Routability Booster–Synthesize a Routing Friendly Standard Cell Library by Relaxing BEOL Resources

Bing-Xun Song, Ting Xin Lin, & Yih-Lang Li
National Yang Ming Chiao Tung University
Computer Science Department
ISPD 2024

Electrical Design Automation Lab.

Outline

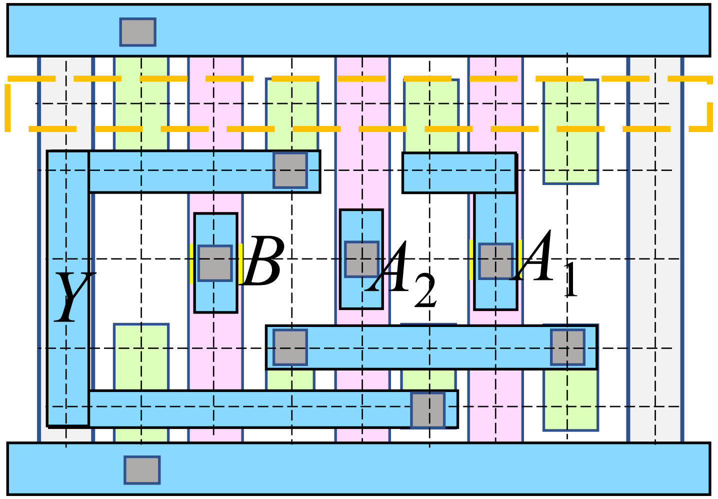
- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

Outline

- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

Introduction

- In advanced technology nodes, routing resources significantly determine routability in physical design stage.
- In this work, we propose a method to synthesize a standard cell library that has two key features to increase routability in P&R stage.
  - Offer more routing resources for the following P&R stage.
  - Increase pin accessibility when synthesizing the cells.

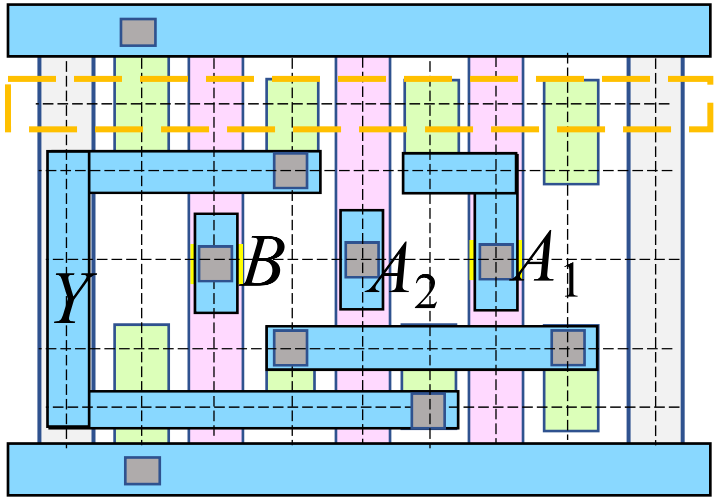


Outline

- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

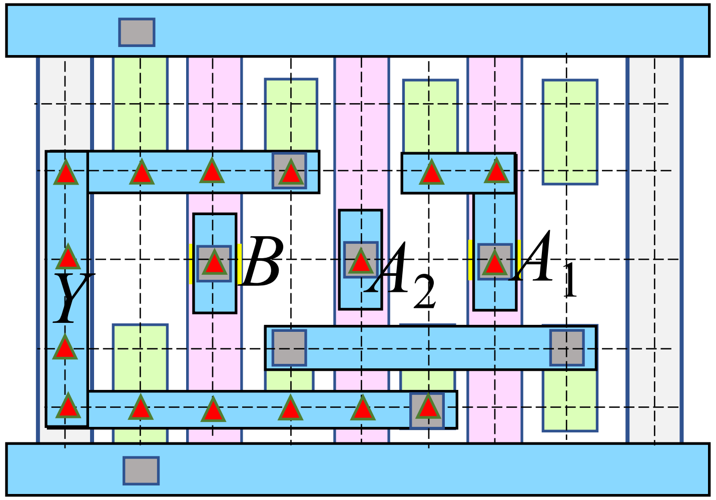
Spare Track

- To offer more routing resources, we define spare tracks before synthesizing cell libraries.
- A spare track is an M1-layer routing track that is a specific track and kept empty during cell routing as an available M1 routing track for the upper-level routing.

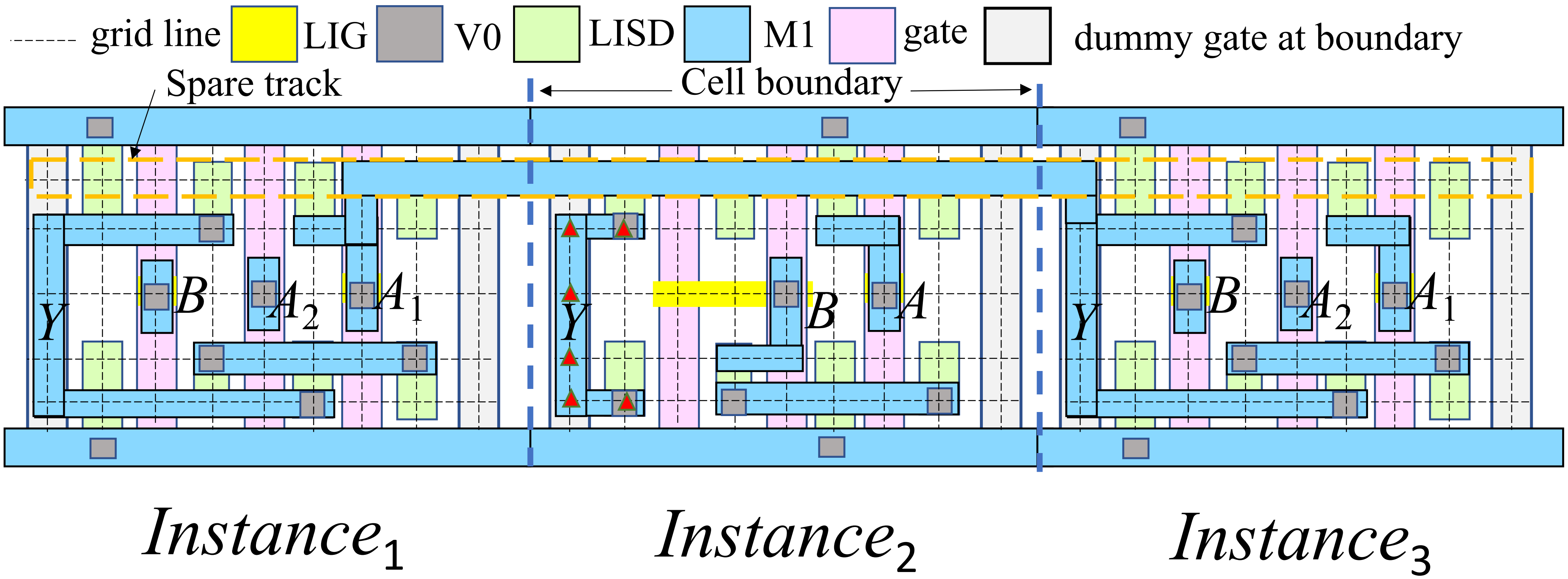


Pin Accessibility

- For evaluating the pin accessibility of a synthesized cell, we calculate its on-grid access points of each pin metal.
  - Vertical access points: access points that can be accessed by using a V1 via.
  - Planar access points: access points that can be accessed by using the spare track.



Example of P&R Results



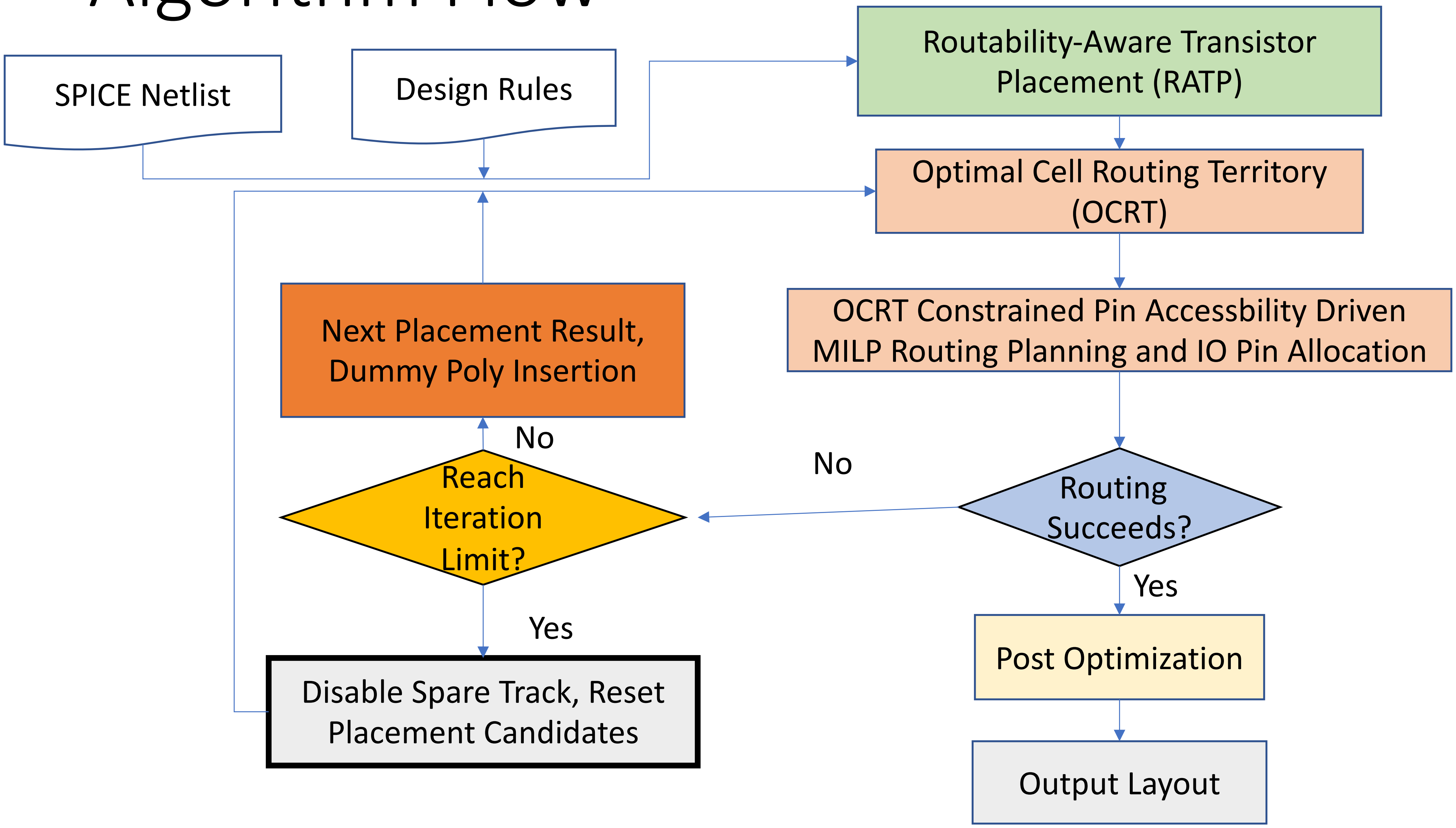
Problem Formulation

- Input: Cell netlist and design rule
- Output: Physical Cell Layout that is DRC-clean and passes LVS verification with
  1. Designated spare track unused.
  2. Maximize the pin accessibility
  3. Minimize the cell area, wirelength and M2 metal usage.

Outline

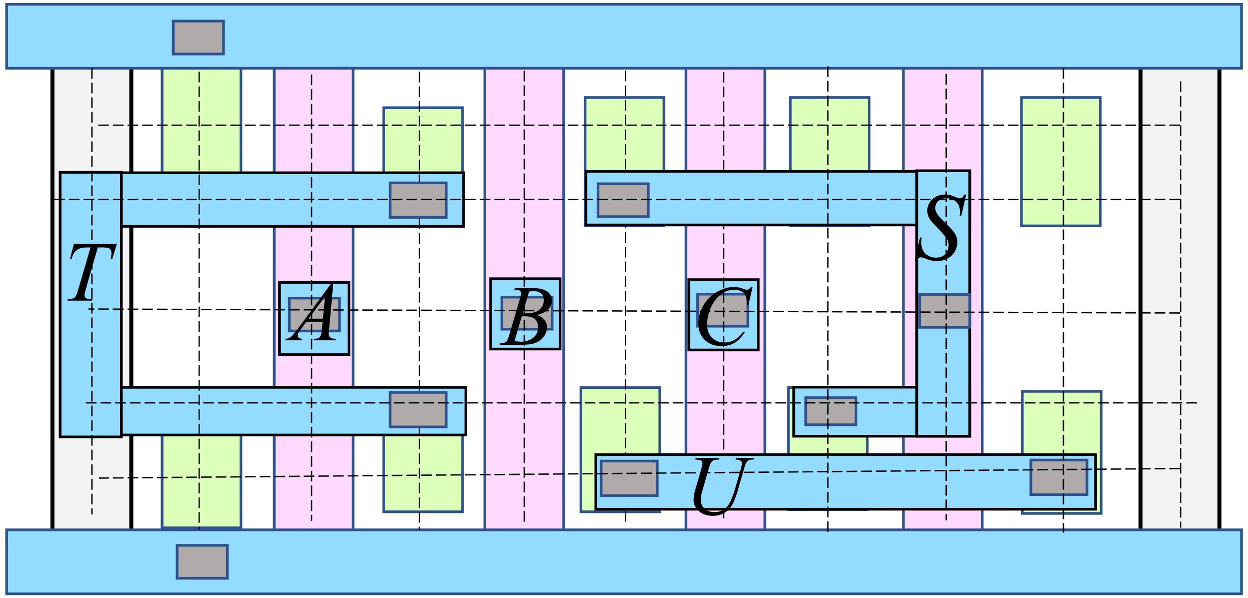
- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

Algorithm Flow



Track Assignment Based Routing Estimation

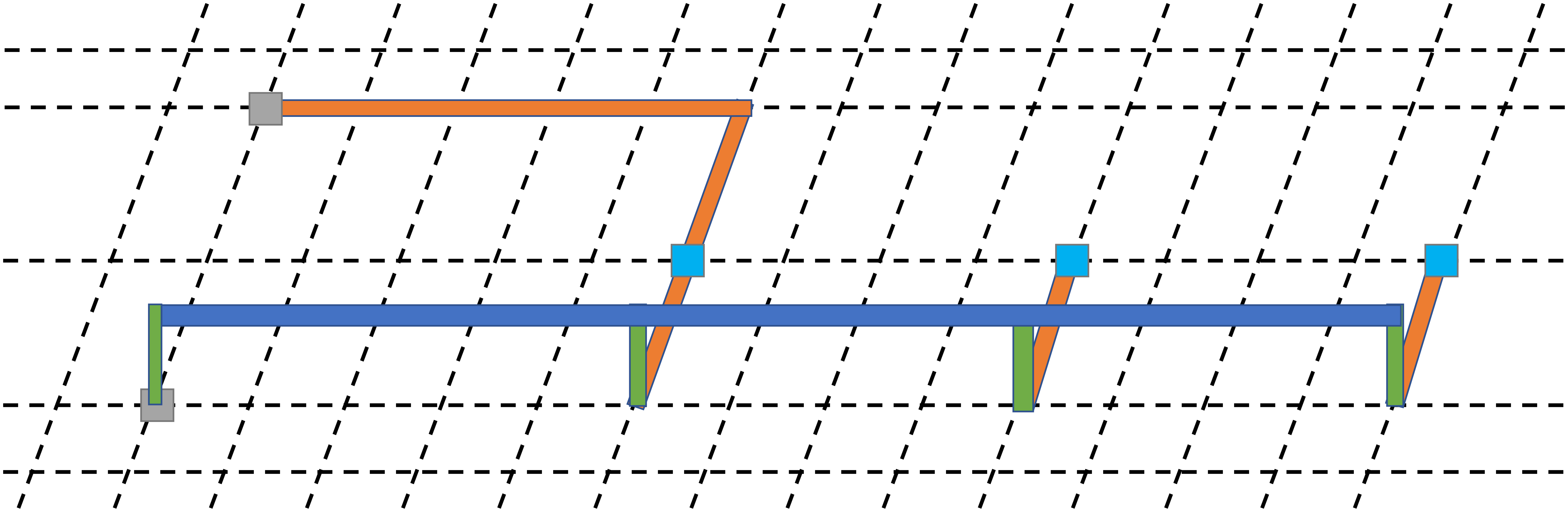
- For each sub-chain, we use track assignment to get a fast routing estimation without using the spare track.
  - Generate segments by the current chain pin allocation.
  - Use greedy algorithm to place segments onto a minimum cost track in non-descending order of segment length.



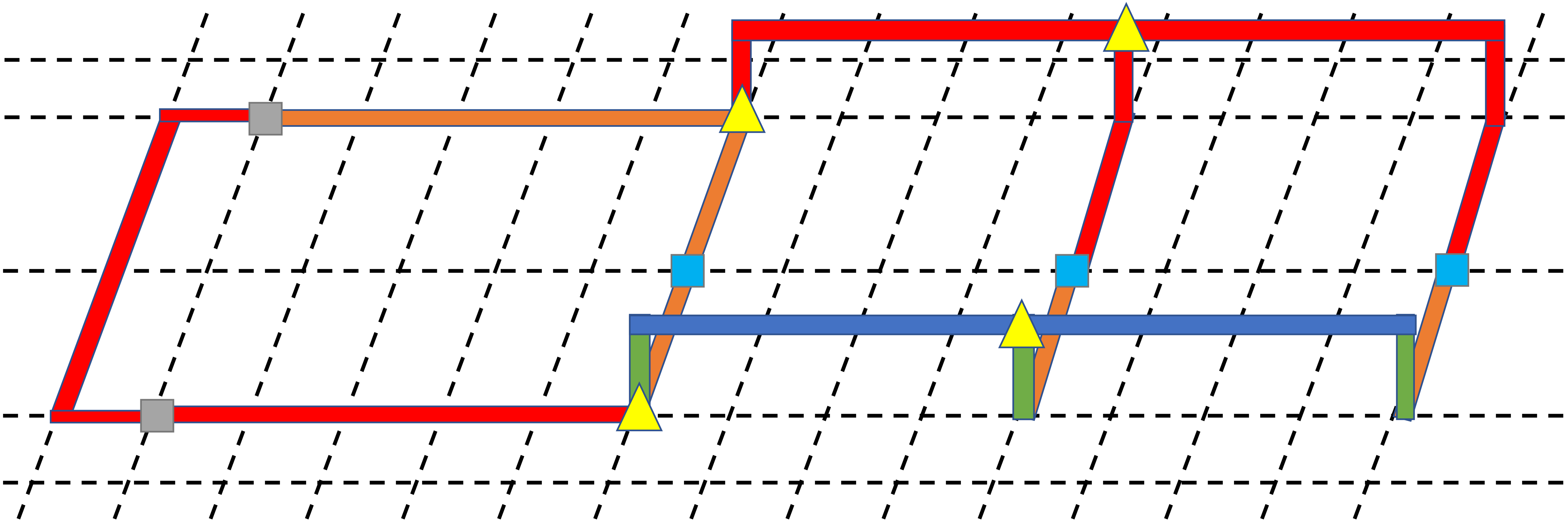
Optimal Cell Routing Territory (OCRT)

- After the transistor placement is done.
- For each net, we enhance the track assignment results to find a possibly best topology quickly.
  1. Track assignment results
  2. Extend to 3D contour graph
  3. Find a Steiner Minimum Tree (SMT) on the 3D contour graph
- The found SMT is an important implication for the following MILP routing.

Track Assignment Results

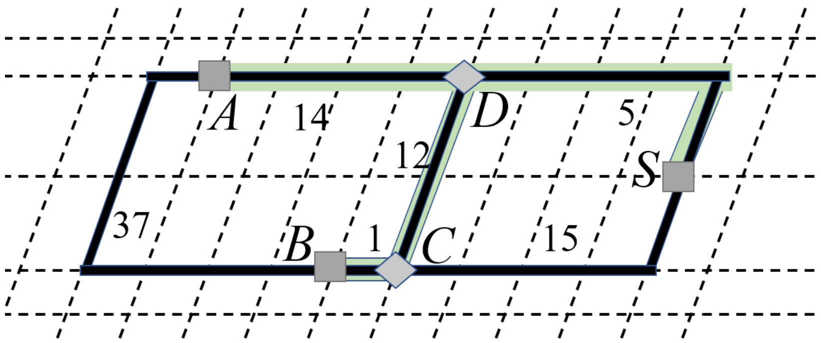


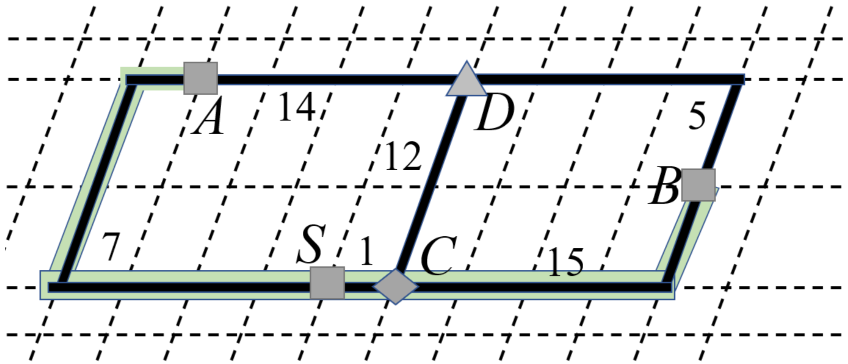
3D Contour Graph Example



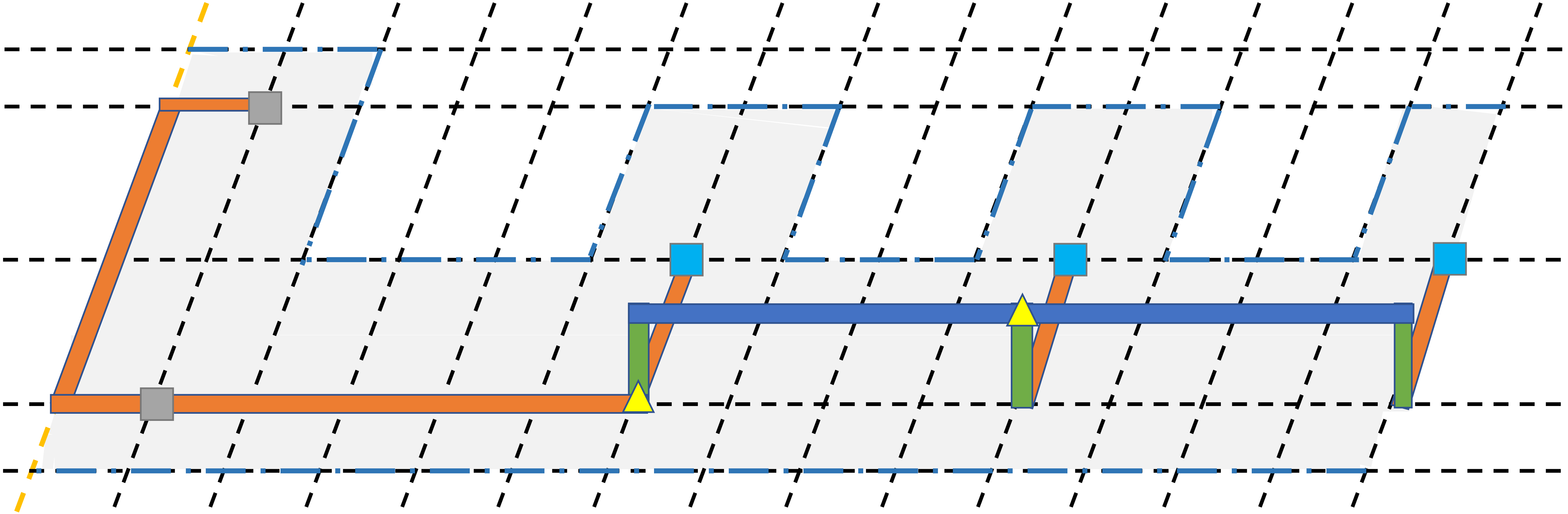
Steiner Minimum Tree (SMT)

- On a 3D contour graph, we use a mixed algorithm of Dijkstra's algorithm and Prim's algorithm to find a Steiner Minimum tree.
- For a pin or a confirmed Steiner point => propagate using Prim's algorithm
- For a potential Steiner point => propagate using Dijkstra's algorithm





SMT Result and OCRT Example



MILP Cell Routing and IOPA

- MILP: Mixed Integer Linear Programming
- IOPA: IO Pin Metal Allocation
- In our MILP formulation, spacing and short violations are allowed, while the open violations are not allowed. To plan IO Pin metals more accurately
  - All nets are given starting solution based on the SMT results.
  - All nets are constrained to only find solutions in the OCRT.

MILP Constraints

- On our routing graph, each net routing is expelled to use the spare track initially.
- Forbidden Region Constraints
  - Prevent nets from routing outside OCRT.

$$\forall n_s \in N_s, \forall g(x,y,z) \notin \mathcal{R}_r(n_s), \mu_{gsn}(x,y,z,n_s) = 0$$

- Connectivity Constraints
  - Make sure the nets are not open.

$$\forall n_s \in N_s, \mu_{gsn}(x,y,z,n_s) = \bigvee_{\forall e \in E_{in}(x,y,z)} \mu_{esn}(e,n_s)$$

$$\forall p_f, \sum_{\forall node\ v_p \in p_f} b_{fp}(v_p) = 1$$

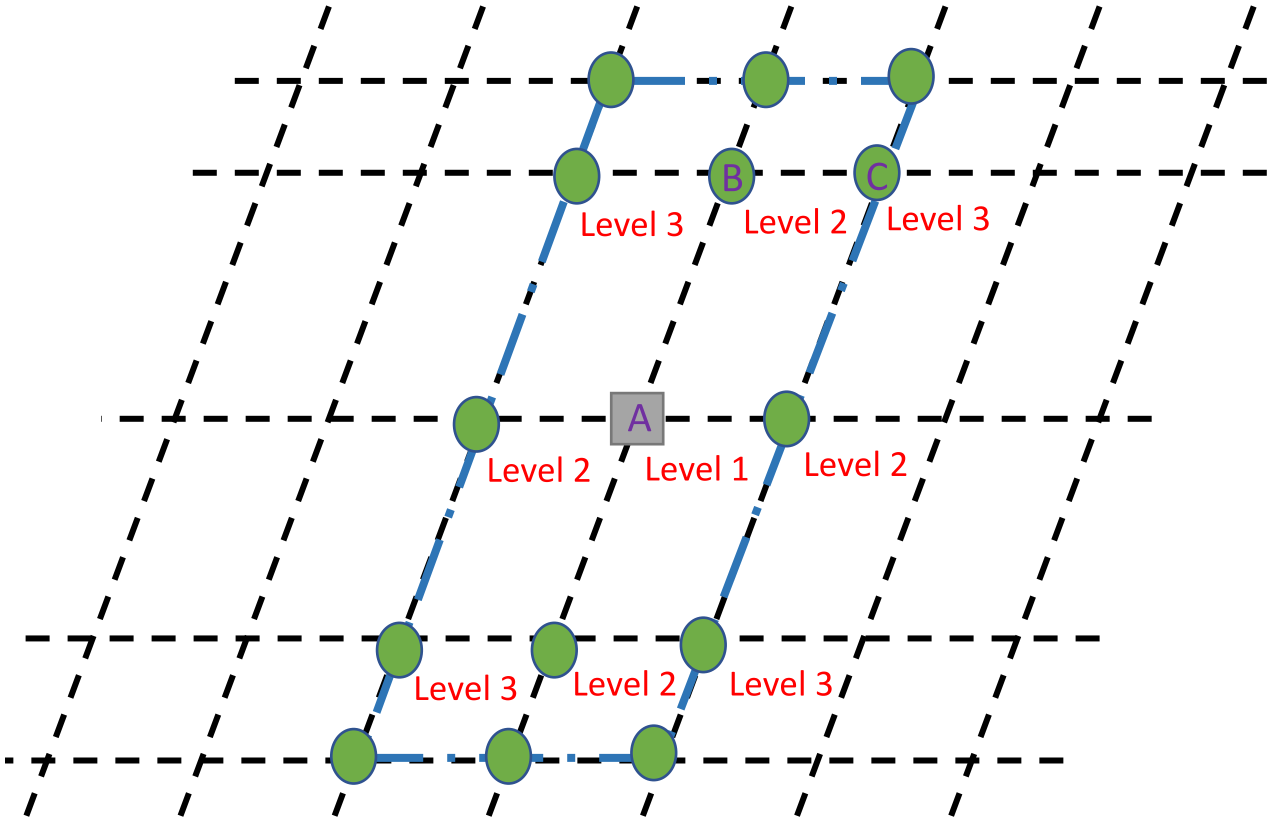
$$\left(deg(x,y,z,sn(p_f)) + b_{fp}(v_p)\right) = 0 \ \vee \left(deg(x,y,z,sn(p_f)) + b_{fp}(v_p)\right) = 2$$

$$deg(x,y,z,n_s) = 0 \vee deg(x,y,z,n_s) = 2$$

- Pin Accessibility Constraints
  - Make sure the length of each pin metal can be maximized legally.
  - Single-Terminal Input Nets Constraints
  - Cyclic Avoiding Constraints

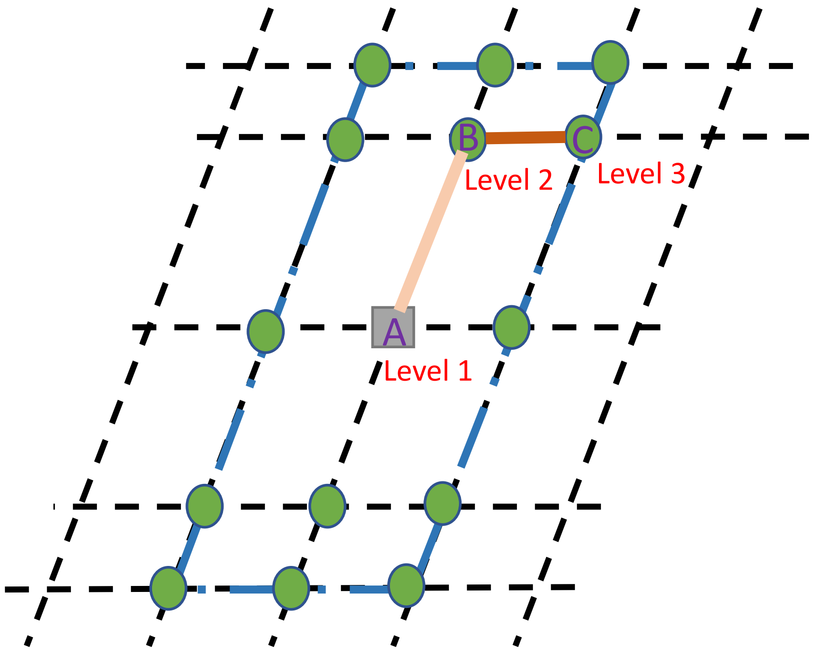
Single Terminal Input Nets Constraints

- BFS propagation



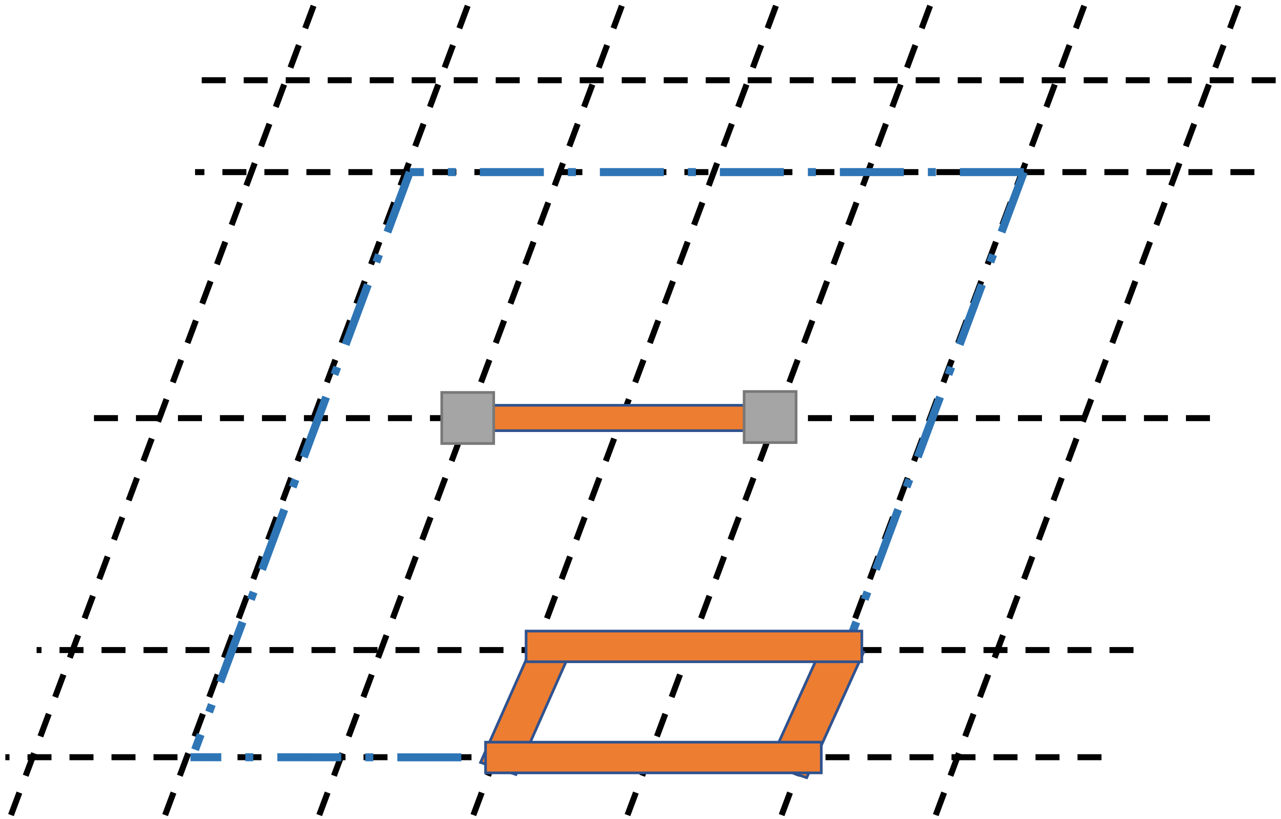
Single Terminal Input Nets Constraints

- $\forall$edge e connects $g_1(x_1,y_1,z_1)$ and $g_2(x_2,y_2,z_2)$ in the routing region ($level$ of $g_1 < level$ of g_2)
  - $\mu_{esn}(e, n_s) \Rightarrow \bigvee_{e' \in E_{bt} \; of \; g_1} \mu_{esn}(e', n_s)$. E_{bt} => backtrack edges
  - If edge BC is used, edge AB must be used as well.



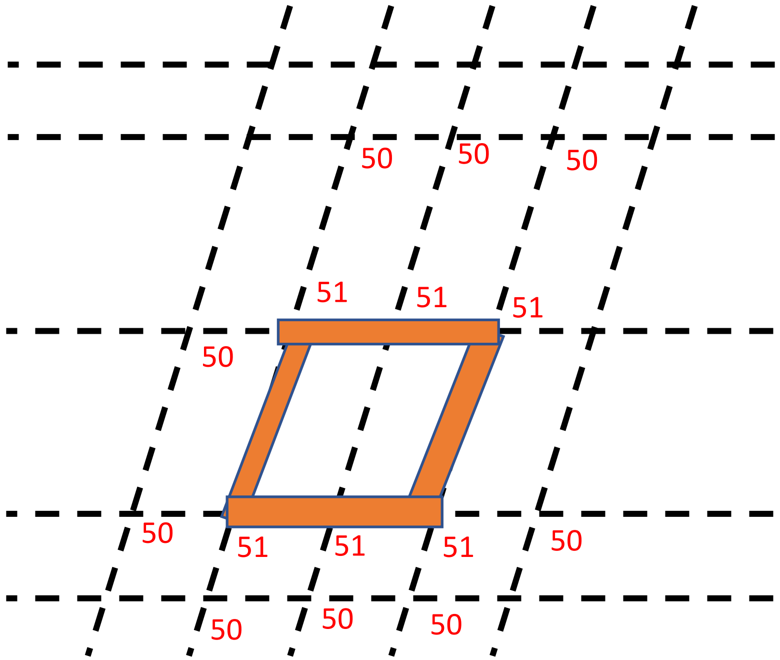
Cyclic Avoiding Constraints

- For multi-pin I/O nets



Cyclic Avoiding Constraints

- $\forall n_s \in$ multi-terminal I/O pin net, $\forall g(x,y,z) \in \mathcal{R}_r(n_s)$
  - $len(x,y,z,n_s) = \begin{cases} 1, if\ deg(x,y,z,n_s) = 1 \\ L_{MAX}, if\ deg(x,y,z,n_s) = 0 \\ \min_{\forall (x',y',z') \in G_{nbr}(x,y,z)} len(x',y',z',n_s) + 1\ , o/w \end{cases}$
  - $len(x,y,z,n_s) \le L_{MAX}$
  - $L_{MAX} = 50$



MILP Objectives

- Minimize max and total short violations
  - $\alpha_2 \times \max_{\forall g(x,y,z)} \sum_{\forall n \in N} \mu_g(x, y, z, n)$
  - $\beta_2 \times \sum_{\forall g(x,y,z)} (\sum_{\forall n \in N} \mu_g(x, y, z, n) > 1)$
- Minimize spacing violations
  - $\gamma_2 \times \sum_{n \in N} (\mu_g(x_1, y_1, z_1, n) \wedge \bigvee_{n' \in N, n' \neq n} \mu_g(x_2, y_2, z_2, n'))$
- Maximize Pin Accessibility
  - $\forall net\ n \in \text{IO pins},\ slack_{ap}(n) = \begin{cases} 0, if\ N_{ap}(n) \geq n_{TA} \\ \theta_2 \times (n_{TA} - N_{ap}(n)), o/w \end{cases}$
- Minimize M2 Usage
  - $\delta_2 \times \sum_{n \in N} \sum_{e} \mu_e(e, n)$
- Minimize Wirelength
  - $\varepsilon_2 \times \sum_{n \in N} \sum_{e \in E_{M2}} \mu_e(e, n)$

Maze Routing Optimization

- After the MILP routing and IOPA, we use the maze router to eliminate existing violations, with the lengthened pin metals remaining fixed.
- If there are still violations after maze routing, we insert a dummy poly for this placement candidate and deal with next placement candidate.

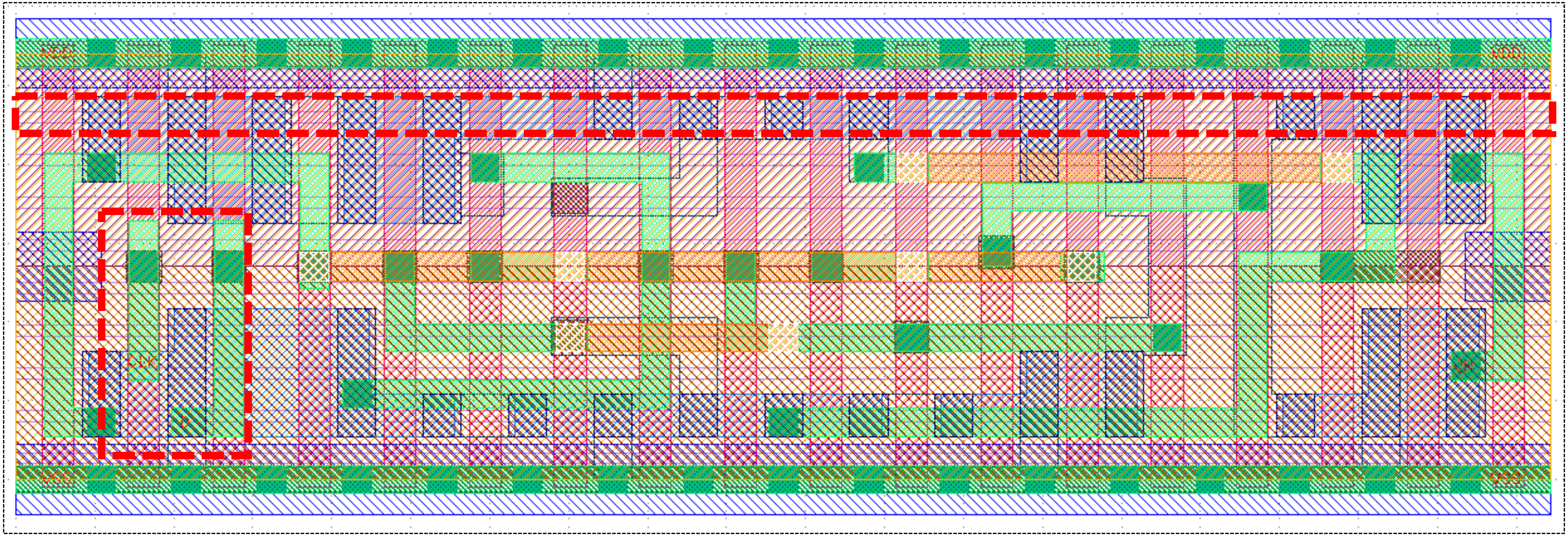
Outline

- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

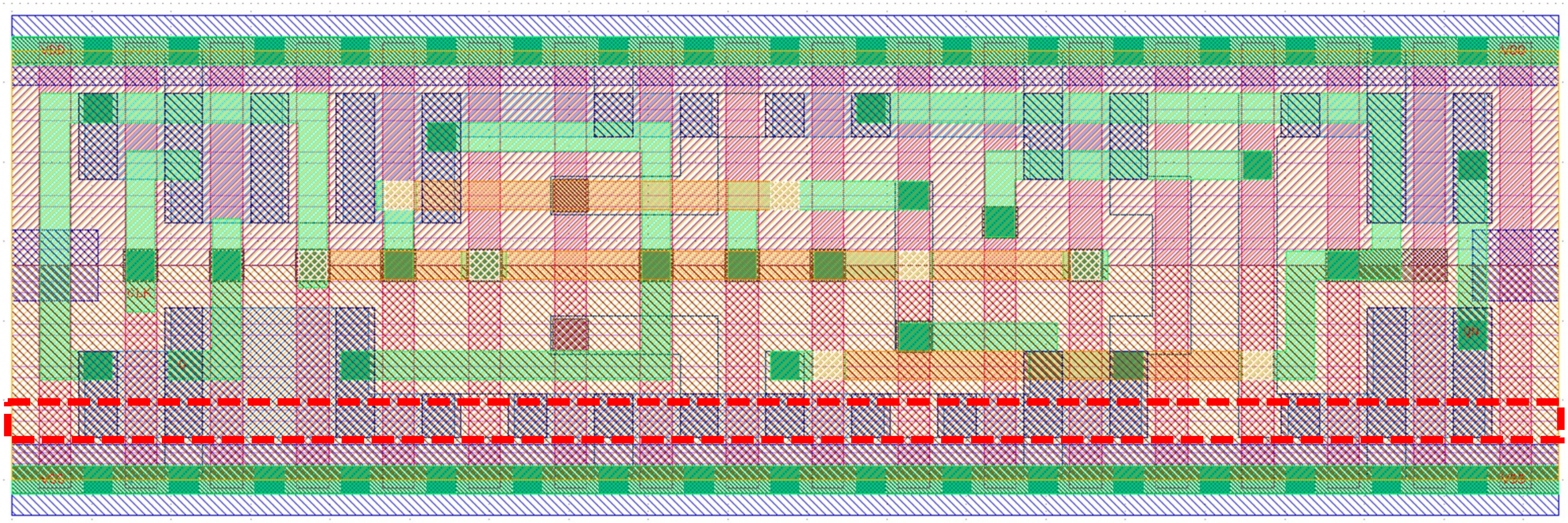
Environments

- Machines
  - 2.1 GHz twenty-core CPU and 252GB of memory
- MILP Solver
  - CPLEX
- Open source ASAP 7nm PDK.
  - https://github.com/The-OpenROAD-Project/asap7
- LVS/DRC verification
  - Mentor Calibre
- Cell characterization
  - Synopsys SiliconSmart

Spare PMOS Track Example



Spare NMOS Track Example



Cell Quality Comparison

NCTUcell ASAP7 PDK

Baseline Values

	Booster_P						[10]		ASAP_ASU		Booster_N			
	#T	#C	M	W	R	#F	M	W	M	W	M	W	R	#F
INV/BUF	13	23	18.63	0.47	28	0	0	1	0	1	0.63	1	40	0
AND/OR	12	17	50.22	0.46	49	0	0.13	0.98	0	1.01	1.25	1	296	2
NAND/NOR	8	16	18.14	5.83	25	0	0	0.96	0	0.96	1.54	1	46	1
AO/AOI	15	14	7.884	7.88	733	11	0	0.96	0	0.97	3.9	0.99	194	0
OA/OAI	14	14	66.26	7.18	101	1	0	0.98	0	0.98	0.37	1.01	324	9
MAJ/MAJI	13	3	29.48	1.51	475	1	0	0.93	0	0.92	0.78	0.96	384	2
XOR/XNOR	13	4	31.27	2.11	587	1	0.66	0.97	0.79	1.08	1.13	1	247	1
HA/FA	17	2	29.16	1.24	7208	1	1.27	1	1.15	1	0.98	1	7231	1
DFF/Latch	24	14	199.3	13.2	532	0	1.29	1.07	1.38	1.11	1.13	1	551	0
SDF/ICG	30	6	117.1	7.45	7363	0	1.53	1.11	2.05	0.96	1.12	0.98	8212	1

[10] Y.-L. Li, et al., "NCTUcell: A DDA-Aware Cell Library Generator for FinFET Structure with Implicitly Adjustable Grid Map," In Proc. of DAC, Jun, 2019

P&R Results

- IWLS'05 benchmarks

TABLE 3. P&R RESULT COMPARISON AMONG FOUR LIBRARIES.

Circuit	Booster_P	Booster_P_No_PinAc			ASAP_ASU			[10] NCTUcell		
	#DRC	#DRC	WL	#Via	#DRC	WL	#Via	#DRC	WL	#Via
b10	0	12	1.14	1.09	0	1.03	1.01	34	1.13	1.22
b12	0	11	1.04	1.04	0	1.03	1.01	218	1.09	1.19
b14	0	33	1.06	1.04	1	1.01	1.01	319	1.14	1.23
b15	0	90	1.07	1.07	0	0.98	1.01	1013	1.05	1.16
b20	0	86	1.04	1.05	0	1	1.04	62	1.02	1.06
b20_1	0	1093	1.28	1.1	0	1.02	0.99	542	1.12	1.13

[10] Y.-L. Li, et al., "NCTUcell: A DDA-Aware Cell Library Generator for FinFET Structure with Implicitly Adjustable Grid Map," In Proc. of DAC, Jun, 2019

Outline

- Introduction
- Preliminary
- Methodology
- Experimental Results
- Conclusions

Conclusions

- We present a cell synthesis methodology that allows us to keep the spare track unused during cell synthesis.
- The proposed MILP can efficiently lengthen the I/O pin metals to improve pin accessibility.
- The experimental results show that the synthesized cell library successfully outperforms the handcrafted cell library that has very high pin accessibility.

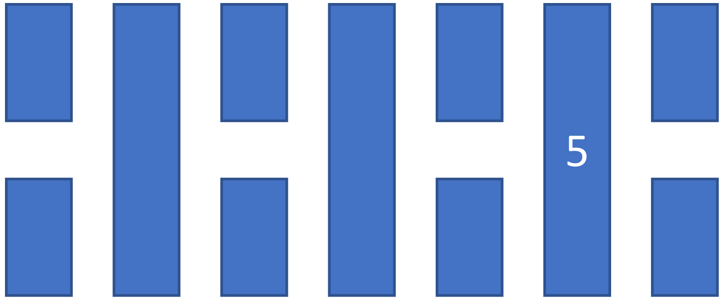
Thanks for listening.
Q&A.

Appendix

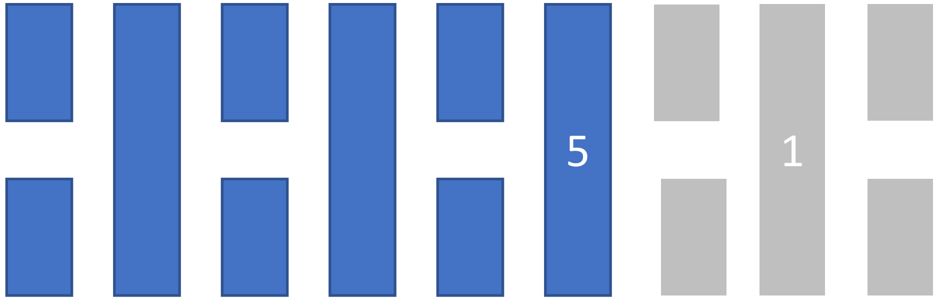
Routability-Aware Transistor Placement (RATP)

- Use dynamic programming to do transistor placement
  - Permutate possible transistor pairs for pMOS and nMOS transistors.
  - DP state: $S[n, k]$, the minimum cost transistor placement result of n transistor pairs with the k^{th} transistor pair placed at the rightmost of the placement (a sub-chain)

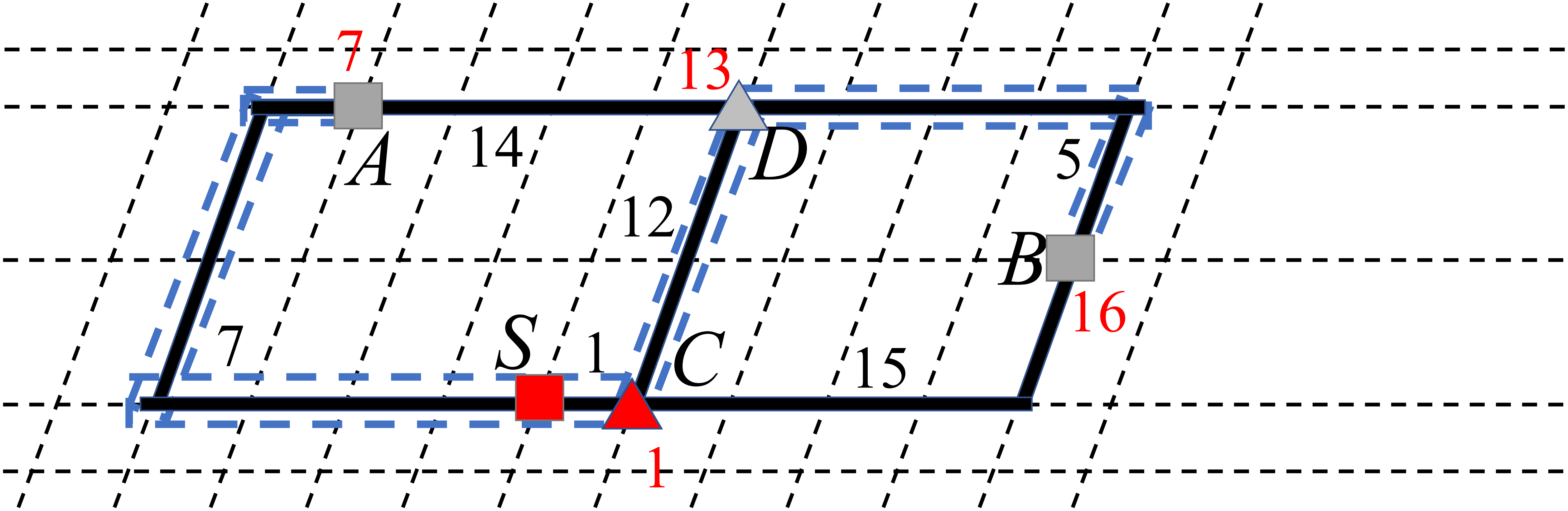
$S[3,5]$



$S[4,1]$

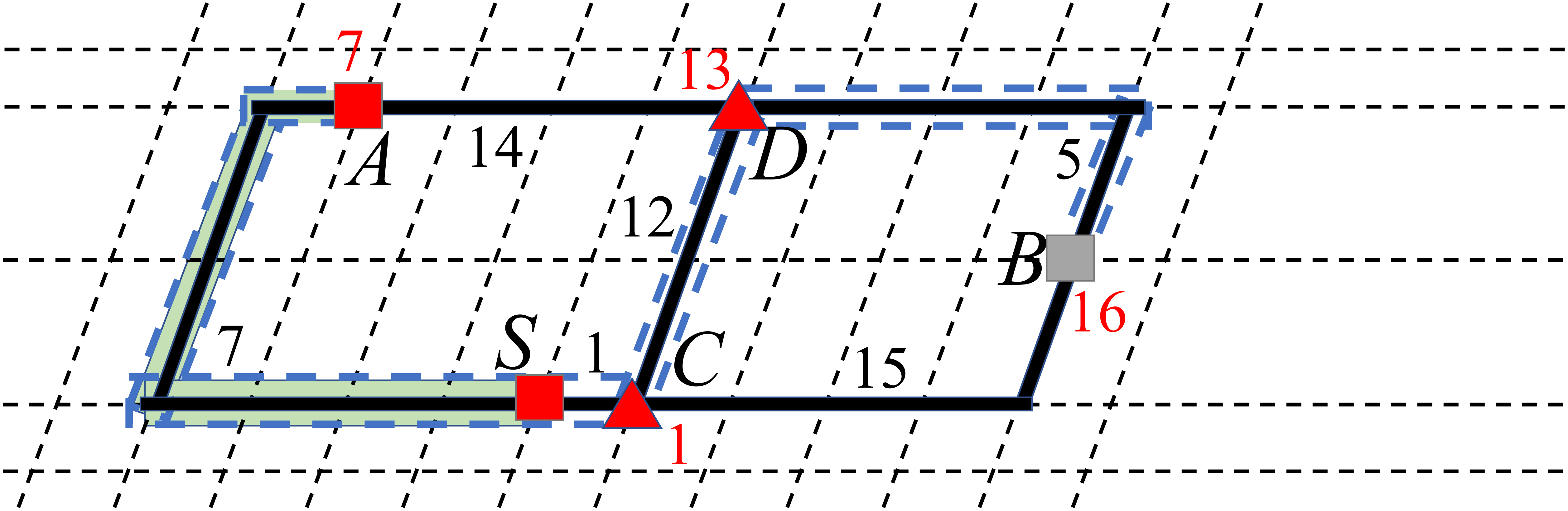


Why Not Using Prim's Directly



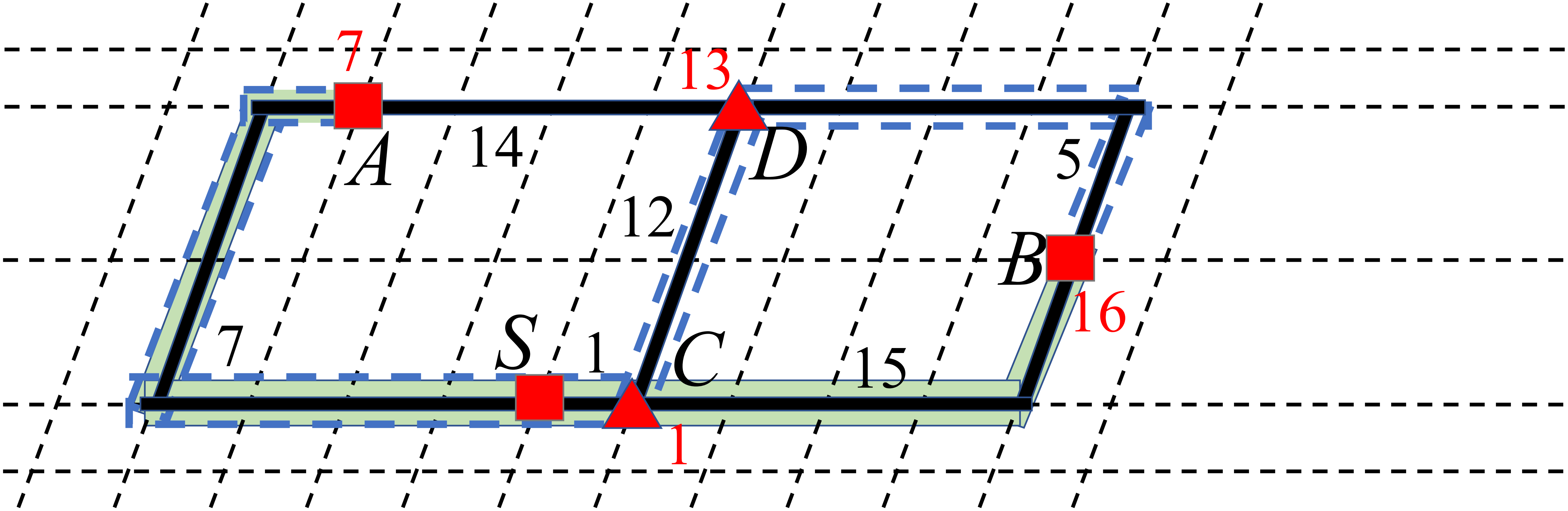
Weight=25

Why Not Using Prim’s Directly



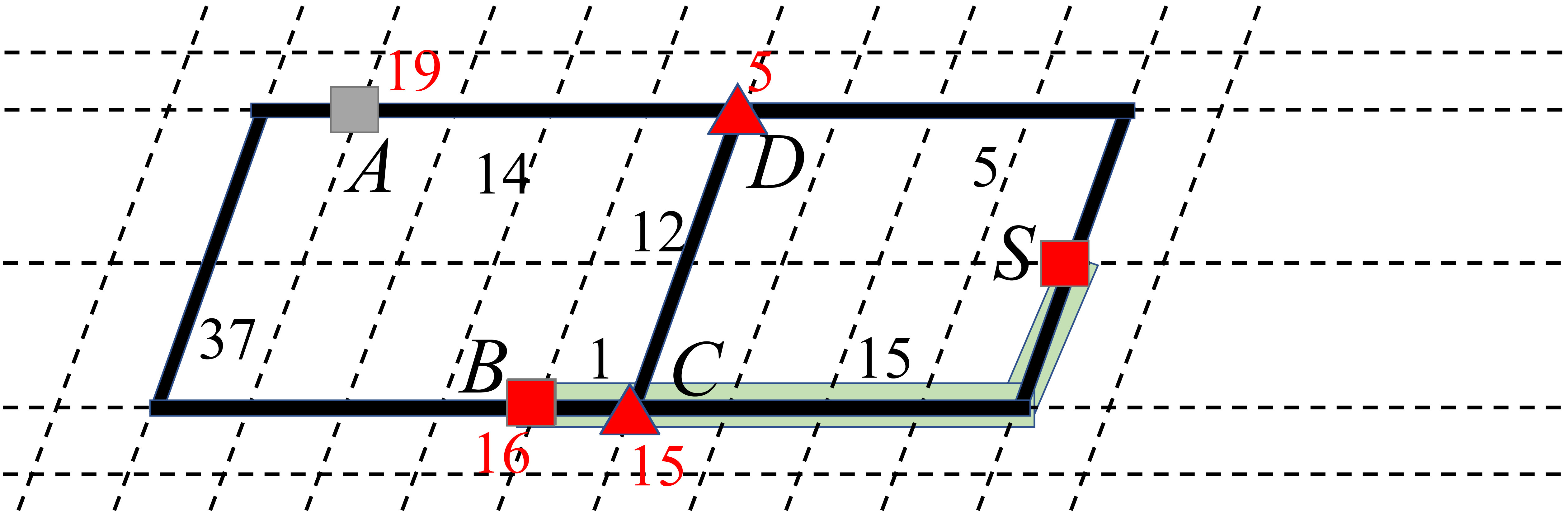
Weight=25

Why Not Using Prim's Directly

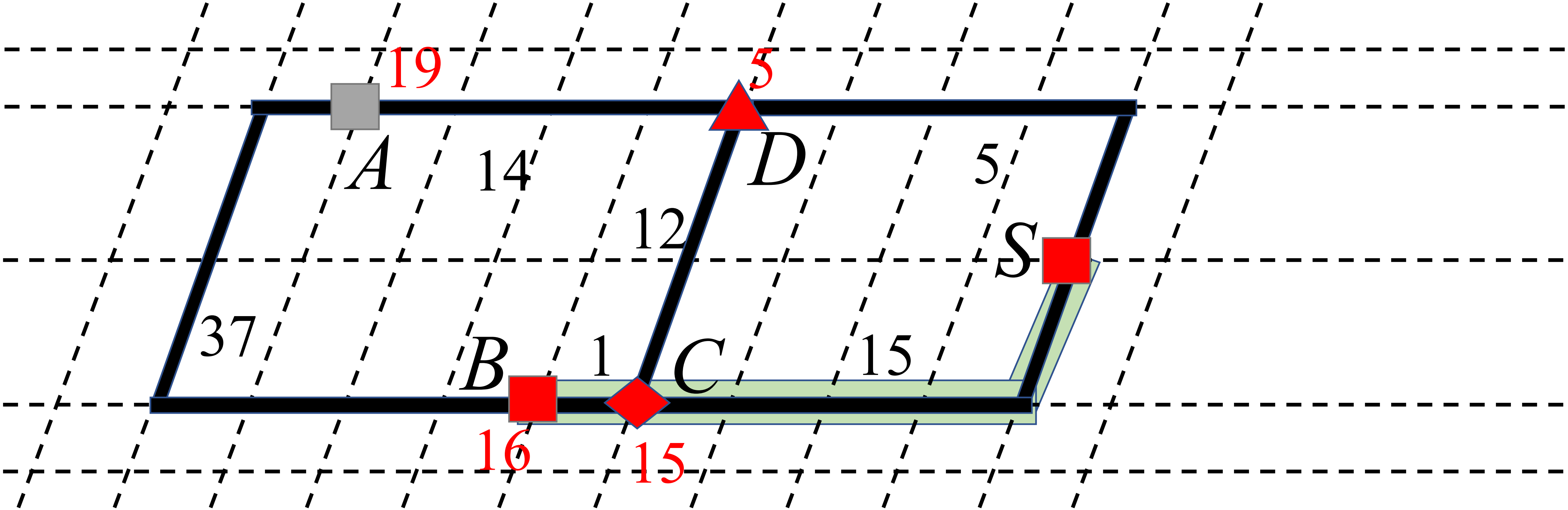


Weight=23 < 25

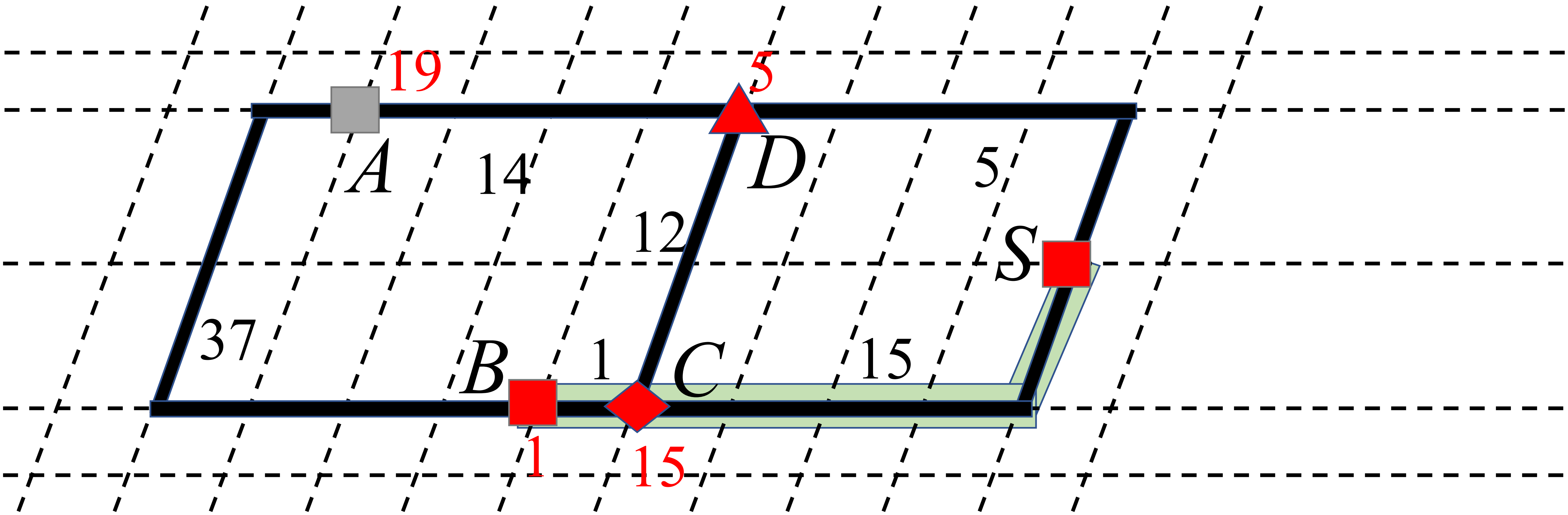
SMT Finding Example



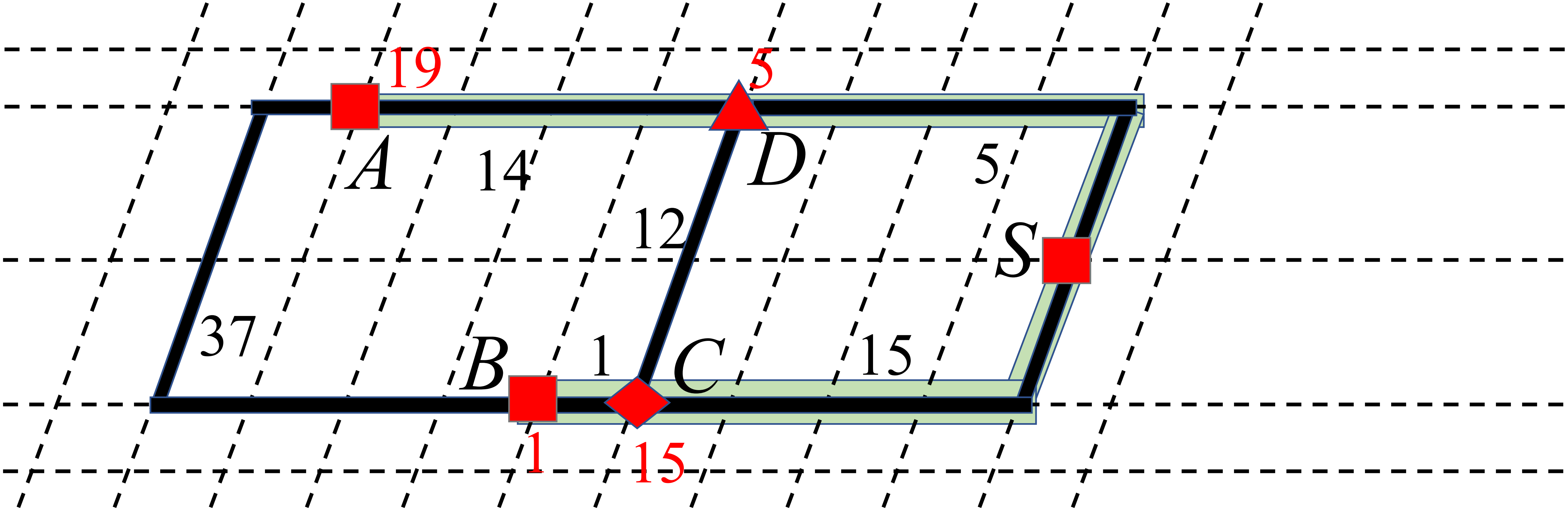
SMT Finding Example



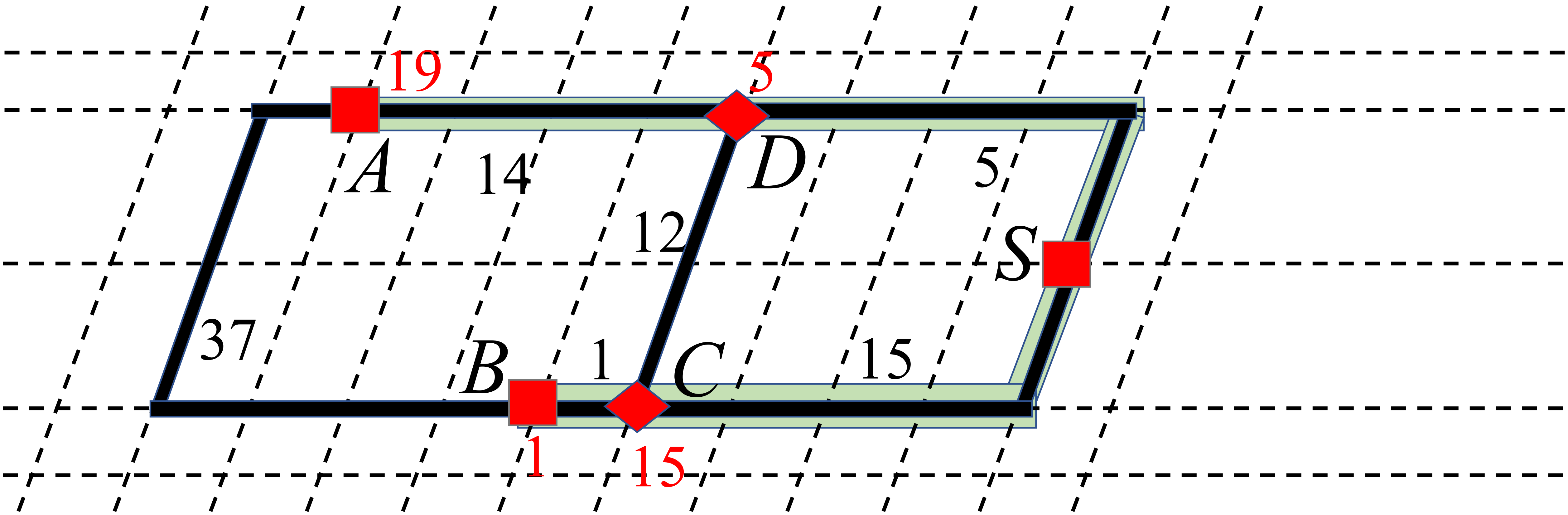
SMT Finding Example



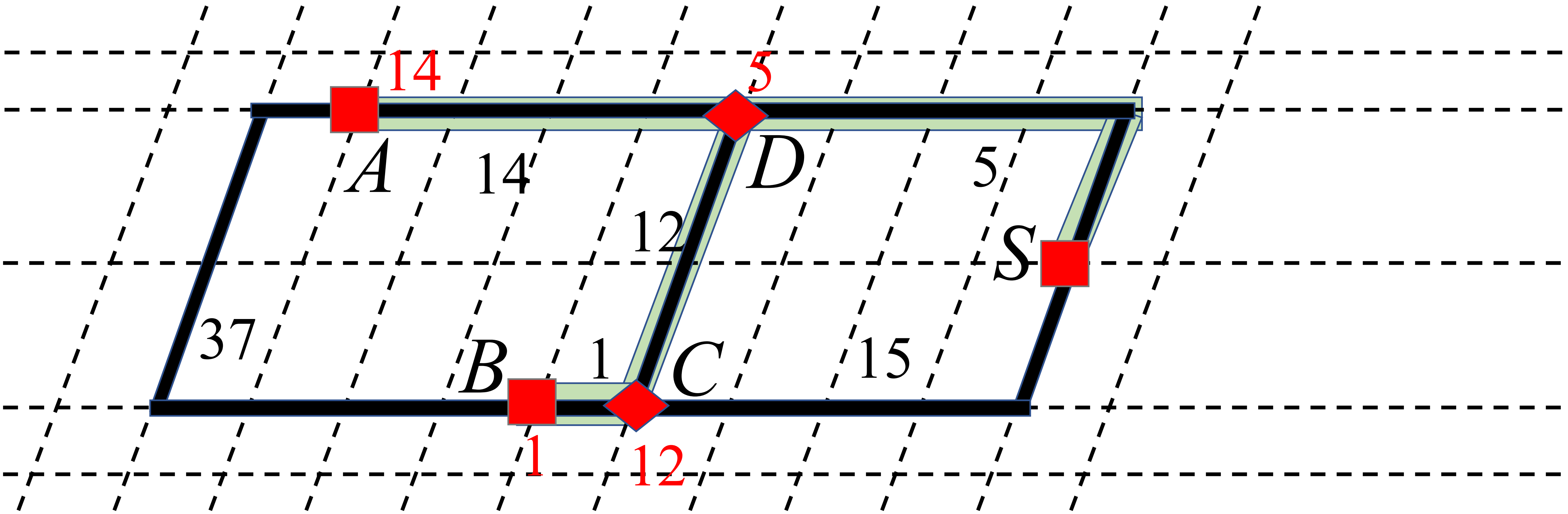
SMT Finding Example



SMT Finding Example



SMT Finding Example



Single Terminal Input Nets Constraints