

# A Fast, Robust Network Flow-based Standard-Cell Legalization Method for Minimizing Maximum Movement

Nima Karimpour Darav  
Laleh Behjat

Ismail Bustany

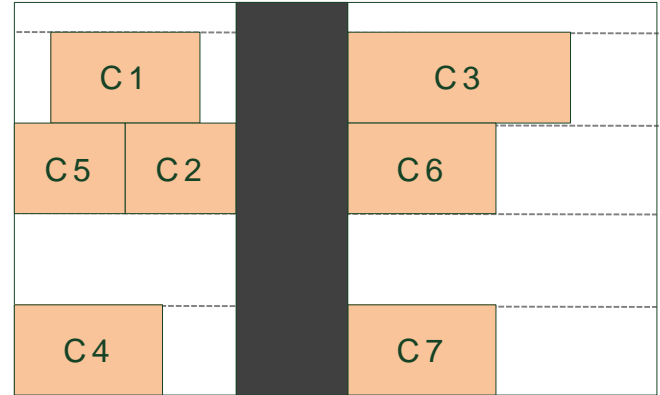
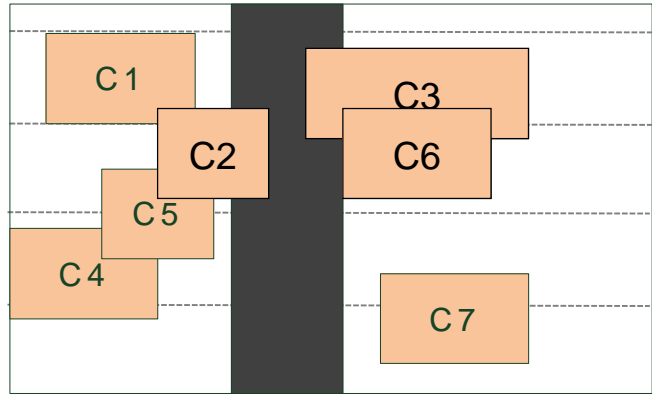
Andrew Kennings





# Outline

- **Standard Cell Legalization Problem**
- **Problem Analogy**
- **Challenges**
- **The Proposed Network Flow Based Algorithm**
- **Illustrative Example**
- **Empirical Results**
- **Conclusion**

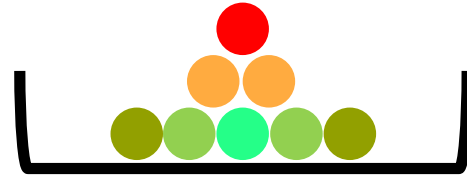
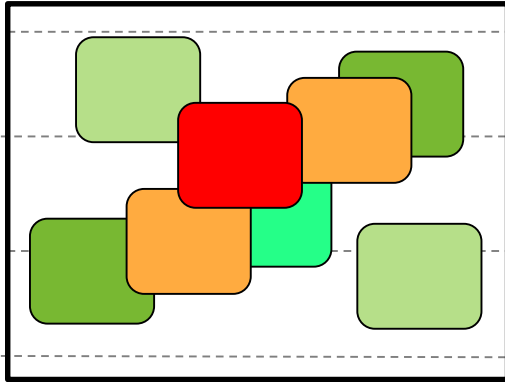
# Standard Cell Legalization Problem



-  Blocked area
-  Movable cells

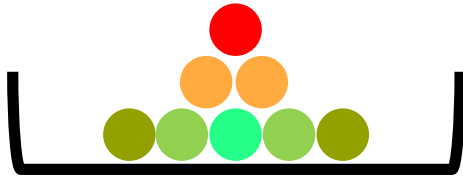
# Problem Analogy

# Problem Analogy

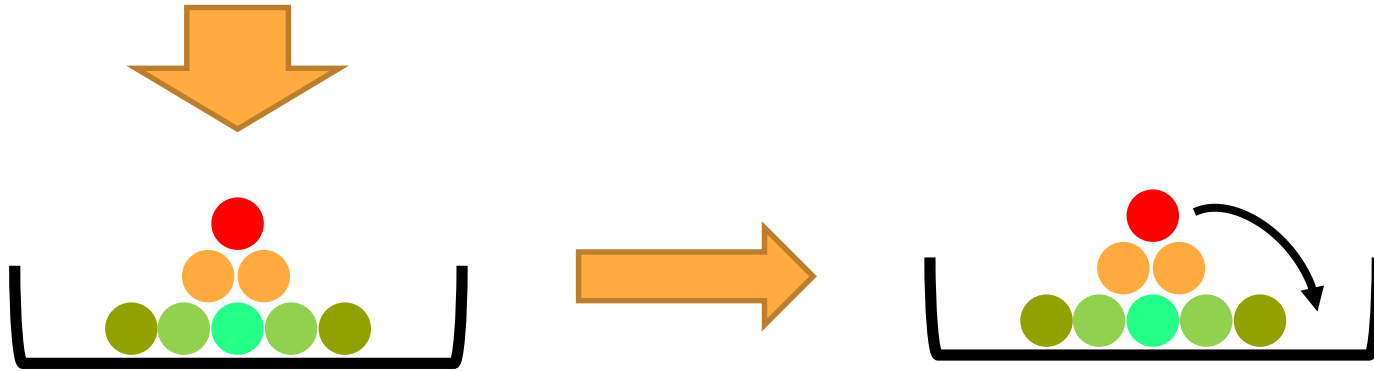


**The legalization process is similar to the process of flattening humps**

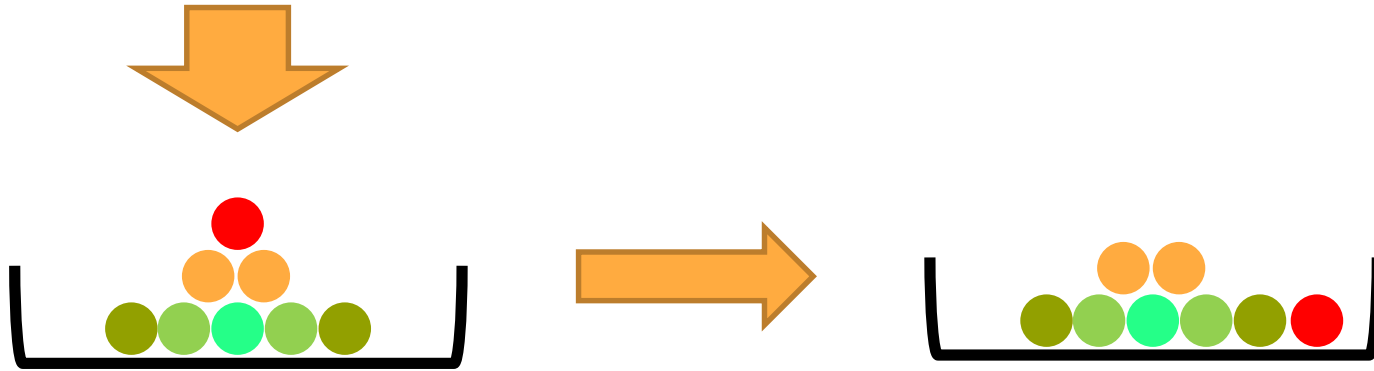
# How To Flatten A Hump?



# How To Flatten A Hump Using Traditional Approaches?

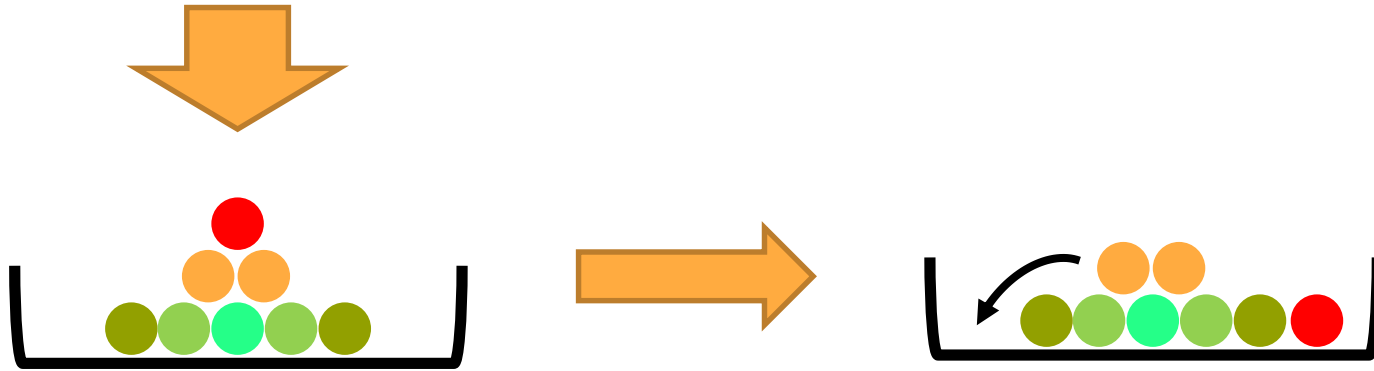


# How To Flatten A Hump Using Traditional Approaches?

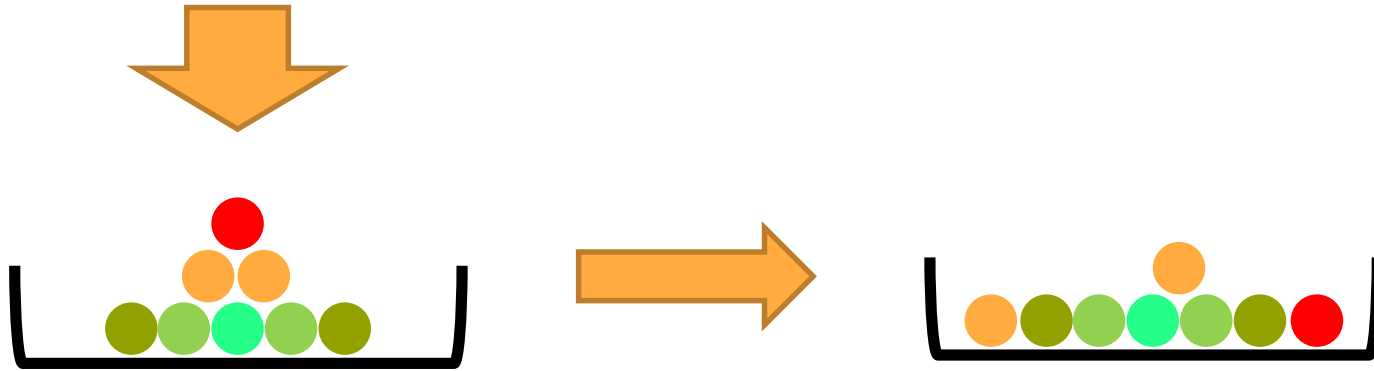




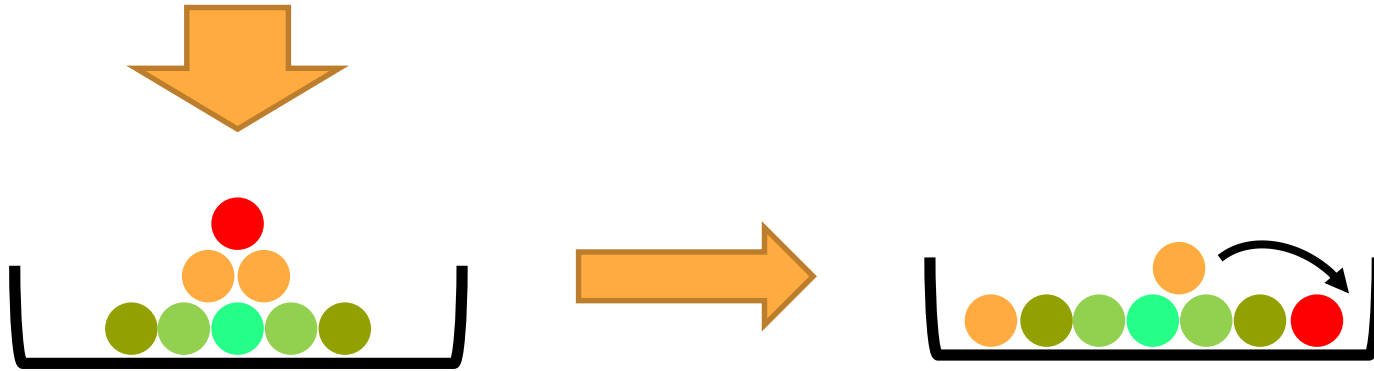
# How To Flatten A Hump Using Traditional Approaches?



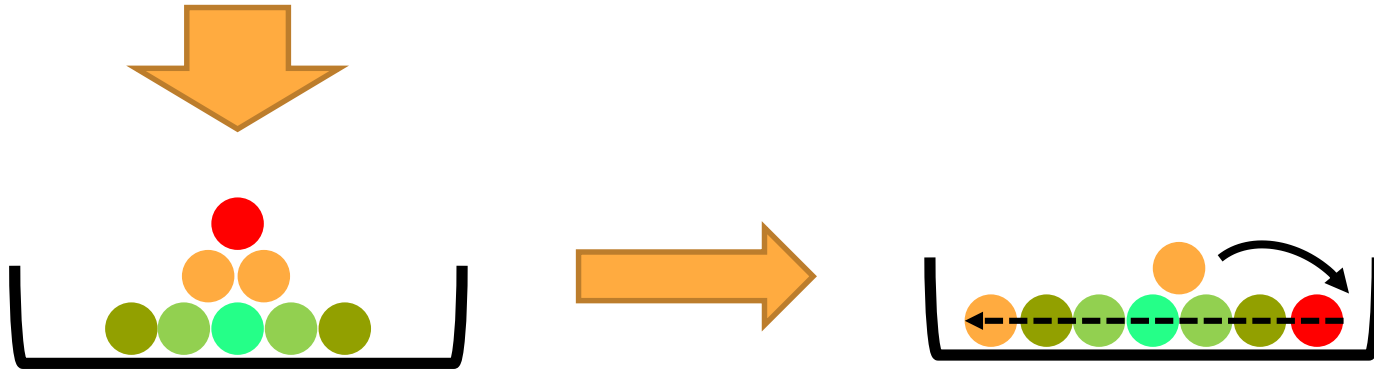
# How To Flatten A Hump Using Traditional Approaches?



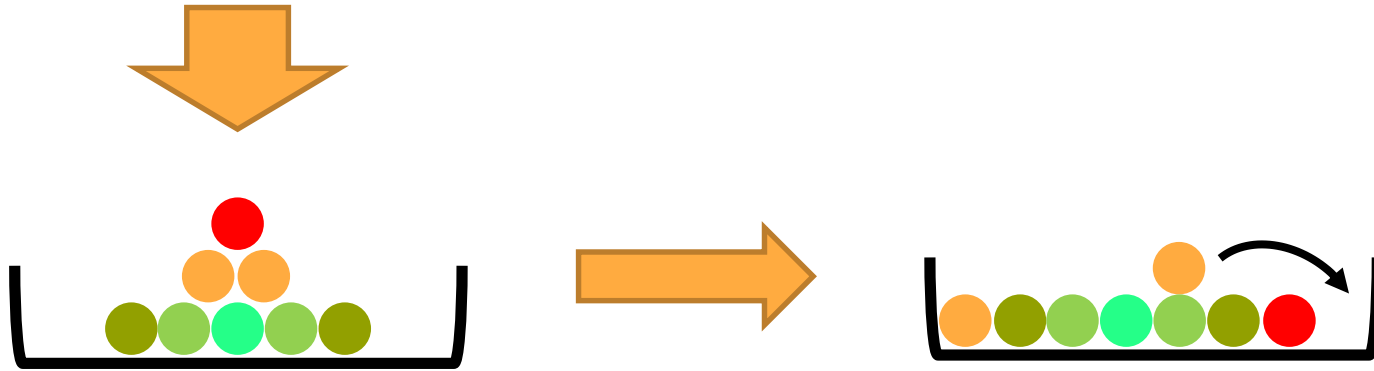
# How To Flatten A Hump Using Traditional Approaches?



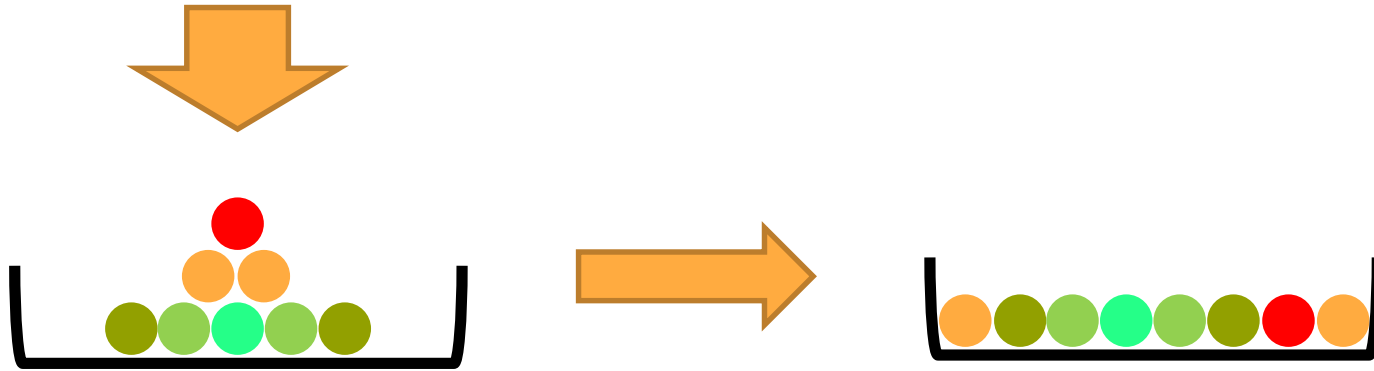
# How To Flatten A Hump Using Traditional Approaches?



# How To Flatten A Hump Using Traditional Approaches?

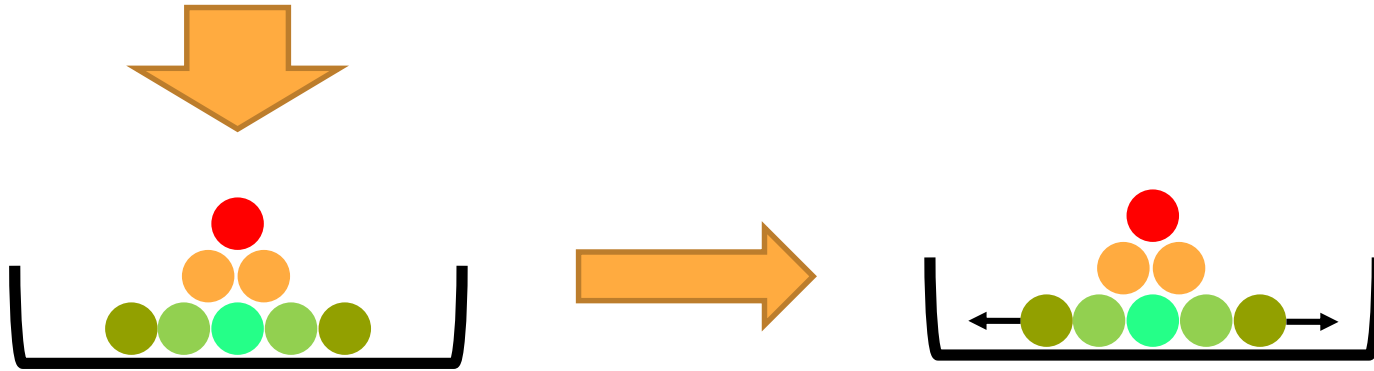


# How To Flatten A Hump Using Traditional Approaches?



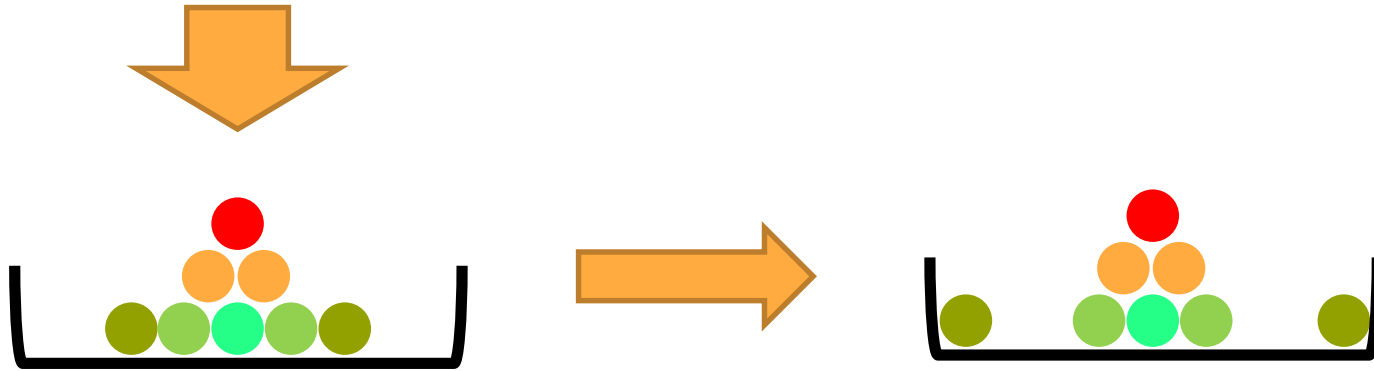
# Proposed Network Flow Based Approach

# Proposed Network Flow Based Approach

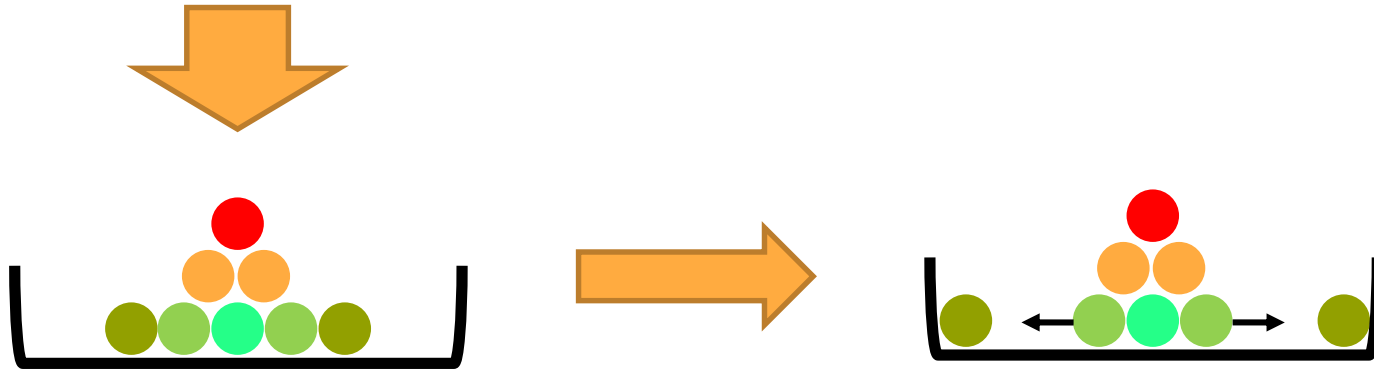




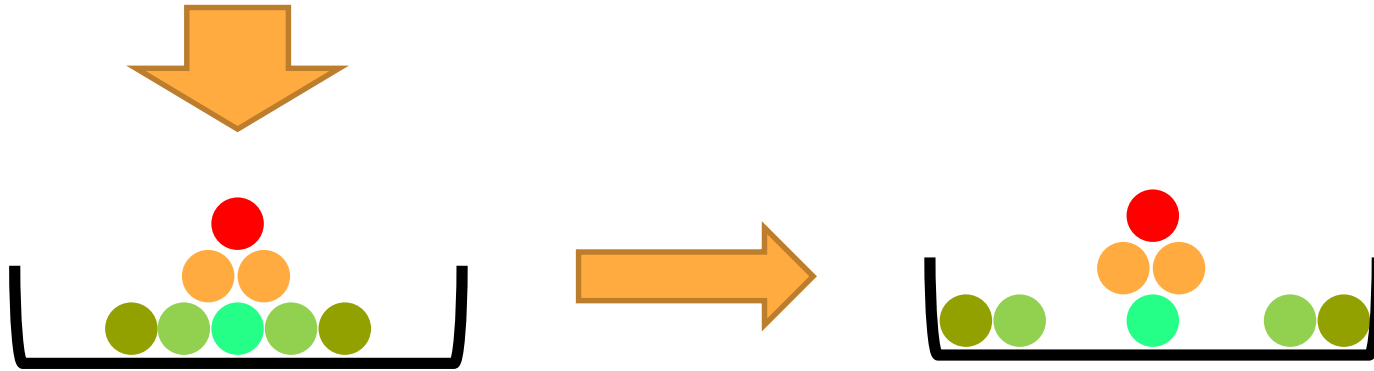
# Proposed Network Flow Based Approach



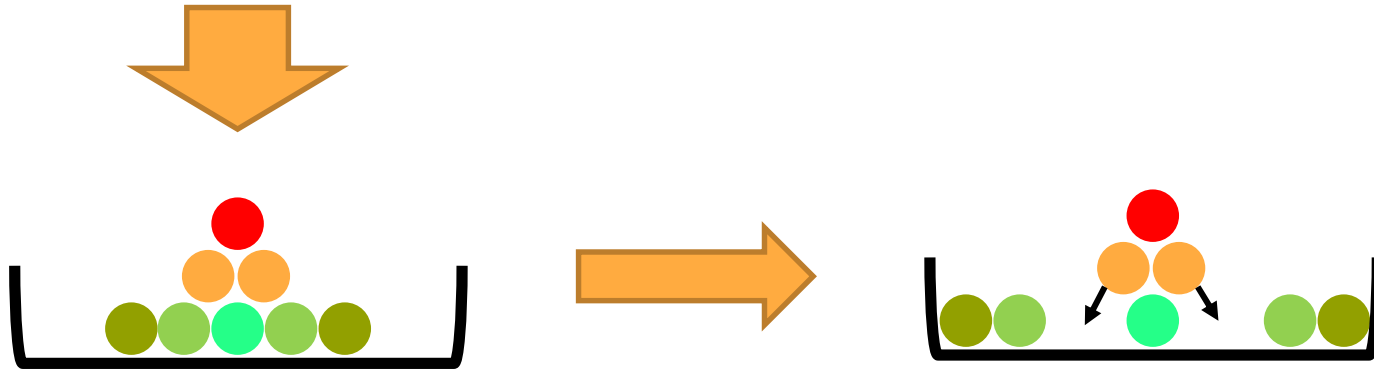
# Proposed Network Flow Based Approach



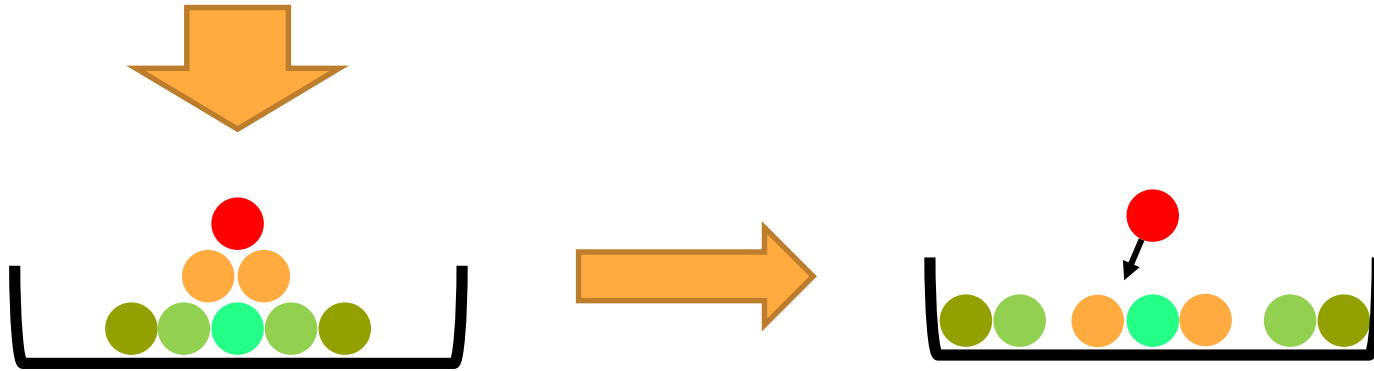
# Proposed Network Flow Based Approach



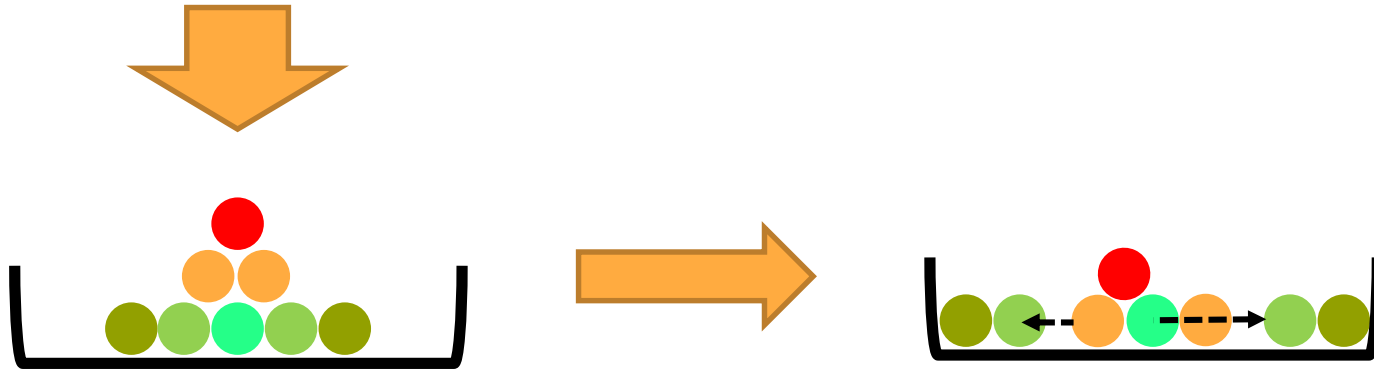
# Proposed Network Flow Based Approach



# Proposed Network Flow Based Approach



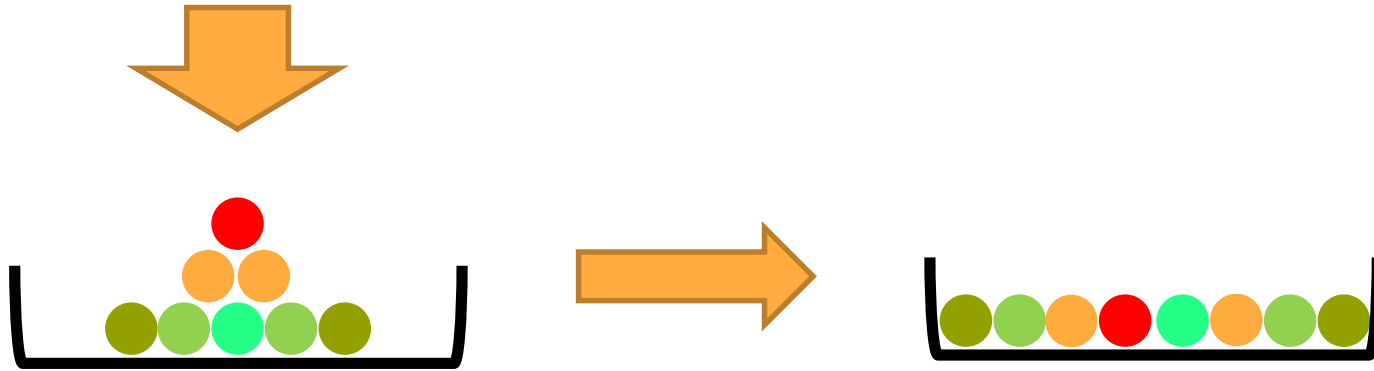
# Proposed Network Flow Based Approach



# Proposed Network Flow Based Approach

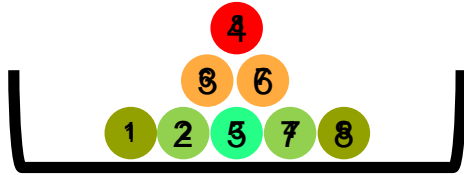


# Proposed Network Flow Based Approach





# Proposed Method vs. Traditional Methods



The order is based on x positions.



Traditional Methods Example

Max movement = 3



Proposed Method Example

Max movement = 1.5

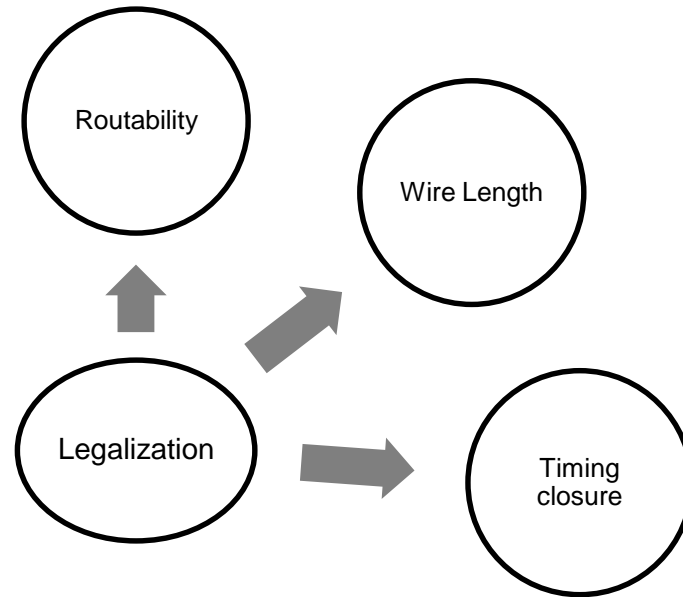
- Network Flow:

- minimizes maximum movement.

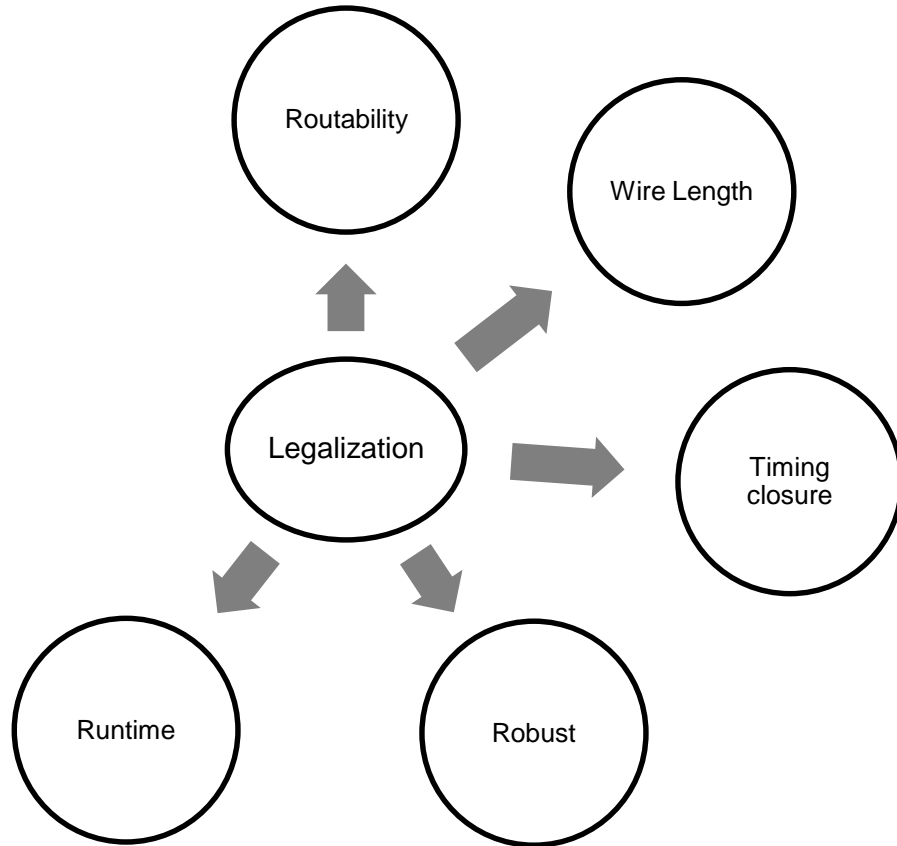
- better preserves the order.

# Legalization Challenges

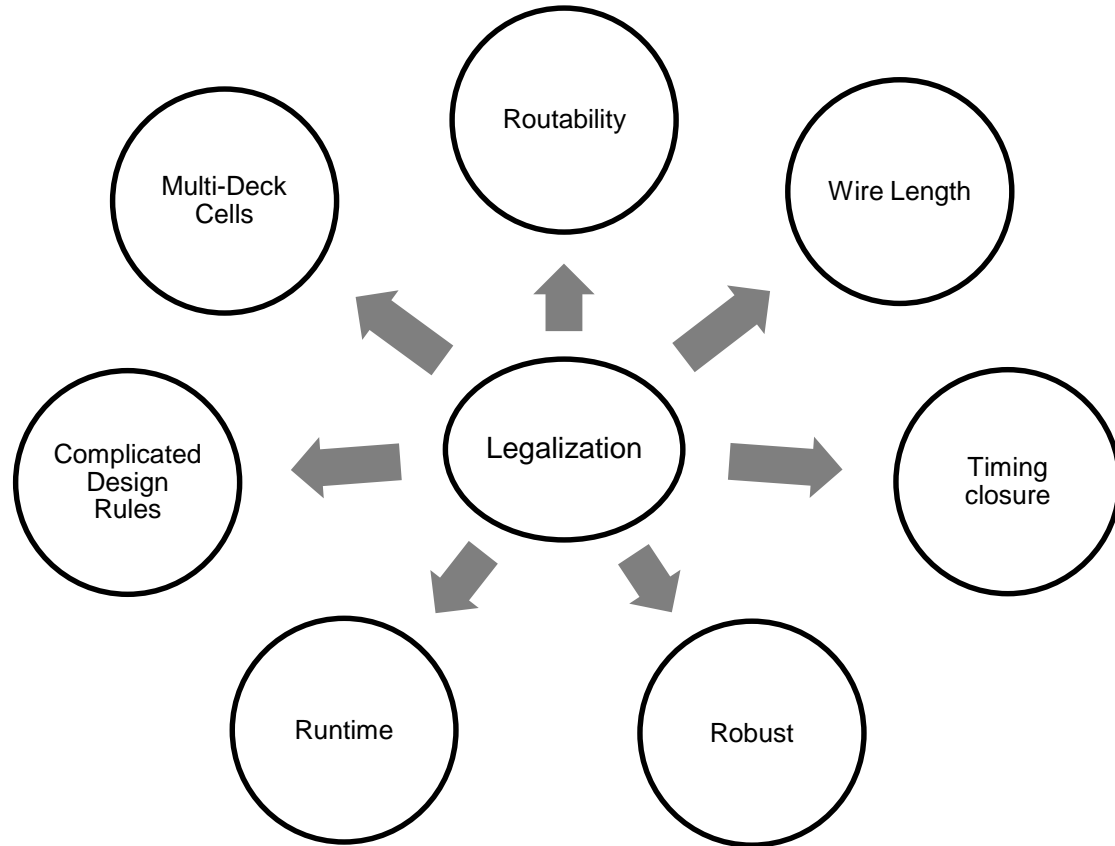
# Challenges



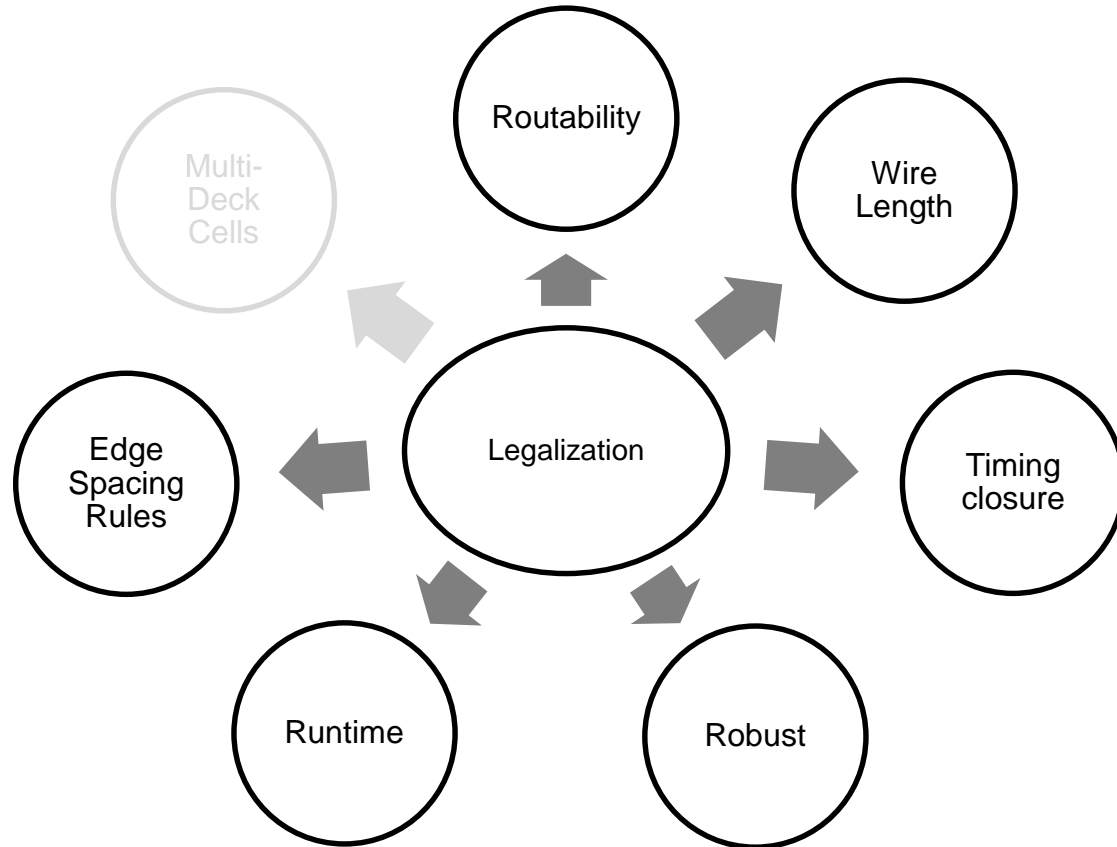
# Challenges



# Challenges

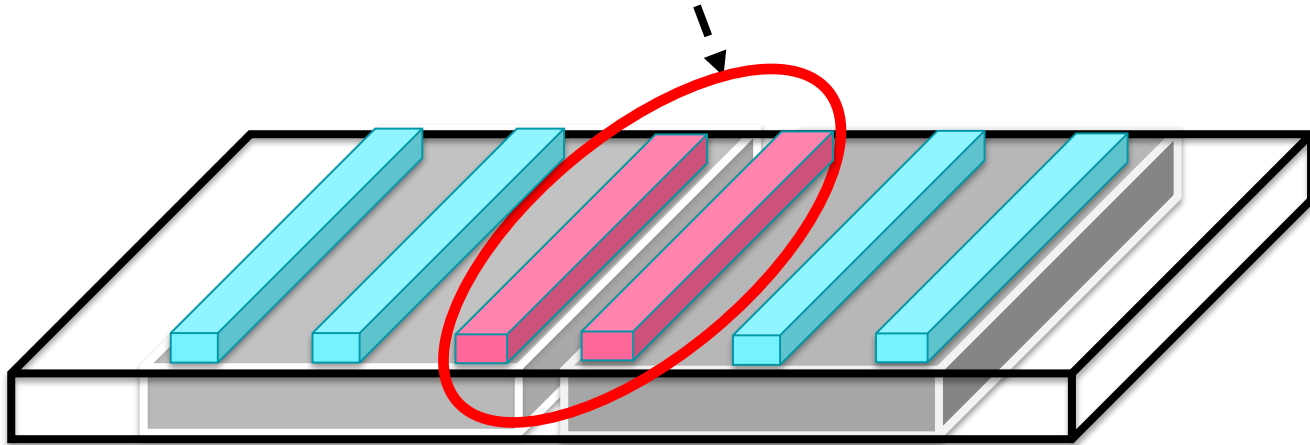


# Challenges



# Challenges – Edge Spacing Constraint

Prone to pin access and short problems



Two cells are too close to each other

# Objectives

- **Minimize**
  - **Maximum Cell Movement**
  - **Average Cell movement**

**But it is not easy to minimize the above objectives because these are two NP hard problems.**



# **Our Network-Flow Based Approach**

# Our Network-Flow Based Approach

- Inspired by the path augmentation algorithm, reference [8] (Brenner 2012).
- Objective is to flatten humps subject to the constraints.
- Set maximum movement as a constraint.
- Uses BFS to identify multiple paths subject to the maximum movement constraint.

# Our Network-Flow Based Approach

## Preparation Step

Initial solution



Build a grid of connected  
bins

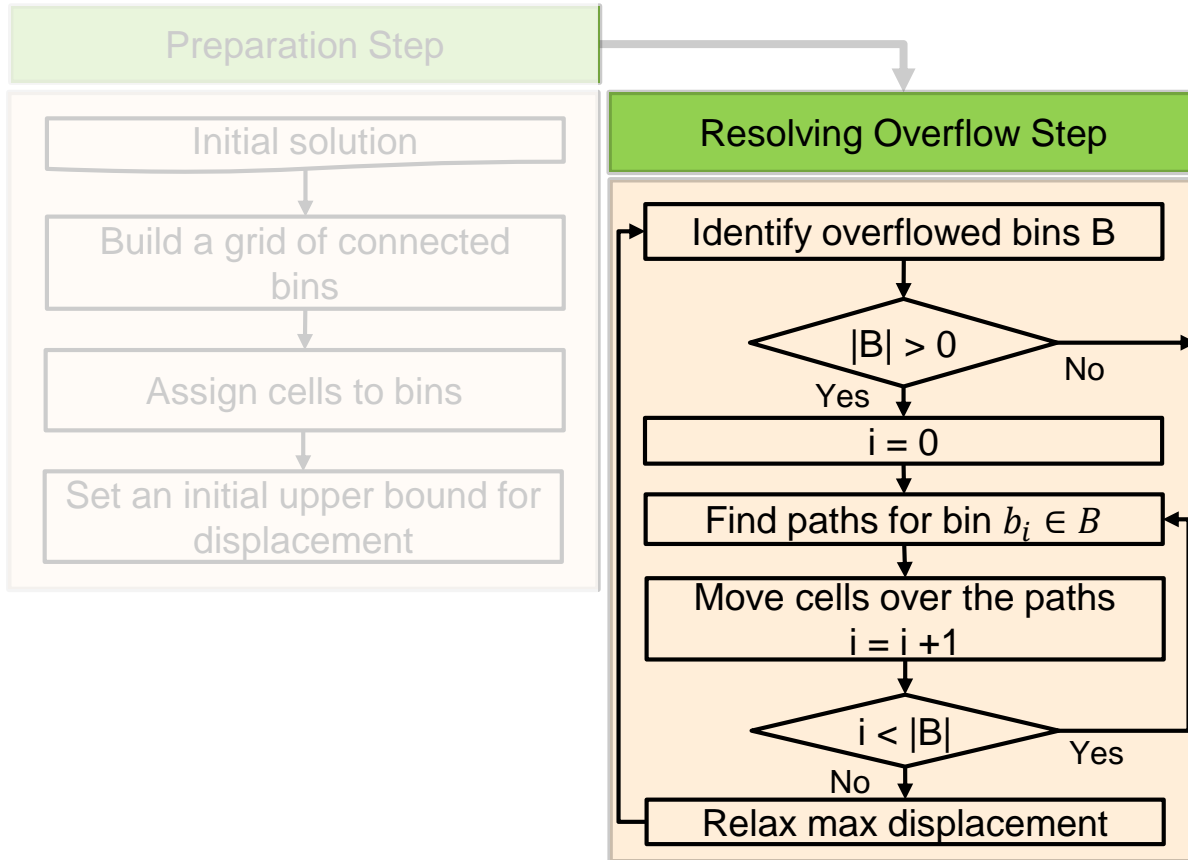


Assign cells to bins

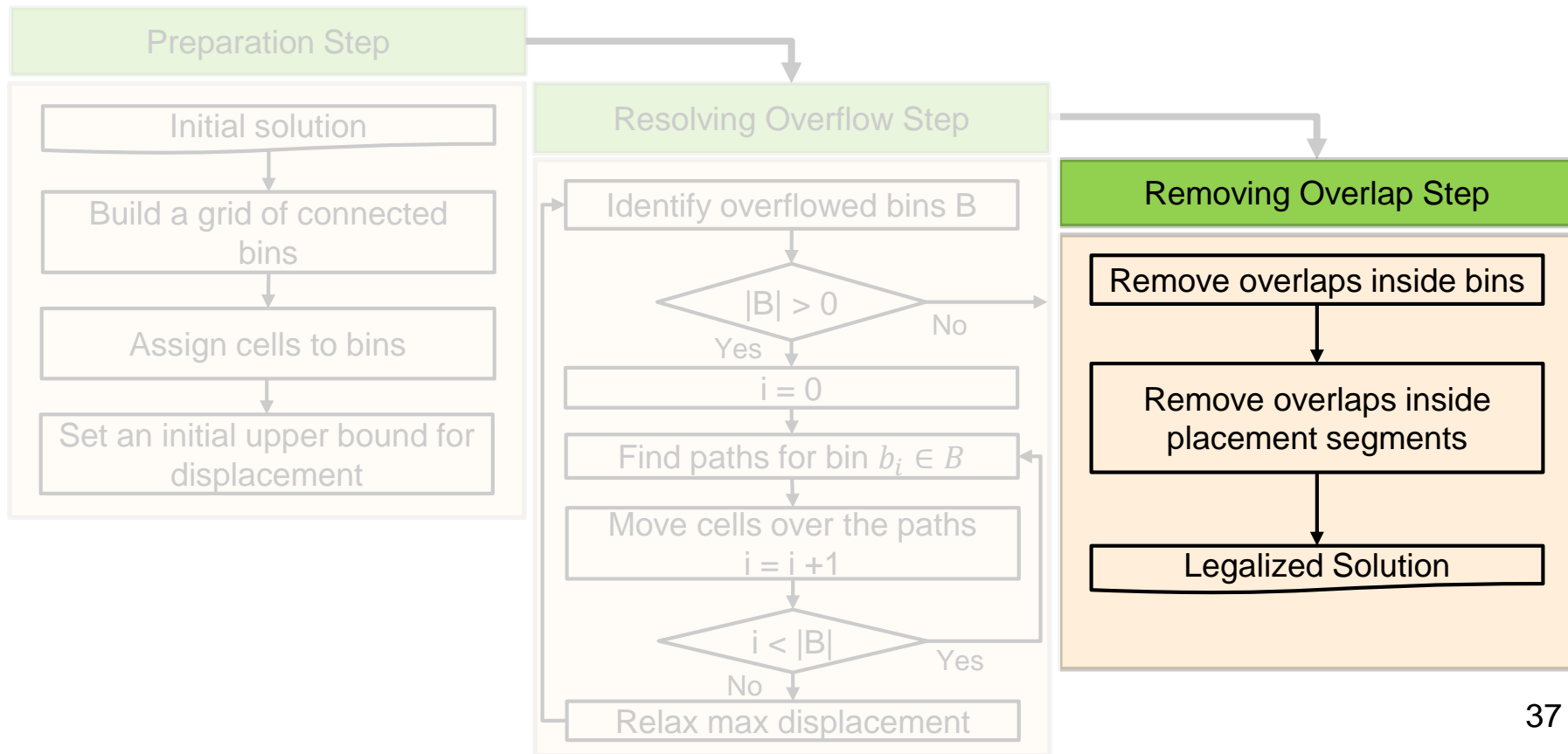


Set an initial upper bound for  
displacement

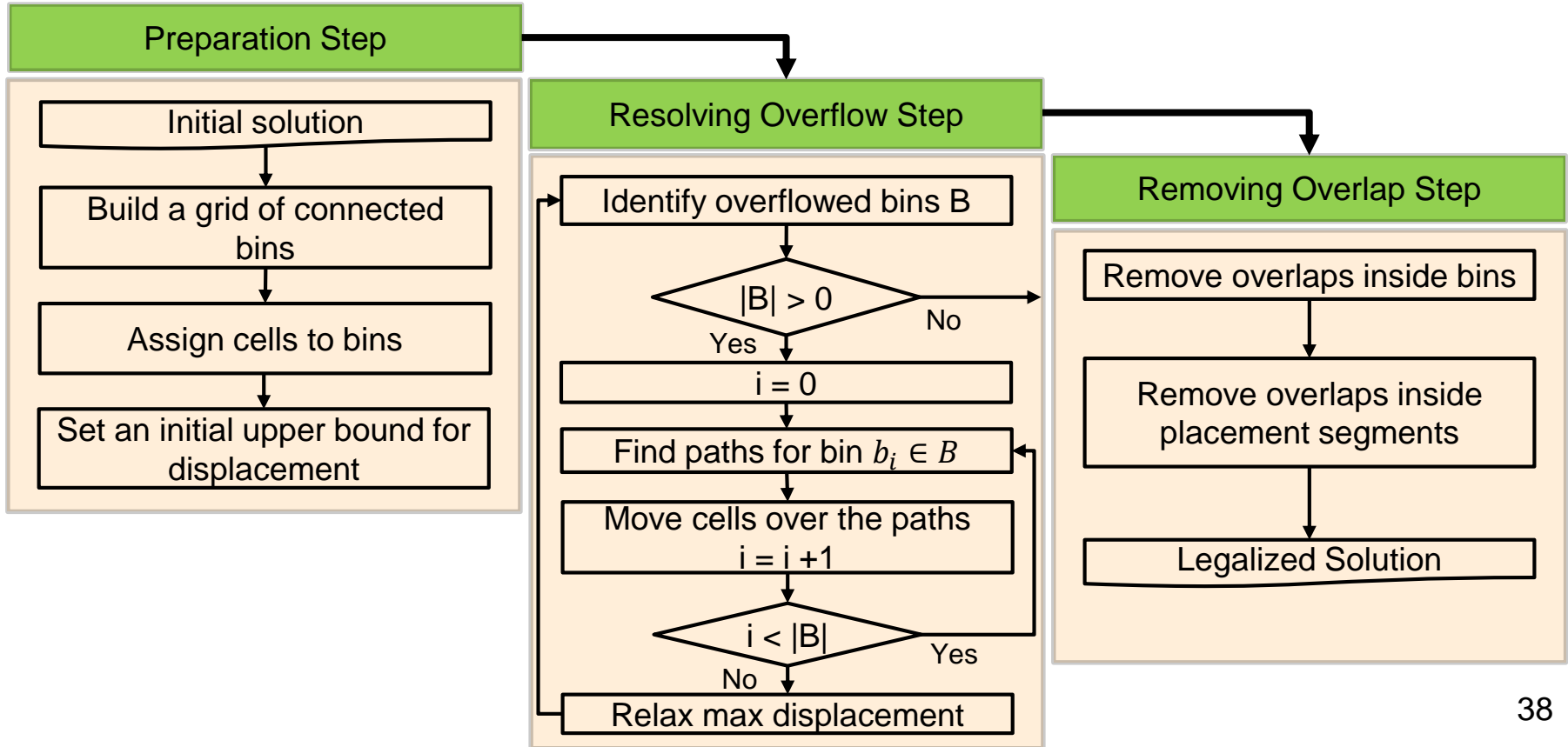
# Our Network-Flow Based Approach



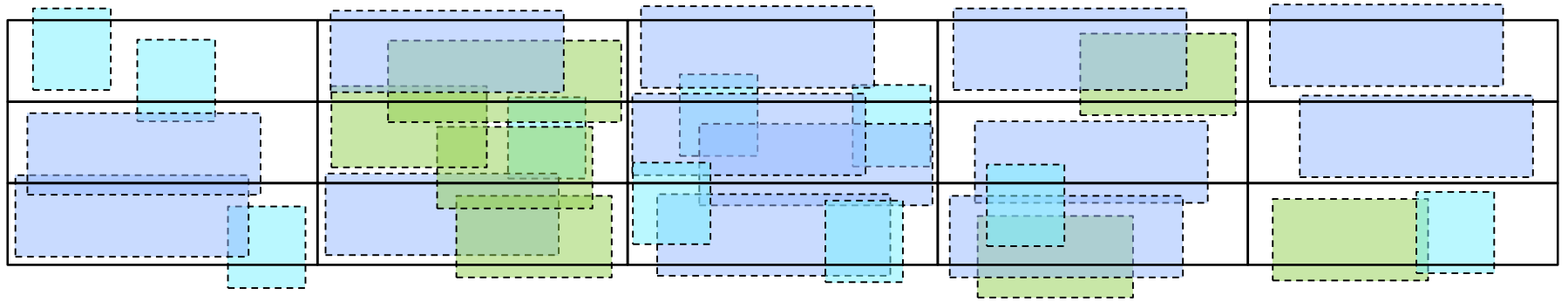
# Our Network-Flow Based Approach



# Our Network-Flow Based Approach

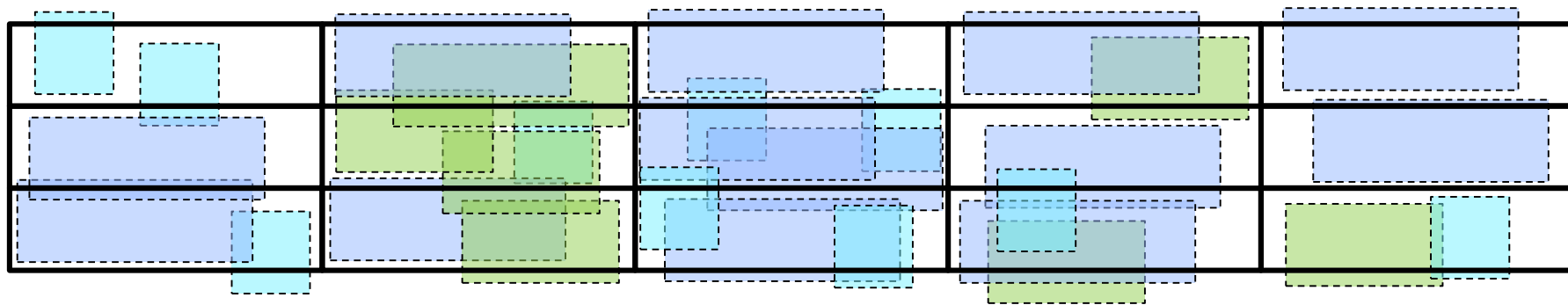


# Illustrative Example



**Initial illegal placement: off rows and overlapping cells**

# Illustrative Example (The Capacity Map)

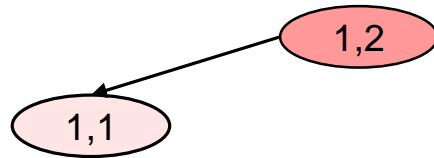


0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1
	0	1	2	3	4



# Illustrative Example (Finding Paths)

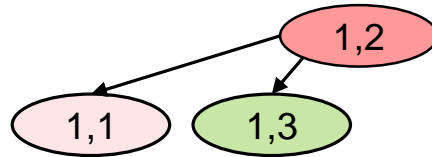
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

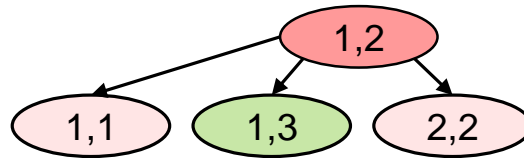
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

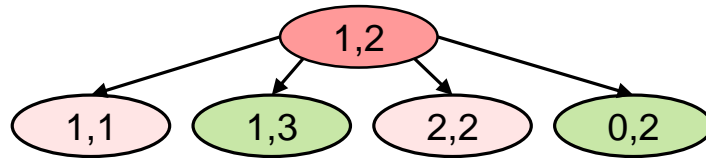
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

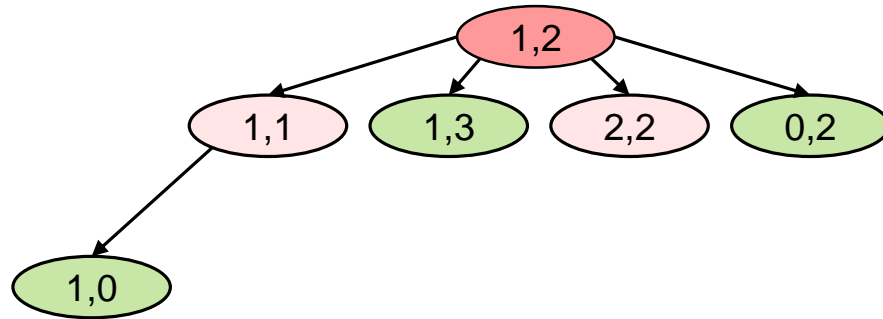
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

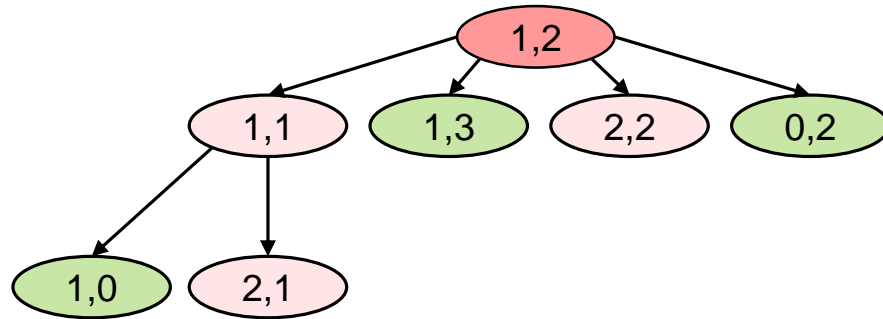
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

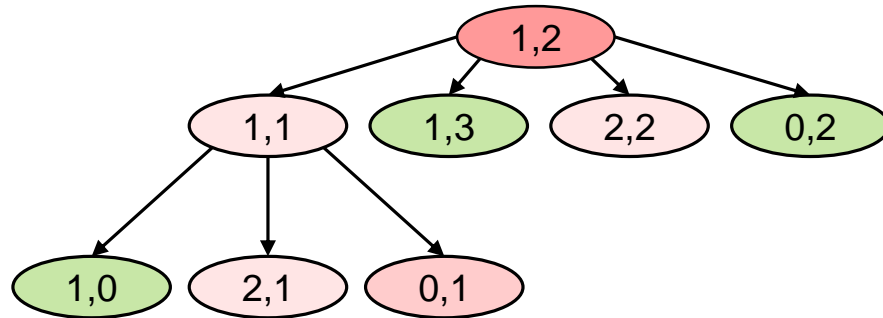
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

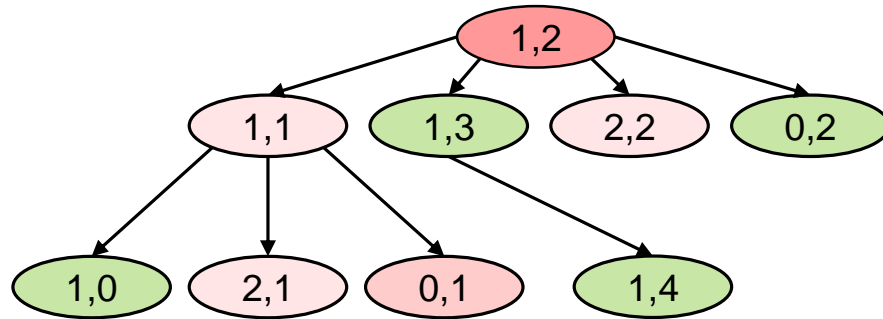
	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1



The proposed BFS-based method for finding paths for an overflowed bin.

# Illustrative Example (Finding Paths)

	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1

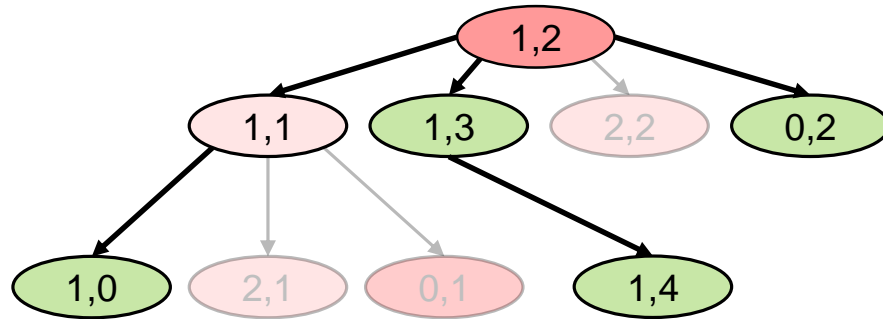


The proposed BFS-based method for finding paths for an overflowed bin.



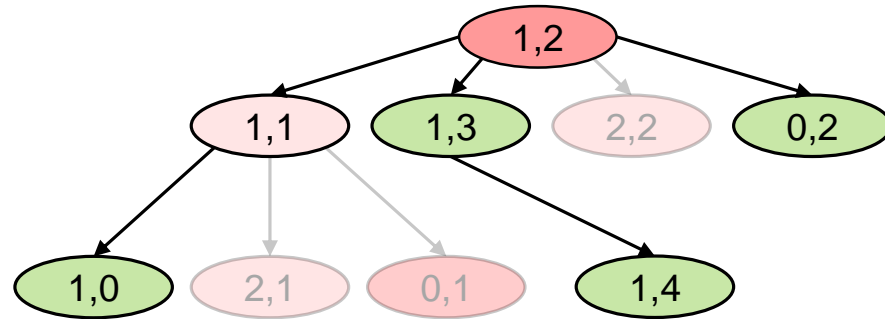
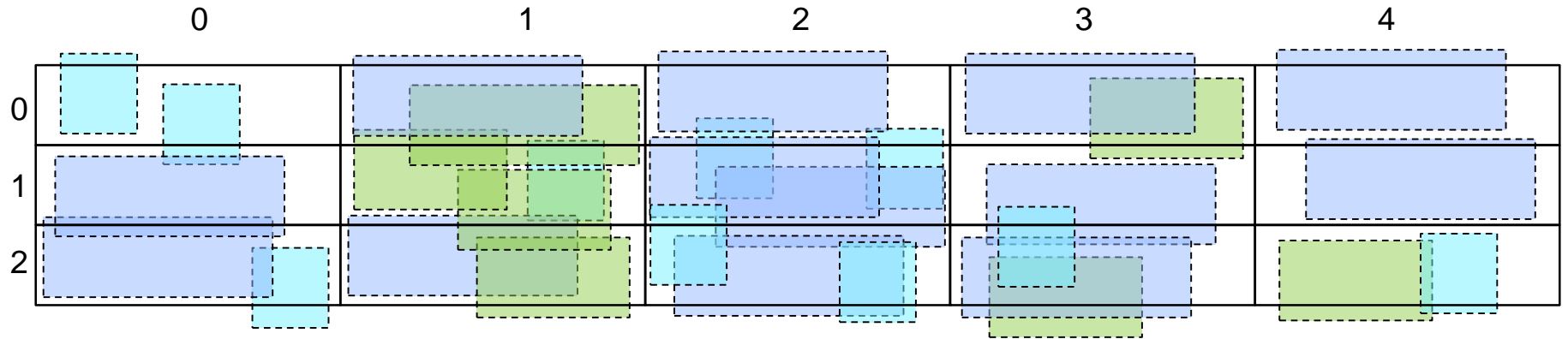
# Illustrative Example (Finding Paths)

	0	1	2	3	4
0	2	-2	1	-1	1
1	1	-1	-4	1	1
2	0	-1	-1	-2	1

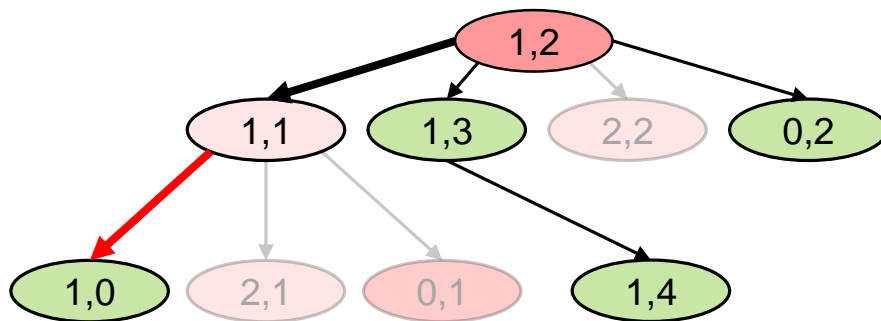
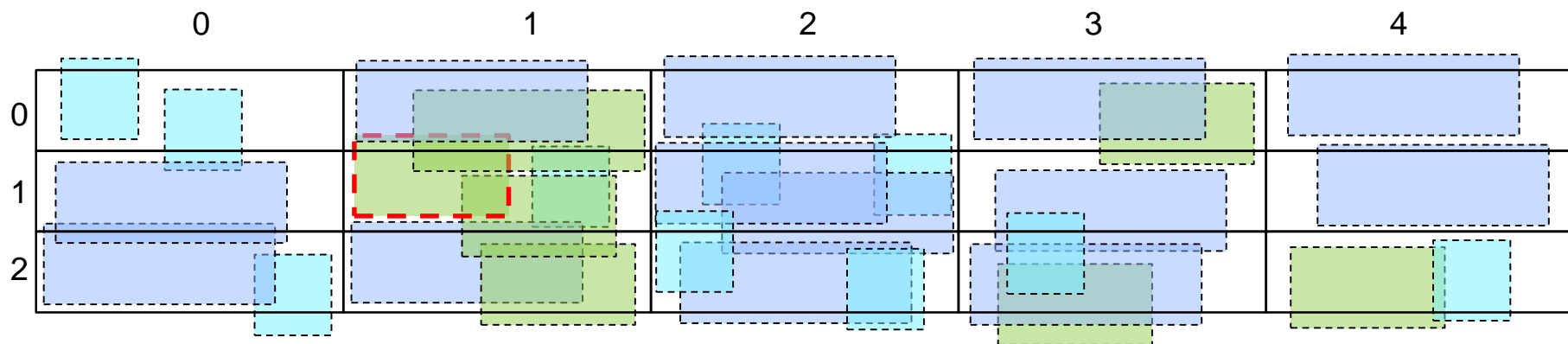


The proposed BFS-based method for finding paths for an overflowed bin.

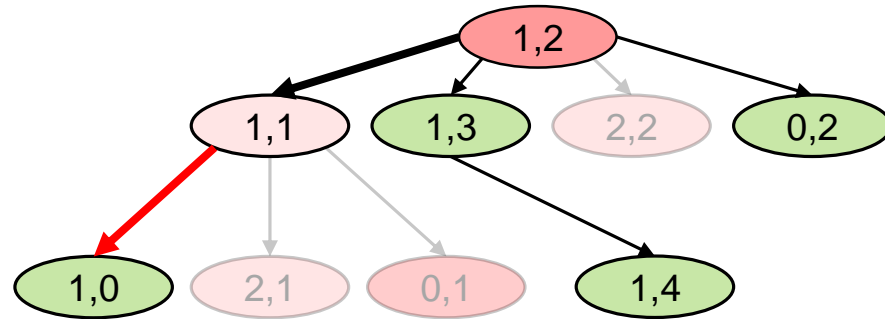
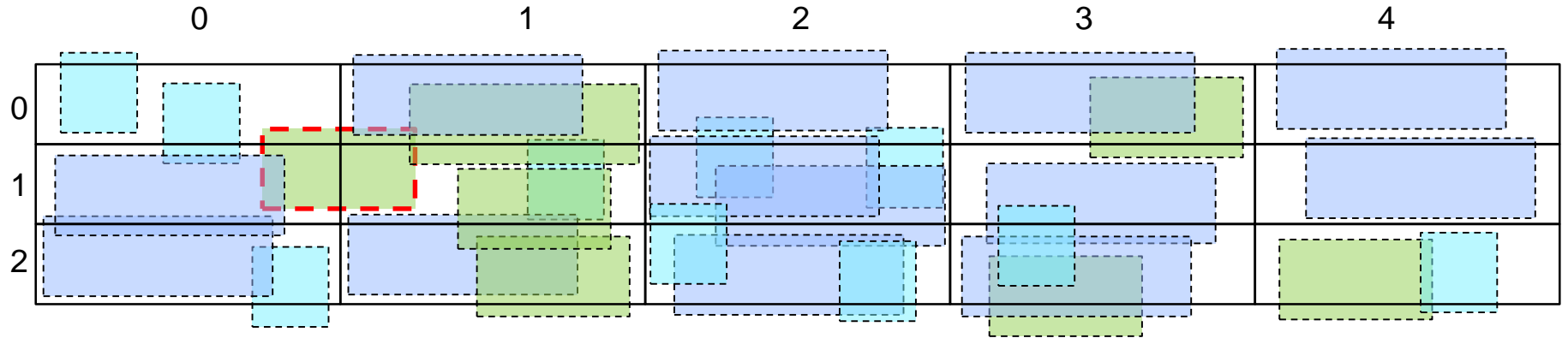
# Illustrative Example (Moving Cells)



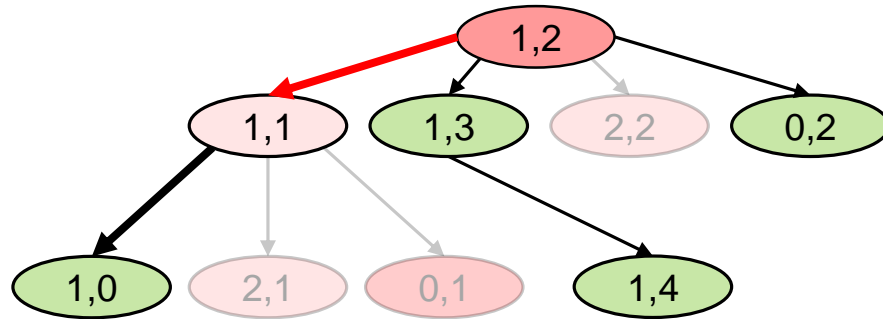
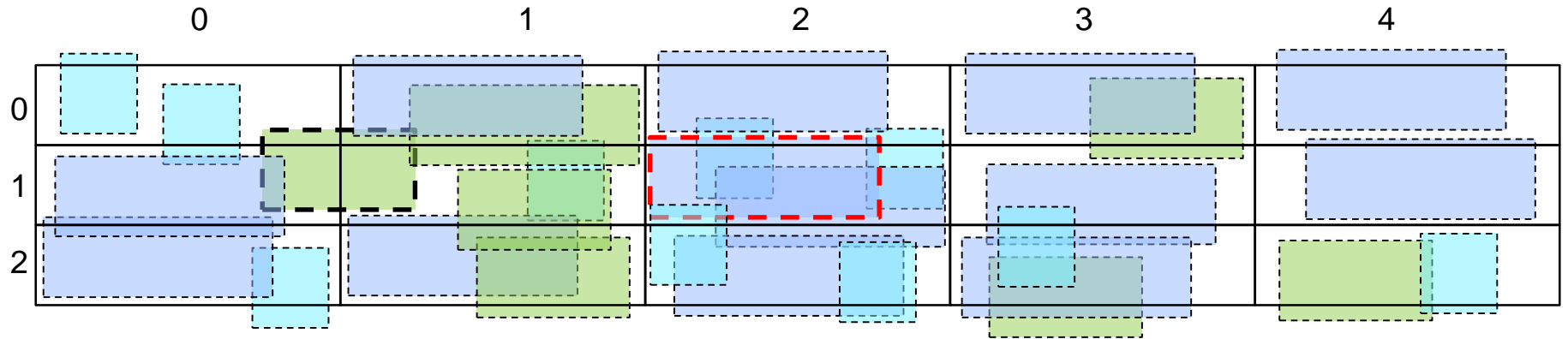
# Illustrative Example (Moving Cells)



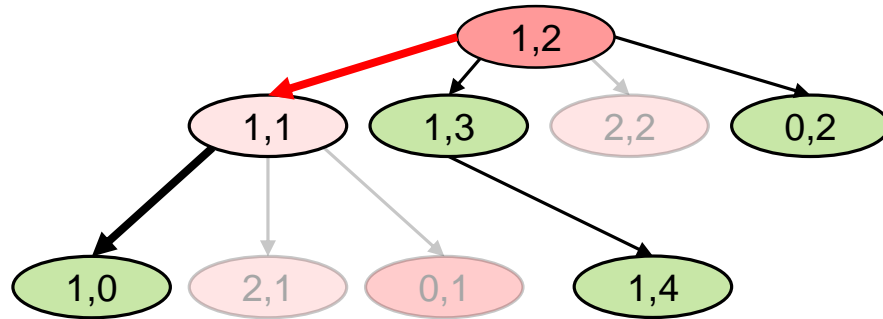
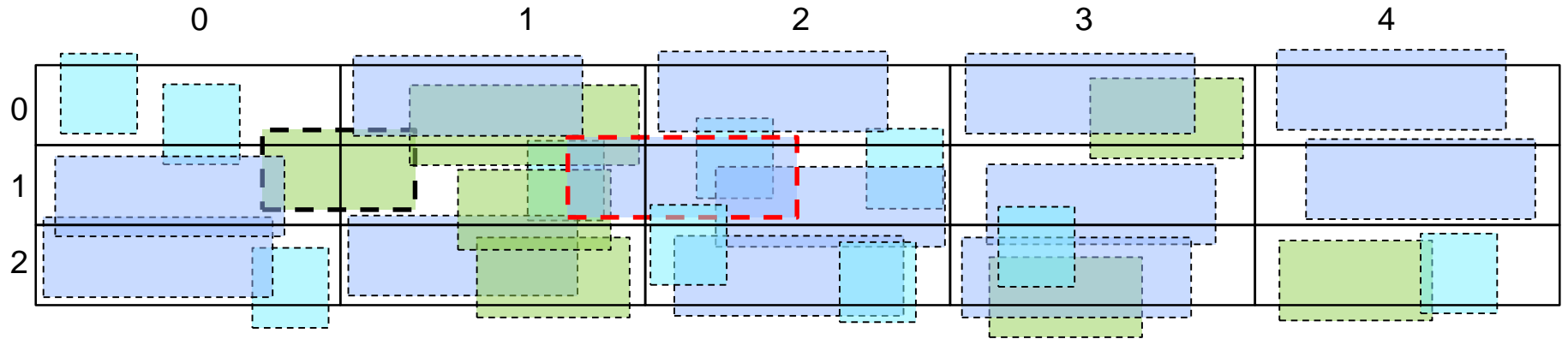
# Illustrative Example (Moving Cells)



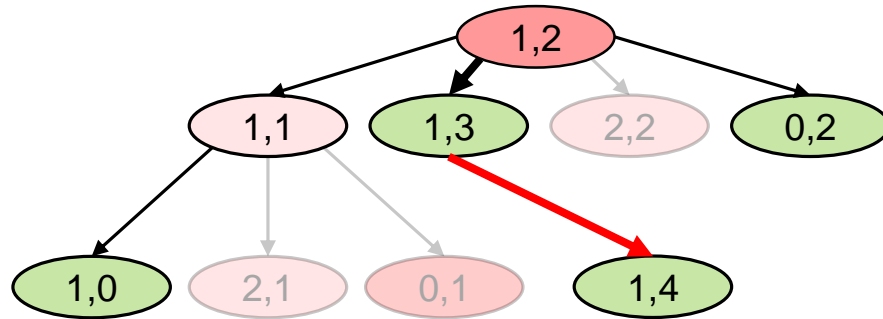
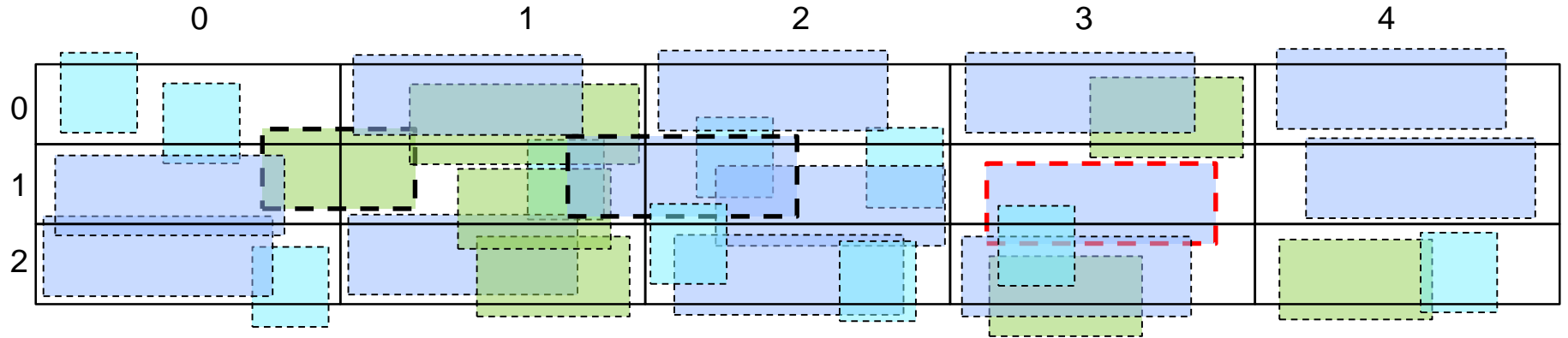
# Illustrative Example (Moving Cells)



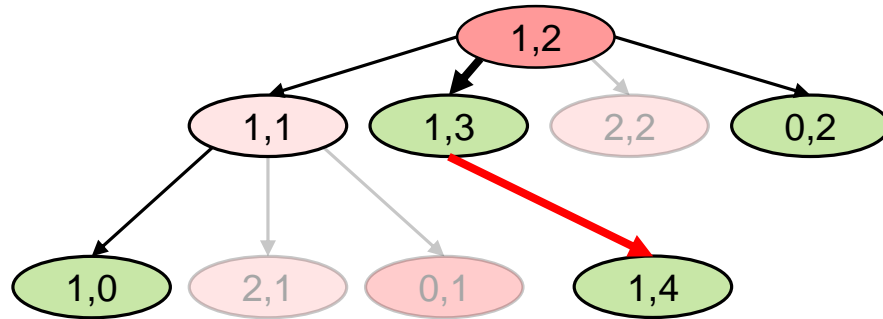
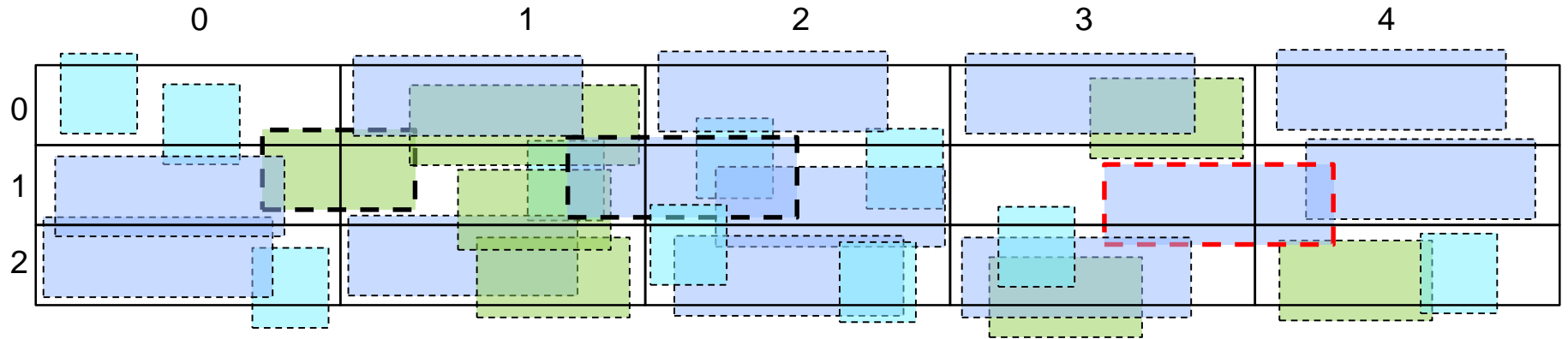
# Illustrative Example (Moving Cells)



# Illustrative Example (Moving Cells)

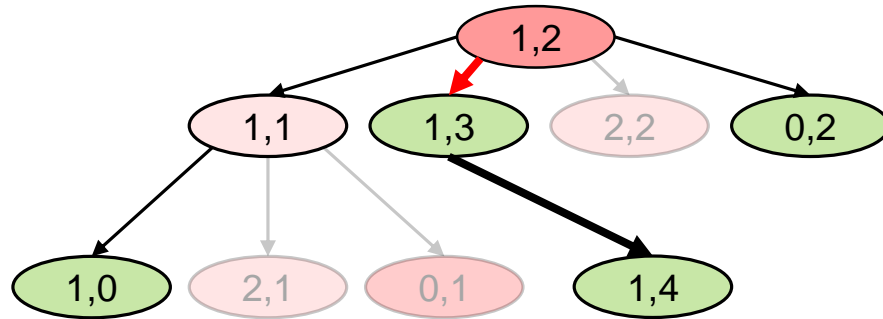
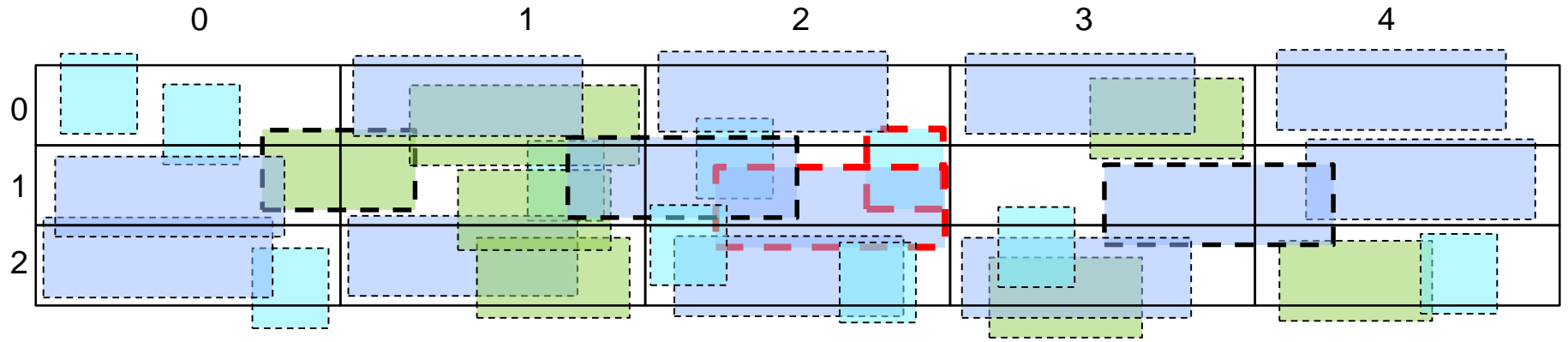


# Illustrative Example (Moving Cells)

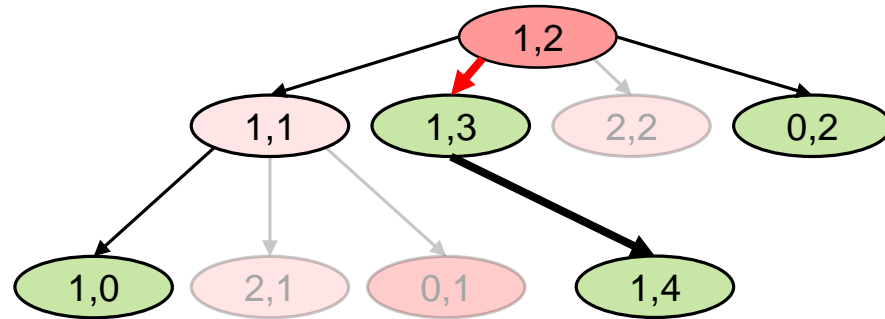
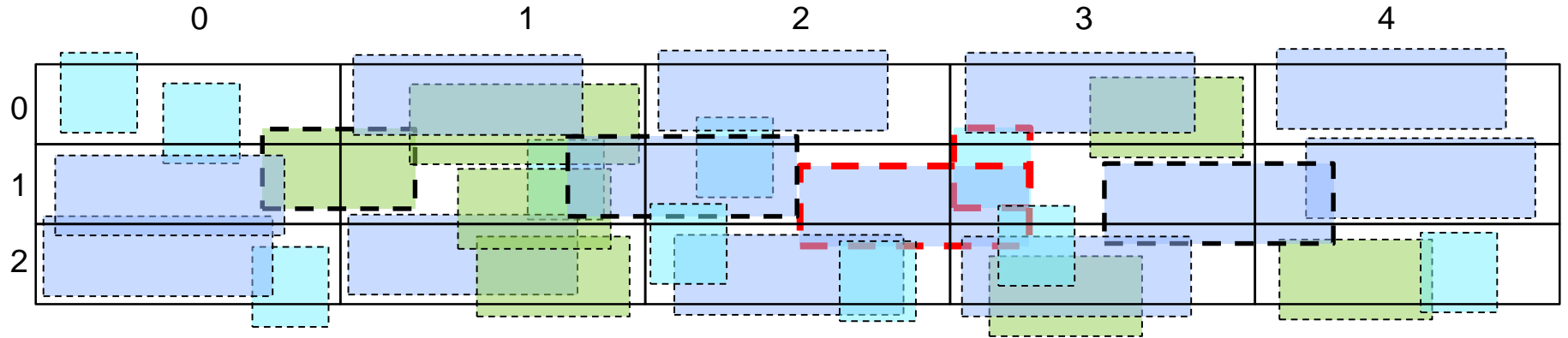




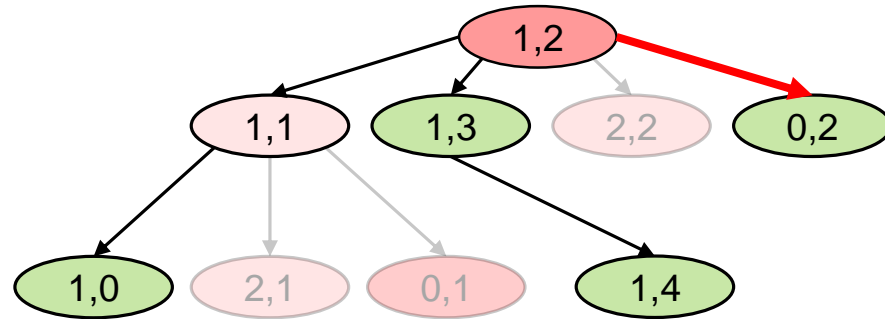
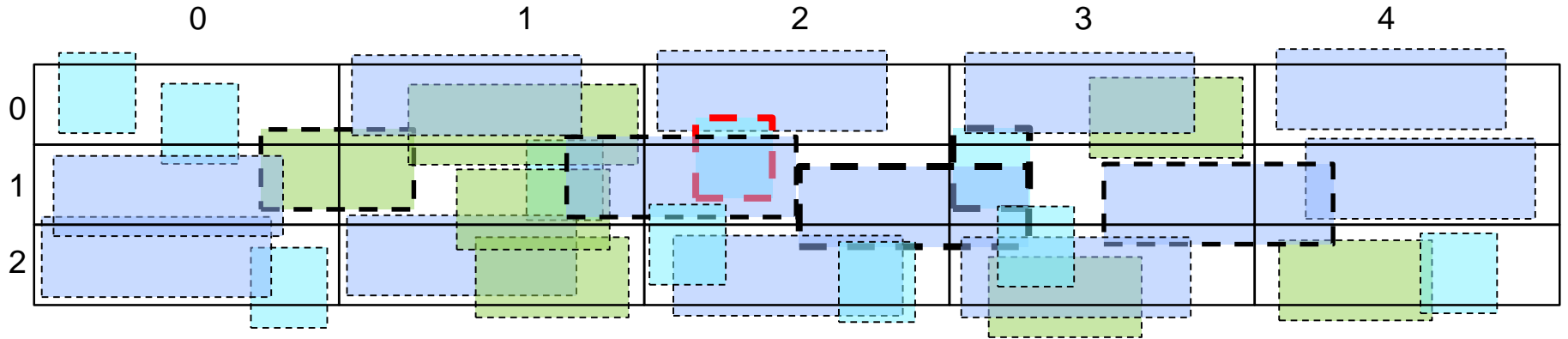
# Illustrative Example (Moving Cells)



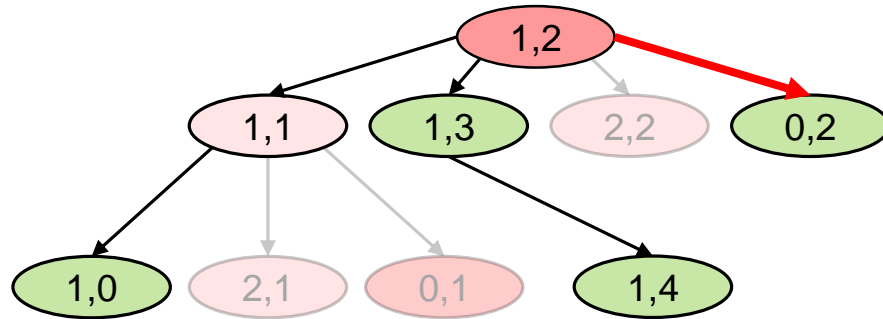
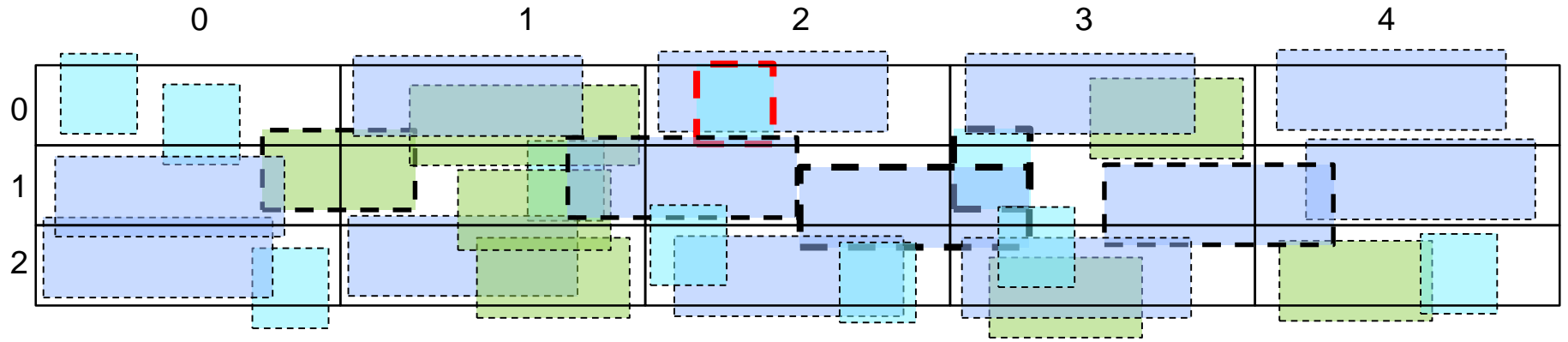
# Illustrative Example (Moving Cells)



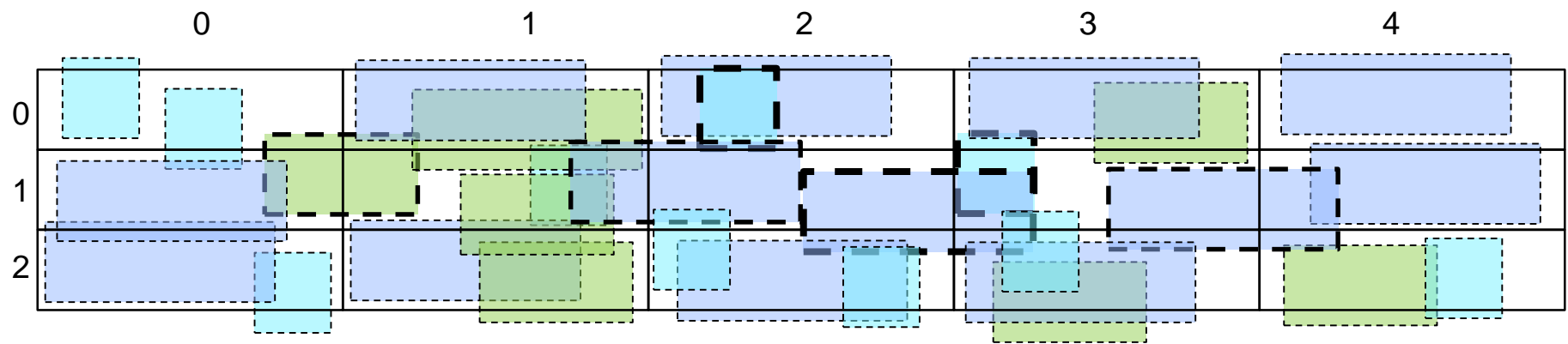
# Illustrative Example (Moving Cells)



# Illustrative Example (Moving Cells)



# Illustrative Example (The Capacity Map)



2	-2	0	-1	1
0	-1	0	0	0
0	-1	-1	-2	1

# Experimental Results

## First Set

- ISPD 2014 benchmarks.
- GP generated by Eh?Placer.
- LEF/DEF format.
- All DR rules removed.

# Experimental Results

## First Set

- ISPD 2014 benchmarks.
- GP generated by Eh?Placer.
- LEF/DEF format.
- All DR rules removed.

## Second Set

- ISPD 2014 benchmarks.
- GP generated by Eh?Placer.
- LEF/DEF format.
- **Edge spacing rules enabled.**

# Experimental Results

## First Set

- ISPD 2014 benchmarks.
- GP generated by Eh?Placer.
- LEF/DEF format.
- All DR rules removed.

## Second Set

- ISPD 2014 benchmarks.
- The same GPs.
- LEF/DEF format.
- Edge spacing rules enabled.

## Third Set

- ISPD 2006 benchmarks with movable single deck cells.
- GP generated by POLAR.
- Bookshelf format.



# Experimental Results

## First Set

- ISPD 2014 benchmarks.
- GP generated by Eh?Placer.
- LEF/DEF format.
- All DR rules removed.

## Second Set

- ISPD 2014 benchmarks.
- The same GPs.
- LEF/DEF format.
- Edge spacing rules enabled.

## Third Set

- ISPD 2006 benchmarks with movable single deck cells.
- GP generated by POLAR.
- Bookshelf format.

# The ISPD 2014 Benchmarks

Global Placements generated by Eh?Placer

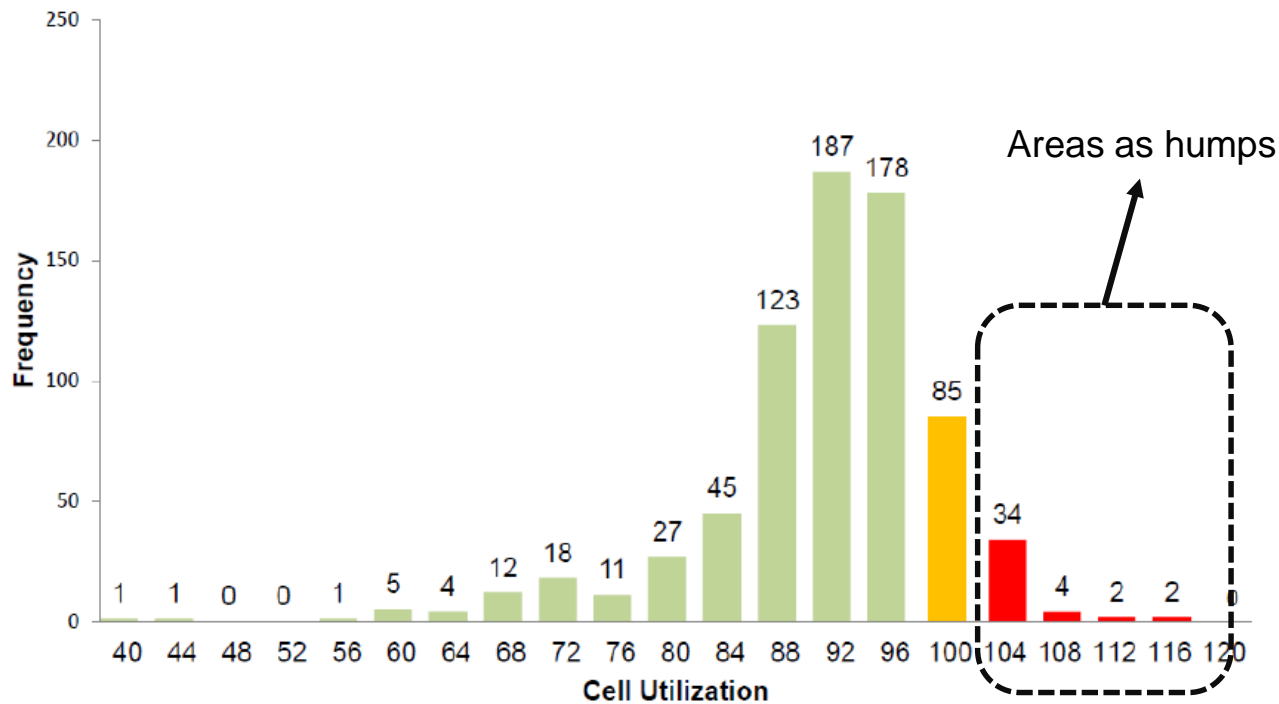
ISPD 2014 Benchmark	#Cells	#Fixed	Utilization
mgc_des_perf_1	112,644	0	<b>0.91</b>
mgc_des_perf_2	112,644	0	0.85
mgc_edit_dist_1	130,661	0	0.40
mgc_edit_dist_2	130,661	0	0.43
mgc_fft	32,281	0	0.84
mgc_matrix_mult	155,325	0	0.80
mgc_pci_bridge32_1	30,675	0	0.84
mgc_superblue11	925616	<b>1458</b>	0.41
mgc_superblue12	<b>1286948</b>	89	0.44
mgc_superblue16	680450	419	0.46

# The ISPD 2006 Benchmarks

## Global Placements generated by POLAR

ISPD 2006 Benchmark	#Cells	#Fixed	Utilization
newblue3	482833	11178	0.26
newblue4	642717	3422	0.46
newblue5	1228177	4881	0.50
newblue6	1248150	6889	0.39
newblue7	<b>2481372</b>	<b>26582</b>	0.49

# Sample Illegal Placement from ISPD 2014



(a) des\_perf\_1

# Results on the ISPD 2014 (Set 1)

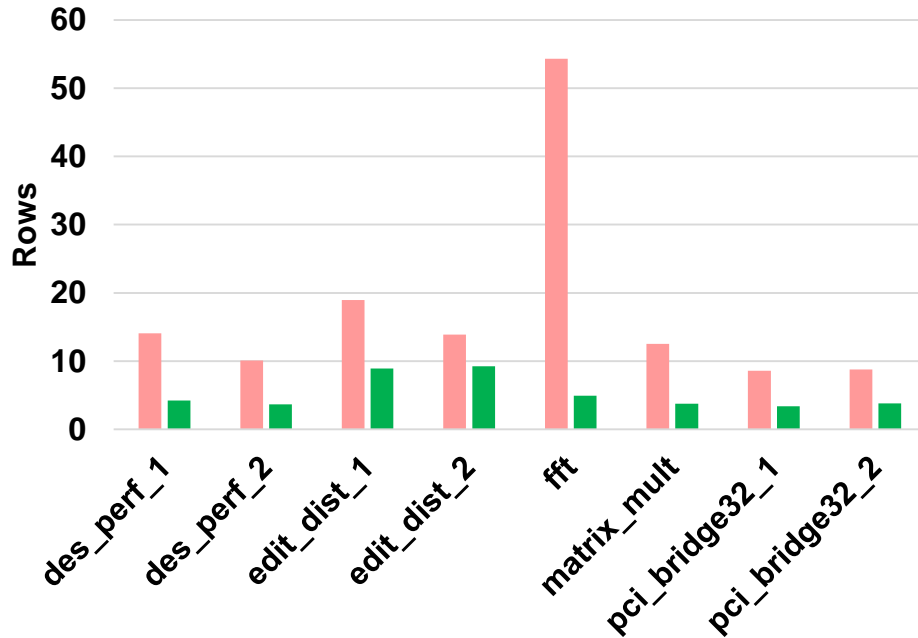
Ours vs Eh?Placer (EP) and RippleDP (RDP)

Benchmark	HPWL Increase (%)			Avg. Disp (rows)			Max. Disp (rows)			Runtime (sec)		
	RDP	EP	Ours	RDP	EP	Ours	RDP	EP	Ours	RDP	EP	Ours
des_perf_1	<b>5.4</b>	14.2	6.1	3.6	1.3	<b>0.8</b>	10.0	11.1	<b>4.3</b>	89.1	5.5	<b>2.5</b>
des_perf_2	<b>4.5</b>	12.7	5.9	3.6	1.2	<b>0.8</b>	10.0	13.2	<b>3.7</b>	80.1	5.1	<b>2.1</b>
edit_dist_1	-	5.8	<b>3.3</b>	-	1.2	<b>1.1</b>	--	17.8	<b>8.3</b>	--	5.9	<b>1.0</b>
edit_dist_2	-	5.7	<b>3.4</b>	-	1.2	<b>1.1</b>	--	12.2	<b>7.8</b>	--	6.1	<b>1.1</b>
fft	10.4	17.5	<b>8.5</b>	3.8	1.3	<b>0.9</b>	10.0	17.9	<b>3.9</b>	10.9	1.2	<b>0.4</b>
matrix_mult	9.2	15.0	<b>7.9</b>	3.8	1.2	<b>0.8</b>	10.0	12.5	<b>3.8</b>	52.8	7.8	<b>3.0</b>
pci_bridge32_1	<b>12.3</b>	25.2	12.7	4.0	1.2	<b>0.8</b>	10.0	8.7	<b>3.0</b>	17.8	0.9	<b>0.3</b>
pci_bridge32_2	<b>10.4</b>	23.9	12.5	4.1	1.1	<b>0.8</b>	10.0	8.2	<b>3.8</b>	17.1	0.8	<b>0.3</b>
superblue11	-	32.4	<b>4.3</b>	-	21.2	<b>3.0</b>	--	951.6	<b>388.6</b>	--	3088.1	<b>154.7</b>
superblue12	-	26.6	<b>4.8</b>	-	6.6	<b>1.9</b>	--	428.0	<b>180.6</b>	--	1537.5	<b>112.1</b>
superblue16	-	18.1	<b>4.0</b>	-	10.6	<b>2.6</b>	--	666.9	<b>259.0</b>	--	2123.9	<b>85.5</b>
Norm	<b>1.0</b>	2.7	<b>1.0</b>	4.6	3.3	<b>1.0</b>	2.7	2.5	<b>1.0</b>	31.1	18.7	<b>1.0</b>

# Results on the ISPD 2014 (Set 2)

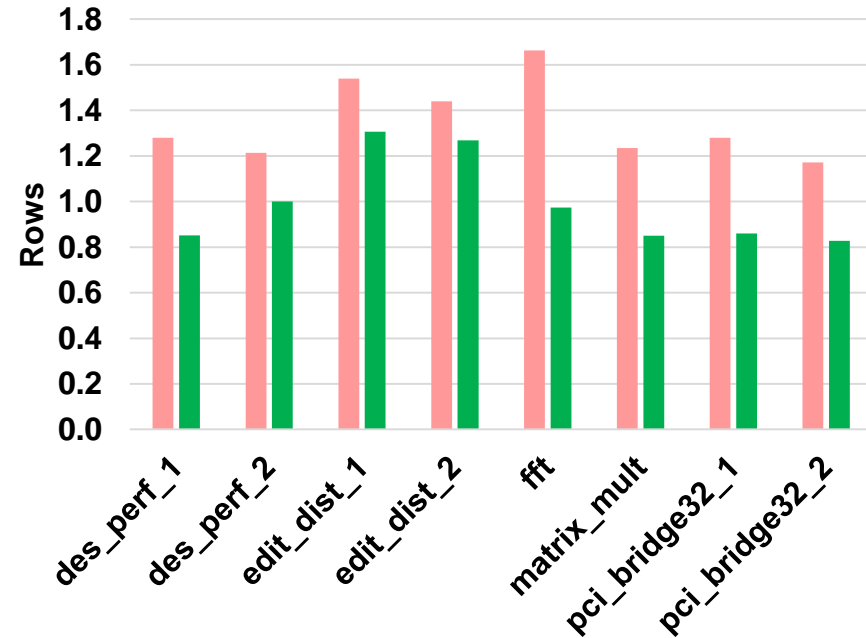
Max. Displacement

Eh?Placer Ours



Avg. Displacement

Eh?Placer Ours

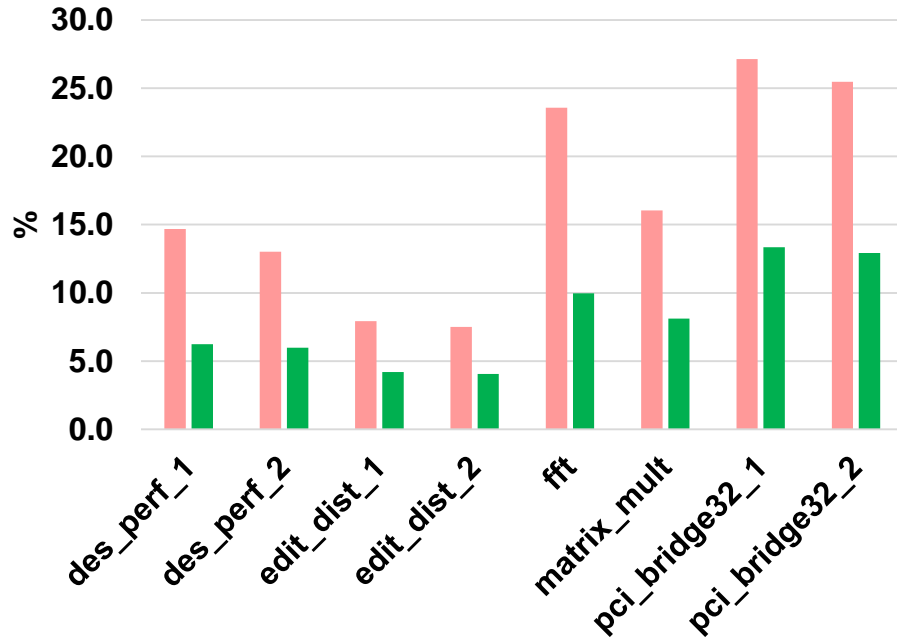


Edge spacing rules were Enabled.

# Results on the ISPD 2014 (Set 2)

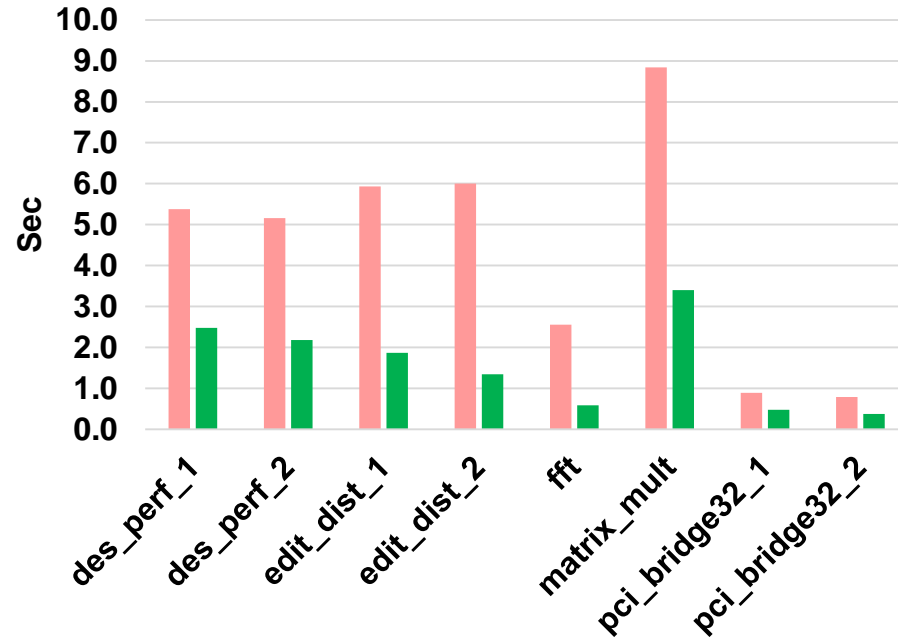
## HPWL Increase

■ Eh?Placer ■ Ours



## Runtime

■ Eh?Placer ■ Ours

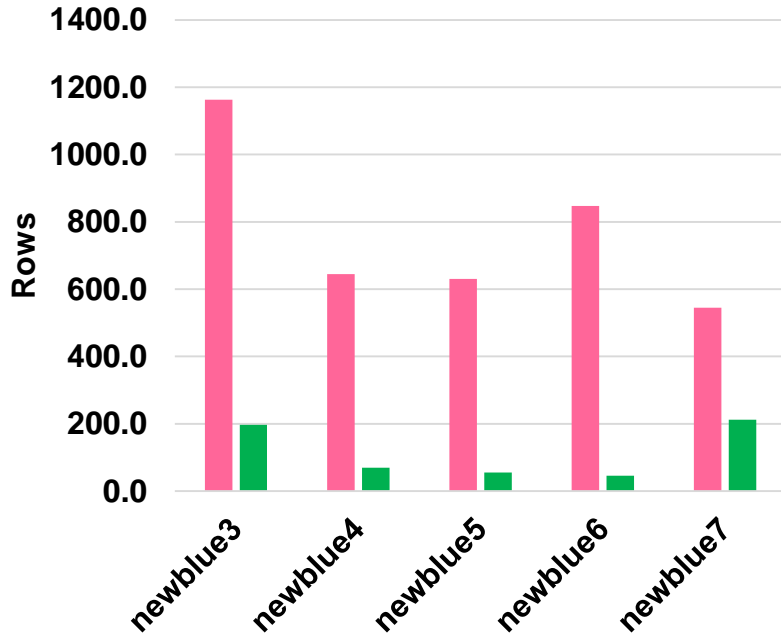


Edge spacing rules were Enabled.

# Results on the ISPD 2006 (Set 3)

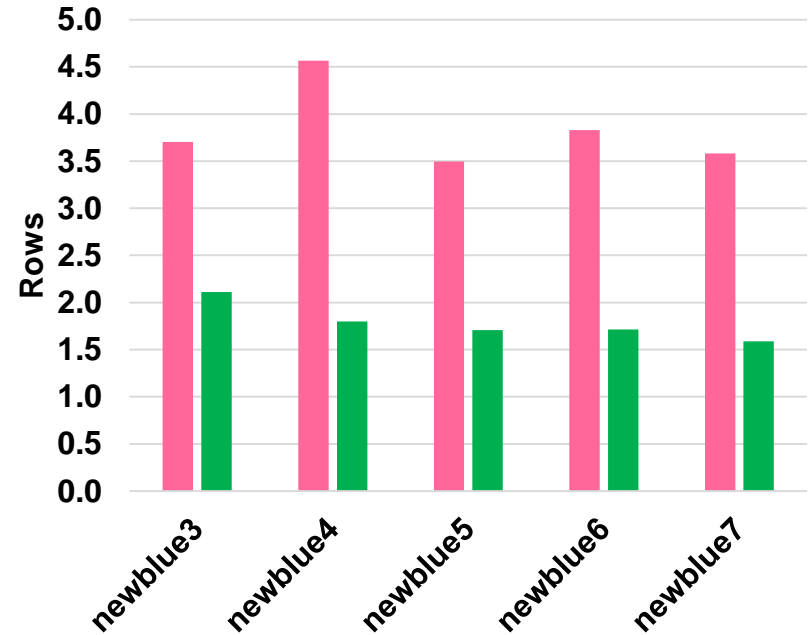
## Max. Displacement

FastPlace Ours



## Avg. Displacement

FastDP Ours

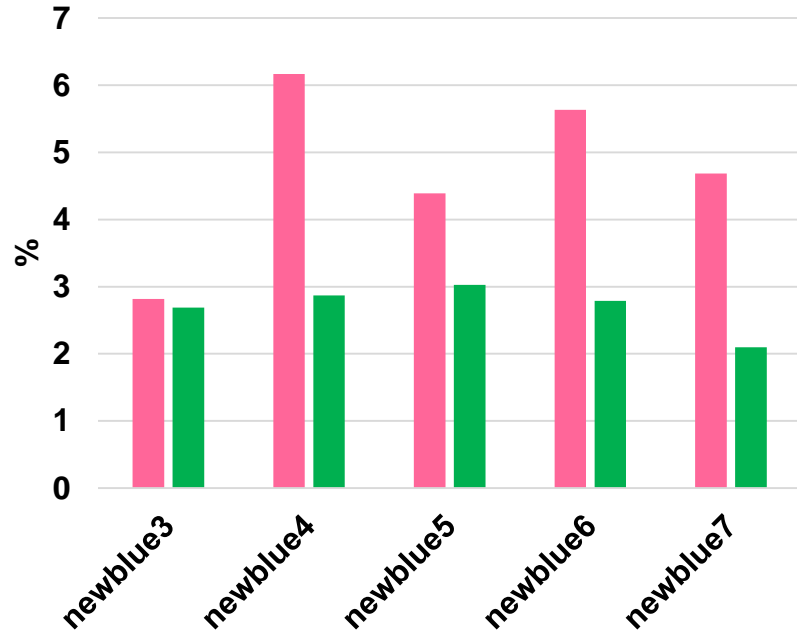




# Results on the ISPD 2006 (Set 3)

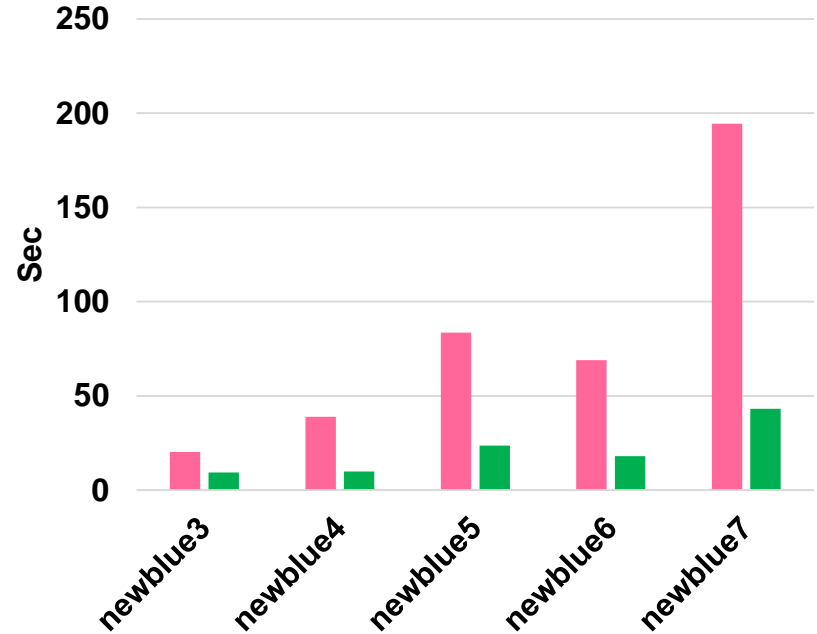
## HPWL Increase

FastPlace Ours

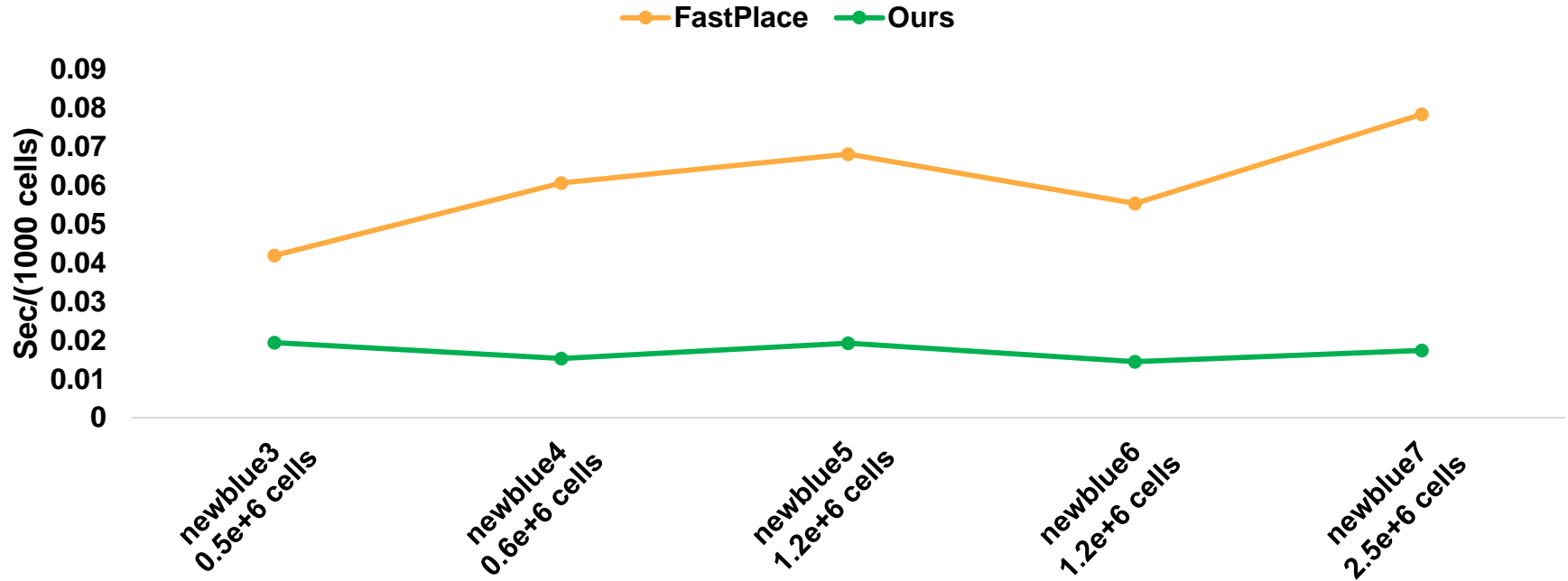


## Runtime

FastPlace Ours



# Scalability and Robustness



Our approach is scalable and robust with respect to design size and floorplan complexity

# Conclusion

- The proposed legalizer results in
  - less maximum cell movement.
  - less average cell movement.
- The proposed legalizer is
  - fast.
  - scalable and robust (independent of the floorplan complexity).

# A Fast, Robust Network Flow-based Standard-Cell Legalization Method for Minimizing Maximum Movement

Nima Karimpour Darav  
Laleh Behjat

Ismail Bustany

Andrew Kennings

