Graph-Based Subfield Scheduling for Electron-Beam Photomask Fabrication

Shao-Yun Fang, Wei-Yu Chen and Yao-Wen Chang

Mar 26, 2012

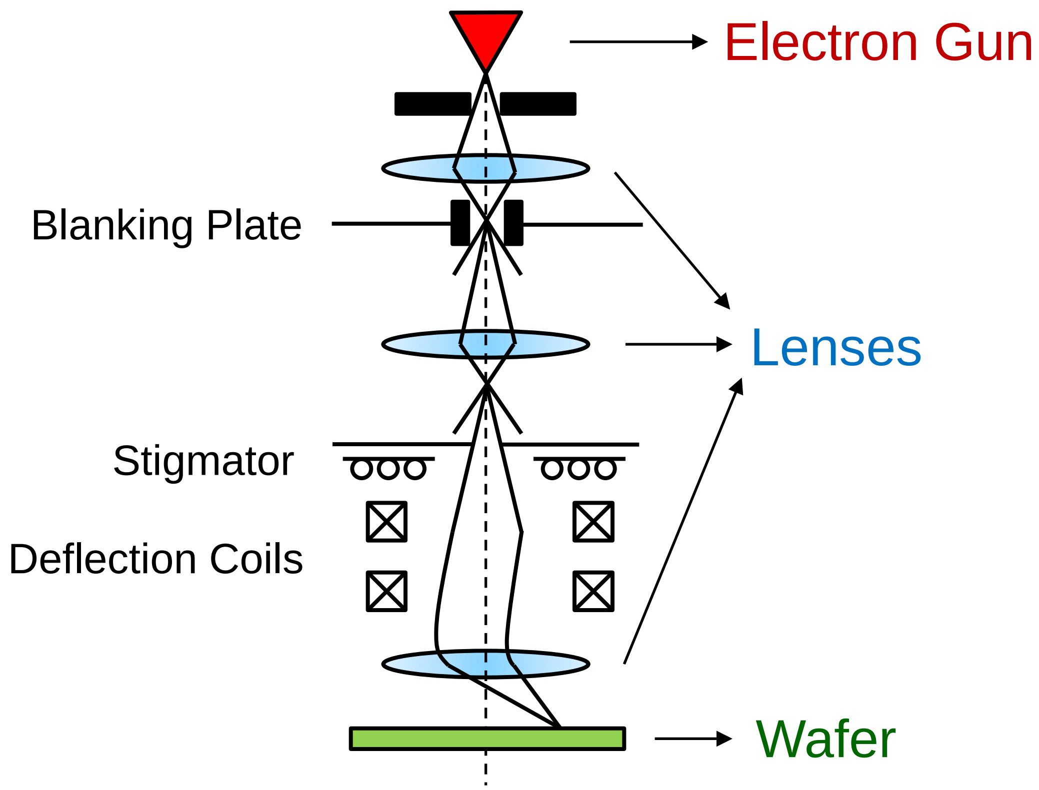
Graduate Institute of Electronics Engineering
Department of Electrical Engineering
National Taiwan University
Taiwan

Outline

- Introduction
- Preliminaries and Problem Formulation
- An Exact MSTSP Algorithm for a Special Case
- Graph-Based Subfield Scheduling Algorithm
- Experimental Results and Conclusion

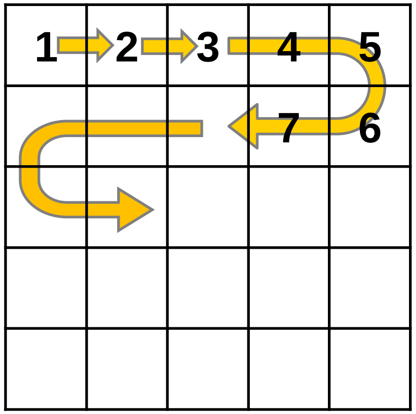
Electron Beam Lithography

- Electron beam lithography is one of the most expected next generation lithography technologies
  - Can avoid suffering from the diffraction limitation of light
  - Can define very fine patterns

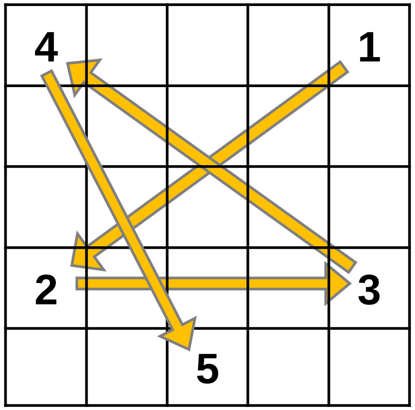


Subfield Scheduling

- **Contiguously sequential writing**: the writing proceeds in order from one subfield to the next adjacent subfield
  - High voltage beams with contiguously sequential writing cause resist heating effects and critical dimension (CD) distortion
- **Subfield scheduling**: the writing proceeds in a non-contiguously-sequential way to avoid successive writing [Babin et al., SPIE'03]



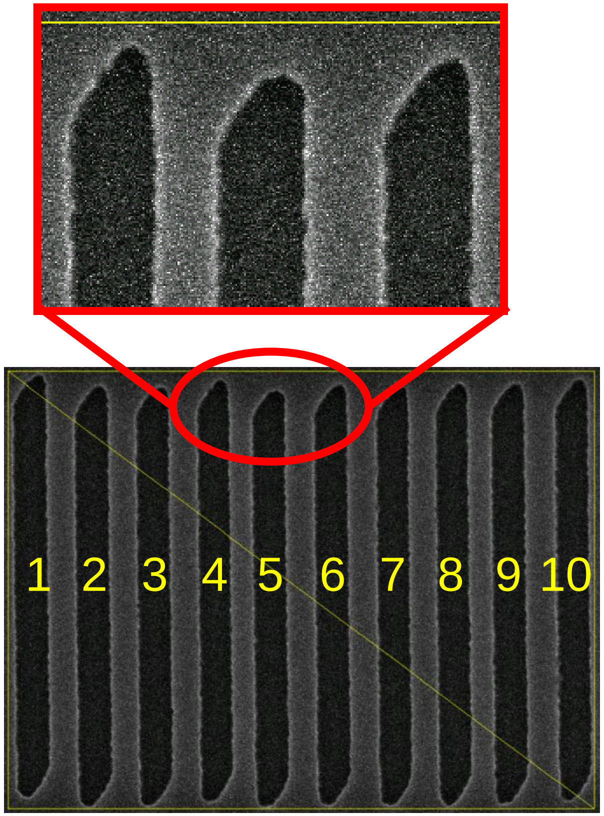
Contiguously Sequential Writing



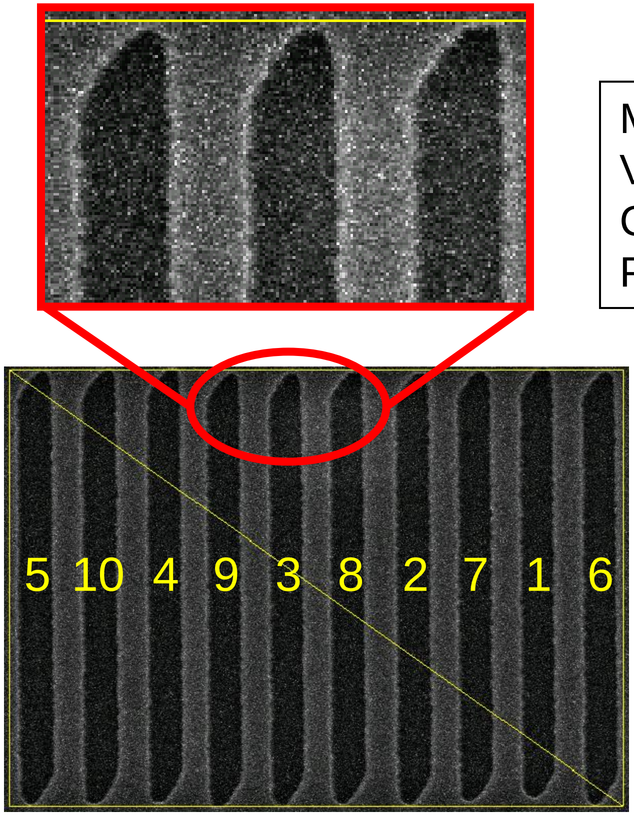
Subfield Scheduling (Non-Contiguously-Sequential Writing)

Effectiveness of Subfield Scheduling

- Line end roughness is improved (much smaller) with subfield scheduling (non-contiguously-sequential writing)



Contiguously Sequential

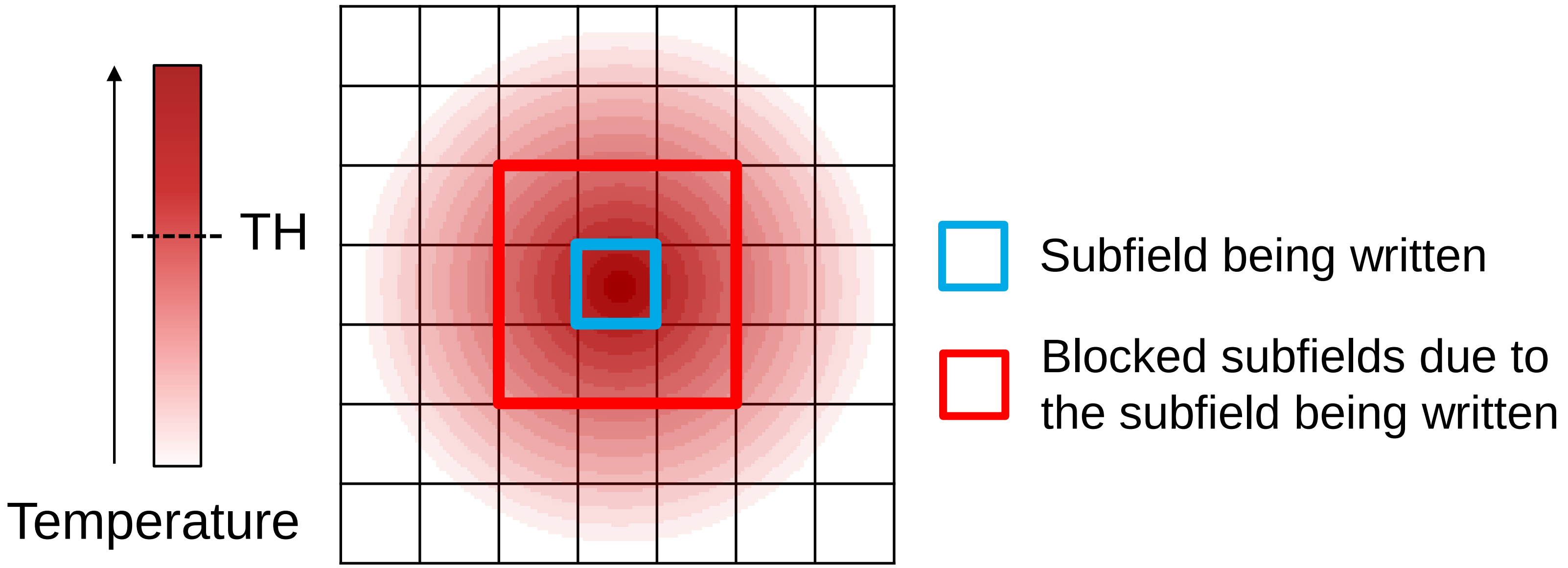


Subfield Scheduling
(Non-Contiguously-Sequential)

Machine:ELS7500
Voltage: 50keV
Current: 600pA
Photoresist: ZEP520A

Motivation

- The writing of a subfield raises the temperature of neighboring subfields
- **Blocked subfields**: the subfields with temperature higher than a threshold value
  - Should not be written in the following writing processes before the temperature drops below the threshold



Contribution

- Present the **first work** to solve the subfield scheduling problem with blocked region consideration
- Compared to previous work, the proposed algorithm is more elegant and systematic
  - Formulate the problem as a *constrained maximum scatter travelling salesman problem* (***constrained MSTSP***)
  - Decompose the problem into sub-problems and optimally solve each sub-problem
  - Merge sub-solutions into the final scheduling solution
- Experimental results show that our method achieves about 30% improvement in temperature reduction

Outline

Introduction

Preliminaries and Problem Formulation

An Exact MSTSP Algorithm for a Special Case

Graph-Based Subfield Scheduling Algorithm

Experimental Results and Conclusion

Thermal Model [Babin et al., SPIE'03]

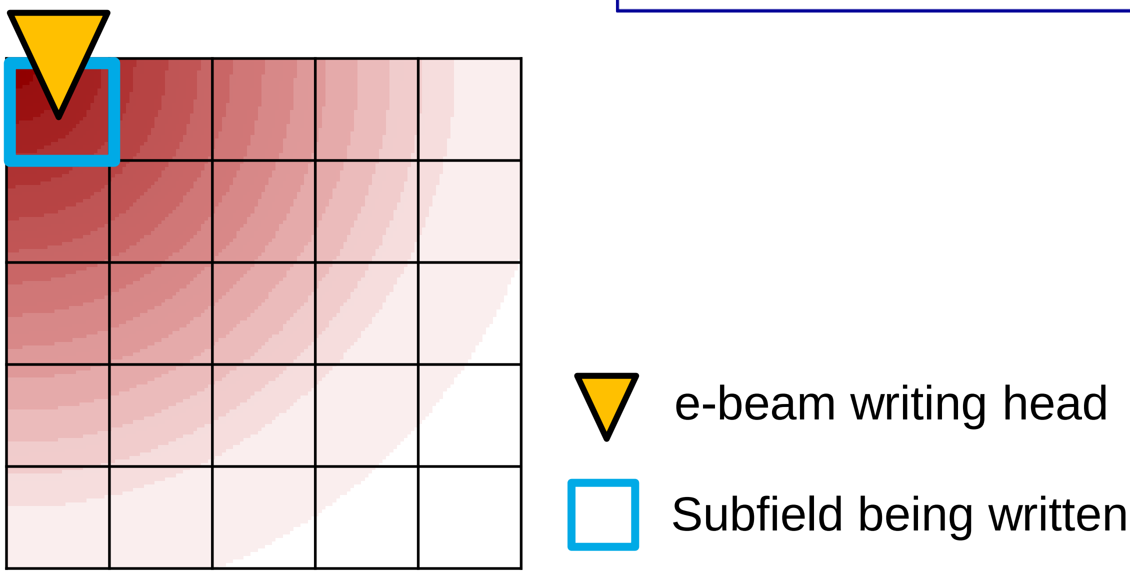
- The raised temperature of a subfield π_i due to the writing of a subfield π_j

$$T_{i,j}^{rise} \propto \frac{\overbrace{T_{j,j} + T_{j,j}^{rise} - T_{i,j}}^{\text{Temperature difference}}}{\underbrace{dist(\pi_i, \pi_j)^2}_{\text{Squared Euclidean distance}}}$$

$T_{i,j}$: temperature of π_i before the writing of π_j

$T_{i,j}^{rise}$: raised temperature of π_i due to the writing of π_j

$T_{j,j}^{rise}$: temperature of π_j due to its own writing



Thermal Model (cont'd)

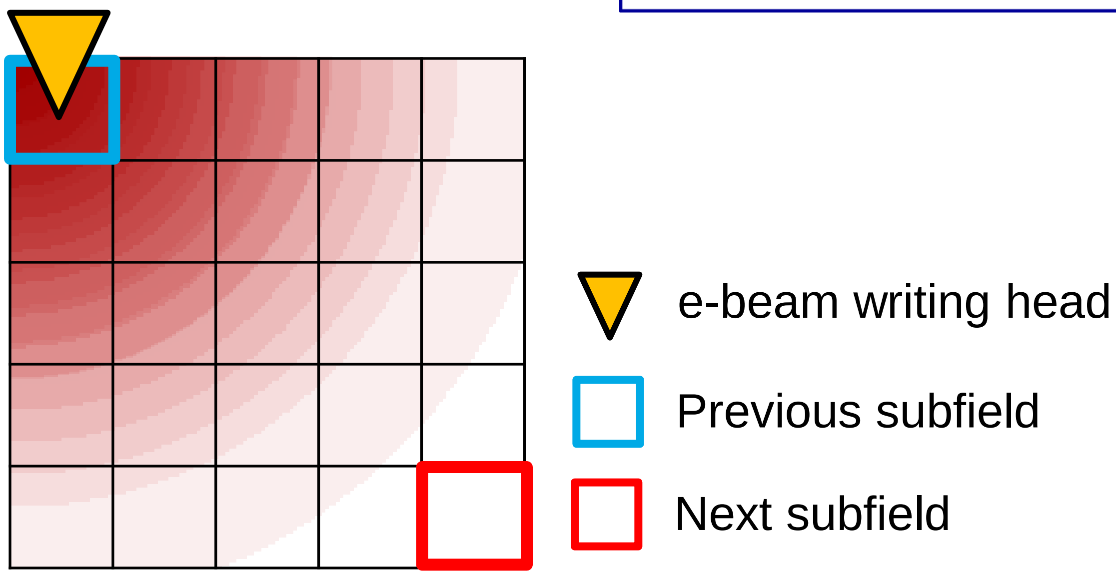
- The temperature of each subfield decays exponentially during the traveling time of an e-beam writer

$$T_{i,j} = (T_{i,j-1} + T_{i,j-1}^{rise}) \cdot \underbrace{f}_{\text{Decay factor}}$$

$T_{i,j}$: temperature of π_i before the writing of π_j

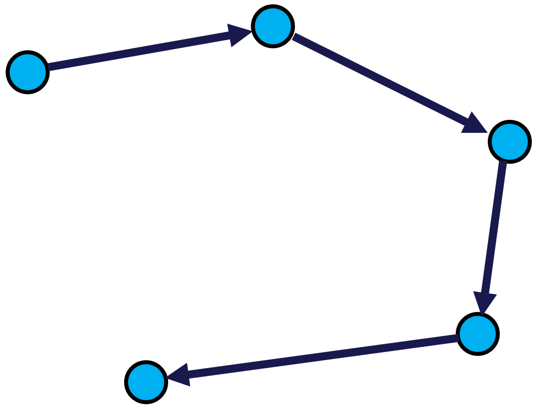
$T_{i,j}^{rise}$: raised temperature of π_i due to the writing of π_j

$T_{j,j}^{rise}$: temperature of π_j due to its own writing

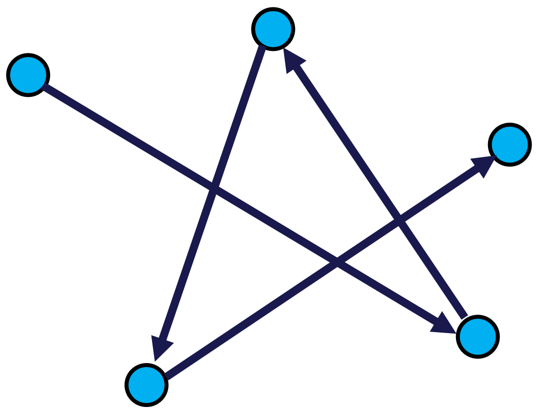


Maximum Scatter TSP (MSTSP)

- The maximum scatter traveling salesman problem (MSTSP) finds a Hamiltonian path that is most scattered
- MSTSP can be used to maximize the distance between two subfields being successively written
- No polynomial-time algorithm for MSTSP



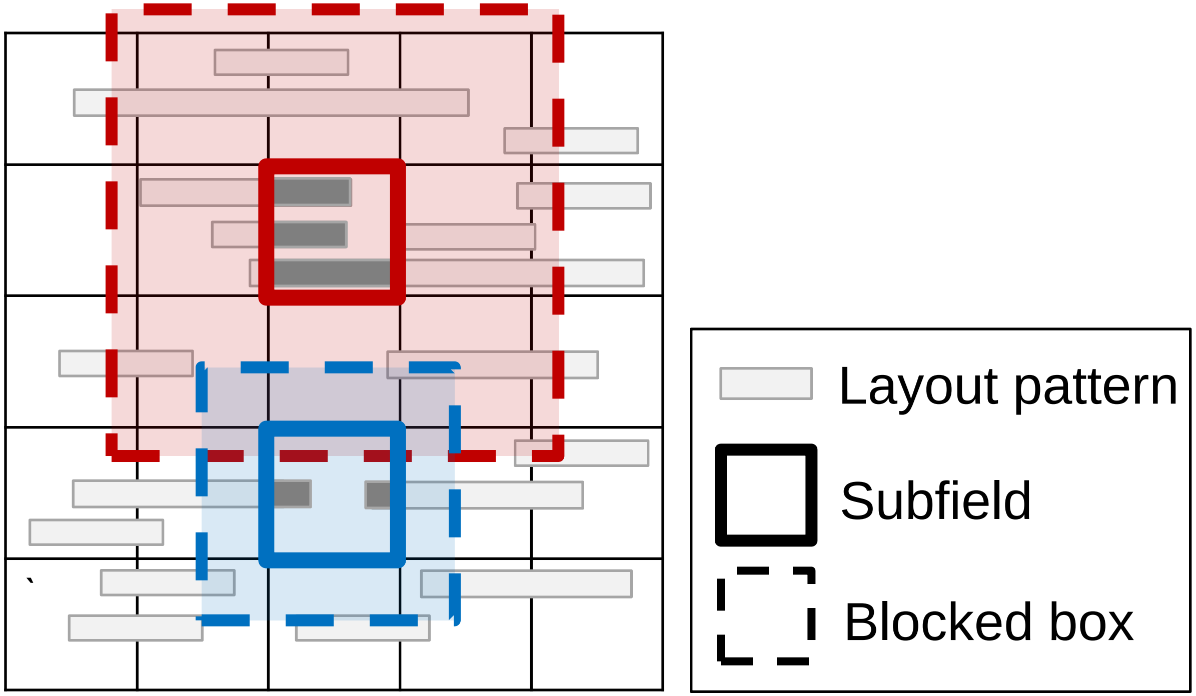
Traditional TSP



MSTSP

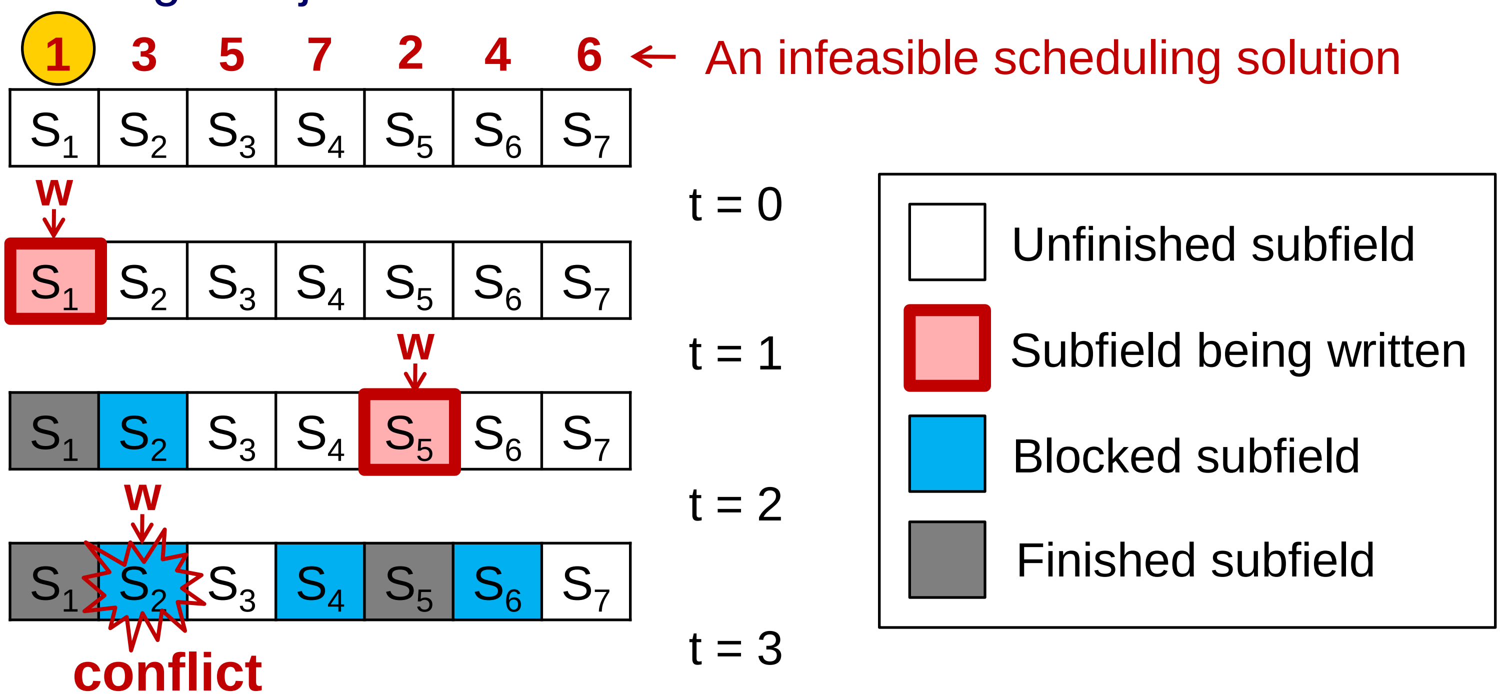
Blocked Box

- A blocked box of a subfield indicates the block coverage caused by its writing process



Constrained MSTSP

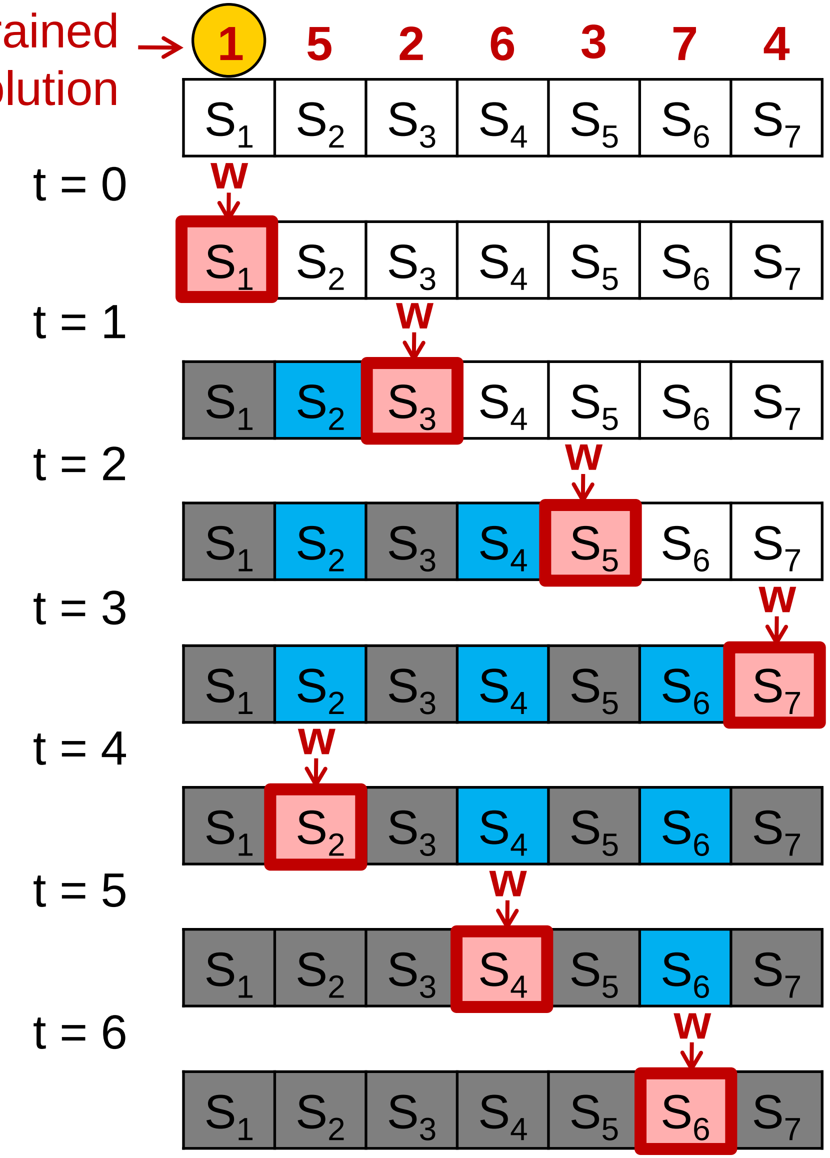
- **Constrained MSTSP**: the MSTSP with the blocked box constraint
- A solution of the MSTSP may not be feasible for the constrained MSTSP
  - An infeasible example: writing time = 1; blocked time = 2; blocked coverage: adjacent subfields

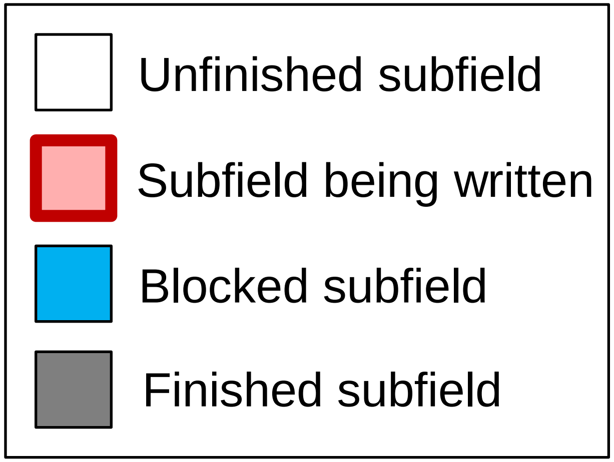


Constrained MSTSP (cont'd)

A feasible constrained MSTSP solution

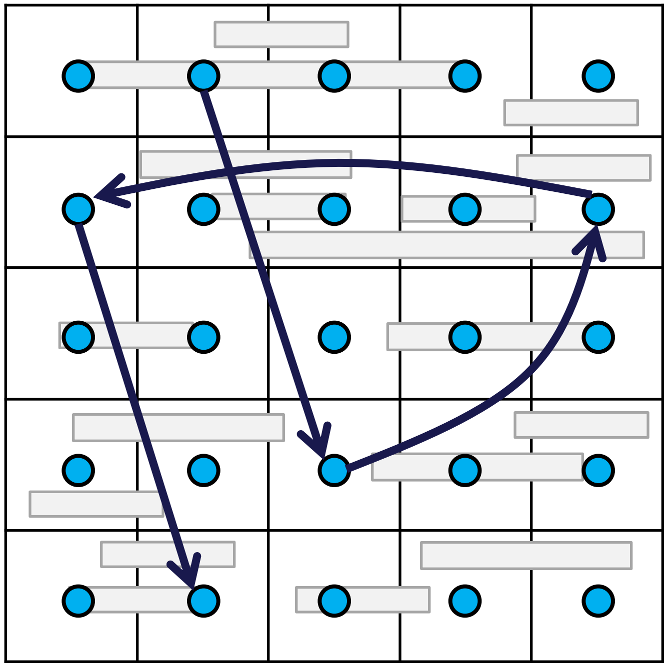
- A feasible constrained MSTSP solution can be found with blocked box consideration





Problem Formulation

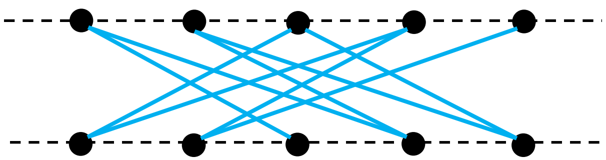
- Given
  - A post-mask layout
  - Predefined subfield size
- Objective
  - An MSTSP solution where the length of the shortest edge in the path is maximized
- Constraint
  - Blocked box constraint



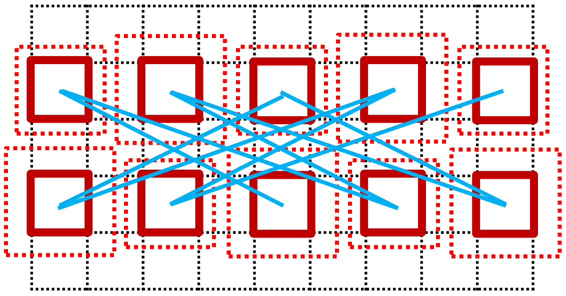
Methodology

- Constrained MSTSP is NP-complete
- To tackle the high complexity and the blocked box constraint:

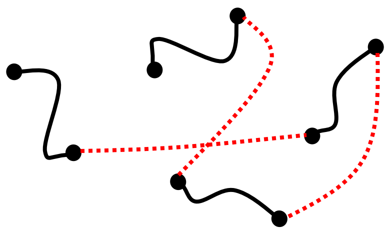
◆ Develop a linear time algorithm for a special case of the MSTSP



◆ Extract sub-problems:
1. Conform to the special case
2. Satisfy the blocked box constraint



◆ Solve each sub-problem independently
◆ Merge sub-solutions into the final solution



Outline

Introduction

Preliminaries and Problem Formulation

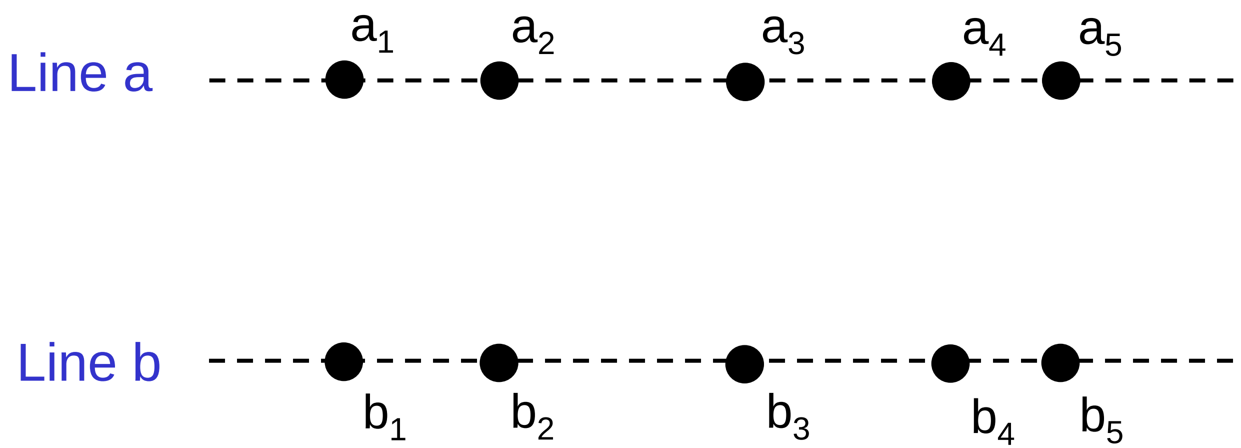
An Exact MSTSP Algorithm for a Special Case

Graph-Based Subfield Scheduling Algorithm

Experimental Results and Conclusion

A Special Case of the MSTSP

- In the case that
  - Points are on two parallel lines
  - Each line has an odd number of points
  - Points on different lines are aligned



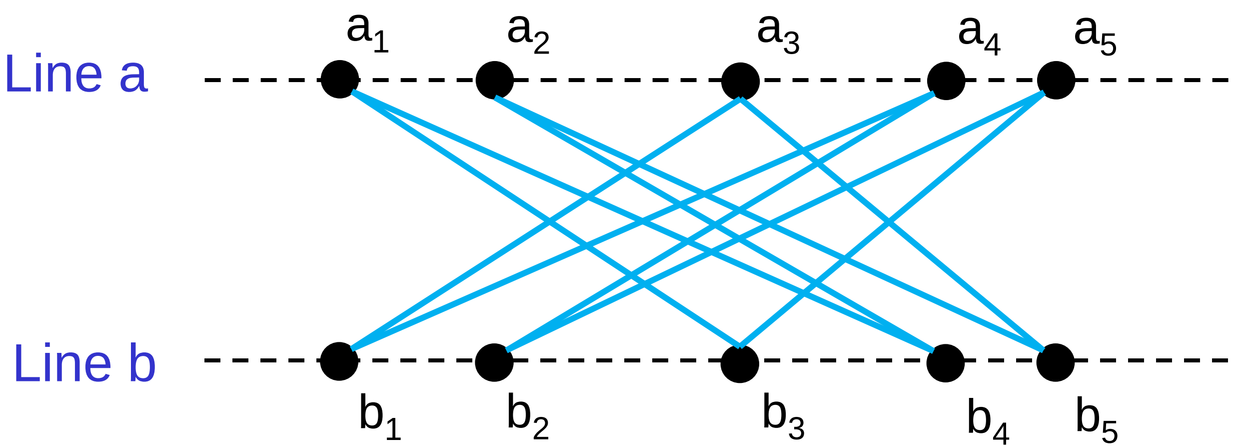
- A linear time algorithm can optimally solve the MSTSP in this special case

Exact MSTSP Algorithm

- V: a set of input points
  - |V|=2n, n=2k+1
- **Exact MSTSP algorithm**
  - **Step 1.** For each point a_h, connect a_h to b_i and b_j, where

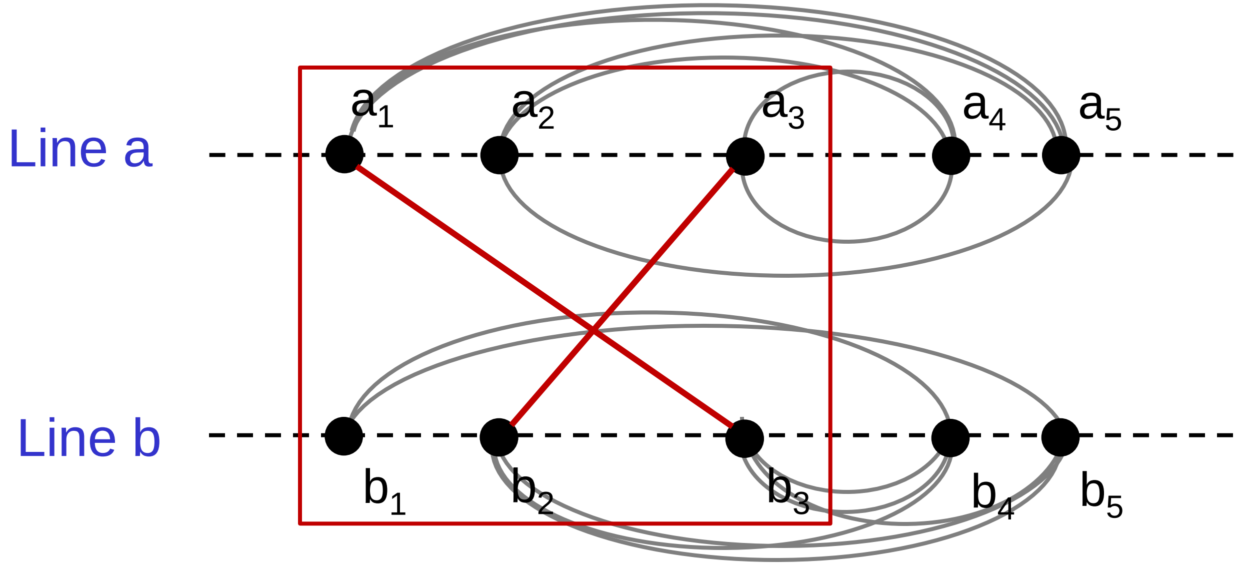
$$i = \begin{cases} h+(1+k) & \text{, if } h+(1+k) \le n \\ h+(1+k)-n\text{,} & \text{if } h+(1+k) > n \end{cases}\text{, and}\quad j = \begin{cases} h+k & \text{, if } h+k \le n \\ h+k-n\text{,} & \text{if } h+k > n \end{cases}.$$

  - **Step 2.** Delete the shortest edge in the generated cycle



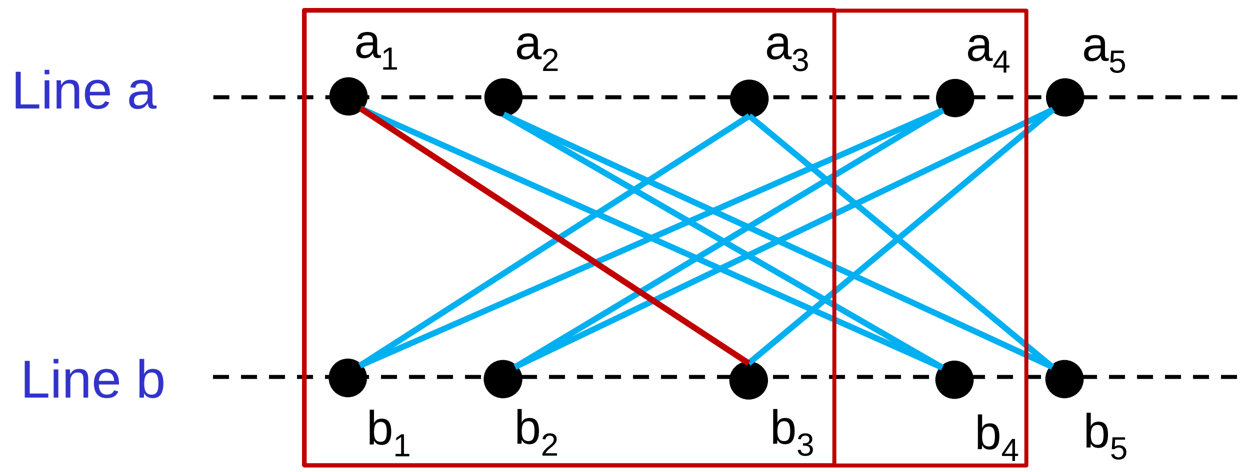
Proof of the Optimality

- Definition
  - d*: the length of the shortest edge of an optimal Hamiltonian path
  - Diameter of a set of points *R*: the longest edge in R
- **Lemma 1** [Arkin et al., SIAM'99]
  - *Let* $R \subseteq V$ *be a subset of points with* $|R| > \left\lceil \frac{|V|}{2} \right\rceil$
    Then, in any Hamiltonian path on V,
    there must exist an edge joining two points of R



Proof of the Optimality (cont'd)

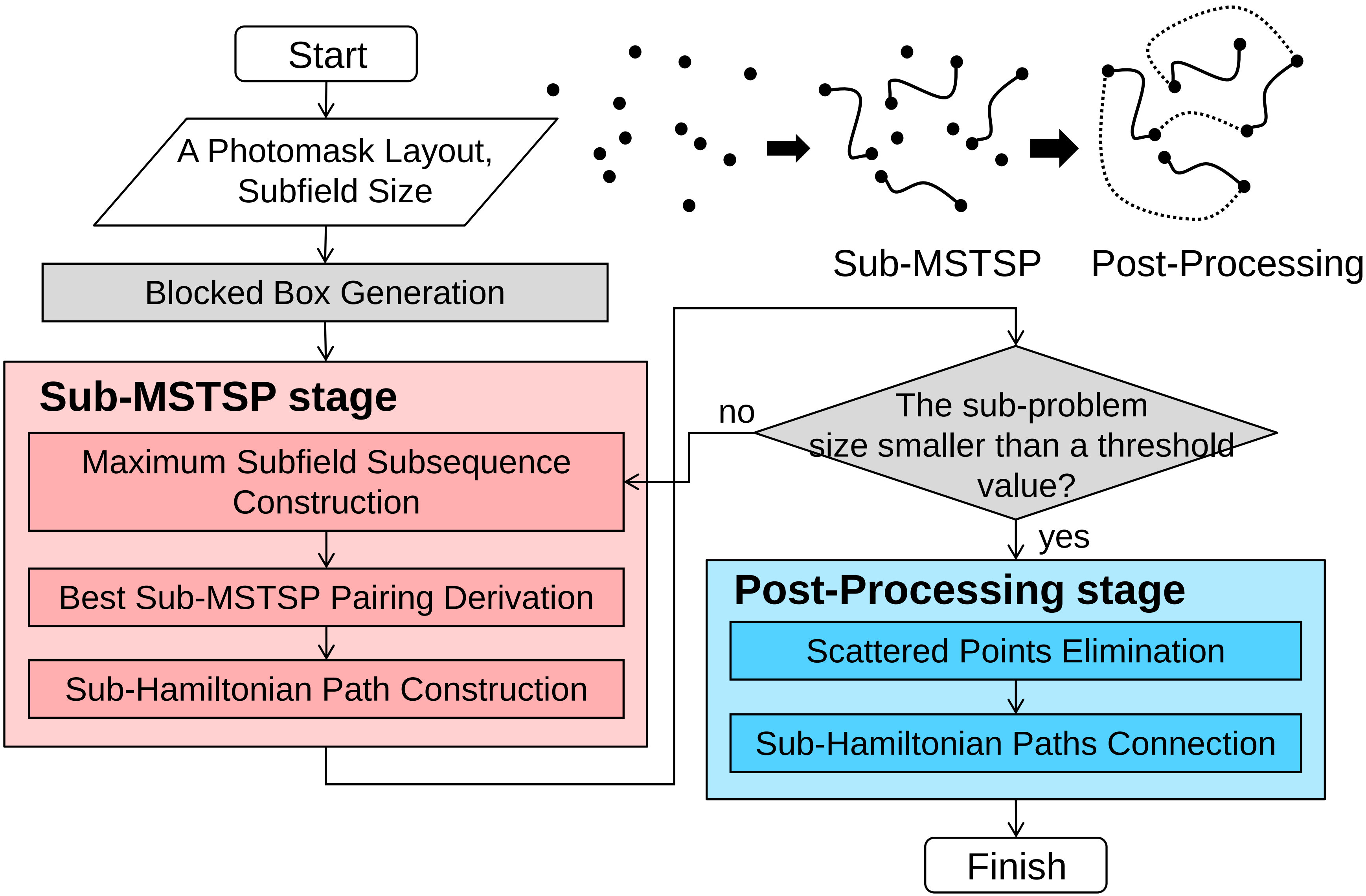
- Key idea of the proof
  - Each edge is a diameter of a set $R_i \subseteq V$ of size $|R_i| > \left\lceil \frac{|V|}{2} \right\rceil$
  - Each edge is an upper bound of *d** => the generated Hamiltonian path is optimal
- Running time: O(n)



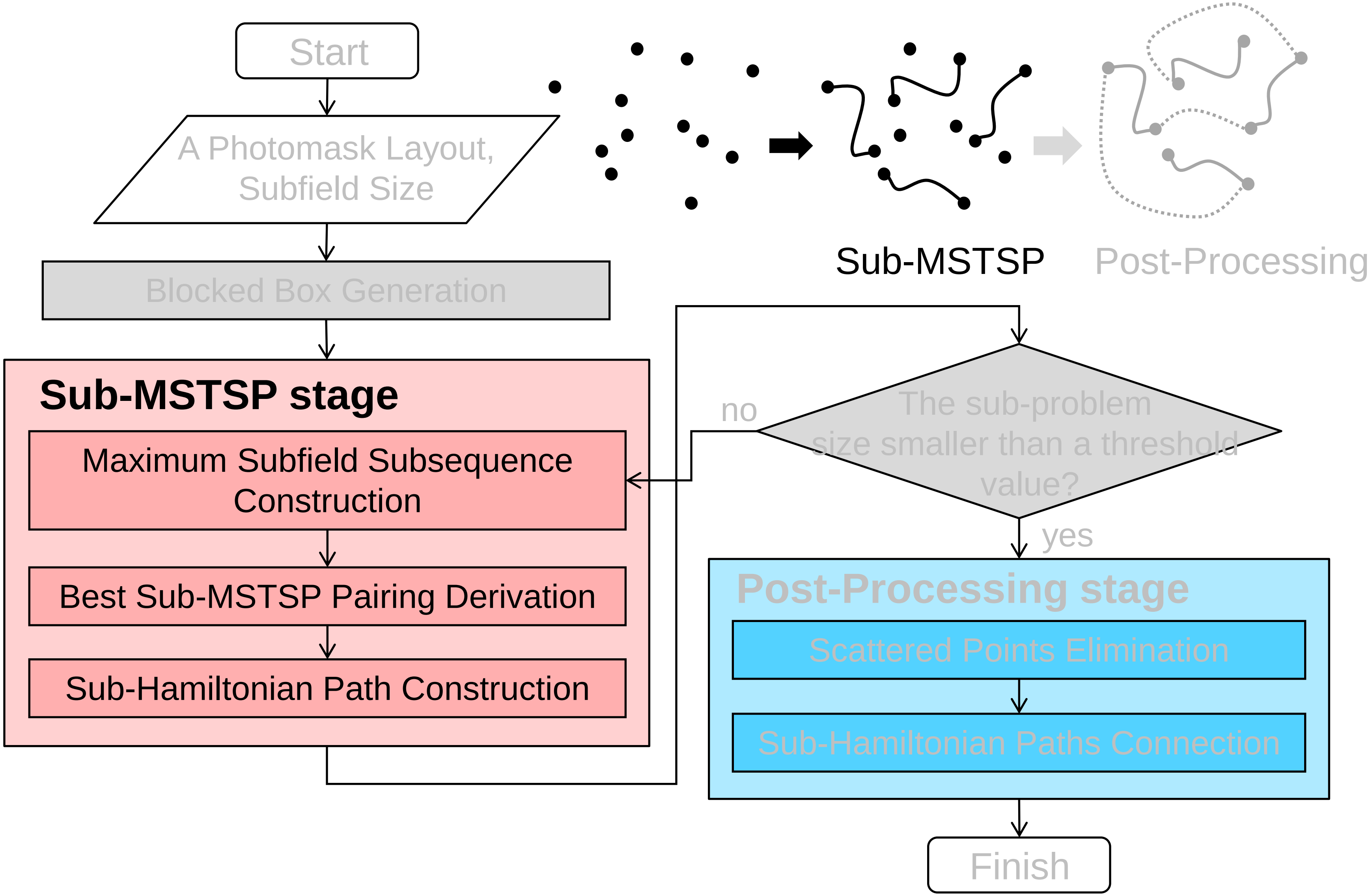
Outline

- Introduction
- Preliminaries and Problem Formulation
- An Exact MSTSP Algorithm for a Special Case
- **Graph-Based Subfield Scheduling Algorithm**
- Experimental Results and Conclusion

Graph-Based Subfield Scheduling Flow

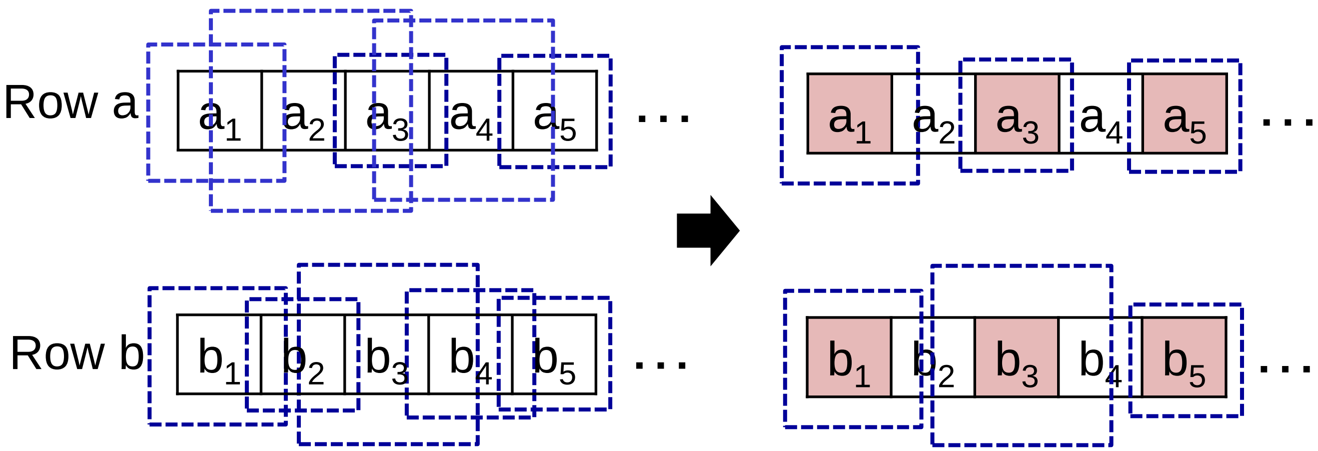


Graph-Based Subfield Scheduling Flow



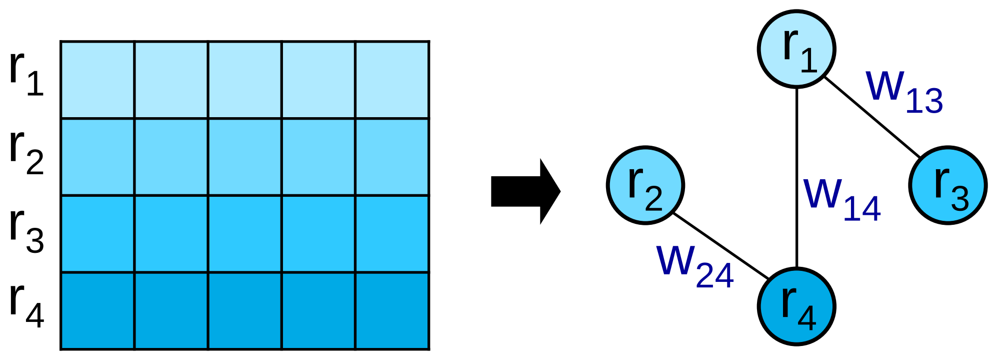
Maximum Subfield Subsequence Construction

- **Maximum subfield subsequence construction**: for a pair of rows, find a set of subfields
  - Conform to the special case
  - Satisfy the blocked box constraint
  - The total area of blocked boxes is maximized



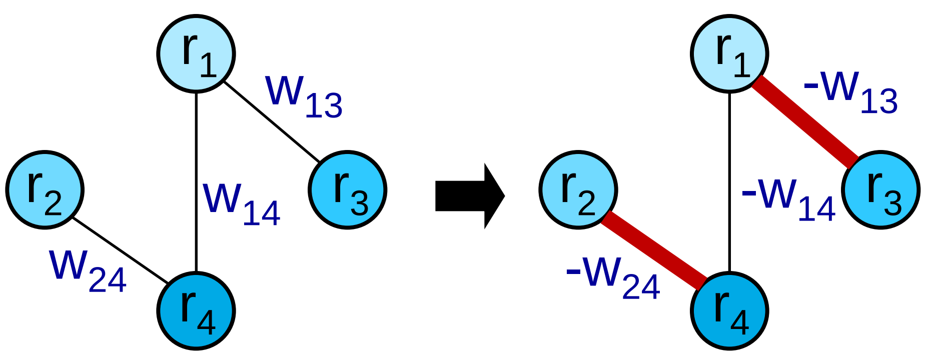
Best Sub-MSTSP Pairing Derivation (1/2)

- **Best sub-MSTSP pairing deviation**: find a pairing of all subfield rows maximizing the length of the shortest edge among all sub-Hamiltonian paths
- **Step 1.** Construct a compatible graph
  - A vertex: a subfield row
  - An edge: the two rows are not overlapped vertically, and the edge weight equals the length of the shortest edge in their optimal sub-Hamiltonian path

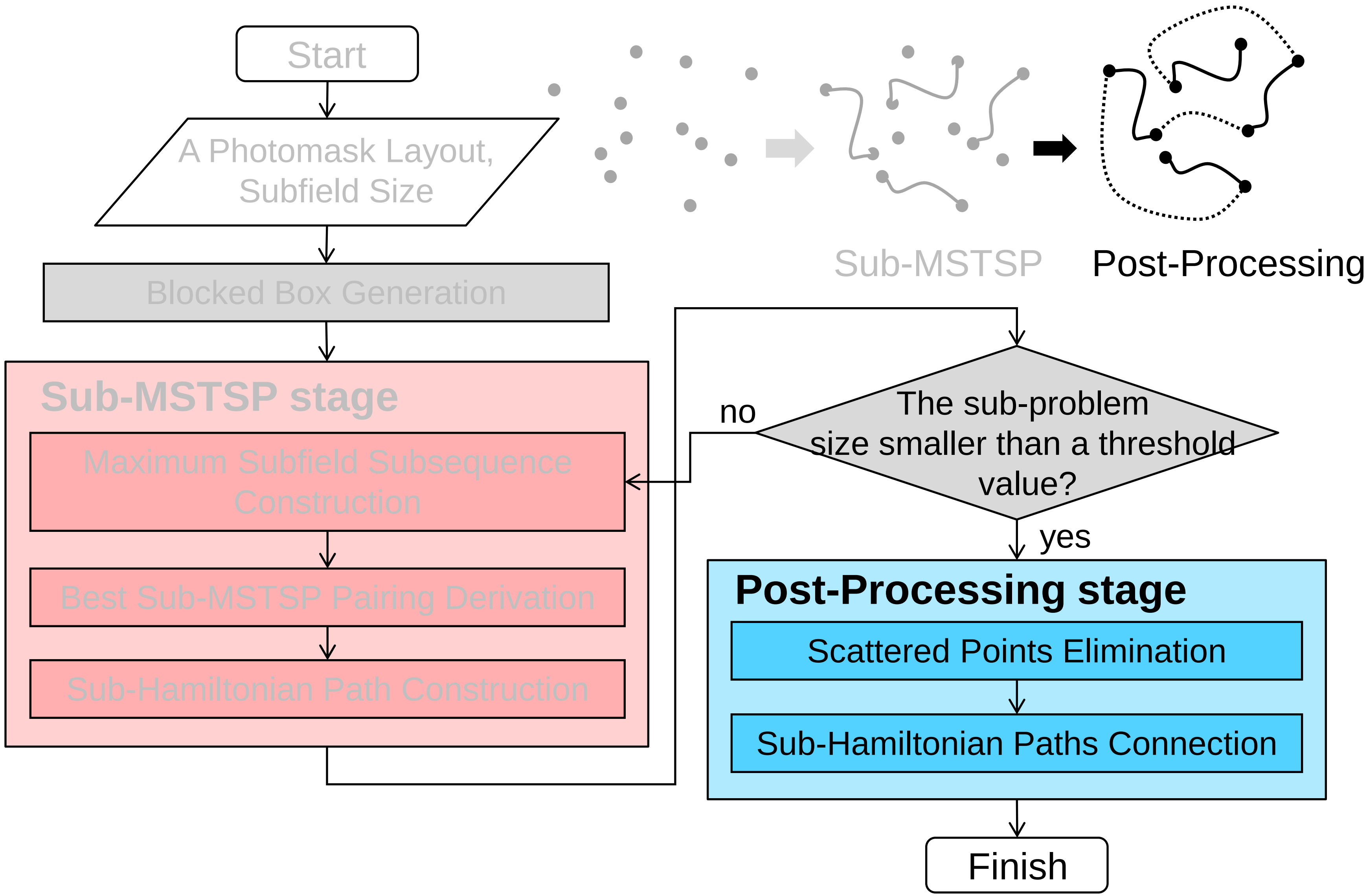


Best Sub-MSTSP Pairing Derivation (2/2)

- **Step 2.** Find a maximum matching where the smallest edge weight is maximized
  - Multiply the edge weights by -1
  - Apply a maximum cardinality bottleneck matching algorithm [Gabow and Tarjan, J. Algorithms'88]
    - Maximum cardinality bottleneck matching: find a maximum matching where the largest edge weight is minimized

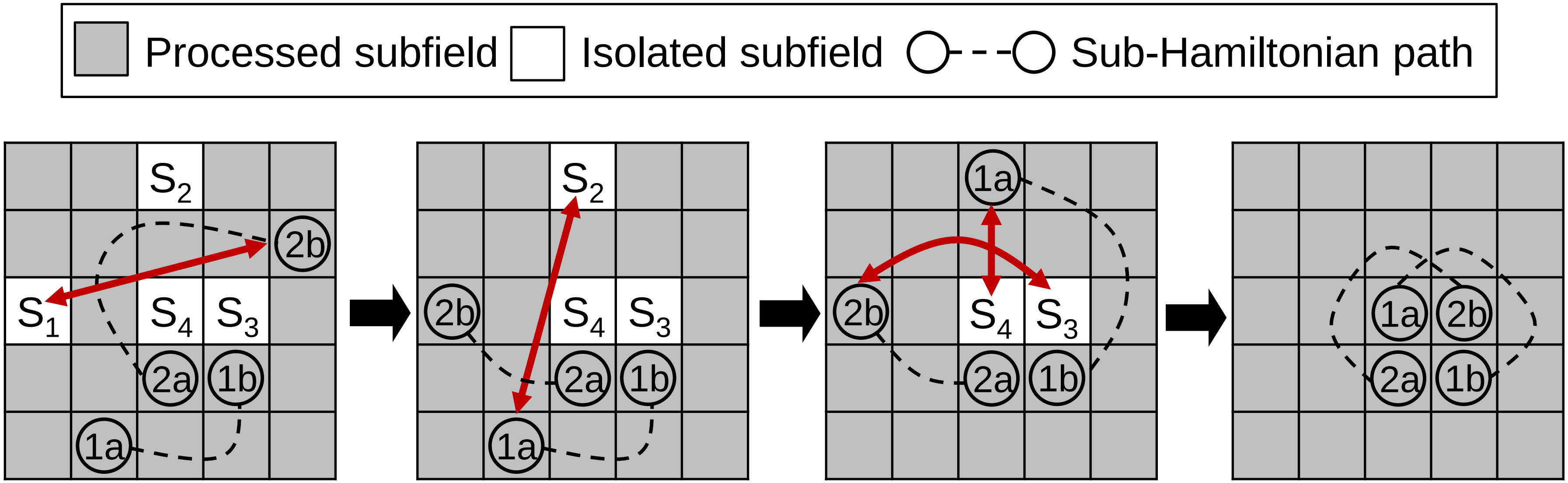


Graph-Based Subfield Scheduling Flow



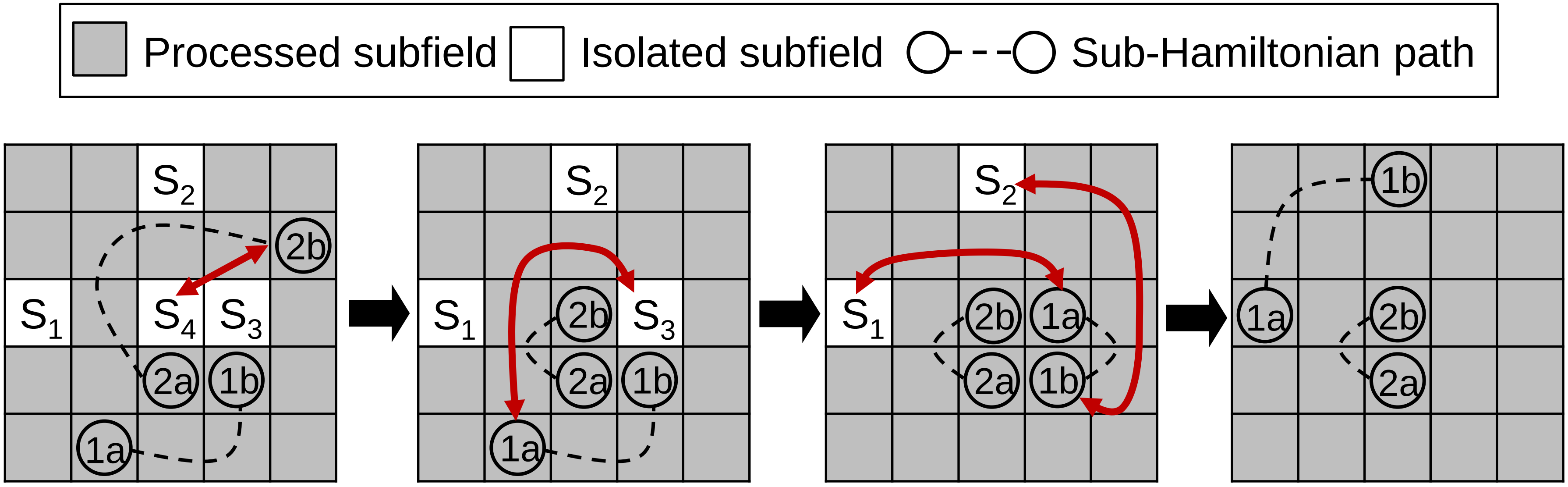
Isolated Subfields Connection (1/2)

- Eliminating isolated subfields from the margin to the center of a chip may generate an undesired result
  - Connection order: <S_1, S_2, S_3, S_4>
  - The end points of sub-Hamiltonian paths are close



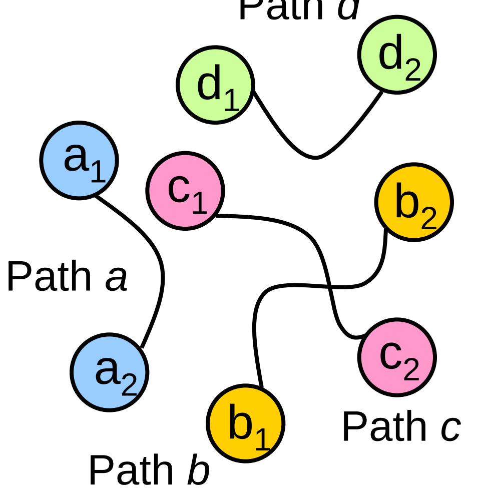
Isolated Subfields Connection (2/2)

- Eliminating isolated subfields from the center to the margin of a chip may generate a better result
  - Connection order: <S_4, S_3, S_2, S_1>
  - The end points of sub-Hamiltonian paths are scattered

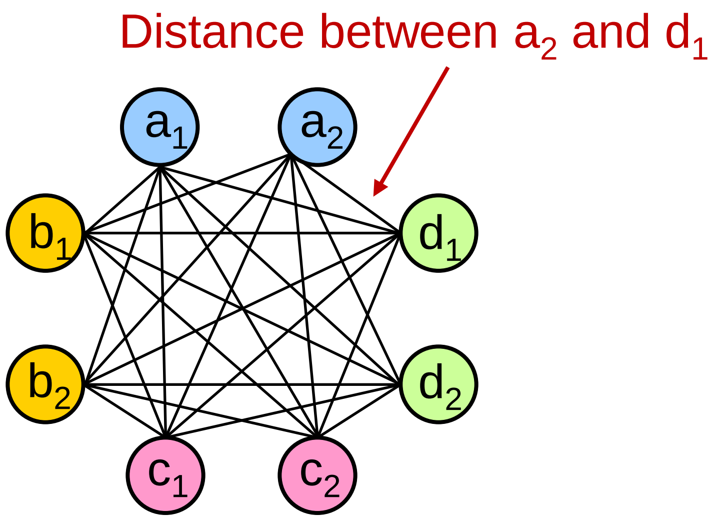


Sub-Hamiltonian Paths Merging (1/2)

- Merge sub-Hamiltonian paths and maximize the lengths of edges connecting sub-Hamiltonian paths
- **Step 1.** Construct a compatible graph
  - A vertex: an endpoint of a sub-Hamiltonian path
  - An edge: the two endpoints belong to different paths, and the edge weight equals their distance



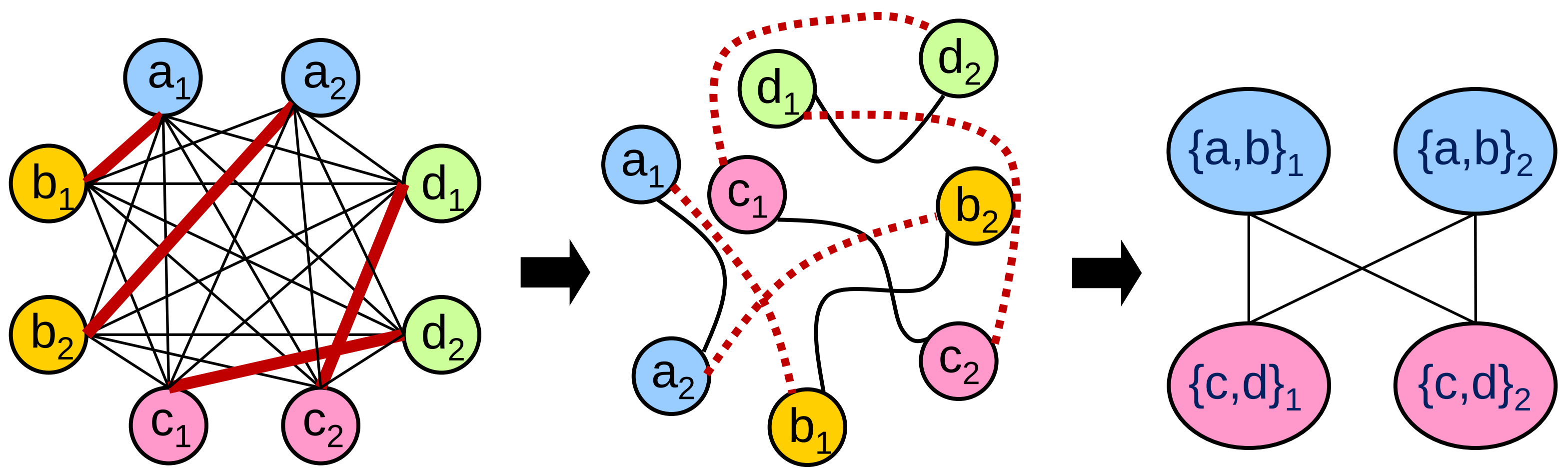
Sub-Hamiltonian paths



Compatible graph

Sub-Hamiltonian Paths Merging (2/2)

- **Step 2.** Merge paths by using a maximum cardinality bottleneck matching algorithm
- **Step 3.** Break resulting cycles and update the compatible graph
  - Repeat Step 1 to Step 3 until all sub-Hamiltonian paths are merged into one Hamiltonian path



Maximum cardinality bottleneck matching | The resulting graph with two cycles | Updated compatible graph after breaking cycles

Outline

Introduction

Preliminaries and Problem Formulation

An Exact MSTSP Algorithm for a Special Case

Graph-Based Subfield Scheduling Algorithm

Experimental Results and Conclusion

Experimental Setup

- Platform
  - C++ programming language
  - 2.13GHz Linux workstation with 16GB memory
- Benchmark
  - 11 benchmark circuits

Circuit	Size (μm^2)	# Layers	# Nets	# Pins
Mcc1	45000×39000	4	1694	3101
Mcc2	152400×152400	4	7541	25024
Struct	4903×4904	3	3551	5471
Primary1	7522×4988	3	2037	2941
Primary2	10438×6488	3	8197	11226
S5378	4330×2370	3	3124	4734
S9234	4020×2230	3	2774	4185
S13207	6590×3640	3	6995	10562
S15850	7040×3880	3	8321	12566
S38417	11430×6180	3	21035	32210
S38584	12940×6710	3	28177	42589

Comparison of Scheduling Results

- Compare the minimum and the average edge lengths of scheduling solutions obtained from GSS and those obtained from our algorithm flow
  - GSS: a greedy subfield scheduling algorithm [Babin et al., SPIE'03]
  - Min.: shortest length between successive subfields
  - Avg.: average length between successive subfields

#subfields	GSS			Ours		
	Min.	Avg.	CPU (s)	Min.	Avg.	CPU (s)
32 x 32	3.16	24.36	33.56	18.46	22.30	0.03
64 x 64	3.61	48.84	>1hr	32.19	43.63	0.42
128 x 128	2.00	86.43	>1hr	63.03	85.17	7.33
256 x 256	1.00	164.92	>1hr	128.79	166.95	66.84
Avg.	**0.08**	**1.05**	**-**	**1.00**	**1.00**	**-**

(measured in unit subfield edge length)

Comparison of Thermal Results (1/2)

- Compare the maximum temperatures computed from scheduling solutions
  - Thermal model: simplified Green's function [Babin et al., SPIE'03]

Circuit	Maximum Temperature					
	32 x 32 (#subfields)			64 x 64		
	GSS [3]	Ours	Imp. Rate	GSS [3]	Ours	Imp. Rate
Mcc1	104.66	97.11	7.22%	132.81	73.16	44.92%
Mcc2	123.96	100.09	19.26%	158.38	88.46	44.14%
Struct	99.76	99.48	0.28%	133.11	95.98	27.89%
Primary1	163.37	165.52	-1.31%	177.39	150.08	15.39%
Primary2	138.49	128.28	7.38%	165.71	131.18	20.83%
S5378	109.46	106.75	2.47%	130.71	94.86	27.43%
S9234	111.58	107.09	4.03%	131.53	79.89	39.26%
S13207	106.20	111.11	-4.62%	133.75	112.97	15.54%
S15850	123.31	124.89	-1.28%	134.95	107.49	20.35%
S38417	91.61	78.90	13.88%	119.89	78.60	34.44%
S38584	93.66	82.01	12.43%	142.17	118.50	16.65%
Avg.			**5.43%**			**27.89%**

Comparison of Thermal Results (2/2)

- Our algorithm achieves much more temperature reduction as the number of subfields increase

Circuit	Maximum Temperature					
	128 x 128 (#subfields)			256 x 256		
	GSS [3]	Ours	Imp. Rate	GSS [3]	Ours	Imp. Rate
Mcc1	108.52	64.06	40.97%	88.08	55.17	37.36%
Mcc2	138.79	80.11	42.28%	104.07	61.47	40.93%
Struct	109.92	88.78	19.23%	85.76	60.18	29.83%
Primary1	150.15	108.43	27.79%	117.17	84.45	27.93%
Primary2	153.82	130.24	15.33%	126.61	104.73	17.28%
S5378	109.29	67.85	37.91%	74.85	44.06	41.13%
S9234	106.55	79.66	25.24%	75.69	45.14	40.36%
S13207	115.78	71.48	38.26%	77.55	45.19	41.72%
S15850	101.33	73.16	27.80%	85.88	62.46	27.28%
S38417	109.18	72.10	33.96%	82.29	56.14	31.78%
S38584	80.01	62.46	21.94%	92.77	74.62	19.56%
Avg.			**30.07%**			**32.29%**

Conclusion

- This paper presents the **fist work** for the subfield scheduling problem with the blocked box constraint

- Compared to previous work, our algorithm
  - Systematically solve the problem by formulating the problem into the constrained MSTSP
  - Elegantly derive a scheduling solution by using graph algorithms

- Experimental results show that our algorithm can
  - Efficiently derive a subfield scheduling solution
  - Effectively mitigate the heating problem

Thank You!