

## 1 Code report

### (1) Class data member

#### Interval

```
private:
    mutable bool    ready;
    mutable bool    done;
    size_t          intervalID;
    size_t          trackNum;
    size_t          startTime;
    size_t          endTime;
    Intervallist    priorList;
    vector<size_t>  timeList;
```

#### Router

```
private:
    size_t          watermark;
    IdList          upper;
    IdList          bottom;
    Intervallist    intervallist;
    vector<Intervallist> trackList;
```

### (2) Constraint Left-edge algorithm

#### I Routing

```
void
Router::routing()
{
    vector<Interval*> list = intervallist;
    size_t trackNum = 0;
    while(!list.empty()){
        trackNum++;
        watermark = 0;
        Intervallist track;
        for(size_t i=0; i<list.size(); i++){
            list[i]->update();
            if( list[i]->isReady())
                if( watermark == 0 || (list[i]->getStart() > watermark) ){
                    list[i]->setTrack(trackNum);
                    list[i]->setDone();
                    watermark = list[i]->getEnd();
                    track.push_back(list[i]);
                    list.erase( list.begin()+i );
                    i--;    //keep searching start from the original next interval
                }
        }
        trackList.push_back(track);
    }
}
```

當 list 不為空時就會再開一個 track，並把 watermark 歸零，並進行 for 迴圈去找出可以放在該 track 裡的 Intervals，最後把 track 放到 trackList 裡。

在 for 迴圈裡，因為 intervalList 已經用 startTime sort 過(在 readNet 裡)，所以先找到的一定是離 watermark 最近的 Interval，每次比對 Interval，都要先將他 update，確保 Interval 是否是 ready(也就是上方的 Interval 都 routing 過)，且如果 startTime 比 watermark 晚，就把 trackNum assign 給這個 Interval，並把這個 Interval 標記為已經 routing，把這個 Interval 放入 track，並更新 watermark 成這個 Interval 的 endTime，把這個 Interval 從 list 刪除，把 i--，然後繼續往下找可以放進這個 track 的 Interval。

## II Update

```
void update() {
    ready = true;
    if(priorList.empty())
        return;
    for(size_t i=0;i<priorList.size();i++){
        if(!priorList[i]->isDone()){
            ready = false;
            break;
        }
    }
}
```

如果 priorList 裡的 Interval 都已經 routing 好，就把 ready 標記為 true，不然就是 false。

## 2 Results

Track1: i1  
Track2: i2  
Track3: i3

Channel Routing Graph:

```
0 2 1 0 2
*-----*
|_|_|_|_|
|_|_|_|_|
|_|_|_|_|
|_|_|_|_|
|_|_|_|_|
*-----*
1 0 0 3 3
```

Track1: i1 i6 i4  
Track2: i2  
Track3: i5  
Track4: i3

Channel Routing Graph:

```
1 1 1 2 2 5 6 3 0 4 0
*-----*
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
*-----*
2 5 0 5 5 3 3 0 6 0 4
```

Track1: i1  
Track2: i2 i3  
Track3: i5  
Track4: i4

Channel Routing Graph:

```
1 2 3 5 1 4 5
*-----*
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|
*-----*
2 5 4 0 3 0 4
```