CSCI5240 – Combinational Search and Optimization with Constrains
Fall 2015
Assignment 1

1155070976 ZHANG Wanying (Sophie)

1. Graph Coloring Problem

(a) *Variables:* V_1, V_2, V_3, V_4, V_5, V_6, V_7 *(the vertex of this graph)*

Domains: R, G, B

Constraints:

There is a set E in which each element indicates the edge between two vertexes. The definition of set E in this problem is:

$E=\{(V_1, V_2),(V_1, V_4),(V_1, V_5),(V_2, V_3),(V_2, V_7),(V_4, V_5), (V_4, V_6), (V_6, V_7)\}$

① *For each element* (V_i, V_j) *in E,*

$(V_i, V_j) \in \{(R,G),(R,B),(G,B),(G,R),(B,G),(B,R)\}$

② *One vertex has only one color.*

(b) *Variables: R, G, B (colors)*

Domains: V_1, V_2, V_3, V_4, V_5, V_6, V_7

Constraints:

There is a set E in which each element indicates the edge between two vertexes. The definition of set E in this problem is:

$E=\{(V_1, V_2),(V_1, V_4),(V_1, V_5),(V_2, V_3),(V_2, V_7),(V_4, V_5), (V_4, V_6), (V_6, V_7)\}$

Thus, the constraints can be defined as following,

·*For each element* (V_i, V_j) *in E,*

① *If* $R= V_i$, *then* $G= V_j$ *or* $B= V_j$; *if* $R= V_j$, *then* $G= V_i$ *or* $B= V_i$

② *If* $G= V_i$, *then* $R= V_j$ *or* $B= V_j$; *if* $G= V_j$, *then* $R= V_i$ *or* $B= V_i$

③ *If* $B= V_i$, *then* $G= V_j$ *or* $R= V_j$; *if* $B= V_j$, *then* $G= V_i$ *or* $R= V_i$

(c) ① *We can suppose one solution to the first model is:*

$V_1=R$, $V_2=G$, $V_3=R$, $V_4=G$, $V_5=B$, $V_6=R$, $V_7=B$

We put this solution into the constraints of the second model, e.g. for $e = (V_1, V_2)$, *we have* $V_1=R$, *thus, we can infer* $V_2=G$ *or* $V_2=B$. *We do this action for every element (edge) in set E.*

Finally, we will have $R=\{V_1, V_3, V_6\}$, $G= \{V_2, V_4\}$, $B= \{V_5, V_7\}$, *that's the solution to the second model.*

② *On the other hand, we can suppose one solution to the second model is:*

$R=\{V_1, V_3, V_6\}$, $G= \{V_2, V_4\}$, $B= \{V_5, V_7\}$.

Also we put this solution into the constraints of the first model, e.g. for $e = (V_1, V_2)$, *we have* $V_1=R$, *thus, we can infer* $(V_1, V_2)= (R,G)$ *or* $(V_1, V_2)= (R,B)$, *thus,* $V_2=G$ *or* $V_2=B$. *We do this action for every element (edge) in set E.*

Finally, we will have $V_1=R$, $V_2=G$, $V_3=R$, $V_4=G$, $V_5=B$, $V_6=R$, $V_7=B$,

that's the solution to the first model.

2. Killer Sudoku

CSP Model:

Please NOTE that in this model, all lines and columns begin with 1, however, in the code, all begin with 0.

Variables: $G_{i,j}$ *represents the value of a cell, which can be located in the* i_{th} *row and* j_{th} *column.*

Domains: [1..9]

Constraints:

·for $i \in [1..9]$, all_different($G_{i,1}, G_{i,2}, G_{i,3}, G_{i,4}, G_{i,5}, G_{i,6}, G_{i,7}, G_{i,8}, G_{i,9}$)

·for $j \in [1..9]$, all_different($G_{1,j}, G_{2,j}, G_{3,j}, G_{4,j}, G_{5,j}, G_{6,j}, G_{7,j}, G_{8,j}, G_{9,j}$)

·for $G_{i,j} \in \{ G_{2,2}, G_{2,5}, G_{2,8}, G_{5,2}, G_{5,5}, G_{5,8}, G_{8,2}, G_{8,5}, G_{8,8} \}$,

all_different($G_{i-1,j-1}, G_{i-1,j}, G_{i-1,j+1}, G_{i,j-1}, G_{i,j}, G_{i,j+1}, G_{i+1,j-1}, G_{i+1,j}, G_{i+1,j+1}$)

·$G_{1,1}+G_{1,2}+G_{2,2}=20$

·$G_{1,3}+G_{1,4}+G_{1,5}+G_{2,3}=19$

·$G_{1,6}+G_{1,7}+G_{1,8}=18$

·$G_{1,9}+G_{2,9}+G_{3,9}=12$

·$G_{2,1}+G_{3,1}+G_{4,1}=7$

·$G_{2,4}+ G_{3,2}+G_{3,3}+G_{3,4}=28$

·$G_{2,5}+G_{2,6}+G_{2,7}=9$

·$G_{2,8}+G_{3,7}+G_{3,8}=16$

·$G_{3,5}+G_{3,6}=10$

·$G_{4,2}+G_{5,1}+G_{5,2}=16$

·$G_{4,3}+G_{4,4}+G_{4,5}+G_{5,3}=23$

·$G_{4,4}+G_{5,4}+G_{5,5}+G_{5,6}+G_{6,4}=32$

·$G_{4,7}+G_{4,8}+G_{4,9}=12$

·$G_{5,7}+G_{6,5}+G_{6,6}+G_{6,7}=17$

·$G_{5,8}+G_{5,9}+G_{6,8}=17$

·$G_{6,1}+G_{6,2}+G_{6,3}=8$

·$G_{6,9}+G_{7,9}+G_{8,9}=23$

·$G_{7,1}+G_{8,1}+G_{9,1}=23$

·$G_{7,2}+G_{7,3}+G_{8,2}=16$

·$G_{7,4}+G_{7,5}=11$

·$G_{7,6}+G_{7,7}+G_{7,8}+G_{8,6}=15$

·$G_{8,3}+G_{8,4}+G_{8,5}=9$

·$G_{8,7}+G_{9,5}+G_{9,6}+G_{9,7}=26$

·$G_{8,8}+G_{9,8}+G_{9,9}=10$

·$G_{9,2}+G_{9,3}+G_{9,4}=8$

3. Imperfect Squared Square

Variables:

$x_1, x_2, .., x_{13}$, x_i: *the x-coordinate of the bottom left vertex* i_{th} *square;*

$y_1, y_2, .., y_{13}$, y_i: *the y-coordinate of the bottom left vertex* i_{th} *square;*

Domains: [0,22]

Constraints:

Notations: size[i] is the size of the i_{th} *small square*

① *Restrict position according to square size:*

For i∈[1..13], x_i *<=23-size[i],* y_i *<=23-size[i]*

② *Well Covered Constraints*

(1). In the coordinate system, for each line x=cx, cx∈[0..23), the sum of the size of small square must equal to the size of master square. This constraint can be written in pseudocode as following:

```
For cx∈[0..23 )
    For i∈[1..13]
            if( cx-size[i]+1<=x[i] && x[i]<= cx)      bx[i]=1;
            else bx[i]=0;
    For i∈[1..13]
        Size of master square=Σ(bx[i]*size[i]);
```

(2). In the coordinate system, for each line y=cy, cy∈[0..23), the sum of the size of small square must equal to the size of master square. This constraint can be written in pseudocode as following:

```
For cy∈[0..23 )
    For i∈[1..13]
            if( cy-size[i]+1<=y[i] && y[i]<= cy)      by[i]=1;
            else by[i]=0;
    For i∈[1..13]
        Size of master square=Σ(by[i]*size[i]);
```

③ *No Overlap Constraints*

for i∈[1..12] and j∈[i+1..13]

(x[i]+length[i]<=x[j]) or (x[j]+length[j]<=x[i]) or

(y[i]+length[i]<=y[j]) or (y[j]+length[j]<=y[i])