Algorithm Engineering – Exercise 1

Team 5: Jonas Köppeler, Julia Maria Lydia Henkel, Zia Badar

1. Implemented Features

We have used the same method explained in the lecture of converting a graph into cluster graph. Complexity of our program is $3^k * n^3$, k is the minimum cost to convert graph to cluster graph, n is the no of nodes in graph, 3^k are the search states possible, n^3 is the time for finding a p3.

Psuedo code of our program is:

Algorithm 1 Cluster graph

```
1: function SOLVE
2:     while BRANCH(k) != CLUSTER_GRAPH do
3:         k ← k+1
4:     end while
5: end function
//————————————
1: function BRANCH(k)
2:     if k < 0 then return NONE
3:     end if
4:
5:     u, v, w ← GET_P3
6:
7:     if BRANCH_EDGE(u, v, k) == CLUSTER_GRAPH
   then return CLUSTER_GRAPH
8:     end if
9:     if BRANCH_EDGE(v, w, k) == CLUSTER_GRAPH
   then return CLUSTER_GRAPH
10:    end if
11:    if BRANCH_EDGE(u, w, k) == CLUSTER_GRAPH
   then return CLUSTER_GRAPH
12:    end if
13:
14:    return NONE
15: end function
//————————————
1: function BRANCH_EDGE(u, v, k)
2:     if WEIGHT(u, v) == ALREADY_MODIFIED
   then return NONE
3:     end if
4:
5:     weight ← WEIGHT(u, v)
6:
7:     if WEIGHT(u, v) > 0 then
8:         DELETE_EDGE(u, v)
9:     end if
10:    if WEIGHT(u, v) < 0 then
11:        ADD_EDGE(u, v)
12:    end if
13:
14:    if BRANCH(k-abs(weight)) == CLUS-
   TER_GRAPH then return CLUSTER_GRAPH
15:    end if
16:
17:    WEIGHT(u, v) ← weight              ▷ backtracking
18:
19:    return NONE
20: end function
```

2. Data Structures

Graph is implemented as a adjancency matrix of size n^3, positive weights represents connection, negative weights represents no connection, DO_NOT_DELETE preprocessor directive is replaced with INT32_MAX and represents weights of edge that has been added and should not be removed to avoid cycles in search space, similarly DO_NOT_ADD is replaced with INT32_MIN and represents weight of edge that has been removed and should not be added.

3. Experiments

Time dependence on the value of n(vertices) and k(optimal cost) is actually not seen from the data acquired from test except for real world data which shows time dependence a little bit. Our algorithm time complexity is $O(3^k n^3)$

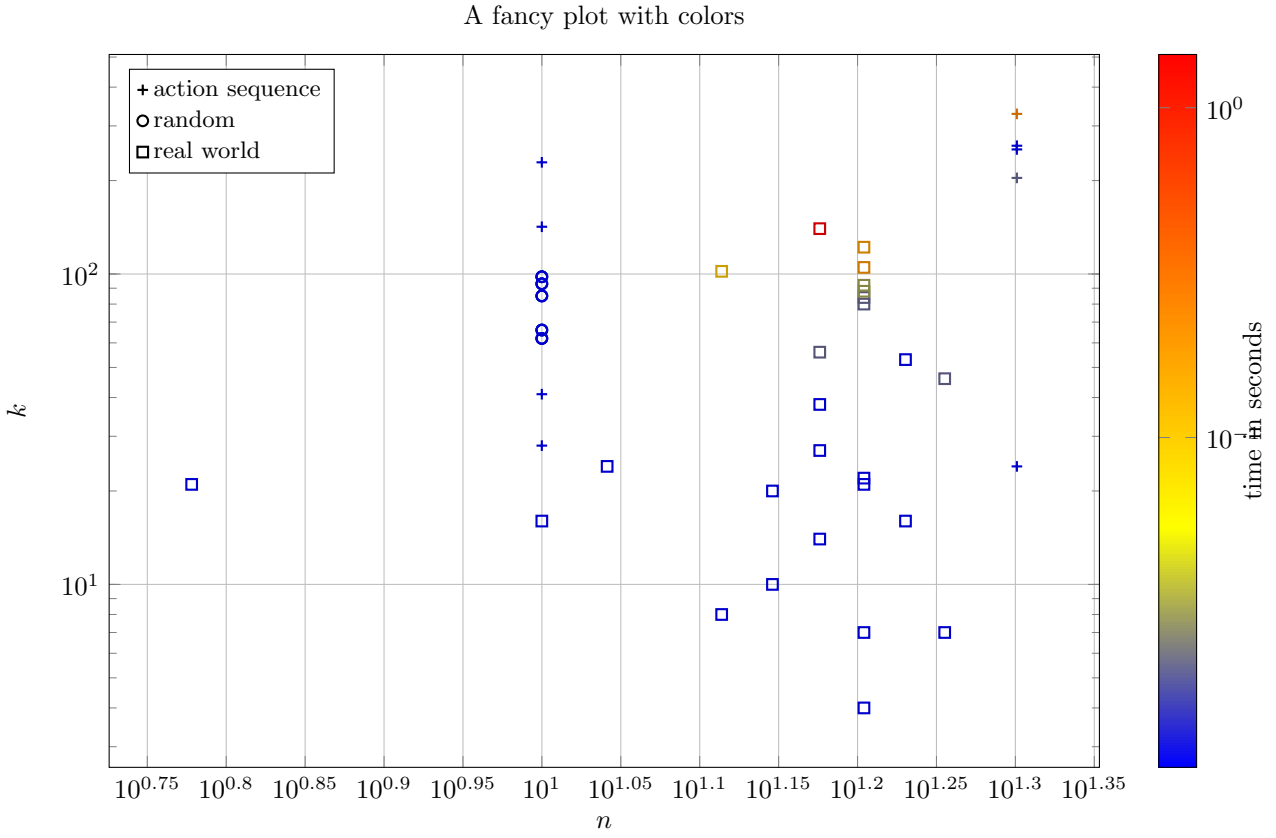

Figure 1: **a**lgorithm ran on different datasets, plus symbol representing action sequence data set, circle symbol representing random data set and square symbol representing real world data set.