Programmation par contraintes
BADGE F
La tomographie discrète

Seyed Amir JOKAR NARAGHI

18 novembre 2020

Objectif

Implémenter un programme permettant de reconstituer une image à partir de projection selon 4 directions.

Implémentation

1. Création des variables et modele

```java
public class ProjF1 {
    Run | Debug
    public static void main(String[] args) {
        // dimention de image
        int N = 10;
        // creation du modele
        Model model = new Model("reconstituer_une_image:" +N+ "." +N);

        // les valeurs associées à chaques lignes, colonnes, diagonales
        int[] lig = {5,4,4,4,3,3,4,4,4,4};
        int[] col = {2,2,7,7,3,2,3,4,4,5};
        int[] dUp = {0,1,1,2,1,3,2,3,3,7,4,3,1,2,1,2,2,0,1};
        int[] dDwn = {0,0,1,2,2,2,5,2,4,4,4,3,2,1,1,0,3,2,1};

        // initialisation du tableau
        // creation des variables
        IntVar[][] x = new IntVar[N][N];
        for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                x[i][j] = model.intVar("X"+i+j, 0, 1);
            }
        }
```

2. Implémentation des contraintes

2.1 Contraintes pour les lignes

```
// Une contrainte est associée à chaque ligne
// on tester les valeurs de ligne i de tableau x
// avec notre initial données
for(int i = 0; i < N; i++) {
    model.sum(x[i], "=", lig[i]).post();
}
```

2.2 Contraintes pour les colonnes

```
// Une contrainte est associée à chaque colonne
for(int j = 0; j < N; j++) {
    IntVar[] colo = new IntVar[N];
    for(int i = 0; i < N; i++) {
        colo[i] = x[i][j];
    }
    // on mettre les valeurs de colonnes dans notre tableau
    model.sum(colo, "=", col[j]).post();
}
```

2. 3 Contraintes pour les diagonales

```
// diagonales montantes
ArrayList<IntVar>[] dUpp = new ArrayList[2 * N-1];

for(int i = 0; i < 2 * N-1; i++) {
    dUpp[i] = new ArrayList<IntVar>();
}
// remplir de liste de diagonale montantes
for(int i = 0; i < N; i++) {
    for(int j = 0; j < N; j++) {
        dUpp[i+j].add(x[i][j]);
    }
}
// comparer les valeurs avec notre valeurs de tableau
for(int k = 0; k<2*N - 1; k++) {
    IntVar[] tmpM = dUpp[k].toArray(new IntVar[0]);
        model.sum(tmpM, "=", dUp[k]).post();
}
```

```
// diagonales descendantes
// crée le liste de diagonal descendanetes
ArrayList<IntVar>[] dDwnn = new ArrayList[2 * N-1];
for(int i = 0; i < 2 * N-1; i++) {
    dDwnn[i] = new ArrayList<IntVar>();
}

for(int i = 0; i < N; i++) {
    for(int j = 0; j < N; j++) {
        dDwnn[i-j+N-1].add(x[i][j]);
    }
 }
for(int k = 1; k < 2*N; k++) {
    IntVar[] tmpD = dDwnn[k-1].toArray(new IntVar[0]);
        model.sum(tmpD, "=", dDwn[2*N-1-k]).post();
}
```

3. Résolution

```
// Résolution
Solution solution = model.getSolver().findSolution();
    if(solution != null) {
        System.out.println(solution.toString());
        for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                if(x[i][j].getValue() == 1){
                    System.out.print("[1]");
                }
                else{
                    System.out.print("[0]");
                }
            }
            System.out.print("");
            System.out.print("\n");

        }
    }
```

```
→  TP git:(master) ✗  cd "/Users/hafezjokar/Library/Mobile Documents/com~apple~CloudDocs/Hafez/UB BDIA/UE7 Outils de l\'IA/TP" ; /Library/Java/JavaVirtua
lMachines/jdk-15.0.1.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 @/var/folders/q0/rnd345c10
qn1t8rnprsg1q_r0000gn/T/cp_84w0bbo7xnc2xdijp6zxflru7.argfile ProjetF.ProjF1
cd: no such file or directory: /Users/hafezjokar/Library/Mobile Documents/com~apple~CloudDocs/Hafez/UB BDIA/UE7 Outils de l\'IA/TP
Solution: X00=0, X01=1, X02=0, X03=1, X04=0, X05=0, X06=0, X07=1, X08=1, X09=1, X10=0, X11=0, X12=1, X13=0, X14=0, X15=1, X16=0, X17=0, X18=1, X19=1, X20
=1, X21=0, X22=1, X23=0, X24=0, X25=0, X26=0, X27=1, X28=0, X29=1, X30=0, X31=0, X32=1, X33=1, X34=1, X35=0, X36=1, X37=0, X38=0, X39=0, X40=0, X41=1, X4
2=0, X43=1, X44=0, X45=0, X46=1, X47=0, X48=0, X49=0, X50=1, X51=0, X52=0, X53=1, X54=1, X55=0, X56=0, X57=0, X58=0, X59=0, X60=0, X61=0, X62=1, X63=1, X
64=0, X65=0, X66=0, X67=1, X68=1, X69=0, X70=0, X71=0, X72=1, X73=1, X74=1, X75=0, X76=0, X77=0, X78=0, X79=1, X80=0, X81=0, X82=1, X83=0, X84=0, X85=1,
X86=0, X87=1, X88=1, X89=0, X90=0, X91=0, X92=1, X93=1, X94=0, X95=0, X96=1, X97=0, X98=0, X99=1,
[0][1][0][1][0][0][0][1][1][1]
[0][0][1][0][0][1][0][0][1][1]
[1][0][1][0][0][0][0][1][0][1]
[0][0][1][1][1][0][1][0][0][0]
[0][1][0][1][0][0][1][0][0][0]
[1][0][0][1][1][0][0][0][0][0]
[0][0][1][1][0][0][0][1][1][0]
[0][0][1][1][1][0][0][0][0][1]
[0][0][1][0][0][1][0][1][1][0]
[0][0][1][1][0][0][1][0][0][1]
```

[0][1][0][1][0][0][0][1][1][1]
[0][0][1][0][0][1][0][0][1][1]
[1][0][1][0][0][0][0][1][0][1]
[0][0][1][1][1][0][1][0][0][0]
[0][1][0][1][0][0][1][0][0][0]
[1][0][0][1][1][0][0][0][0][0]
[0][0][1][1][0][0][0][1][1][0]
[0][0][1][1][1][0][0][0][0][1]
[0][0][1][0][0][1][0][1][1][0]
[0][0][1][1][0][0][1][0][0][1]

```
→  TP git:(master) ✗
```

Image exporte