Programing Assignment #4

CSCE 625 - Artificial Intelligence
Spring 2015

Name: Xiaoshu Zhang

Email: kallen5208@gmail.com

1. Problem 1 (the 'jobs' puzzle):

1.1 Variables and domains

Choose jobs as variables, each has 4 values. People are the values.

Table 1 Variables

CHEF = 0	GUARD = 1	NURSE = 2	CLERK = 3
POLICE = 4	TEACHER = 5	ACTOR = 6	BOXER = 7

Table 2 Domains

ROBERTA = 0	THELMA = 1	STEVE = 2	PETE = 3

1.2 Assignment

Assignment is a one dimension array, int[] asm. An index is a job, asm[job] value is its assignment. Initialized with 4, represents NULL.

1.3 Domain

Domain is a two dimension array, int[][] domain. First index is job. domain[CHEF] = {1, 1, 0, 1, 3} represents "the job CHEF has 3 remaining values: ROBERTA, THELMA, PETE". Initialized with {1, 1, 1, 1, 4}.

1.4 Consistency function

(1). Node consistency: [BOXER, ROBERTA] represents "ROBERTA is not a BOXER".

Function:

```
/**
        Node consistency check.
        @param j The variable being checked.
        @param p The value can not choose.
        @domainNext The domain to be checked and modified.
        @return -1: if the modified domain is empty, 0: if not changed, 1: if changed.
*/
                    private int NC(int j, int p, int[][] domainNext).
```

Table 3 All the node consistency

[BOXER, ROBERTA]	[NURSE, PETE]	[TEACHER, PETE]
[POLICE, PETE]	[CHEF, ROBERTA]	[POLICE, ROBERTA]

(2). Arc consistency: [ACTOR, CHEF] represents "ACTOR and CHEF are not a same person".

Function:

```
/**
        Arc consistency check.
        @param j1 The variable being checked.
        @param j2 The pair variable.
        @domainNext The domain to be checked and modified.
        @return -1: if the modified domain is empty, 0: if not changed, 1: if changed.
*/
    private int AC(int j1, int j2, int[][] domainNext)
```

Table 4 All the arc consistency

[ACTOR, CHEF]	[CLERK, CHEF]	[NURSE, CHEF]	[POLICE, CHEF]

1.5 Solutions

Using MRV can reduce the number of states explored. Since this problem is small, MRV's influence is small.

Table 5 The 'jobs' puzzle solutions

	With MRV	Without MRV
Number of states explored	8	12
Roberta	nurse, clerk	teacher, actor
Thelma	police officer, teacher	chef, boxer
Steve	chef, boxer	nurse, police officer
Pete	guard, actor	guard, clerk

2. Problem 2 (the 'houses' puzzle):

1.1 Variables and domains

Choose house positions as variables. Color, nation, snack, drink, pet are attributes, each attribute has 5 values.

Table 6 Variables and domains

Variables	FIRST = 0	SECOND = 1	THIRD = 2	FOURTH = 3	FIFTH = 4
Attributes	COLOR = 0	NATION = 1	SNACK = 2	DRINK = 3	PET = 4
COLOR	RED = 0	GREEN = 1	IVORY = 2	YELLOW = 3	BLUE = 4

NATION	BRITAIN = 0	SPAIN = 1	NORWAY = 2	UKRAINE = 3	JAPAN = 4
SNACK	HERSHEYS = 0	KITKATS = 1	SMARTIES = 2	SNICKERS = 3	MILKWAYS = 4
DRINK	JUICE = 0	TEA = 1	COFFEE = 2	MILK = 3	WATER = 4
PET	DOG = 0	FOX = 1	SNAIL = 2	HORSE = 3	ZEBRA = 4

1.2 Assignment

Assignment is a two dimension array, int[][] asm. First dimension is a house position, second dimension is attribute. asm[pos][attribute] value is its assignment. Initialized with 5, represents NULL.

1.3 Domain

Domain is a three dimension array, int[][][] domain. First dimension is a house position, second dimension is attribute. domain[FIRST][COLOR] = {1, 1, 0, 1, 1, 4} represents "the first house has 4 remaining values: RED, GREEN, YELLOW, BLUE". Initialized with {1, 1, 1, 1, 1, 5}.

1.4 Consistency function

(1). Node consistency: $[FIRST Norwegian] represents "FIRST has Norwegian".

Function:

```
    /**
                Node constraints. Combine requirement. Assignment actually.
                @param pos House position.
                @param value The only value legal.
                @param domainNext The domain to be checked and directly modified.
                @return
         */

         private boolean NCCombine(int pos, int attribute, int value, int[][][] domainNext)
```

Table 7 All the node consistency

$[FIRST Norwegian]	$[THIRD Milk]

(2). Combine arc consistency: $[red Englishman] represents "RED and Englishman are together".

Function:

```
/**
        Arc consistency check. Combine requirement.
        @param pos The house position.
        @param attribute1 The attribute being checked.
        @param attribute2 The module attribute.
```

```
            @param value1 attribute1's value.
            @param value2 attribute2's value.
            @param domainNext The domain to be checked and directly modified.
            @return -1: if attribute1's domain is empty, 0: if not changed, 1: if changed.
        */
```

private int ACCombine(int pos, int attribute1, int attribute2, int value1, int value2, int[][][] domainNext)

Table 8 All the combine arc consistency

$[red Englishman]	$[yellow KitsKats]	$[green Coffee]	$[Spaniard dog]
$[Ukranian tea]	$[Japanese MilkyWays]	$[Smarties snails]	$[Snickers Juice]

(3). Exclude arc consistency: $[Hersheys !fox] represents "Hersheys and fox are not together".

Function:

```
/**
            Arc consistency check. Exclude requirement.
            @param pos The house position.
            @param attribute1 The attribute being checked.
            @param attribute2 The module attribute.
            @param value1 attribute1's value.
            @param value2 attribute2's value.
            @param domainNext The domain to be checked and directly modified.
            @return -1: if attribute1's domain is empty, 0: if not changed, 1: if changed.
*/
```

private int ACExclude(int pos, int attribute1, int attribute2, int value1, int value2, int[][][] domainNext)

Table 9 All the exclude arc consistency

$[Hersheys !fox]	$[Norwegian !blue]	$[KitKats !horse]

(4). Global consistency: $[ivory => green] represents "ivory is in the left of green", $[Hersheys <=> fox] represents "Hersheys and fox are neighbors".

Function:

```
/**
            Do global constraint check when a variable is assigned with a value.
            @param pos
            @param attribute
            @param value
            @param domainNext The domain to be checked and directly modified.
            @return Modified domain is not empty return true, empty return false.
*/

    private boolean GCCheck(int pos, int attribute, int[][][] domainNext)
```

Table 10 All the global consistency

$[ivory => green]	$[Hersheys <=> fox]	$[Norwegian <=> blue]	$[KitKats <=> horse]

1.5 Solutions

Using MRV can significantly reduce the number of states explored. Since this problem is bid, MRV's influence is big.

Table 11 The 'houses' puzzle solutions

	With MRV	Without MRV
Number of states explored	1151	29926
first	yellow, Norwegian, Kit Kats, water, fox	yellow, Norwegian, Kit Kats, water, fox
second	blue, Ukranian, Hershey bars, tea, horse	blue, Ukranian, Hershey bars, tea, horse
third	red, Englishman, Smarties, milk, snails	red, Englishman, Smarties, milk, snails
fourth	ivory, Spaniard, Snickers, orange juice, dog	ivory, Spaniard, Snickers, orange juice, dog
fifth	green, Japanese, Milky Ways, coffee, zebra	green, Japanese, Milky Ways, coffee, zebra