to calculate in Problem C. In the same way, I set the constraints for o, m, k and use if and for loops to get the value of k, l, m, n, o. After that, I print the result and complete the program, which is satisfied with all constraints.

● Describes strategies (if you employed any) you employed to reduce the runtime of your program, measured by the final value of the variable nva.

The strategies I employed to reduce the runtime of my program are used the idea of Variable Elimination. I conducted a mathematical pre-analysis to eliminate variables. Therefore, I can reduce the number of loops and search complexity. Before I used Variable Elimination, my nva value is high, which is more than 1,000,000. By using the Variable Elimination, I can reduce the nva value to 1057, which is a big difference.

● If you conducted a mathematical pre-analysis to eliminate variables, to obtain additional ‘<’ constraints to reduce search complexity or came up with other problem complexity reduction strategies based on such a pre-analysis, describe the results of the pre-analysis you conducted, and how the results of this pre-analysis were used for reducing the search complexity.

We have E+F<A and the biggest value of A is 50. So, I can write e<=28 and f<=28

Because A is an integer value, the result of (d \* d - 417) / (e \* e) need to be an integer. So, I set the mode of the divisible of the function (d \* d - 417) / (e \* e)

I change the A+D=(F-G)\*\*21 to g = f - sqrt(a + d + 1) to calculate the value of G through the value of A and D from the Problem 1

After got the value of g, I change the 4\*J=G\*\*2+39 to j = (g \* g + 39) / 4 to calculate j through the value of g

Next, I change the H\*J+E\*12=(G+I)\*\*2 to h = (pow(g + i, 2) - e \* 12) / j to calculate the value of h through the value of g, I and e we got below

In the Problem C, I change 2\*M=K\*\*2 – 6 and N\*\*2=M\*\*2 + 291 to k = sqrt(2 \* m + 6) and m = sqrt(n \* n – 291) to simplify the constraints of the value m and k.

● If your program takes advantage of the hierarchical structure of the three CSP problems also explain how this was done.

Like I mentioned in the beginning, I used the idea of Variable Assignment Ordering to solve my program. By using constraints, for loops and if function, I can get a satisfactory.

● If the program you developed is generic in the sense that its code could be reused to solve constraint satisfactions which have a similar structure but different constraints, include a paragraph presenting evidence why your program has this property and what you did to make your program ‘generic’…

I think my program can be reused to solve constraint satisfactions that have a similar structure but different constraints. In my program, I also attach the “No solution exists” function, which is can use in other similar structures. I also believe that my program is visual and clear to change the constraints easily and solve the other problem.

**APPENDIX:**

**I write this Program on JavaScript**

**To run this file:**

**1. Download the file**

**2. Unzip file**

**3. Move your mouse to the blank space in the folder**

**4. Keep the SHIFT BUTTION and RIGHT CLICK ON THE MOUSE IN THE SAME TIME**

**5. Click on OPEN POWERSHELL WINDOWN HERE**

**6. Type node app.js**

**7. The result will be appeared**