

<https://doi.org/10.7236/JIIBC.2024.24.2.105>
JIIBC 2024-2-16

격자 그래프의 최소선형배열 알고리즘

Algorithm for a Minimum Linear Arrangement(MinLA) of Lattice Graph

이상운*

Sang-Un Lee *

요약 격자 그래프의 최소 선형 배열(MinLA)은 선형 복잡도 $O(n)$ 의 근사 알고리즘이 적용되고 있으며, 33×33 격자의 최적 MinLA는 31,680으로 알려져 있다. 본 논문은 격자의 정확한 해 MinLA를 복잡도 $O(1)$ 으로 구하는 분할배열 알고리즘을 제안하였다. 분할배열 알고리즘은 컨테이너에 박스를 넣는 방법으로 m 행을 r_1, r_2, r_3 로, n 열을 c_1, c_2, c_3 로 분할하여 7개 컨테이너를 얻고 규칙을 가지도록 분할한다. 분할된 박스들에 있는 정점들 위치 순서로 번호를 부여하여 MinLA를 구한다. $m, n \geq 11$ 에 대해 C_2, C_4, C_6 박스 크기를 2씩 증가시키면서 MinLA가 증가할 때까지 반복 수행한다. 이 과정은 $m, n \leq 100$ 에 대해 최대 4회 반복 수행하는 특징이 있다. 제안된 알고리즘은 $m=n$ 과 $m \neq n$ 인 모든 격자에 적용할 수 있다. 분할배열 알고리즘을 $2 \leq n \leq 100$ 격자에 적용하였으며, 33×33 과 100×100 격자에 대해 기존 알고리즘들보다 월등히 좋은 최적의 결과를 얻었다. 제안된 알고리즘은 간단하면서도 보다 정확한 해를 얻을 수 있어 m, n 이 무한히 크더라도 쉽게 해를 얻을 수 있어 VLSI 회로 설계 분야에 응용이 될 수 있을 것이다.

Abstract This paper deals with the minimum linear arrangement(MinLA) of a lattice graph, to which an approximate algorithm of linear complexity $O(n)$ remains as a viable solution, deriving the optimal MinLA of 31,680 for 33×33 lattice. This paper proposes a partitioning arrangement algorithm of complexity $O(1)$ that delivers exact solution to the minimum linear arrangement. The proposed partitioning arrangement algorithm could be seen as loading boxes into a container. It firstly partitions m rows into r_1, r_2, r_3 and n columns into c_1, c_2, c_3 , only to obtain 7 containers. Containers are partitioning with a rule. It finally assigns numbers to vertices in each of the partitioned boxes location-wise so as to obtain the MinLA. Given $m, n \geq 11$, the size of boxes C_2, C_4, C_6 is increased by 2 until an increase in the MinLA is detected. This process repeats itself 4 times at maximum given $m, n \leq 100$. When tested to lattice in the range of $2 \leq n \leq 100$, the proposed algorithm has proved its universal applicability to lattices of both $m=n$ and $m \neq n$. It has also obtained optimal results for 33×33 and 100×100 lattices superior to those obtained by existing algorithms. The minimum linear arrangement algorithm proposed in this paper, with its simplicity and outstanding performance, could therefore be also applied to the field of Very Large Scale Integration circuit where m, n are infinitely large.

Key Words : Minimum linear arrangement, Optimal linear ordering, Layout problem, Lattice, Mesh

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2023년 10월 21일, 수정완료 2024년 3월 10일
게재확정일자 2024년 4월 5일

Received: 21 October, 2023 / Revised: 10 March, 2024 /
Accepted: 5 April, 2024

*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National
University, Korea

I. 서 론

그래프 배치 문제(graph layout problems, GLP)는 주어진 그래프 $G=(V,E)$ 에 대해 특정 목표 비용(objective cost)을 최적화할 수 있도록 정점들을 선형 배치(linear layout) 하는 문제이다. 여기서 목표 비용은 대역폭(bandwidth), 최소선형배열(minimum linear arrangement, MinLA), 절단 폭(cutwidth), 수정된 절단(modified cut), 정점 분리(vertex separation), 절단 합 프로파일(sumcut profile), 간선 양분(edge bisection), 정점 양분(vertex bisection)이 될 수 있다. 이들 문제는 망 최적화, VLSI 회로 설계, 정보 검색, 수치해석, 계산 생물학, 그래프 이론, 일정, 고고학 등 다양한 분야에 응용되고 있다.^[1-3] 배치(layout)는 선형 배열(linear arrangement), 분류(labeling) 또는 번호부여(numbering)라고도 부른다.

대역폭 문제는 무 방향 그래프(undirected graph) $G=(V,E)$, $n=|V|$ 의 각 정점들을 $f: V \rightarrow \{1,2,\dots,n\}$ 함수로 선형으로 배열 하였을 때 각 정점들에 부속된 간선들의 길이 중에서 최대 길이를 최소화하도록 정점들을 배열하는 문제로 대역폭의 최대 거리는 $B(f,G) = \max\{|f(v) - f(w)| : \{u,v\} \in E\}$ 로 정의되며, 찾고자 하는 대역폭은 $B(G) = \min\{B(f,G)\}$ 이다. 반면에, 최소선형배열문제는 대역폭 문제의 최대(maximum)를 합(sum)으로 교체하는 개념으로 모든 인접 정점들(adjacent vertices) 간의 거리인 간선 가중치(edge weight)의 합을 최소로 하는 $MinLA = \min \sum_{vw} \{|f(v) - f(w)| : \{v,w\} \in E\}$ 이 되도록 정점들을 선형으로 배열하는 문제이다. MinLA를 최적 선형 순서(optimal linear ordering), 간선 합 문제(edge sum problem) 또는 최소-1-합(minimum-1-sum)이라고도 한다. MinLA 문제는 VLSI 배치 설계에서 총 배선의 길이를 최소화하는 문제로 적용된다.

MinLA 문제는 일반적으로 NP-난제(NP-Hard)로 알려져 있다. 그러나 특정 그래프인 Rectangular mesh와 Square mesh에 대해서는 $O(n)$ 의 다항시간 알고리즘이 제안되어 있다.^[1,2]

본 논문은 격자(lattice, grid, square mesh 또는 rectangular mesh) 그래프에 대해 최적의 MinLA를 $O(1)$ 으로 구하는 알고리즘을 제안한다. 2장에서는 격자에 대해 MinLA를 구한 연구 결과를 고찰한다. 3장에서는 최적의 MinLA를 $O(1)$ 으로 구하는 알고리즘을 제안한다. 4장에서는 다양한 격자들을 대상으로 제안된 알고리즘을 적용하여 성능을 검증한다.

II. 관련 연구와 연구 배경

Petit^[1,4]와 Díaz et al.^[2]에서 인용된 그림 1의 5×5 격자를 대상으로 MinLA 개념을 고찰해 보자. 5×5 의 25개 정점들을 (a)와 같이 일반적인 배열 방법으로 배치할 경우 인접 정점들 간의 거리의 합은 120이 된다. 이를 상한 값(upper bound)^[5]이라 하자. 그러나 (b)와 같이 최적으로 배열하면 116을 얻을 수 있다.

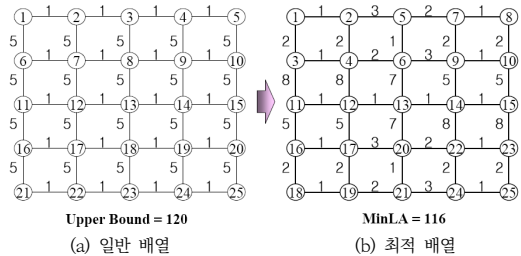


그림 1. 5×5 격자 그래프의 MinLA
Fig. 1. MinLA for 5×5 lattice graph

2차원 격자 구조에 번호를 부여 하는 방법을 Space filling problem이라 한다. Luttamaguzi et al.^[6]은 그림 2와 같이 z -곡선, Gray 곡선과 Hilbert 곡선을, Niedermeier^[7]는 Hilbert 곡선을, Wikipedia^[8]은 Dragon 곡선, Gosper 곡선, Hilbert 곡선, Moore 곡선, Sierpiński 곡선, z -곡선 등 다양한 방법을 제시하고 있다.

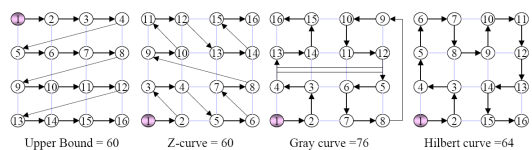


그림 2. 4×4 격자 번호 부여 방법
Fig. 2. Labeling methods for 4×4 lattice

MinLA 문제를 해결하는데 이들 곡선을 적용하면 4×4 격자의 경우 상한 값 60에 대해 z -곡선은 60으로 동일한 결과를 얻지만 다른 곡선들은 가중치 합이 60을 초과한다. 또한, 5×5 에 z -곡선을 적용하는데도 어려움이 있으며, 최적 값 116을 얻지 못한다. 결국, MinLA를 구하기 위해 이들 곡선을 적용할 수 없다.

3장에서는 다양한 격자의 MinLA를 간단히 구하는 상수 알고리즘을 제안한다.

III. 분할배열 MinLA 알고리즘

주어진 $m \times n (m = n \text{ or } m \neq n)$ 격자에 대한 MinLA를 수행 복잡도 $O(1)$ 로 구하는 알고리즘은 다음과 같이 수행된다.

- (1) $n, m \geq 5$ 인 격자에 대해 컨테이너 분할을 수행한다. 이를 표준분할이라 하자.
 - 행: $r_1 = r_3 = m/3$ 반올림, $r_2 = n - (r_1 + r_3)$ 의 컨테이너 분할.
 - 열: $c_1 = c_3 = n/3$ 반올림, $c_2 = m - (c_1 + c_3)$ 의 컨테이너 분할.
 - 컨테이너: $C_1 = r_1 c_1, C_2 = r_1 c_2, C_3 = r_1 c_3, C_4 = r_2 (c_1 + c_2 + c_3), C_5 = r_3 c_1, C_6 = r_3 c_2, C_7 = r_3 c_3$
- (2) 컨테이너에 들어 있는 정점들에 대해 박스 분할을 수행한다.
 - C_1 컨테이너 : $\{(1,1), (1,2), (2,1), (2,2)\}, \{(1,3), (2,3)\}, \{(3,1), (3,2), (3,3)\}, \dots$ 으로 상 삼각행렬 (upper triangle matrix)은 열 박스로, 하 삼각행렬 (lower triangle matrix)은 행 박스 형태로 분할.
 - C_3 : C_1 과 좌우 대칭, C_5 는 C_1 과 상하 대칭, C_7 은 C_5 와 좌우 대칭이 되도록 분할.
 - C_2, C_6 : 세로 박스로 분할, C_4 : 가로 박스로 분할.
- (3) C_1 컨테이너로부터 박스가 쌓여져 있는 순서대로 정점들에 번호를 부여하여 MinLA를 구한다.
- (4) $n, m \geq 11$ 에 대해 $r_1 = r_1 - 1, r_3 = r_3 - 1, r_2 = r_2 + 2, c_1 = c_1 - 1, c_3 = c_3 - 1, c_2 = c_2 + 2$ 로 분할하면서 MinLA가 증가할 때까지 (1) ~ (3) 과정을 반복 수행한다. 이를 추가 분할이라 하자.

제안된 알고리즘을 분할 배열 (divide and arrangement) 알고리즘이라 하자. 분할 배열 알고리즘은 $m, n \leq 10$ 은 항상 표준분할만을 수행하며, $11 \leq m, n \leq 33$ 은 추가분할을 2회 수행한다. $m, n = 66$ 은 추가분할을 3회, $m, n = 99$ 와 100 은 추가분할을 4회 수행한다. 결국, 분할 배열 알고리즘은 수행 복잡도는 $O(1)$ 이다. 12×12 격자에 제안된 알고리즘을 적용한 결과는 그림 3에 제시되어 있다.

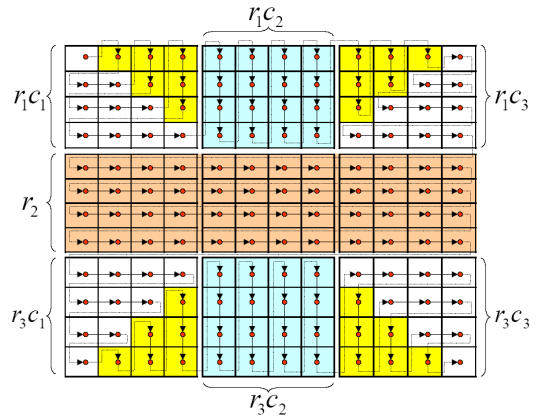
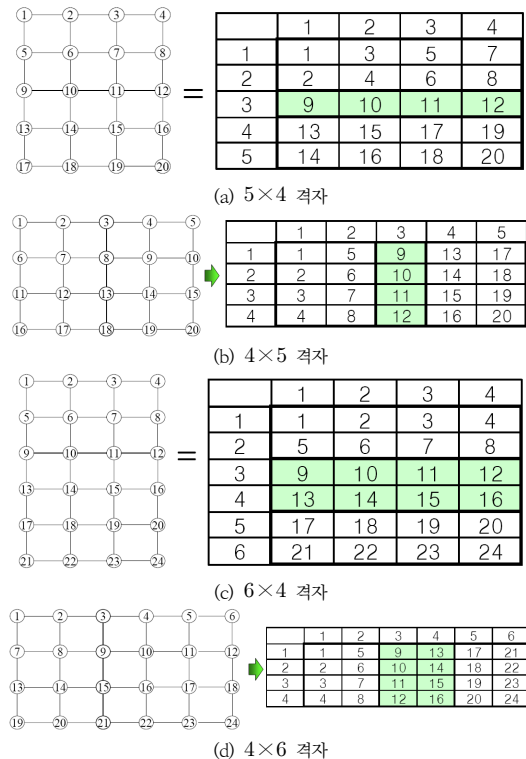
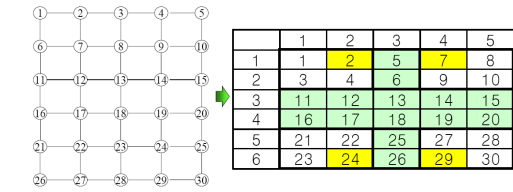


그림 3. 12×12 격자의 분할배열 알고리즘
 Fig. 3. Divide and arrangement algorithm for 12×12 lattice

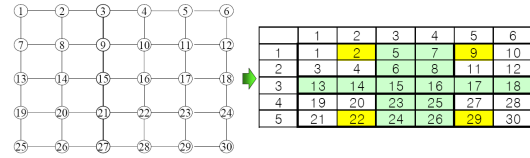
IV. 알고리즘 적용 및 결과 분석

본 장에서는 먼저, $m, n \leq 15$ 인 $m \neq n$ 과 $m = n$ 인 다양한 격자에 대해 MinLA를 구하여 본다. 분할 배열 결과는 그림 4와 그림 5에, MinLA는 표 1에 제시하였다.

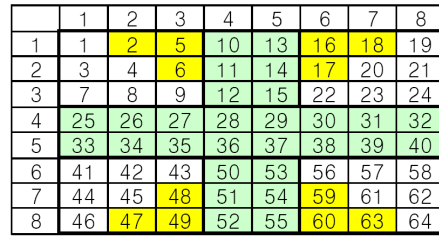




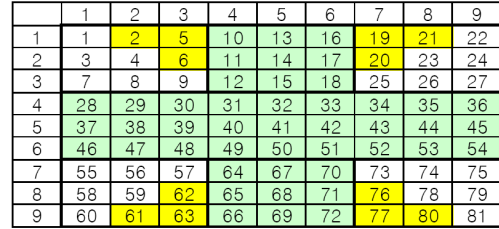
(e) 6×5 격자



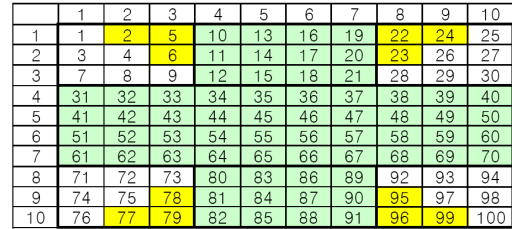
(f) 5×6 격자



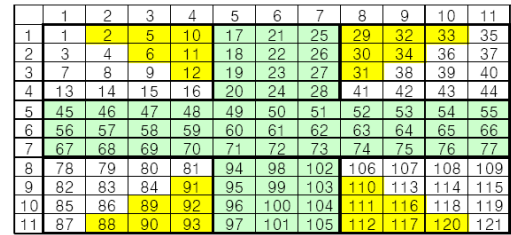
(g) 8×8 격자



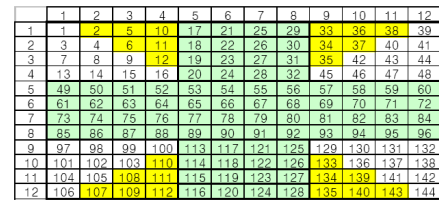
(h) 9×9 격자



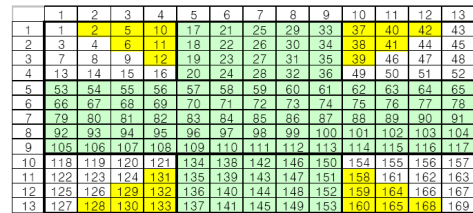
(i) 10×10 격자



(j) 11×11 격자

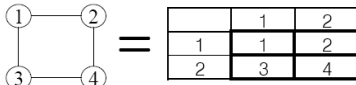


(k) 12×12 격자

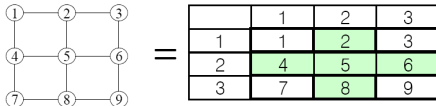


(l) 13×13 격자

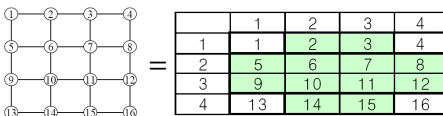
그림 4. $n \neq m$ 격자
Fig. 4. $n \neq m$ lattice



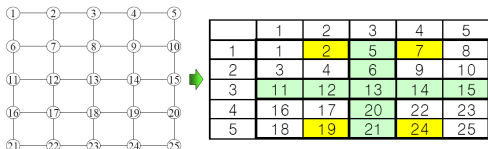
(a) 2×2 격자



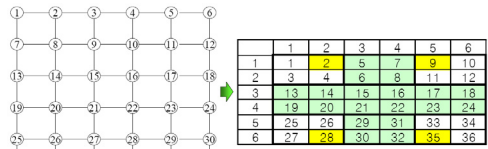
(b) 3×3 격자



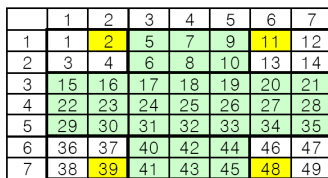
(c) 4×4 격자



(d) 5×5 격자



(e) 6×6 격자



(f) 7×7 격자

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	1	2	5	10	17	21	25	29	33	37	41	44	46	47
2	3	4	6	11	18	22	26	30	34	38	42	45	48	49
3	7	8	9	12	19	23	27	31	35	39	43	50	51	52
4	13	14	15	16	20	24	28	32	36	40	53	54	55	56
5	57	58	59	60	61	62	63	64	65	66	67	68	69	70
6	71	72	73	74	75	76	77	78	79	80	81	82	83	84
7	85	86	87	88	89	90	91	92	93	94	95	96	97	98
8	99	100	101	102	103	104	105	106	107	108	109	110	111	112
9	113	114	115	116	117	118	119	120	121	122	123	124	125	126
10	127	128	129	130	131	132	133	134	135	136	137	138	139	140
11	141	142	143	144	145	146	147	148	149	150	151	152	153	154
12	145	146	147	154	158	162	166	170	174	178	185	188	189	190
13	148	149	152	155	159	163	167	171	175	179	186	191	193	194
14	150	151	153	156	160	164	168	172	176	180	187	192	195	196

(m) 14×14 격자

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	1	2	5	10	17	26	31	36	41	46	51	55	58	60	61
2	3	4	6	11	18	27	32	37	42	47	52	56	59	62	63
3	7	8	9	12	19	28	33	38	43	48	53	57	64	65	66
4	13	14	15	16	20	29	34	39	44	49	54	67	68	69	70
5	21	22	23	24	25	30	35	40	45	50	71	72	73	74	75
6	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
7	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
8	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
9	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
10	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
11	151	152	153	154	155	176	181	186	191	196	201	202	203	204	205
12	156	157	158	159	172	177	182	187	192	197	206	210	211	212	213
13	160	161	162	169	173	178	183	188	193	198	207	214	217	218	219
14	163	164	167	170	174	179	184	189	194	199	208	215	220	222	223
15	165	166	168	171	175	180	185	190	195	200	209	216	221	224	225

(n) 15×15 격자

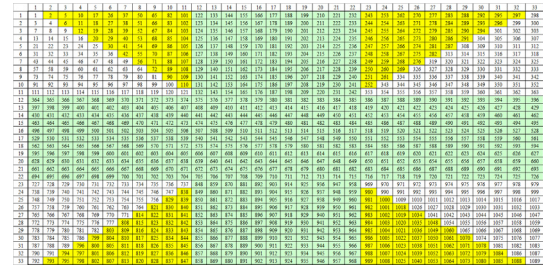
그림 5. $n=m$ 격자
 Fig. 5. $n=m$ lattice

표 1. 실험 격자의 MinLA
 Table 1. MinLA for experimental lattices

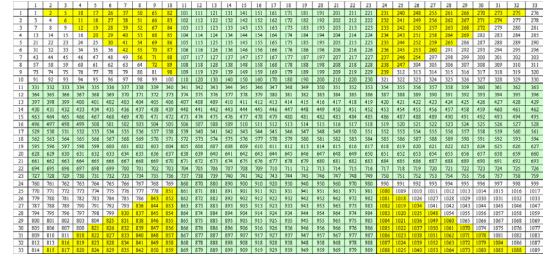
격자 그래프	분할배열			
	분할비율	간선 가중치 합 (MinLA)	상한치	비용 감소
5x4	행: 2-1-2	79	79	0
4x5	열: 2-1-2	79	91	-12
6x4	행: 2-2-2	98	98	0
4x6	열: 2-2-2	98	128	-30
6x5	행: 2-2-2, 열: 2-1-2	145	149	-4
5x6	행: 2-1-2, 열: 2-2-2	159	169	-10
3x3	행: 1-1-1, 열: 1-1-1	24	24	0
4x4	행: 1-2-1, 열: 1-2-1	60	60	0
5x5	행: 2-1-2, 열: 2-1-2	116	120	-4
6x6	행: 2-2-2, 열: 2-2-2	200	210	-10
7x7	행: 2-3-2, 열: 2-3-2	318	336	-18
8x8	행: 3-2-3, 열: 3-2-3	472	504	-32
9x9	행: 3-3-3, 열: 3-3-3	668	720	-56
10x10	행: 3-4-3, 열: 3-4-3	914	990	-76
11x11	행: 4-3-4, 열: 4-3-4 행: 3-5-3, 열: 3-5-3	1,215 1,216	1,320	-105 -104
12x12	행: 4-4-4, 열: 4-4-4 행: 3-6-3, 열: 3-6-3	1,568 1,580	1,716	-148 -136
13x13	행: 4-5-4, 열: 4-5-4 행: 3-7-3, 열: 3-7-3	1,988 2,058	2,184	-196 -126
14x14	행: 5-4-5, 열: 5-4-5 행: 4-6-4, 열: 4-6-4 행: 3-8-3, 열: 3-8-3	2,482 2,480 2,518	2,730	-248 -250 -212
15x15	행: 5-5-5, 열: 5-5-5 행: 4-6-4, 열: 4-6-4	3,040 3,050	3,360	-320 -310

일반 배열 방법의 상한 값과 분할 배열로 변환시켜도 동일한 형태인 5x4, 2x2, 3x3, 4x4 격자를 제외한 다른 격자들은 모두 분할 배열로 MinLA를 얻는다.

추가로 33x33, 66x66, 99x99, 100x100 격자에 적용하였으며, 33x33 격자의 분할 배열 방법은 그림 6에, MinLA는 표 2에 제시되어 있다.



(a) 11-11-11 분할



(b) 10-13-10 분할

그림 6. 33x33 격자의 분할배열 알고리즘
 Fig. 6. Divide and arrangement algorithm for 33x33 lattice

표 2. 추가 실험 격자의 MinLA
 Table 2. MinLA for additional experimental lattices

격자 그래프	분할배열			
	분할비율	간선 가중치 합 (MinLA)	상한치	비용 감소
33x33	행: 11-11-11, 열: 11-11-11	31,724	35,904	-4,180 -4,224 -4,176
	행: 10-13-10, 열: 10-13-10	31,680		
	행: 09-15-09, 열: 09-15-09	31,728		
66x66	행: 22-22-22, 열: 22-22-22	251,240	287,430	-36,190 -36,520 -36,670 -36,636
	행: 21-24-21, 열: 21-24-21	250,910		
	행: 20-26-20, 열: 20-26-20	250,760		
	행: 19-28-19, 열: 19-28-19	250,794		
99x99	행: 33-33-33, 열: 33-33-33	844,884	970,200	-125,316 -126,174 -126,764 -127,082
	행: 32-35-32, 열: 32-35-32	844,026		
	행: 31-37-31, 열: 31-37-31	843,436		
	행: 30-39-30, 열: 30-39-30	843,118		
100x100	행: 29-41-29, 열: 29-41-29	843,542	999,900	-126,658 -129,536 -130,324 -131,080 -131,040
	행: 33-34-33, 열: 33-34-33	870,364		
	행: 32-36-32, 열: 32-36-32	869,576		
	행: 31-38-31, 열: 31-38-31	869,060		
	행: 30-40-30, 열: 30-40-30	868,820		
행: 29-42-29, 열: 29-42-29	868,860			

제안된 알고리즘의 적합성을 검증하기 위해 그림 7과 같이 33x33 격자에 대한 기존의 알고리즘^[1,3,4,9-16]과 성능을 검증해 본다. 33x33 격자의 상한 값은 35,904이며, SAN, BFS, Spec, Random, Neive, Hillc2, Worst 인 경우는 상한 값을 초과하기 때문에 알고리즘으로 채택이 불가하다.

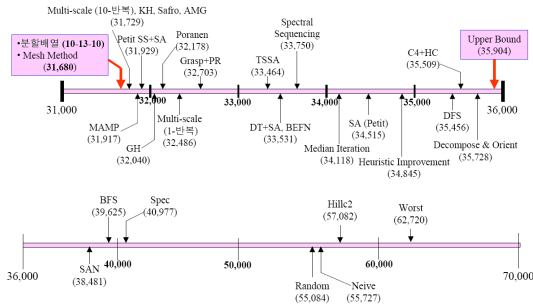


그림 7. 33x33 격자의 MinLA 알고리즘 성능
Fig. 7. MinLA algorithm's performance for 33x33 lattice

$32,000 \leq MinLA < 35,9045,904$ 에도 다양한 알고리즘들이 존재한다. $31,680 \leq MinLA < 32,000 < 32,000$ 범위에 속하는 알고리즘들은 제안된 알고리즘을 포함하여 8개에 불과함을 알 수 있다. 제안된 알고리즘의 10-13-10 분할은 Mesh Method와 동일하게 31,680의 가장 좋은 성능을 나타냄을 알 수 있다. 그러나 Mesh Method는 실험 결과 정점들의 배열 방법을 제시하지 않아 실제로 적용에 어려움이 있다. 또한, 100x100 격자에 대해 Safro et al.^[13]의 결과와 비교하였다. 100x100 격자의 상한치는 999,900이며, 알려진 최적치는 868,820이다. Safro et al.^[13]은 Multilevel Weighted Edge Contraction 방법을 적용하여 880,234를 얻는데 반해 제안된 알고리즘은 정확하게 최적치 868,820을 얻는데 성공하였다. 결국, 제안된 알고리즘은 가장 단순하면서도 최적의 결과를 얻는 MinLA 알고리즘으로 적용 할 수 있다.

V. 결 론

본 논문은 격자 그래프에 대한 MinLA를 구하는 $O(1)$ 복잡도의 분할배열 알고리즘을 제안하였으며, 기존의 근사 알고리즘들에 비해 최적의 성능을 나타냄을 보였다.

제안된 알고리즘은 m 행을 r_1, r_2, r_3 로, n 열을 c_1, c_2, c_3 로 분할하여 $C_1 = r_1c_1, C_2 = r_1c_2, C_3 = r_1c_3, C_4 = r_2(c_1 + c_2 + c_3), C_5 = r_3c_1, C_6 = r_3c_2, C_7 = r_3c_3$ 의 7개 컨테

이너를 얻는다. C_1, C_3, C_5, C_7 컨테이너들은 내부의 정점들을 상 삼각행렬과 하 삼각행렬 형태의 박스들로 분할한다. C_2, C_6 컨테이너는 세로 박스 형태로, C_4 컨테이너는 가로 박스 형태로 분할한다. C_1 컨테이너부터 박스가 쌓여 있는 순서대로 정점들에 번호를 부여하여 MinLA를 얻는다. $m, n \geq 11$ 에 대해 C_2, C_4, C_6 컨테이너의 크기를 2씩 증가시키면서 MinLA가 증가할 때까지 반복 수행하여 최소의 MinLA 값을 선택한다.

제안된 분할배열 알고리즘을 33x33, 66x66, 99x99와 100x100 격자에 적용하였으며, 33x33과 100x100 격자에 대해 기존의 알고리즘들보다 월등히 좋은 최적의 MinLA를 얻는데 성공하였다. 분할 배열 알고리즘은 단순하고 최적의 성능을 나타내어 VLSI 회로 설계분야에 쉽게 적용할 수 있을 것이다.

References

- [1] J. Petit, "Experiments on the Minimum Linear Arrangement Problem," ACM Journal of Experimental Algorithmics, Vol. 82003, Article No. 23, pp. 1-29, Dec. 2003, <https://doi.org/10.1145/996546.996554>
- [2] J. Diaz, J. Petit, and M. Serna, "A Survey on Graph Layout Problems," ACM Computing Surveys, Vol. 34, No. 3, pp. 313-356, Sep. 2002, <https://doi.org/10.1145/568522.568523>
- [3] Y. Koren and D. Harel, "A Multi-Scale Algorithm for the Linear Arrangement Problem," 28th International Workshop on Graph-Theoretic Concepts in Computer Science, pp. 296-309, Jan. 2002, https://doi.org/10.1007/3-540-36379-3_26
- [4] J. Petit, "Approximation Heuristics and Benchmarkings for the MinLA Problem," Building Bridges Between Theory and Applications, pp. 112-128, Feb. 1998.
- [5] M. Yannakakis, "Computing the Minimum Fill-In is NP-Complete," SIAM Journal on Matrix Analysis and Applications, Vol. 2, No. 1, pp. 77-79, Mar. 1981, <https://doi.org/10.1137/0602010>
- [6] J. Luttamaguzi, M. Pelsmajer, Z. Shen, and B. Yang, "Integer Programming Solutions for Several Optimization Problems in Graph Theory," DIMACS Technical Report 2005-02, Jan. 2005.
- [7] R. Niedermeier, K. Reinhardt, and P. Sanders, "Towards Optimal Locality in Mesh-Indexings," Discrete Applied Mathematics, Vol. 117, No. 1-3, pp. 211-237, Mar. 2002, [https://doi.org/10.1016/S0166-218X\(00\)00326-7](https://doi.org/10.1016/S0166-218X(00)00326-7)
- [8] Wikipedia, "Space-Filling Curve," http://en.wikipedia.org/wiki/Space-filling_curve, Retrieved Apr. 2023.

- [9] T. Poranen, "A Genetic Hillclimbing Algorithm for the Optimal Linear Arrangement Problem," *Fundamenta Informaticae*, Vol. 68, No. 4, pp. 333-356, Dec. 2005.
- [10] B. Y. Reuven, E. Guy, F. Jon, and N. Joseph, "Computing an Optimal Orientation of a Balanced Decomposition Tree for Linear Arrangement Problems," *Journal of Graph Algorithms and Applications*, Vol. 5, No. 4, pp. 1-27, Oct. 2001, https://doi.org/10.1142/9789812794741_0016
- [11] R. T. Eduardo, "Best Known Solutions for the MinLA Problem," Information Technology Laboratory, Cinvestav, Tamaulipas, <https://www.tamps.cinvestav.mx/~ertello/MinLA/bestSolutionsMinLA-120107.pdf>, Retrieved Apr. 2023.
- [12] A. R. S. Amaral, A. Caprara, A. N. Letchford, and J. J. Salazar-Gonzalez, "A Cutting Plane Algorithm for the Linear Arrangement Problem," *European Journal of Operational Research*, Vol. 141, No. 2, pp. 274-294, Sep. 2002, [https://doi.org/10.1016/S0377-2217\(02\)00125-X](https://doi.org/10.1016/S0377-2217(02)00125-X)
- [13] I. Safro, D. Ron, and A. Brandt, "Graph Minimum Linear Arrangement by Multilevel Weighted Edge Contractions," *Journal of Algorithms*, Vol. 60, No. 1, pp. 24-41, Jul. 2006, <https://doi.org/10.1016/j.jalgor.2004.10.004>
- [14] J. Diaz, M. D. Penrose, J. Petit, and M. Serna, "Layout Problems on Lattice Graphs," *International Computing and Combinatorics conference*, pp. 103-112, Jan. 1999, https://doi.org/10.1007/3-540-48686-0_10
- [15] R. T. Eduardo, J. K. Hao, and T. J. Jose, "Memetic Algorithms for the MinLA Problem," *International Conference on Artificial Evolution*, pp. 73-84, Oct. 2005, https://doi.org/10.1007/11740698_7
- [16] A. Duarte, J. J. Pantrigo, V. Campos, and R. Marti, "Heuristics for the Minimum Linear Arrangement Problem," *Universidad Rey Juan Carlos, Spain*, pp. 1-13, Jul. 2008.

저 자 소 개

이 상 윤(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr