CS 170 Project Algorithm

Smeet Patel

December 8, 2020

1 QIP

Algorithm 1: Quadratic Integer program for Graph Partitioning with Dynamic Constraints

```
Input: $i, j,\ h_{i,j},\ s_{i,j}\ \forall i, j,$ , R, $S_{max}$
Output: {room$_r$: [people $\in$ room$_r$]}
Initialization:
$e_{i,j_k} \leftarrow \{0, 1\},\ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
$v_{i_k} \leftarrow \{0, 1\},\ \forall i\ \ \&\ \ k = \{1, \dots K\}$
placements ← []
output ← {}
Objective:
max$\{e_{i,j_k} * h_{i,j}\},\ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
Constraints:
$\sum_{k=1}^{K} e_{i,j_k} = 1,\ \forall i, j$
$\sum_{(i,j)} e_{i,j_k} \leq \frac{S_{max}}{K},\ k = \{1, \dots K\}$
$e_{i,j_k} = v_i v_j,\ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
placements ← $split(v_i's, k),\ \ k = \{1, \dots K\}$
for k in range(K) do
    temp = [];
    for i in range(N) do
        if placement[i][k] >0 then
            temp = [i];
            output[k] = temp;
            return output
        end
    end
end
```

2 ILP

Algorithm 2: Integer Linear program for Graph Partitioning with Dynamic Constraints

```
Input: $i, j,\ h_{i,j},\ s_{i,j}\ \forall i, j,$ , K, $S_{max}$
Output: {room$_r$: [people $\in$ room$_r$]}
Initialization:
$e_{i,j_k} \leftarrow \{0, 1\},\ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
$v_{i_k} \leftarrow \{0, 1\},\ \forall i\ \ \&\ \ k = \{1, \dots K\}$
placements $\leftarrow$ []
output $\leftarrow$ {}
Objective:
max$\{e_{i,j_k} * h_{i,j}\},\ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
Constraints:
$\sum_{k=1}^{K} e_{i,j_k} = 1,\ \forall i, j$
$\sum_{(i,j)} e_{i,j_k} \leq \frac{S_{max}}{K},\ \ k = \{1, \dots K\}$
$e_{i,j_k} \leq v_{i_k}\ \ \forall i\ \ \&\ \ k = \{1, \dots K\}$
$e_{i,j_k} \leq v_{j_k}\ \ \forall i\ \ \&\ \ k = \{1, \dots K\}$
$e_{i,j_k} \geq v_{i_k} + v_{j_k} - 1\ \ \forall i, j\ \ \&\ \ k = \{1, \dots K\}$
placements $\leftarrow$ split($v'_i s, k$),  $k = \{1, \dots K\}$
for k in range(K) do
    temp = [];
    for i in range(N) do
        if placement[i][k] >0 then
            temp = [i];
            output[k] = temp;
            return output
        end
    end
end
```

The End.