by Patrick
November 7th, 2019

weeSeepy®: A comparison with choco

At the end of note 35 I wrote:

"There is one thing that begs to be done and that is to compare weeSeepy's performance against choco. The model in Listing 4 should be coded up in choco and a quick comparison made."

Listing 1 gives (again) the weeSeepy model for the (im)perfect squares problem and Listing 2 gives the same model coded up in choco, with the same variable ordering heuristic. That is, they are a similar as can be. When executed both implementations produce the same result, shown in Figure 1. The weeSeepy implementation ran in 58 milliseconds and took 76 nodes to find a first solution. The choco implementation took 82 milliseconds (resolution time) and 75 nodes (913.0 nodes per second).

This is an example of *small scale statistics*.

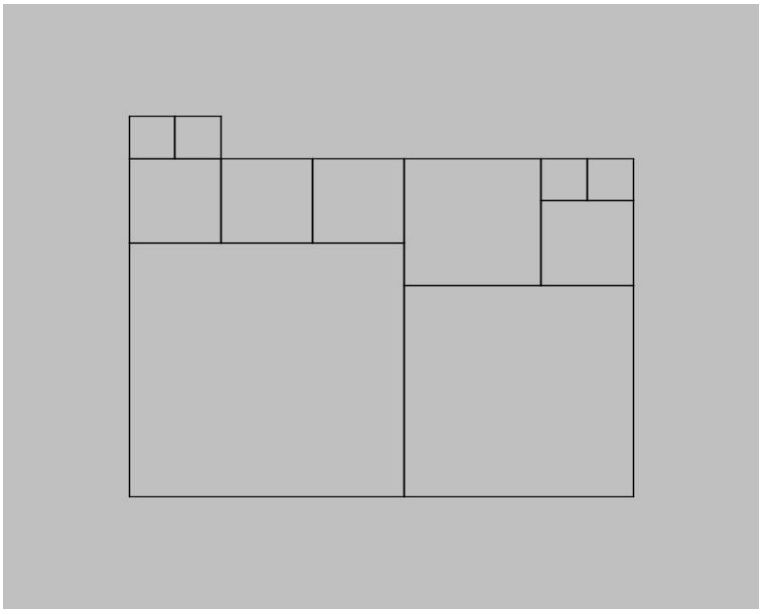

Figure 1: A packing of the squares in problem p01 into an 11 by 9 rectangle.

```
public class Squares {

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(new File(args[0]));
        sc.next(); // pallet
        int maxX = sc.nextInt();
        int maxY = sc.nextInt();
        sc.next(); // n
        int n = sc.nextInt();
        sc.next(); // sizes
        int[] size = new int[n];
        for (int i=0;i<n;i++) size[i] = sc.nextInt();
        sc.close();

        Problem pb = new Problem("Squares");

        IntVar[] x = new IntVar[n];
        IntVar[] y = new IntVar[n];

        for (int i=0;i<n;i++){
            x[i] = pb.intVar("x_" + i,0,maxX-size[i]);
            y[i] = pb.intVar("y_" + i,0,maxY-size[i]);
        }

        for (int i=0;i<n-1;i++)
            for (int j=i+1;j<n;j++){
                Constraint[] C = new Constraint[4];
                C[0] = new GreaterThanOrEqual(pb,x[j],x[i],size[i]); // i left of j
                C[1] = new GreaterThanOrEqual(pb,y[j],y[i],size[i]); // i below j
                C[2] = new GreaterThanOrEqual(pb,x[i],x[j],size[j]); // i right of j
                C[3] = new GreaterThanOrEqual(pb,y[i],y[j],size[j]); // i above j
                new Or(pb,C);
            }

        ArrayList<IntVar> decVars = new ArrayList<IntVar>();
        for (int i=0;i<n;i++){
            decVars.add(x[i]);
            decVars.add(y[i]);
        }
        pb.voh = new SDF(decVars); // variable ordering heuristic is smallest domain first

        boolean satisfied = pb.bt();

        if (satisfied){
            System.out.println("n "+ n +" maxX "+ maxX +" maxY "+ maxY);
            for (int i=0;i<n;i++)
                System.out.println(x[i].getValue() +" "+ y[i].getValue() +" "+ size[i]);
        }
        System.out.println("solved: "+ satisfied +" nodes: "+ pb.nodes
                           +"  fails: "+ pb.fails +"  cpuTime: "+ pb.cpuTime);
    }

}
```

Listing 1: (im)Perfect squares in weeSeepy®.

```java
public class ChocoSquares {

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(new File(args[0]));
        sc.next(); // pallet
        int maxX = sc.nextInt();
        int maxY = sc.nextInt();
        sc.next(); // n
        int n = sc.nextInt();
        sc.next(); // sizes
        int[] size = new int[n];
        for (int i=0;i<n;i++) size[i] = sc.nextInt();
        sc.close();

        Model model  = new Model("ChocoSquares");
        Solver solver = model.getSolver();

        IntVar[] x = new IntVar[n];
        IntVar[] y = new IntVar[n];

        for (int i=0;i<n;i++){
            x[i] = model.intVar("x_" + i,0,maxX-size[i]);
            y[i] = model.intVar("y_" + i,0,maxY-size[i]);
        }

        for (int i=0;i<n-1;i++)
            for (int j=i+1;j<n;j++){
                Constraint[] C = new Constraint[4];
                C[0] = model.arithm(x[j],">=",x[i],"+",size[i]);
                C[1] = model.arithm(y[j],">=",y[i],"+",size[i]);
                C[2] = model.arithm(x[i],">=",x[j],"+",size[j]);
                C[3] = model.arithm(y[i],">=",y[j],"+",size[j]);
                model.or(C).post();
            }

        IntVar[] decVars = new IntVar[2*n];
        for (int i=0,k=0;i<n;i++){
            decVars[k] = x[i];
            decVars[k+1] = y[i];
            k = k + 2;
        }

        solver.setSearch(Search.minDomLBSearch(decVars)); // fail-first

        boolean satisfied = solver.solve();

        if (satisfied){
            System.out.println("n "+ n +" maxX "+ maxX +" maxY "+ maxY);
            for (int i=0;i<n;i++)
                System.out.println(x[i].getValue() +" "+ y[i].getValue() +" "+ size[i]);
        }
        System.out.println(solver.getMeasures());

    }
}
```

Listing 2: (im)Perfect squares in choco.