Term Project – Quadratic Placement

20222912 이재승

1. To do.

Implement quadratic placement

2. Problem

Quadratic Placement Solver

Quadratic placement는 initial placement 방법 중 하나로, metrices 연산을 통해 connectivity 와 netweight를 고려할 수 있어 많이 사용된다. 본 과제에서는 quadratic placement equation Ax = b 의 A와 b행렬을 얻고, 주어진 matrixSolver.h를 이용하여 solution을 얻어 initial placement를 진행한다.

3. Implementation

Prerequisites

주어진 def 파일은 instance를 index로 호출하기 어렵기 때문에, STL인 unordered_map을 이용해서 hashMap을 구성한다.

```
// Give ID to instances
unordered_map<string, int> instMap;
int instCnt = 0;
for (auto &inst : instance_pointers_) {
  instMap.insert(make_pair(inst->getName(), instCnt++));
}
```

Matrix A

Pin의 정확한 위치를 기반으로 문제를 풀기위해 instance iteration을 돌면서 A 행렬을 구성한다.

Matrix A는 adjacent matrix C 와 밀접한 관련이 있기 때문에 pin과 pin사이의 연결을 구하기위해

inst-> pin -> net -> other pin의 접근방식을 이용했다.

```
for(auto &pin : inst->getPins()) {
  // cout << "In inst pin : "<<pin->getCoordinate().first << " : "<< pin->getCoordinate().second<<endl;
  if(pin->getNet()) {
    Net *net = pin->getNet();
    if (net->getSignalType() != "POWER" && net->getSignalType() != "GROUND" && net->getSignalType() != "CLOCK" && net->getSignalType() != "RESET") {
      int netWeight = net->getWeight();
      // For all connected pins
      for(auto &connectedPin : net->getConnectedPins()) {
```

Object function을 미분해서 나온 결과로 A를 구성한다.

- A의 대각 성분

Pin 마다 netweight를 합해주어 대각 값을 계산한다.

```
diagA += (double)netWeight;
```

- A의 연결 성분

```
if(inst1Num != inst2Num) {
  vector_row_idx.push_back(inst1Num);
  vector_col_idx.push_back(inst2Num);
  vector_data.push_back(-1.0 * netWeight);
```

netWeight 의 음수 값을 연결에 맞게 더해준다.

Matrix b

b 벡터

```
bx += netWeight * (connectedPin->getCoordinate().first - pin->getCoordinate().first);
by += netWeight * (connectedPin->getCoordinate().second - pin->getCoordinate().second);
```

Pin의 instance 상대 좌표를 고려하기 위해 위와 같은 식을 구성한다. 다만 block핀일 경우도 계산에 의해 같은 식을 가지게 된다.

데이터 변환 및 계산

사용한 vector를 다음과 같이 valarray로 형변환하고, solve() 메소드를 이용해서 핀의 위치를 구한다.

```
A.row = valarray<int>(vector_row_idx.data(), A.nnz);
A.col = valarray<int>(vector_col_idx.data(), A.nnz);
A.dat = valarray<double>(vector_data.data(), A.nnz);
```

```
// Solve Ax = b
A.solve(bx, x);
A.solve(by, y);
```

Instance placement

Instance의 setCoordinate() 메소드를 이용해, 계산된 위치에 cell을 배치한다.

```
for (auto &inst : instance_pointers_) {
  int inst1Num = instMap.find(inst->getName())->second;
  inst->setCoordinate(x[inst1Num], y[inst1Num]);
}
```