Appendix A

Algorithm 1: Auxiliary function for vertex selection of the instance generator

```
1 function chooseVW(V, L, E)
2     Let sortedVertices be a queue of vertices sorted by their colored degree
3     Let candidates be the first 0.2 * N vertices of sortedVertices
4     for i ← 1 to 101 do
5         Let v, w be two random distinct vertices from candidates
6         if (v, w, l') ∉ E | ∀l' ∈ L then return (v, w)
7         if i = 100 then
8             i ← 1
9             candidates ← candidates ∪ { next vertex of sortedVertices, if any }
```

Algorithm 2: Auxiliary function for label selection of the instance generator

```
1 function chooseLabel(v, w, Klimit, V, L, E)
2     Let sortedLabels be a queue of labels sorted by the amount of edges associated with each of them
3     Let candidates be the first Klimit labels of sortedLabels
4     for i ← 1 to 1000 do
5         Let l be a random label from candidates
6         if (v, w', l) ∉ E | ∀w' ∈ L and (v', w, l) ∉ E | ∀v' ∈ L then return l
7         candidates ← candidates ∪ { next label of sortedLabels, if any }
8     return a random label from L
```

Algorithm 3: G4 Instances Generator

```
procedure G4Generator(V, L, d, initialSizeOfSolution)
    Let E ← ∅ be the set of edges of the graph
    Let LabelsOfSolution ← ∅ be the set of labels of the non-trivial solution
    Let leftPartitionSize be a random integer between 4 and |V|/4
    Let leftPartition be a set of size leftPartitionSize of randomly chosen vertices
    for i ← 1 to (d * |V| * (|V| − 1)/2) do
        Let v, w ← chooseVW(V, L, E)
        if v ∈ leftPartition XOR w ∈ leftPartition then
            if |LabelsOfSolution| = initialSizeOfSolution then
                Let l be a random label from LabelsOfSolution
            else
                Let l ← chooseLabel(v, w, |L|, V, L, E)
                LabelsOfSolution ← LabelsOfSolution ∪ {l}
        else Let l ← chooseLabel(v, w, 0.2 * |L|, V, L, E)
        E ← E ∪ {(v, w, l)}
    Let minTrivial ← min( Labeled degree of v | ∀v ∈ V) − sizeOfSolution
    if minTrivial > 1 then
        for i ← 1 to (random integer between minTrivial/2 and minTrivial − 1) do
            Let l_out be a random label such that l_out ∈ LabelsOfSolution
            Let l_in be a random label such that l_in ∈ L\LabelsOfSolution
            Let P ← {e = (v, w, l_out) | ∀v, w ∈ V, such that v ∈ leftPartition XOR w ∈ leftPartition}
            for i ← 1 to random integer between 1 and 2 do
                if |P| > 1 then
                    Let e' = (v', w', l_out) be a random edge from P
                    P ← P\{e'}
                    E ← E ∪ {(v', w', l_in)}\{e'}
                    LabelsOfSolution ← LabelsOfSolution ∪ {l_in}
    return E
```