Border Length Minimization in DNA Array Design

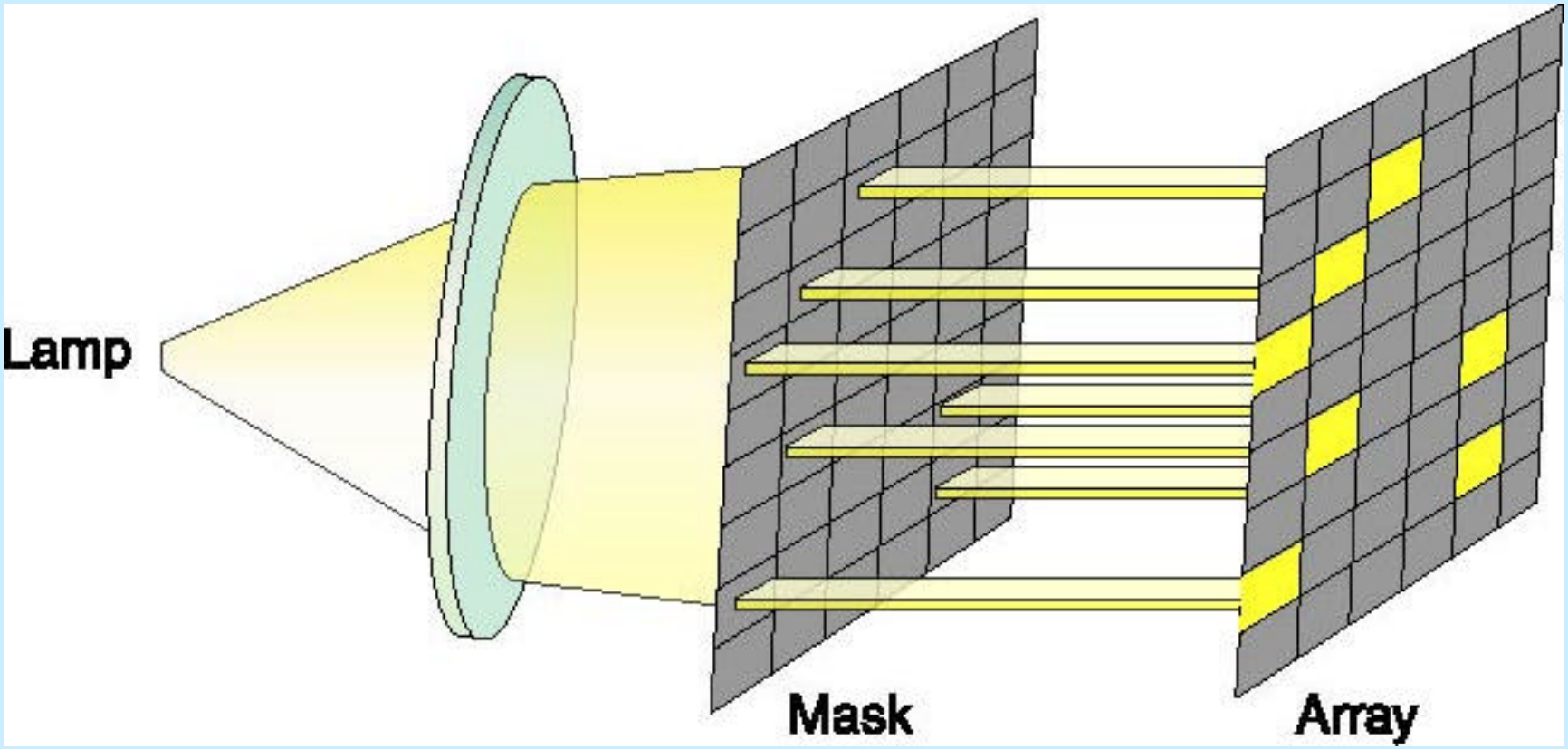
A.B. Kahng, I.I. Mandoiu, P.A. Pevzner,

S. Reda (all UCSD), A. Zelikovsky (GSU)

DNA Probe Arrays

- Used in wide range of genomic analysis
- DNA Probe Arrays up to 1000x1000 sites filled with 25-long probes
- Array manufacturing process
  VLSIPS = very large-scale immobilized polymer synthesis:
  - Sites selectively exposed to light to activate further nucleotide synthesis
  - Selective exposure achieved by sequence of masks M_1, M_2, …, M_K
  - Masks induce deposition of nucleotide (ACTG) at exposed sites
  - Mask sequence
    → nucleotide deposition sequence - typically periodical $(ACTG)^p$
    → supersequence of all probe sequences
- Our concern: Diffraction ➔unwanted illumination ➔yield decrease

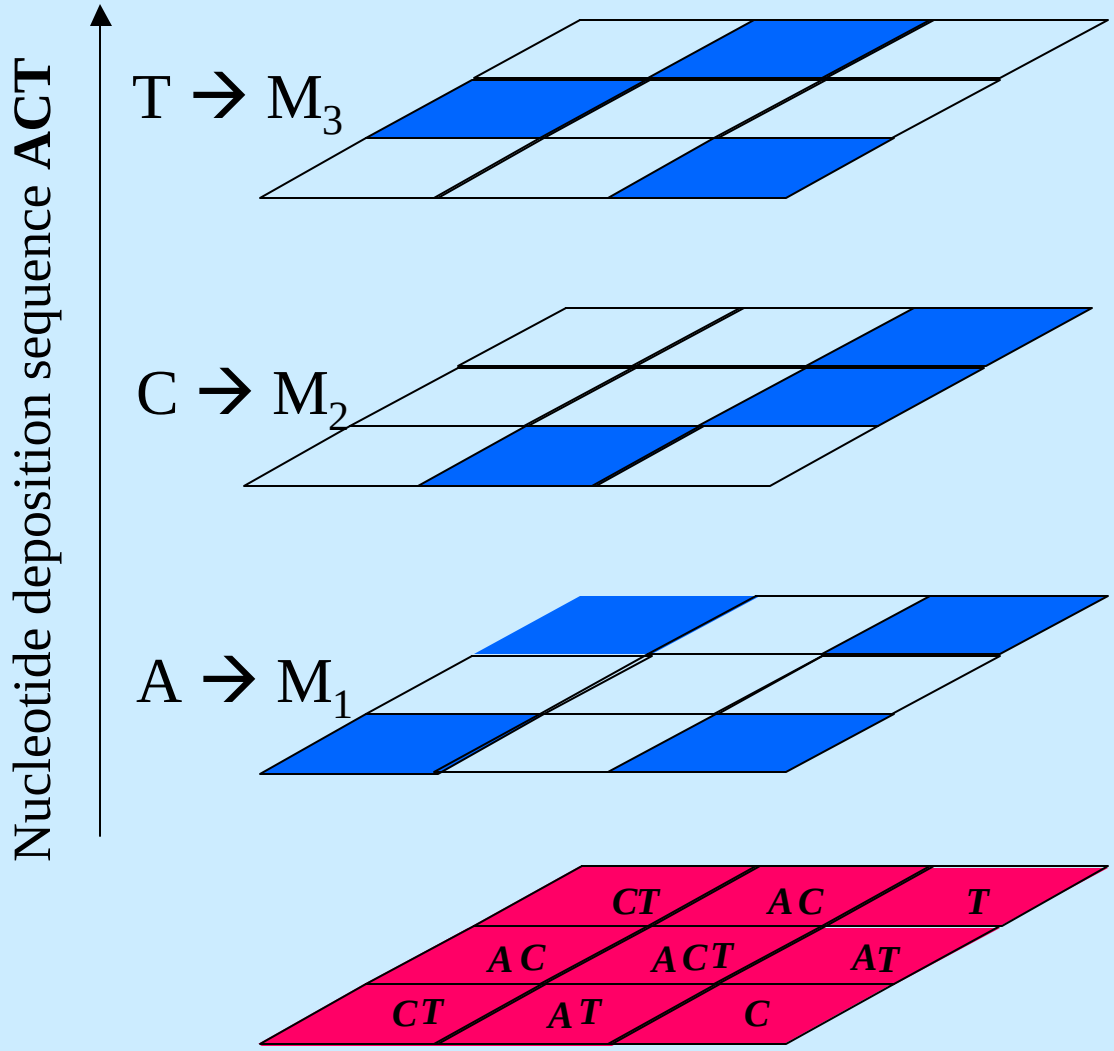
Affymetrics Chip



2-dim Probe Placement and Synthesis

CT	**AC**	**T**
AC	**ACT**	**AT**
CT	**AT**	**C**

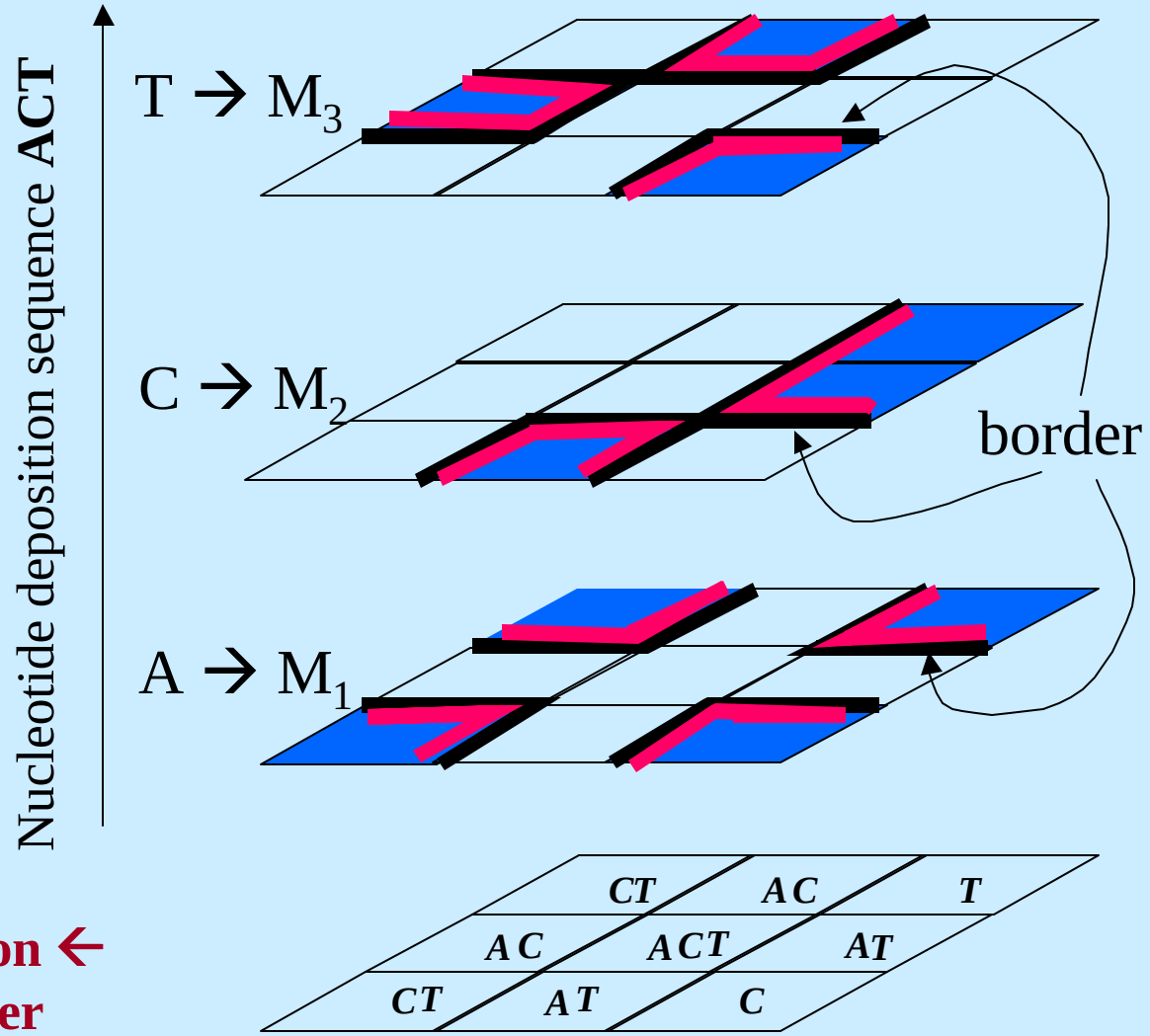
2-dim placement of probes



Unwanted Illumination

CT	AC	T
AC	ACT	AT
CT	AT	C

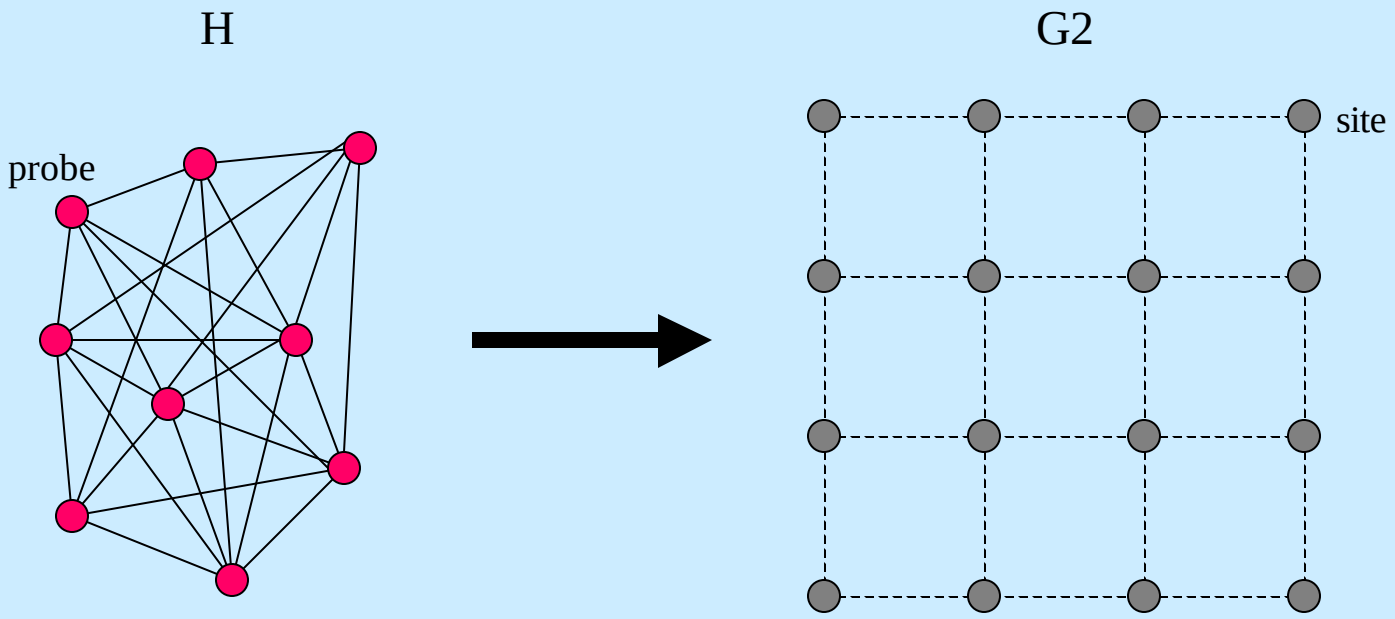
2-dim placement of probes



Unwanted illumination ←
➔Minimize the border

Problem formulation

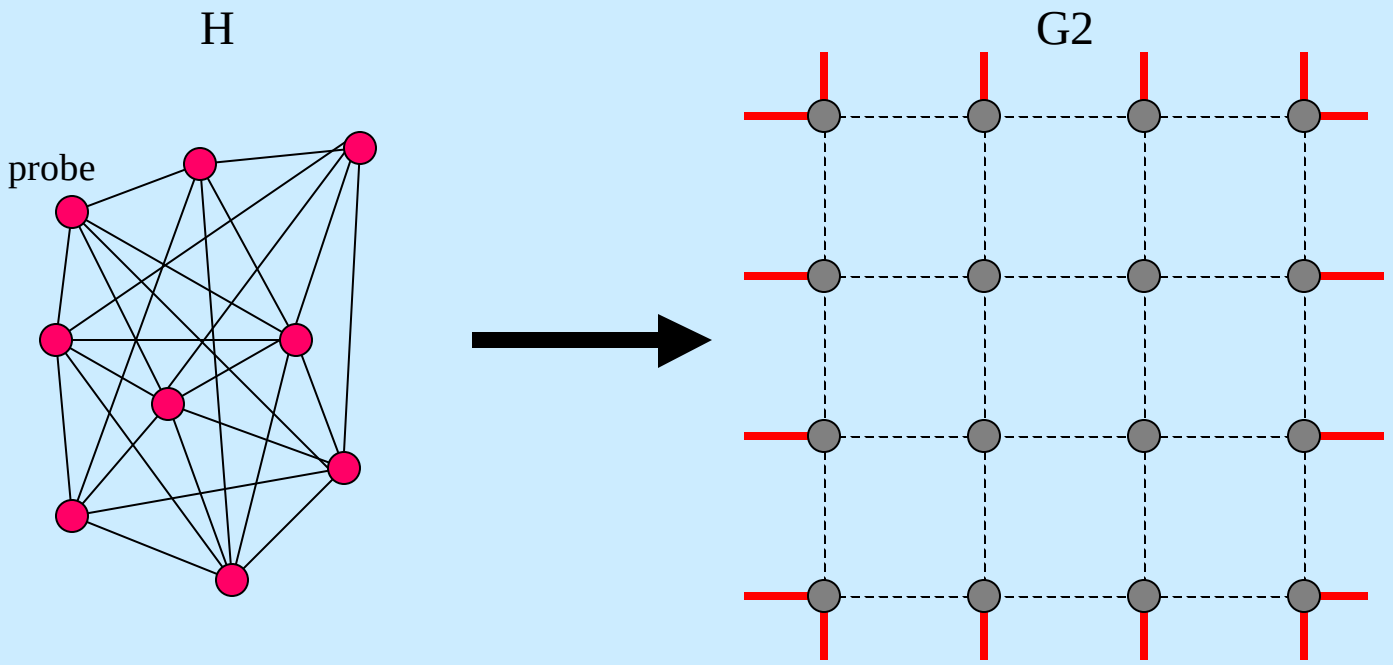
- 2-dim (synchronous) Array Design Problem:
  - Minimize placement cost of Hamming graph H
    - (vertices=probes, distance = Hamming)
  - on 2-dim grid graph G2 (N x N array, edges b/w neighbors)



Lower Bound

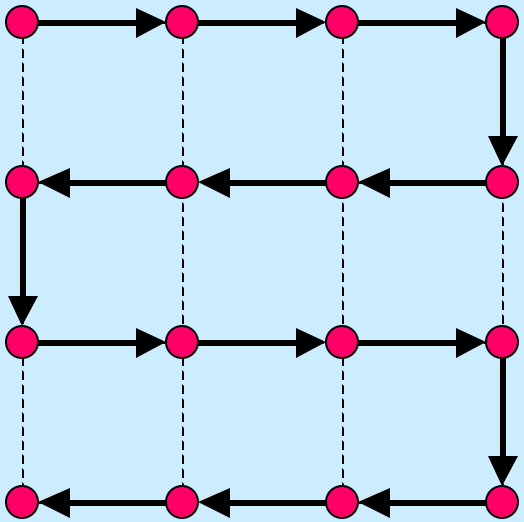
Lower bound for the placement:

Sum of distances to 4 closest neighbors
– weight of 4N heaviest arcs

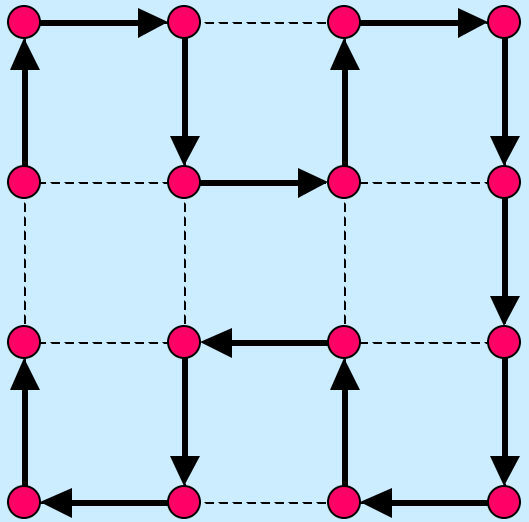


TSP+1-Threading Placement

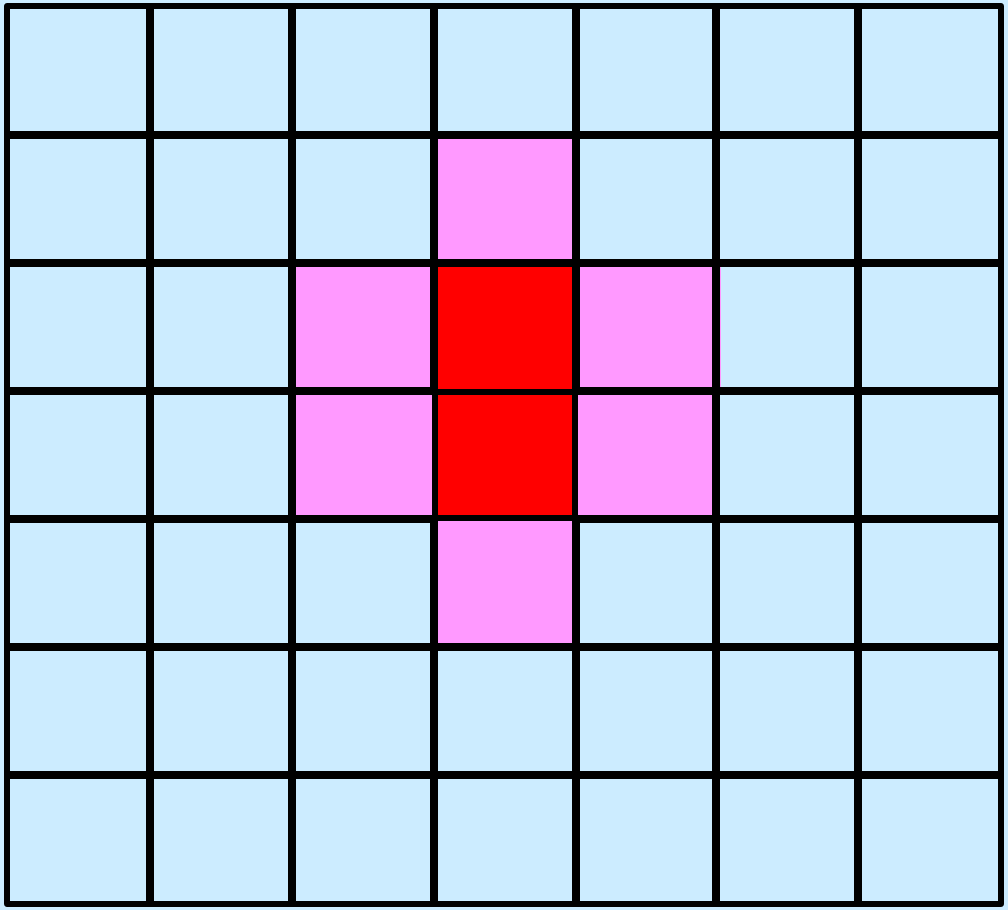
- Hubbel 90's
  - Find TSP tour/path over given probes with Hamming distance
  - Place in the grid following TSP
  - Adjacent probes are similar



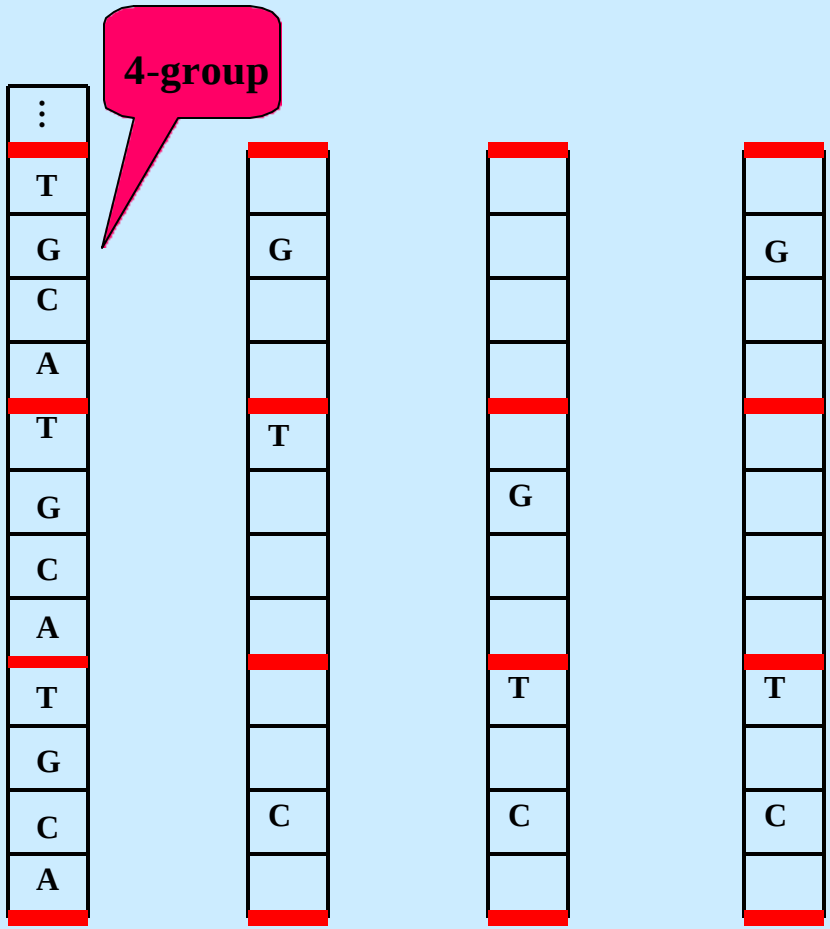
- Hannenhalli,Hubbel,Lipshutz, Pevzner'02:
  - Place the probes according to 1-Threading
  - further decreases total border by 20%



Epitaxial Placement Algorithm



Diving into 3d Dimension: Embedding in Nucleotide Sequence



Periodic nucleotide sequence S

→Synchronous embedding of CTG in S

→Asynchronous leftmost embedding of CTG in S

→Another asynchronous embedding

Problem formulations

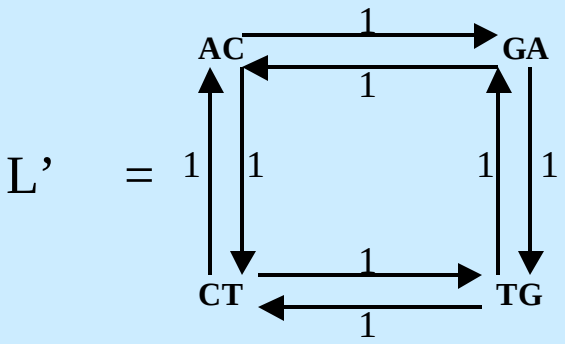
- 2-dim (synchronous) Array Design Problem:
  - Minimize placement cost of Hamming graph H
    - (vertices=probes, distance = Hamming)
  - on 2-dim grid graph G2 (N x N array, edges b/w neighbors)

- 3-dim (asynchronous) Array Design Problem:
  - Minimize cost of placement and embedding of Hamming graph H'
    - (vertices=probes, distance = Hamming b/w embedded probes)
  - on 2-dim grid graph G2 (N x N array, edges b/w neighbors)

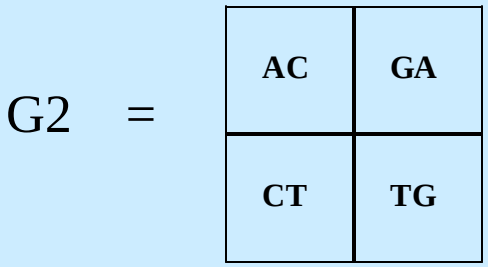
Lower Bound

- Lower bound (LB) for the grid weight:
  Sum of distances to 4 closest neighbors minus weight of 4N heaviest arks
- Synchronous LB distance = Hamming distance
- Asynchronous LB distance =50-|Longest Common Subsequence|
  - Although the LB = 8 conflicts, the best placement has 10 conflicts

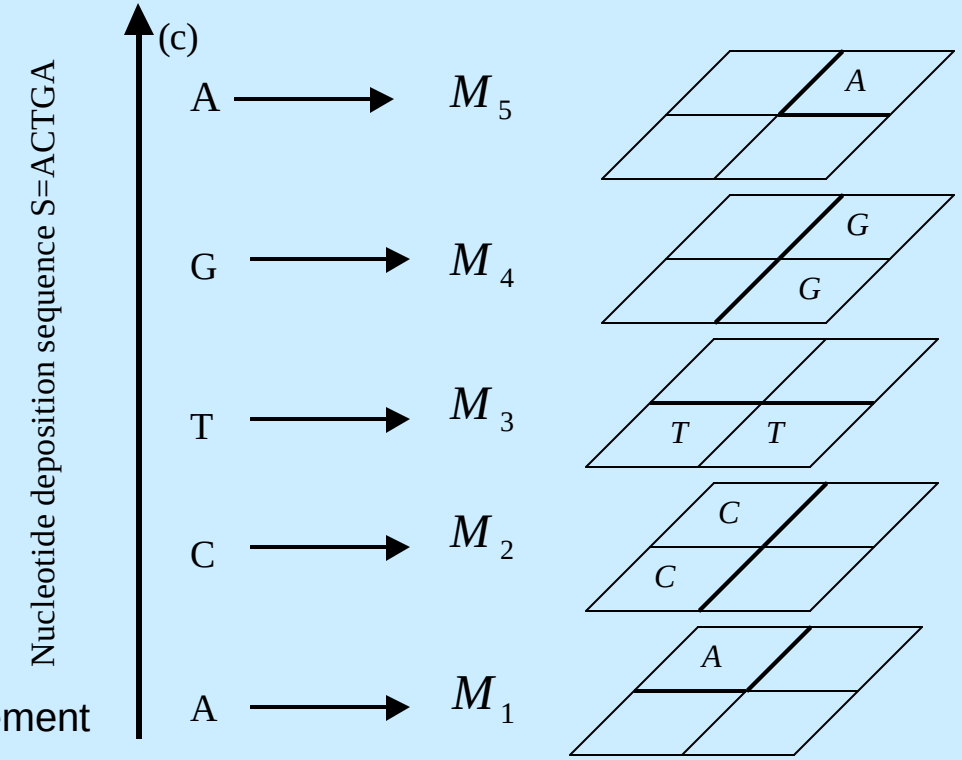
(a)



(b)

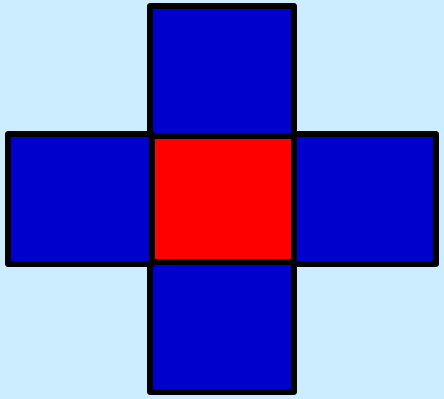


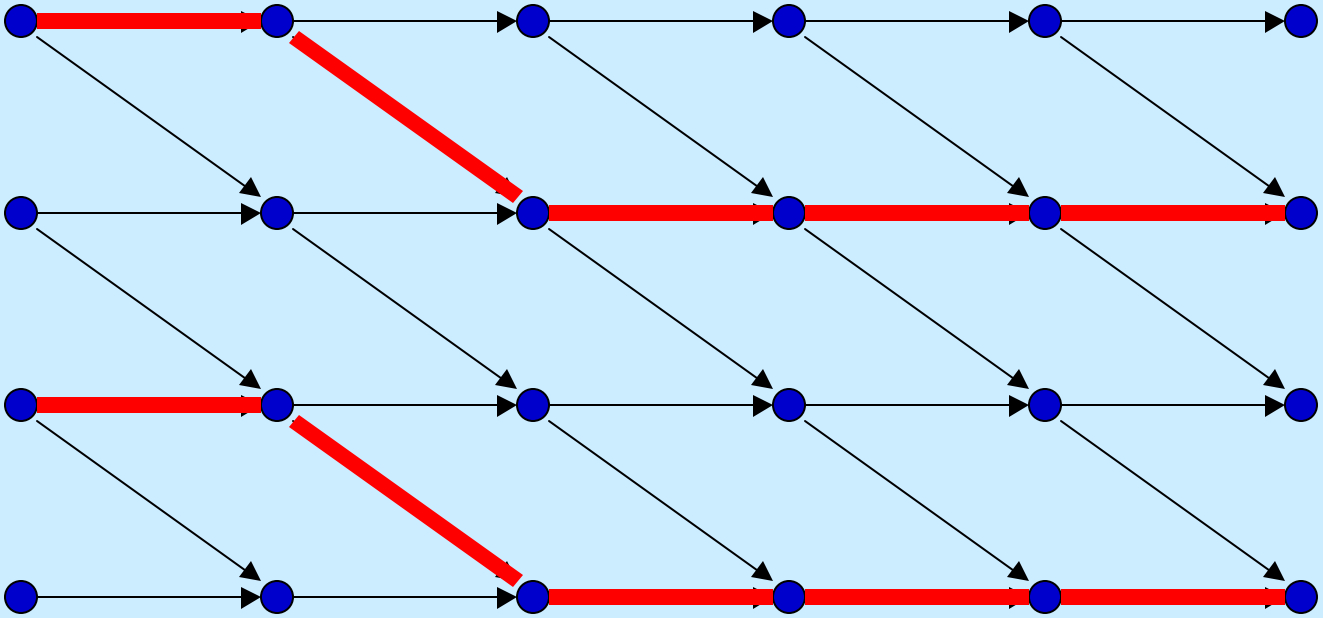
Post-placement LB =
asynchronous LB applied to placement



Optimal Probe Alignment

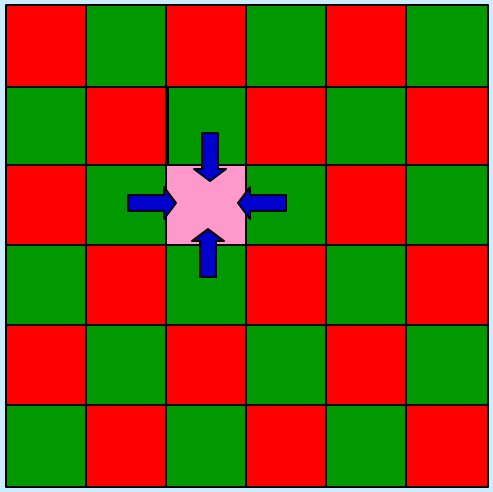
- Given nucleotide deposition sequence
- Find the best alignment of probe with respect to 4 embedded neighbors

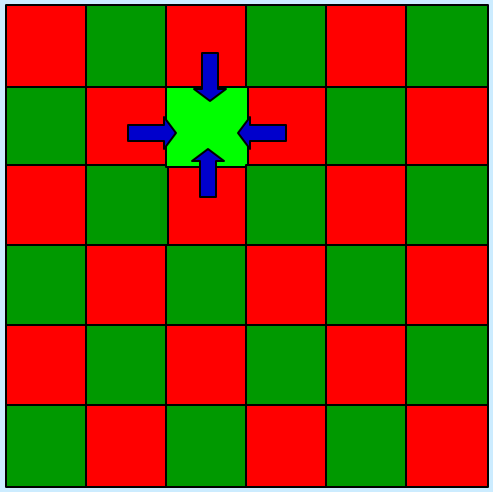




Post-placement Optimization Methods

- Asynchronous re-embedding after 2-dim placement
  - Greedy Algorithm
    - While there exist probes to re-embed with gain
      - Optimally re-embed the probe with the largest gain
  - Batched greedy: speed-up by avoiding recalculations
  - Chessboard Algorithm
    - While there there is gain
      - Re-embed probes in green sites
      - Re-embed probes in green sites





Experimental Results

1. Array size 20x20 – 500x500
2. All results = averages over 10 sets of probes
3. Each probe is of length 25 generated uniformly at random
4. Runtime in CPU seconds of SGI Origin 2000 and 1.4GHz Xeon

Chip Size	Sync LB Cost	TSP+1-Threading			Epitaxial			Async LB	
		Cost	%Gap	CPU	Cost	%Gap	CPU	Cost	%Gap
20	19242	23401	21.6	0	22059	14.6	0	10236	-46.8
40	72240	92267	27.7	3	85371	18.2	6	38414	-46.8
60	156149	204388	30.9	15	187597	20.1	32	83132	-46.8
80	268525	358945	33.7	46	327924	22.1	104	144442	-46.2
100	410019	554849	35.3	113	505442	23.3	274	220497	-46.2
200	1512014	2140903	41.6	1901	1924085	27.3	4441	798708	-47.2
300	3233861	4667882	44.3	12028	—	—	—	—	—
500	8459958	12702474	50.1	109648	—	—	—	—	—

Placement heuristic and lower bounds

Post-placement Experiments

Optimization of the probe embedding after TSP+1-Threading

Chip Size	LCS LB cost	Initial %Gap	Batched Greedy %Gap	Batched Greedy #Iter	Batched Greedy CPU	Chessboard %Gap	Chessboard #Iter	Chessboard CPU	2×1 Chessboard %Gap	2×1 Chessboard #Iter	2×1 Chessboard CPU
20	14859	57.5	24.3	8.8	2	19.1	13.3	2	18.0	11.1	18
40	59500	55.1	25.0	8.9	7	20.0	12.7	9	18.8	11.8	77
60	133165	53.5	25.3	8.5	15	20.2	13.2	20	19.0	11.9	175
80	234800	52.9	25.7	8.7	27	20.5	13.4	35	19.3	12.0	315
100	364953	52.0	25.7	8.8	40	20.5	13.6	54	19.4	11.7	480
200	1425784	50.2	26.3	8.3	154	20.9	13.9	221	19.7	11.8	1915
300	3130158	49.1	26.7	8.0	357	21.5	13.6	522	21.6	11.0	4349
500	8590793	47.9	27.1	8.0	943	21.4	14.0	1423	20.2	12.0	15990

Optimization of the probe embedding after epitaxial placement

Chip Size	LCS LB cost	Initial %Gap	Batched Greedy %Gap	Batched Greedy #Iter	Batched Greedy CPU	Chessboard %Gap	Chessboard #Iter	Chessboard CPU	2×1 Chessboard %Gap	2×1 Chessboard #Iter	2×1 Chessboard CPU
20	14636	50.7	22.7	8.4	1	17.9	12.3	2	17.0	10.5	17
40	58494	45.9	22.4	8.1	6	17.7	12.9	8	16.8	11.4	74
60	130636	43.6	22.0	8.0	14	17.5	12.3	18	16.6	11.2	164
80	230586	42.2	21.7	8.0	24	17.2	12.6	33	16.3	11.1	289
100	357800	41.3	21.6	8.0	37	17.1	12.1	48	16.3	11.3	461
200	1395969	37.8	20.8	7.1	130	16.4	12.0	190	15.6	10.7	1779

Summary and Ongoing Research

- Contributions:
  - Epitaxial placement → reduces by extra 10% over the previously best known
  - Asynchronous placement problem formulation
  - Postplacement improvement by extra 15.5-21.8%
  - Lower bounds
- Further directions:
  - Comparison on industrial benchmarks
  - SNP's
  - Empty cells