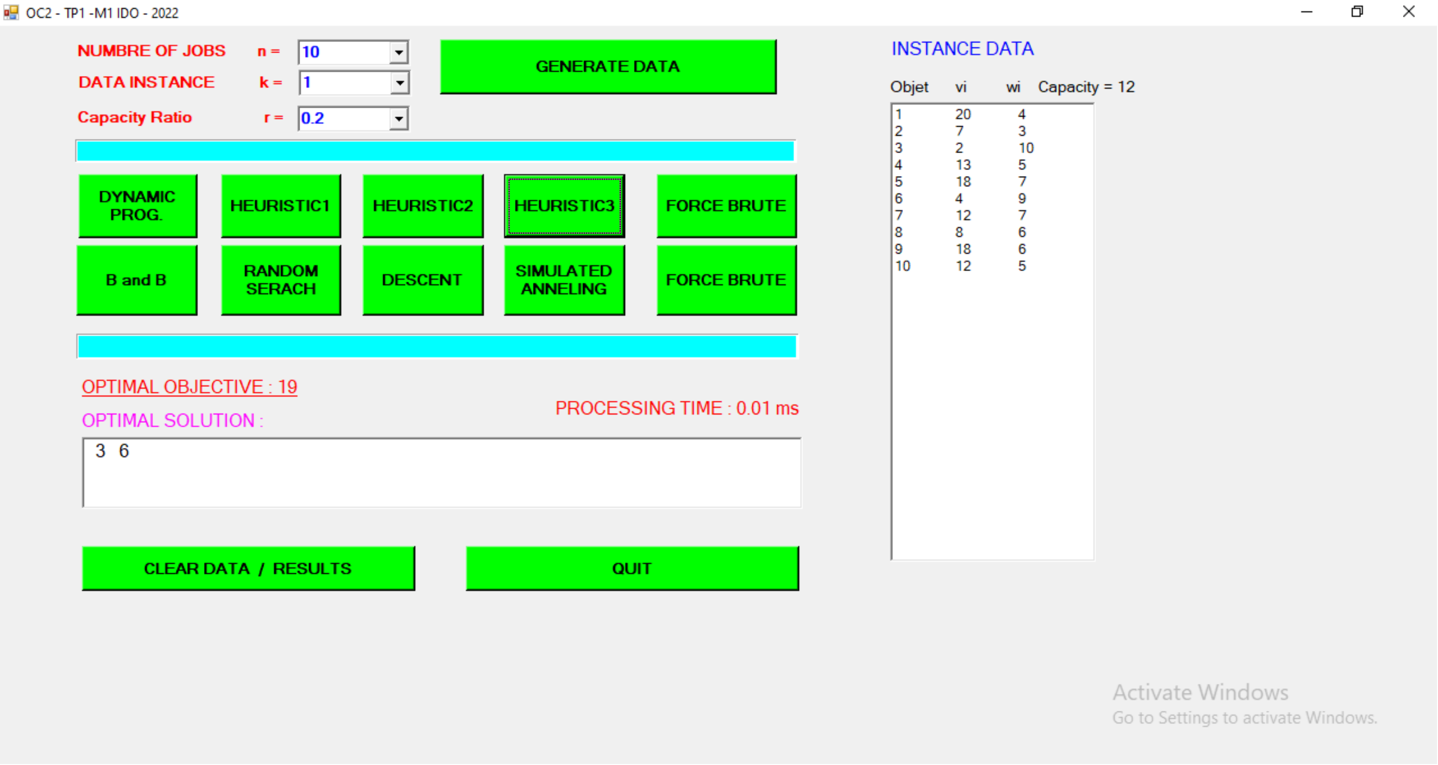
Mohamed Boudiaf university of M'sila
Faculty of MI
Department of informatics
Master 1 IDO
Combinatorial optimization 2 2022
Practice work: solving KSP using exact methods and local search

Create a new folder called PW CO2 2022, then, in NetBeans IDE, create a new project as java application called PWOC2. Rename the main class as KSP.

In order to prepare local search implementation and efficiency measure, we use the following program that allow solving 0-1 KSP by dynamic programming method.

```
public class KSP {
 // maximum of two integers
   static int max(int a, int b)   { return (a > b) ? a : b; }
// main
   public static void main(String args[])
   {
      // input
      int n = 5;
      int val[] = new int[] { 60, 100, 120, 50 , 250 };
      int wt[] = new int[] { 10, 20, 30, 41,65 };
      int sumwt = 0;
      for(int i=0; i<n;i++) sumwt+= wt[i];
      double ratio =0.5;
      int W = (int) ((int) sumwt*ratio);
   // variables
      int i, w;
      int K[][] = new int[n + 1][W + 1];
   // Building table K[][] in bottom up manner
      for (i = 0; i<= n; i++) {
         for (w = 0; w<= W; w++) {
            if (i == 0 || w == 0)
               K[i][w] = 0;
            else if (wt[i - 1]<= w)
               K[i][w] = max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
            else
               K[i][w] = K[i - 1][w];
         }
      }
      System.out.println("The optimum of this instance is : " + K[n][W]);
// look for selected items
      System.out.println("Selected Items : " );
      while (n != 0) {
          if (K[n][W] != K[n - 1][W]) {
                    System.out.println("\tItem " + n + " with Weight = " + wt[n - 1] + " and Value = " + val[n - 1]);
                    W = W - wt[n-1];
          }
                          n--;
     }
   }
}
```

1. **Recall the KSP statement and mathematical formulation.**
2. **Write the recursive formula of DP for solving KSP.**
3. **Type then run the above code.**
4. **Run this code using your own input with different instances.**
5. **Modify this code to enter random values and weights.**
6. **Add necessary code to make your results in text file called KSP_DP_RESULTS.TXT.**
7. **In order to solve this problem using local search methods seen in our course, design and complete the interface below :**



NUMBRE OF JOBS n = 10

DATA INSTANCE k = 1

Capacity Ratio r = 0.2

OPTIMAL OBJECTIVE : 19

OPTIMAL SOLUTION :

PROCESSING TIME : 0.01 ms

3 6

INSTANCE DATA

Capacity = 12

Objet	vi	wi
1	20	4
2	7	3
3	2	10
4	13	5
5	18	7
6	4	9
7	12	7
8	8	6
9	18	6
10	12	5