A Provably Good Approximation Algorithm for Rectangle Escape Problem with Application to PCB Routing

Qiang Ma[1], Hui Kong[1], Martin D. F. Wong[1] and Evangeline F. Y. Young[2]

[1]Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

[2]Dept. of Computer Science and Engineering
The Chinese University of Hong Kong

ASPDAC'11

Outline

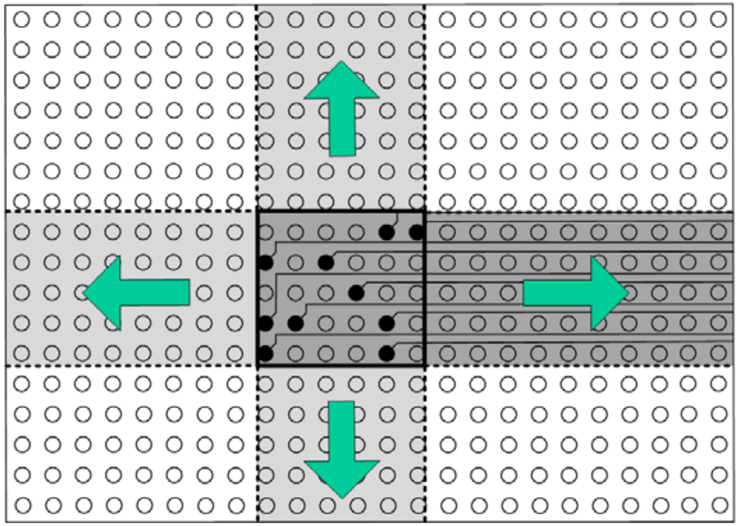
- Introduction
- Rectangle Escape Problem Formulation
- Proof of NP-Completeness
- An Approximation Algorithm
- Experimental Results
- Conclusions

Outline

- **Introduction**
- Rectangle Escape Problem Formulation
- Proof of NP-Completeness
- An Approximation Algorithm
- Experimental Results
- Conclusions

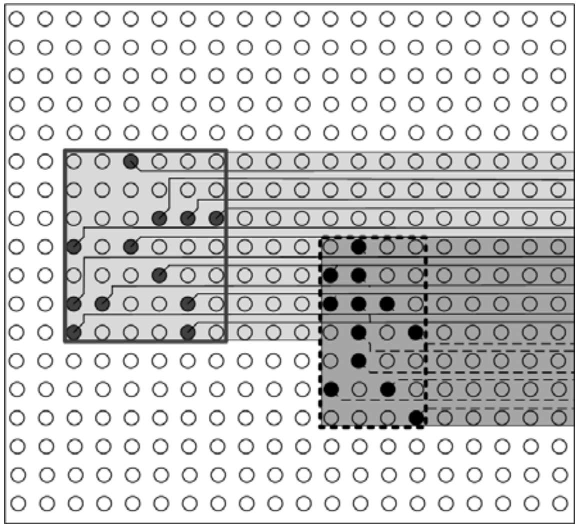
PCB Bus Escape Routing

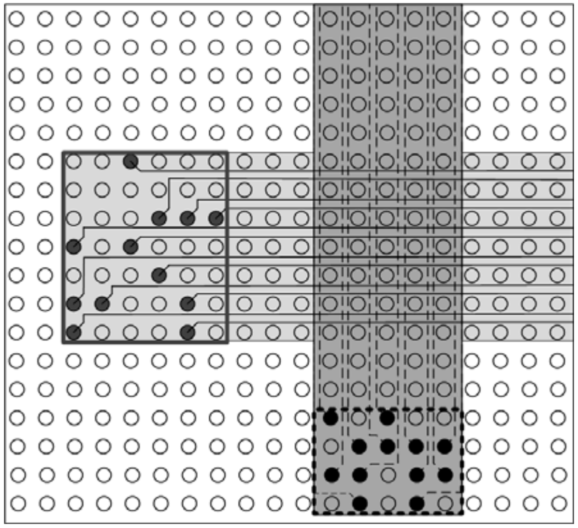
- In PCB bus escape routing, the nets of a bus are preferred to be routed together
- We observe that the escape routes of a bus are typically within one of its projection rectangles
  - Projection rectangles can be obtained by extending the bounding box of the pin cluster to the component boundaries



Conflict of the Buses

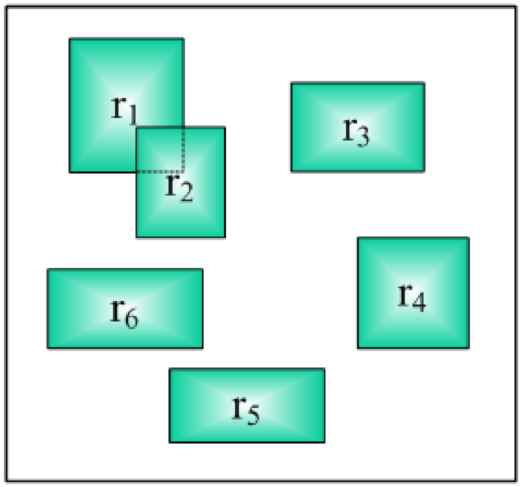
- When the escape routes of two buses conflict, they have to be routed on different layers.
- We want to use minimum number of layers
  - The fabrication cost increases with the number of layers
- According to our experience, the maximum density is usually a good indicator of the number of layers needed.



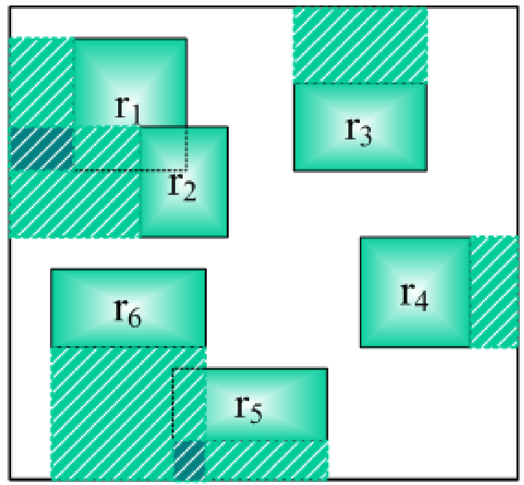


Density Minimization

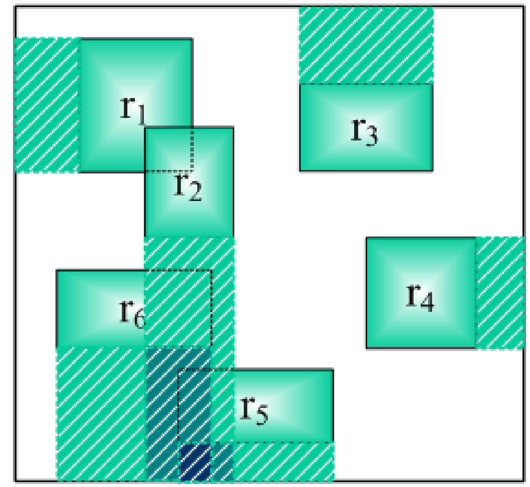
- We want to decide an escape direction for each bus, such that the maximum density is minimized.



An example



density = 2



density = 3

- Note that the actual layer assignment will have to be deferred until bus planning between various components is done so that a more global view of all bus intersections is available

Outline

- Introduction
- **Rectangle Escape Problem Formulation**
- Proof of NP-Completeness
- An Approximation Algorithm
- Experimental Results
- Conclusions

Rectangle Escape Problem (REP)

- An REP instance R contains
  - a rectangular region R
  - a set S of n rectangles $\{r_1, r_2, \ldots, r_n\}$.
- The objective is to
  - decide a escape direction for each rectangle
  - such that the maximum density d_{max} is minimized

RECTANGLE ESCAPE PROBLEM (REP)

INSTANCE: A rectangular region R and a set S of n rectangles $\{r_1, r_2, \ldots, r_n\}$ residing within R.

QUESTION: Each rectangle $r_i \in S$ chooses a direction to escape, such that d_{max} is minimized.

Outline

- Introduction
- Rectangle Escape Problem Formulation
- **Proof of NP-Completeness**
- An Approximation Algorithm
- Experimental Results
- Conclusions

REP is in NP

DECISION VERSION OF REP
INSTANCE: An integer k, a rectangular region R and a set S of n rectangles $\{r_1, r_2, \ldots, r_n\}$ residing within R.

QUESTION: Each rectangle $r_i \in S$ chooses a direction to escape, is there a solution such that $d_{max} \leq k$?

- *Lemma 1: REP is in NP.*

  Proof: Given an escape solution for the REP, where each rectangle *has its escape direction determined,* the maximum density d_{max} *can be checked* in *O(nlogn) time.*

REP is NP-hard

- We prove REP is NP-hard by reduction from 3SAT

3SAT
INSTANCE: A set U of n variables $\{x_1, x_2, \ldots, x_n\}$, a collection C of m clauses $\{c_1, c_2, \ldots, c_m\}$ over U such that $|c_i| = 3$, for $1 \leq i \leq m$.

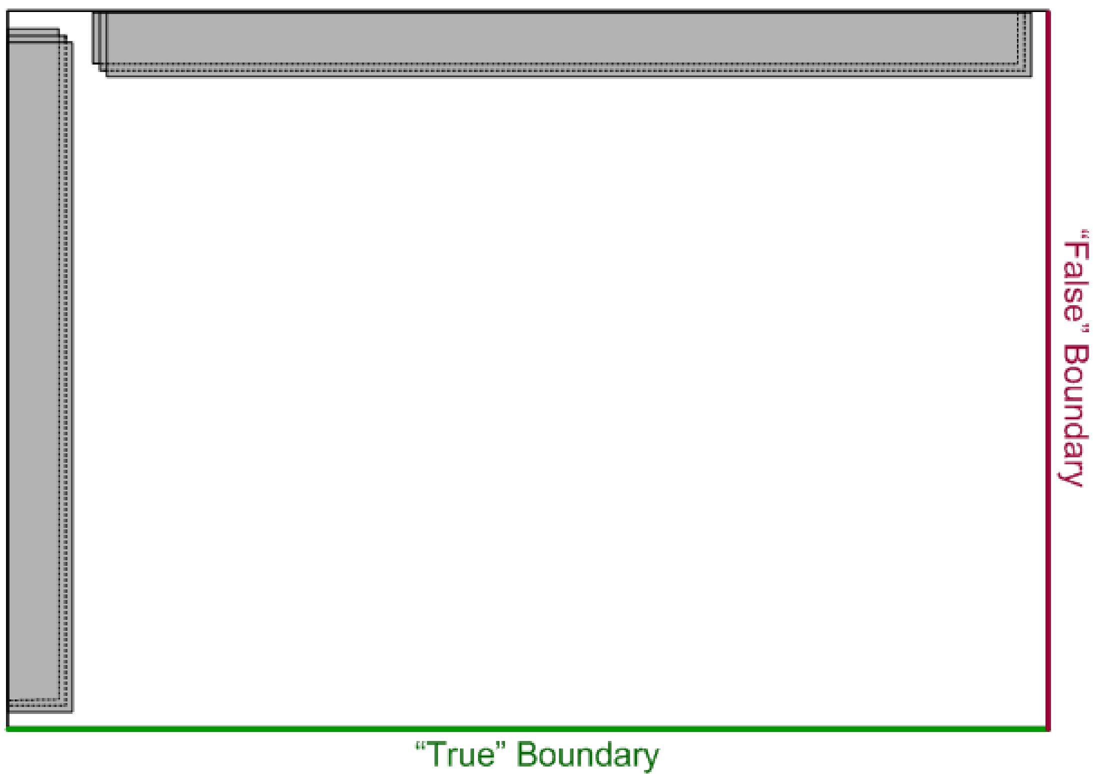
QUESTION: Is there a satisfying assignment for C?

DECISION VERSION OF REP
INSTANCE: An integer k, a rectangular region R and a set S of n rectangles $\{r_1, r_2, \ldots, r_n\}$ residing within R.

QUESTION: Each rectangle $r_i \in S$ chooses a direction to escape, is there a solution such that $d_{max} \leq k$?

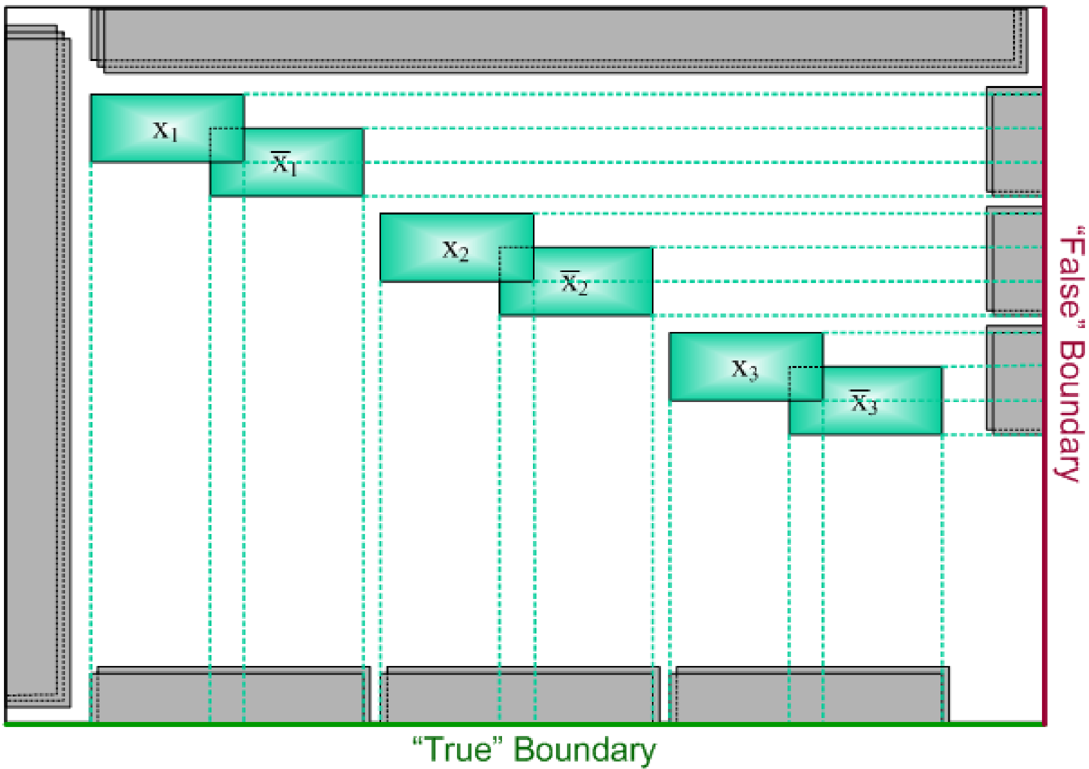
Reduction from 3SAT (1/3)

- Step 1: A rectangular region R
  - “Block” the top boundary and the left boundary of *R*
    - Three overlapping blockage rectangles on top boundary
    - Three overlapping blockage rectangles on left boundary
  - “True” boundary : bottom boundary
  - “False” boundary : right boundary



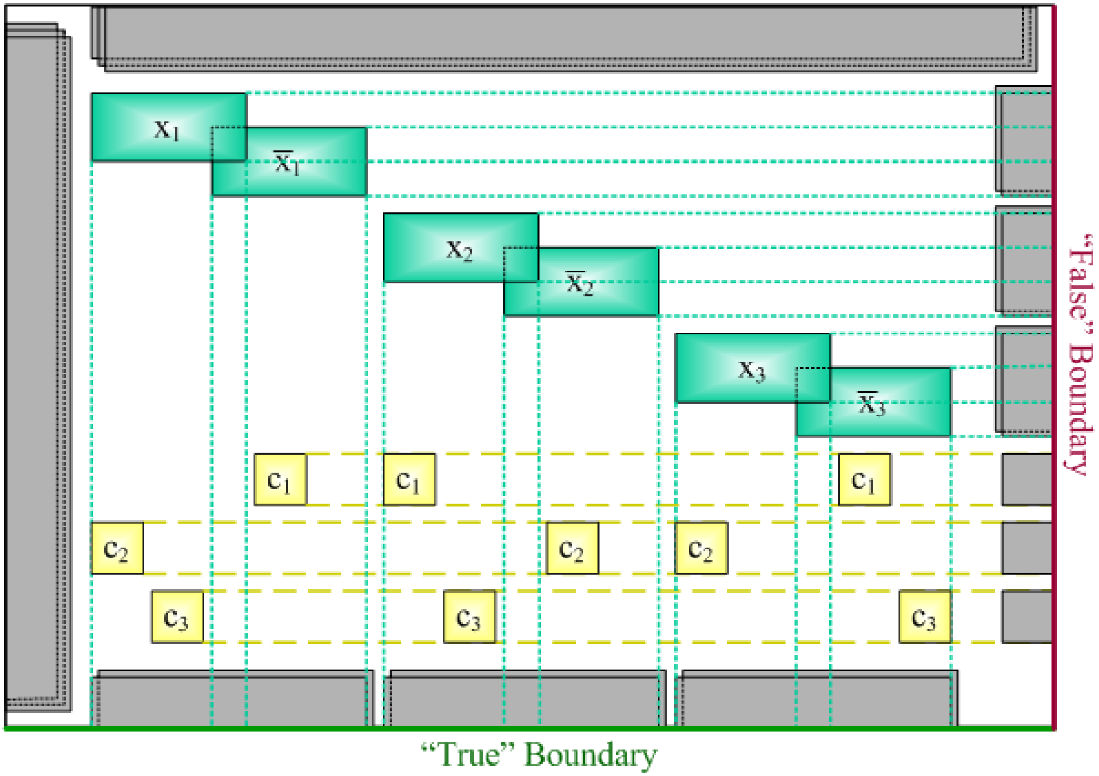
Reduction from 3SAT (2/3)

- Step 2: Variable rectangles
  - For each variable x_i
    - Create two overlapping variable rectangles
    - *Create two overlapping blockage rectangles on “True” boundary*
    - *Create two overlapping blockage rectangles on “False” boundary*
  - The rectangles for different variables cannot overlap



Reduction from 3SAT (3/3)

- Step 3: Clause rectangles
  - Create three clause rectangles for each clause c_i and place them under the three rectangles for the three variables c_i contains
  - Place a blockage rectangle for each clause c_i on "False" boundary
  - The clause rectangles cannot overlap with each other



Three clauses:

$$c_1 = (x_1 \vee \bar{x}_2 \vee x_3)$$

$$c_2 = (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$$

$$c_3 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$$

Proof of NP-completeness

- *Lemma 2: The 3SAT* instance has a satisfying truth assignment if and only if the constructed REP instance has an escaping solution with $d_{max} \leq 3$.

- *Theorem 1*: REP is NP-Complete.

Outline

- Introduction
- Rectangle Escape Problem Formulation
- Proof of NP-Completeness
- **An Approximation Algorithm**
- Experimental Results
- Conclusions

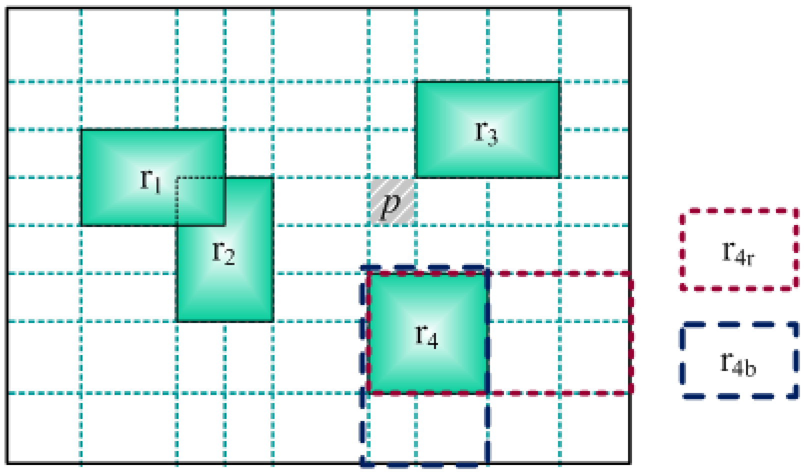
An ILP Formulation (1/2)

- Four 0-1 variables x_{il}, x_{it}, x_{ir} and x_{ib} for each rectangle r_i
  - Each rectangle chooses one direction to escape

$$x_{il} + x_{ir} + x_{it} + x_{ib} = 1, \quad \forall i = 1, 2, \ldots, n.$$

- Construct the Hanan Grid
  - Extending the boundaries of the rectangles
  - The whole region is partitioned into a set P of tiles
- Let r_{il}, r_{it}, r_{ir} and r_{ib} denote the four rectangular regions that r_i occupies after escaping to corresponding boundaries
  - The density in each tile p cannot

$$\sum_{i,*:\, r_{i*} \text{ occupies } p} x_{i*} \le d_{max}, \quad \forall p \in P,$$



An ILP Formulation (2/2)

- The objective is to minimize the density d_{max}, *so* the ILP for the REP can be formulated as follows, with *O(n)* variables and *O(n²)* constraints

$$
\begin{array}{ll}
\text{Minimize} & d_{max} \\
\text{Subject to} & \\
& x_{il} + x_{ir} + x_{it} + x_{ib} = 1, \qquad \forall i = 1, 2, \ldots, n. \\
& \displaystyle\sum_{i,*:\, r_{i*} \text{ occupies } p} x_{i*} \leq d_{max}, \qquad \forall p \in P. \\
& x_{il}, x_{ir}, x_{it}, x_{ib} \in \{0, 1\}, \qquad \forall i = 1, 2, \ldots, n.
\end{array}
$$

An 4-Approximation Algorithm (1/2)

- Solving ILP is also an NP-complete problem
- LP relaxation and Rounding
  - LP relaxation

$$\begin{aligned}
&\text{Minimize} \quad d_{max} \\
&\text{Subject to} \\
&x_{il} + x_{ir} + x_{it} + x_{ib} = 1, \qquad \forall i = 1, 2, \ldots, n. \\
&\sum_{i,*:\, r_{i*} \text{ occupies } p} x_{i*} \le d_{max}, \quad \forall p \in P. \\
&0 \le x_{il}, x_{ir}, x_{it}, x_{ib} \le 1, \qquad \forall i = 1, 2, \ldots, n.
\end{aligned}$$

  - Solve the LP using an LP solver
    - A fractional solution is obtained
  - Rounding
    - r_i's escape direction is set to the be Left (Right, Top, Bottom) if x_{il} (x_{it}, x_{ir}, x_{ib}) is the largest among the four direction variables

An 4-Approximation Algorithm (2/2)

- *Theorem 2: Given an REP* instance, where each rectangle has 4 candidate choices of escape directions, the LP relaxation and rounding method is a 4-approximation algorithm.

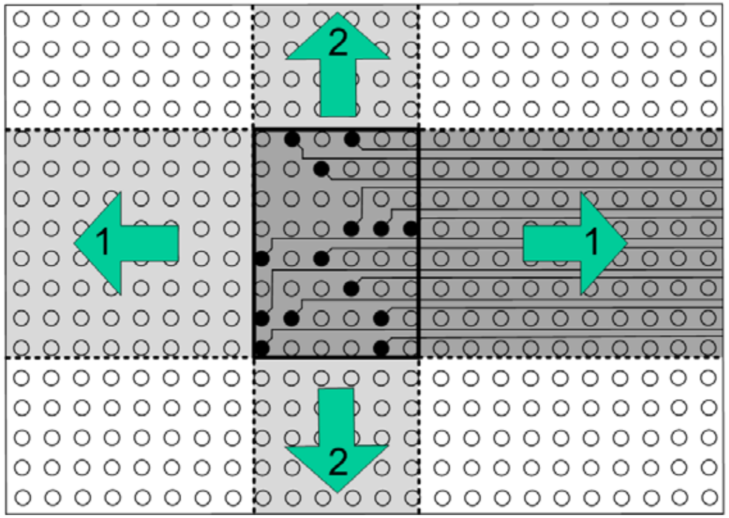
- *Corollary 1:* Given an *REP* instance, where each rectangle has α candidate choices of escape direction, the LP relaxation and rounding method is an α-approximation algorithm.

Weighted REP (1/2)

- Weighted REP is a more general problem
  - Each escape direction is associated with a weight

Weighted REP
INSTANCE: A rectangular region R and a set S of n rectangles $\{r_1, r_2, \ldots, r_n\}$ residing within R. Each rectangle r_i is associated with a weight vector $w_i = [w_{il}, w_{ir}, w_{it}, w_{ib}]$, for $1 \le i \le n$.

QUESTION: Each rectangle $r_i \in S$ chooses a direction to escape, such that d_{max} is minimized.



The bus may occupy different number of layers when escaping in different directions

Weighted REP (2/2)

- ILP

$$
\begin{aligned}
&\text{Minimize} && d_{max} \\
&\text{Subject to} && \\
& && x_{il} + x_{ir} + x_{it} + x_{ib} = 1, && \forall i = 1, 2, \ldots, n. \\
& && \sum_{i,*:\, r_{i*} \text{ occupies } p} w_{i*} \times x_{i*} \le d_{max}, && \forall p \in P. \\
& && x_{il}, x_{ir}, x_{it}, x_{ib} \in \{0, 1\}, && \forall i = 1, 2, \ldots, n.
\end{aligned}
$$

- LP relaxation and Rounding

$$
\begin{aligned}
&\text{Minimize} && d_{max} \\
&\text{Subject to} && \\
& && x_{il} + x_{ir} + x_{it} + x_{ib} = 1, && \forall i = 1, 2, \ldots, n. \\
& && \sum_{i,*:\, r_{i*} \text{ occupies } p} w_{i*} \times x_{i*} \le d_{max}, && \forall p \in P. \\
& && 0 \le x_{il}, x_{ir}, x_{it}, x_{ib} \le 1, && \forall i = 1, 2, \ldots, n.
\end{aligned}
$$

- Approximation ratio still holds

Iterative Refinement

- Iterative refinement is a post-processing procedure to further improve the result
  - In each iteration, we re-escape all the rectangles one by one
    - For each rectangle, pick the best one out of the four directions
  - The procedure is repeated until no improvement is achieved

```
ALGORITHM GREEDYREFINE(REP instance R):
  terminate ← false;
  while ! terminate do
      for each rectangle r_i ∈ S do
          Try all r_i's escape directions;
          Pick the best one;
      if there is no improvement in this iteration then
          terminate ← true;
  return
```

Outline

- Introduction
- Rectangle Escape Problem Formulation
- Proof of NP-Completeness
- An Approximation Algorithm
- **Experimental Results**
- Conclusions

Ex2

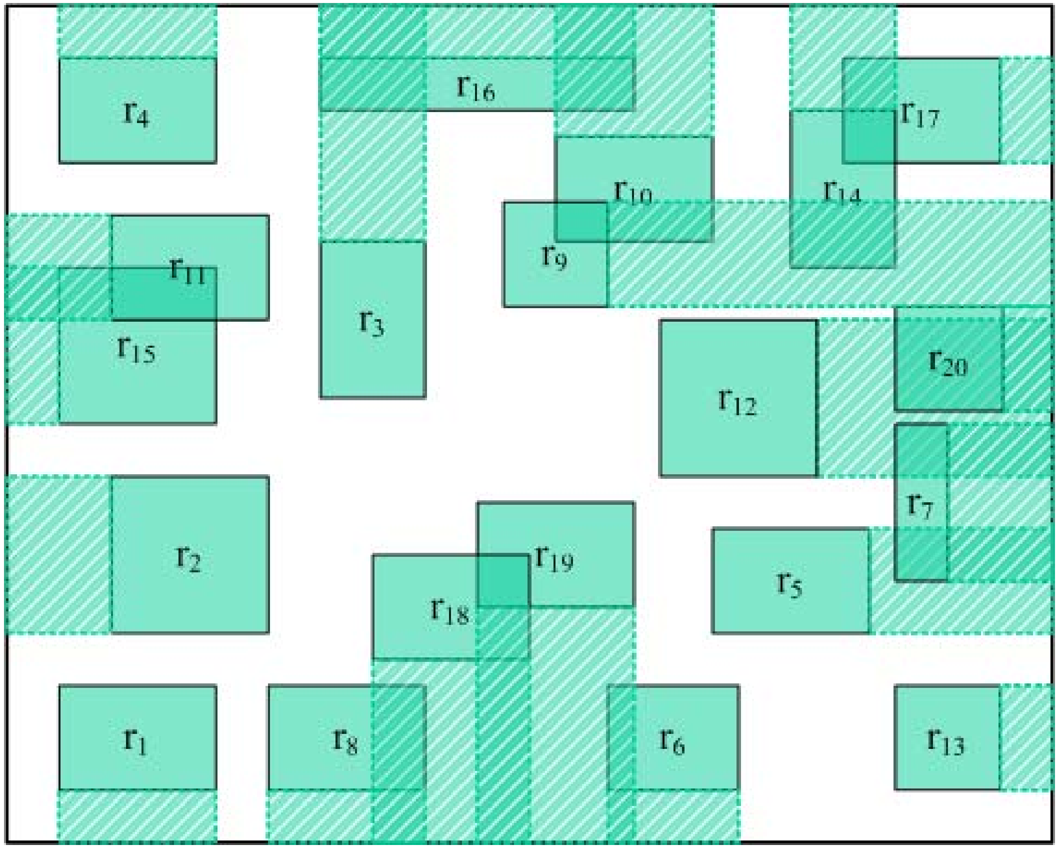
Experimental Results

Test Cases	# Bus	Density by LP	Density After Rounding	Density After Refinement
Ex1	16	1	1	1
Ex2	20	2	2	2
Ex3	24	2	2	2
Ex4	43	3.2	4	4
Ex5	44	3.08	4	4
Ex6	69	3	3	3
Ex7	106	3.14	4	4
Ex8	129	4.12	5	5
Ex9	148	4.4	6	5
Ex10	148	4.5	5	5

Experimental Results

- Sample escape solution for test case Ex2

density = 2



Outline

- Introduction
- Rectangle Escape Problem Formulation
- Proof of NP-Completeness
- An Approximation Algorithm
- Experimental Results
- **Conclusions**

Conclusion

- The Rectangle Escape Problem is introduced
- NP-completeness of REP is proved
- A 4-approximation algorithm is proposed
- It has application in PCB bus escape routing

Thank You!