Journal of The Korea Society of Computer and Information
Vol. 20 No. 7, pp. 41-47, July 2015 www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.7.041

Maximum Degree Vertex Central Located Algorithm for Bandwidth Minimization Problem

Sang-Un Lee*

Abstract

The bandwidth minimization problem (BMP) has been classified as NP-complete because the polynomial time algorithm to find the optimal solution has been unknown yet. This paper suggests polynomial time heuristic algorithm is to find the solution of bandwidth minimization problem. To find the minimum bandwidth $\phi^* = {}_{\min}\phi(G), \phi(G) = {}_{\max}\{|f(v_i) - f(v_j) : v_i, v_j \in E\}$ for given graph $G = (V,E)$, $m = |V|, n = |E|$, the proposed algorithm sets the maximum degree vertex v_i in graph G into global central point (GCP), and labels the median value $\lceil m+1/2 \rceil$ between $[1,m]$ range. The graph G is partitioned into subgroup, the maximum degree vertex in each subgroup is set to local central point (LCP), and we adjust the label of LCP per each subgroup as possible as minimum distance from GCP. The proposed algorithm requires $O(mn)$ time complexity for label to all of vertices. For various twelve graph, the proposed algorithm can be obtains the same result as known optimal solution. For one graph, the proposed algorithm can be improve on known solution.

▸ Keyword : Labeling, bandwidth, Maximum degree, Global central point, Local central point

I. Introductions

대역폭 최소화 문제 (bandwidth minimization problem, BMP)는 주어진 그래프 $G=(V,E)$, $m=|V|, n=|E|, e=(v_i,v_j) \in E$ 의 m개 정점들에 $[1,m]$ 범위의 서로 다른 (distinct) 번호 $f(v_i)$를 부여하여 일렬로 배치할 경우, 최소 대역폭 (간선들 중 최대 거리) $\phi^* = {}_{\min}\phi(G), \phi(G) = {}_{\max}\{|f(v_i) - f(v_j) : v_i, v_j \in E|$을 갖는 배치도 (layout)를 찾는 문제이다[1].

BMP는 정확한 해를 다항시간으로 구하는 알고리즘이 현재까지 제안되지 않고 있어 NP-완전 (NP-complete)으로 분류된 난제이다[2,3]. 이와 같이 난제임에도 불구하고 BMP는 $ax=b$ 의 선형대수 방정식을 갖는 정칙체계를 효율적으로 풀기 위해, 편미분방정식의 근사 해를 유한요소 방법으로 구하기 위해, 대규모 전력전송시스템, 회로 설계, 하이퍼텍스트 배치, 화학 반응속도, 수치화된 지구물리학, 데이터저장소, VLSI 설계와 망 생존성 등 다양한 분야에 적용되고 있는 문제이다[1].

BMP의 최소 대역폭 해 ϕ에 대해, 선 그래프 (line graph) P_m 의 $\phi(P_m)=1$, 완전 그래프 (complete graph) K_m 의 $\phi(K_m)=m-1$, 완전 이분그래프 (complete bipartite graph) $K_{m,n}$ 의 $\phi(K_{m,n}) = \lfloor (m-1)/2 \rfloor + n, m \geq n \geq 1$, 별 그래프 (star graph) $S_k = K_{k,1}$ 의 $\phi(S_k) = \lfloor (k-1)/2 \rfloor + 1$, 2^n 의 초입방체 그래프 (hypercube graph) Q_m 의 $\phi(Q_m) = \sum_{m=0}^{n-1} \binom{m}{\lfloor m/2 \rfloor}$, 정사각형 그리드 그래프 (square grid graph) $P_m \times P_n$ 의 $\phi(P_m \times P_n) = {}_{\min}\{m,n\}$ 으로 알려져 있다[4-8].

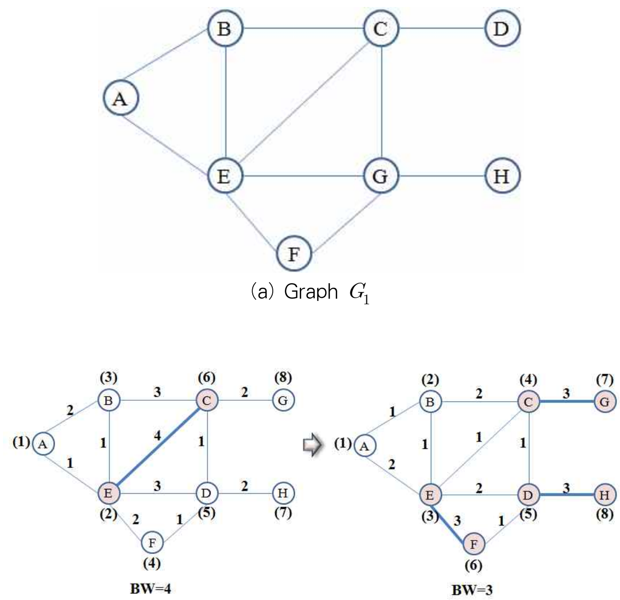
본 논문에서는 다양한 그래프에 대해 BMP의 근사 해를 $O(m)$ 의 다항시간으로 찾아갈 수 있는 규칙을 갖는 휴리스틱 알고리즘을 제안하여 위와 같이 다양한 분야에 쉽게 적용하여 업무 효율성을 극대화시키고자 한다. 2장에서는 BMP의 개념과 관련 연구 문제점을 고찰한다. 3장에서는 BMP에 대해 $O(m)$ 복잡도로 근사 해를 구할 수 있는 휴리스틱 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 실제 데이터에 적용하여 알고리즘 적합성을 평가해 본다.

• First Author: Sang-Un Lee, Corresponding Author: Sang-Un Lee
*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Eng., Gangneung-Wonju National University
• Received : 2015. 05. 22, Revised : 2015. 06. 11, Accepted : 2015. 07. 13.

II. Problem Definitions and Related Works

그림 1의 $G=(V,E)$ 그래프가 주어졌다고 가정하여 보자. (a)의 G_1은 Varshney[1]에서 인용되었다. 여기서 $m=8$, $n=11$ 이다. (b)는 주어진 그래프의 정점 순서대로 번호를 부여한 경우로 BW=4이며, (c)는 최적 해를 갖도록 번호를 다시 부여한 경우로 BW=3이다. 따라서 이 그래프의 최소 대역폭 $\phi(G_1)=3$ 임을 알 수 있다.

BMP는 NP-완전으로 정확한 해를 찾을 수 있는 다항시간 알고리즘이 알려져 있지 않다. 따라서 근사 해를 다항시간으로 얻기 위한 방법으로 메타휴리스틱 기법들이 적용되고 있다.



(a) Graph G_1

(b) Sequential labeling (c) Optimal labeling

Fig. 1. Bandwidth minimization problem

메타휴리스틱 기법들을 BMP에 적용한 사례를 살펴보면, Safro et al.[9]은 다중 레벨 알고리즘을, Pintea et al.[10]은 하이브리드 개미집락 최적화 (ant colony optimization, ACO)를, Esposite et al.[11]은 행렬 대역폭 감소 알고리즘 (matrix bandwidth reduction algorithm, MBRA)을, Lim et al.[12]은 진화 알고리즘 (evolutionary algorithm, EA)을, Lim et al.[13]은 ACO와 언덕 오르기 (hill climbing, HC) 방법을, Lim et al.[14]은 입자무리 최적화 (particle swarm optimization, PSO)와 HC를, Marti et al.[15]은 Tabu 탐색법 (tabu search, TS)을, Pinana et al.[16]은 탐욕적 무작위 적응 탐색법 (greedy randomized adaptive search, GRASP)을 제안하였다.

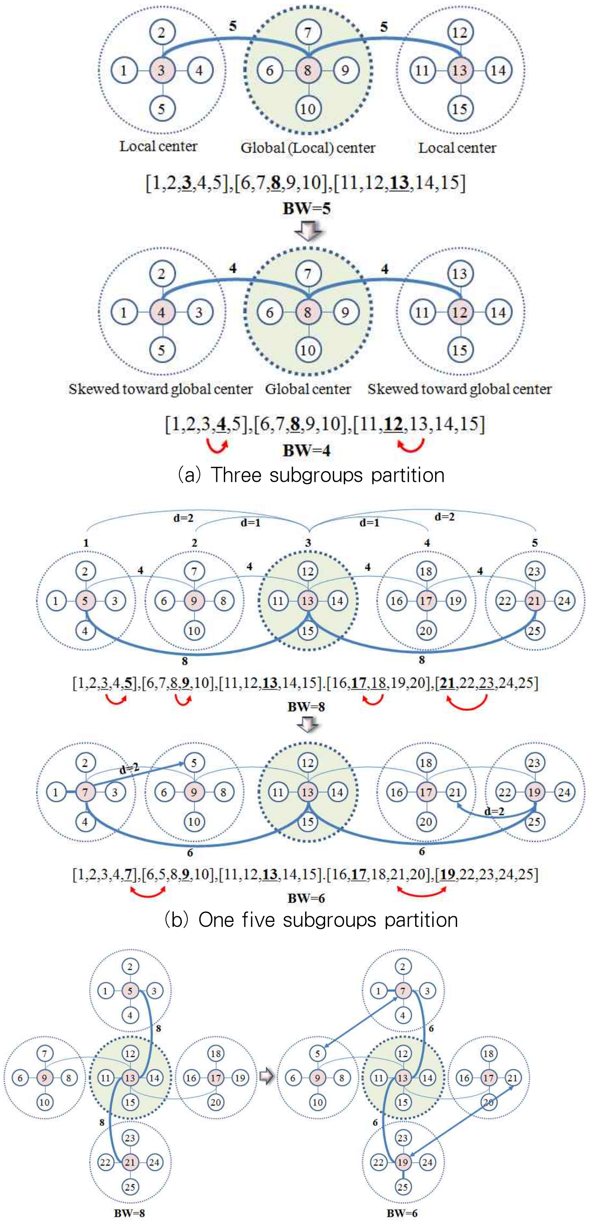
3장에서는 BMP의 근사 해를 $O(m)$ 의 다항시간으로 찾아가는 규칙을 제시한 휴리스틱 알고리즘을 제안한다.

III. Maximum Degree Vertex Partitioned Central Location Algorithm

$G=(V,E)$, $m=|V|$의 m개 정점에 대해 번호 $N=1,2,\cdots,m$ 를 부여하는 경우, 두 정점 번호 i,j 간의 차이 (거리) $|N_i-N_j|$를 계산할 수 있다. 여기서 $\max|N_i-N_j|$를 대역폭 BW라 한다. 주어진 그래프의 대역폭을 최소화시키려면 당연히 최대차수를 가진 정점 $\Delta(G)=v_i$ 를 중간에 위치시키고, 근접한 거리에 인접한 정점들을 배치시켜야만 한다. 이는 별 그래프 (star graph)에서 그 이유가 자명하게 증명될 수 있다. m개 정점으로 구성된 별 그래프에서 최소 대역폭을 갖도록 하려면 당연히 $m-1$ 개의 정점들과 연결된 하나의 정점의 번호 $i=\lceil(m+1)/2\rceil$ 이 되어야만 하기 때문이다. 본 장에서는 별 그래프에서의 이러한 자명한 사실에 기반하여, 일반 그래프에도 이러한 법칙을 어떻게 적용시킬 수 있을까를 연구하였다. 본 장에서 적용한 법칙은 군집 (clustering) 또는 분할 (partition) 기법을 제안하고자 한다. 이들 군집이나 분할의 중심에는 최대차수 정점들이 위치하게 된다.

m개 정점에 번호 $N=1,2,\cdots,m$ 를 부여하는 방법으로 그래프에서의 최대 차수 정점 $d_G(v_i)=\Delta(G)$ 인 v_i의 번호 i를 $i=\lceil(m+1)/2\rceil$ 로 부여하고, m개 정점들을 v_i의 인접 (이웃) $N_G^1(v_i), N_G^2(v_i),\cdots$ 인 근방 그래프로 표현하여 마지막 근방에 위치한 최소 차수 정점 $d_G(v_j)=\delta(G)$ 2개에 대해 $(1,m)$ 을 부여하는 방법을 적용하였다. 남은 n에 대해서는 $N_G^1(v_i)$ 의 정점 수 $k=|N_G^1(v_i)|$로 분할하고 $(n-2)/k$ 로 번호를 분할하였다. $1,m$ 의 인접 정점 v_j,v_k 에 대역폭 상한값 (upper limit) $BW_{UL}=\max\{\lceil n/2\rceil-2,(n-1)-\lfloor n/2\rfloor\}$ 을 초과하지 않는 값으로 번호 j,k 를 배정하였다. 단, 이들 두 정점 간에 인접하면 $|j-k|\le BW_{UL}$ 이 되어야 한다. 이와 같은 방법으로 남아 있는 k개 분할 집합의 정점들에 대해 번호를 부여하였다.

보다 상세히 설명하면, 각 정점의 차수 $d_G(v_i)$를 구하여 최대 차수 정점 $d_G(v_i)=\Delta(G)$ 인 v_i의 번호 $n(v_i)=\lfloor(m+1)/2\rfloor$ 로 부여하고, 최소 차수 정점 $d_G(v_i)=\delta(G)$ 2개에 대해 $(1,m)$ 을 부여한다. 이는 $m-1$ 개 정점이 $\forall N_G^1(v_i)$ 인 경우, 각 정점의 차수를 오름차순으로 정렬하여 최소 차수 정점 $d_G(v_i)=\delta(G)$ 부터 2개씩 $(1,m)$, $(2,m-1),\cdots$ 로 배치한다. 그러나 $\exists N_G^2(v_i)$ 인 경우 m을 $k=|N_G(v_i)|$인 $k+1$ 개의 그룹으로 분할하고, 각 그룹의 중앙에 위치한 정점이 전역 중심점 (global center point, GCP)으로 $\lfloor(m+1)/2\rfloor$ 번호를 부여하며, 다른 지역 중심점 (local center point, LCP)은 GCP로 부터의 배치 순서 차이 $d(group)$ 에 따라 해당 그룹에서 d만큼 GCP 쪽으로 이동하여 번호를 부여한다. 다음으로, $d=2$ 그룹의 중심점을 $d=1$ 그룹의 거리=2인 정점과 교환한다. 이는 그림 2에 제시되어 있다.



(a) Three subgroups partition

(b) One five subgroups partition

(c) Another five subgroups partition

Fig. 2. Maximum degree vertex central location algorithm

제안된 알고리즘을 최대차수 정점의 분할영역 중앙 배치 알고리즘(maximum degree vertex partitioned central location algorithm, MDVPCLA)라 하며, 다음과 같이 수행된다.

$G=(V,E)$, $m=|V|$

번호 : $n=1,2,\cdots,m$

Step 1. 그룹 분할

최대 차수 정점 $d_G(v_i)=\Delta(G)$ 인 v_i 선택

v_i의 인접 (이웃) 정점 $v_j=N_G(v_i)$ 선택

$G_1=N_G[v_i]$ 로 그룹 형성

for $i=2$ to $|N_G(v_i)|$

if $v_j,v_k\in N_G(v_i)$, $v_k\neq N_G(v_j)$ and $|N_G(v_k)|>0, N_G(v_k)\notin N_G(v_i)$ then

$G_i=N_G[v_k]$ 그룹 형성

end

end

Step 2. 그룹 내 번호 부여

(1) 그룹 내 정점 v_i의 차수 $d_G(v_i)$ 오름차순 정렬

(2) $d_G(v_i)=\Delta(G)$ 인 v_i의 번호 v_n을 그룹 정점 수 m_i에 대해 $\lceil m_i/2\rceil$ 이 되도록 각 정점의 차수를 오름차순으로 $d_G(v_i)=\delta(G)$ 부터 2개씩 $(m_{i-1}+1, m_i)$, $(m_{i-1}+2, m_i-1)$, … 로 배정된 번호 부여

Step 3. 그룹 간 번호 교환 최적화

(1) k 그룹의 LCP에 대해 GCP와의 그룹 간 배치순서 차이 $d(group)$ 에 따라 해당 그룹에서 d 만큼 GCP 쪽으로 이동하여 번호 재부여

(2) $d=2,3,\cdots$ 그룹의 LCP 정점 번호를 $d=1,2,\cdots$ 그룹의 거리=$2,3,\cdots$ 인 정점과 교환

Step 1에서는 주어진 그래프의 m개 정점을 최대차수 정점을 중심으로 여러 개의 그룹으로 분할하여 군집을 형성하도록 하는 방법이다. 이 때 최대차수 정점 수만큼 그룹들로 분할된다. Step 2에서는 그룹 내에서 해당 그룹에 속한 정점 수 만큼 부여된 번호들 범위에 대해, 최대차수 정점이 중간에 위치하도록 연속된 번호를 부여하는 방법이다. 이 단계에서는 그래프 전체의 그룹들 간 번호 간격 (전역)을 고려하지 않고, 단지 자신의 그룹 내 범위 (지역)만을 고려하여 최소 대역폭을 얻도록 번호가 부여되었다. 따라서, 전역적인 측면에서의 최소 대역폭을 얻기 위해 Step 3의 최적화 과정을 추가로 수행하였다.

그림 1의 G_1에 대해 제안된 MDVCLA를 수행한 결과는 그림 3에 제시되어 있다. $m=8$, $\Delta_G(G)=E$, $\lceil (m+1)/2\rceil=5$ 로 E 정점 번호 $n(E)=5$ 로 부여된다. $N_G^1(E)=\{A,B,C, D,F\}$, $N_G^2(E)=\{G,H\}$ 로 $n(G)=1$, $n(H)\,n(H)=8$ 이 부여된다. 여기서 남은 번호는 [2,3,4,6,7]로 E로부터 최대 거리는 5-2=3이다. 따라서 $BW_{UL}=3$으로, $N_G(C)=\{G\}$ $N_G(D)=\{H\}$ 에 대해 $n(C)-1\le 3, 8-n(D)\le 3$ 이면서 $e\{C,D\}$ 가 존재하여 $n(C)-n(D)\le 3$ 을 충족하도록 $n(C)=3$, $n(D)=6$ 이 부여된다. 남은 번호 [2,4,7]에 대해 A,B,F에 $BW_{UL}=3$ 을 초과하지 않도록 번호가 부여되었다.

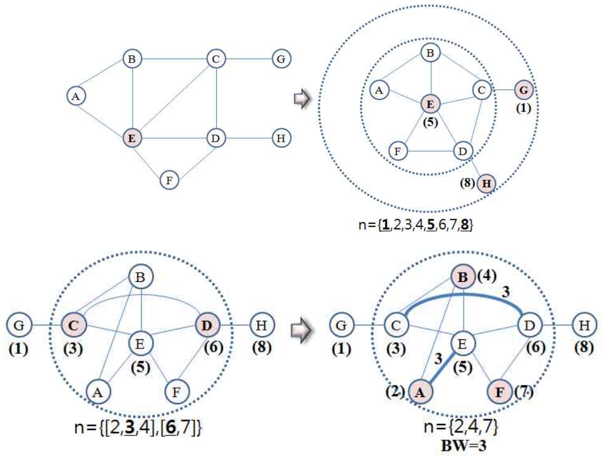
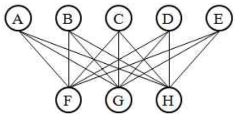


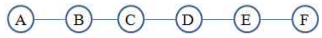
Fig. 3. MDVCLA for G_1

IV. Experiments and Result Analysis

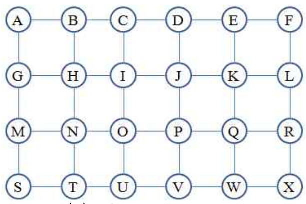
본 장에서는 그림 4의 다양한 그래프들에 대해 제안된 MDVCLA를 적용하여 본다.



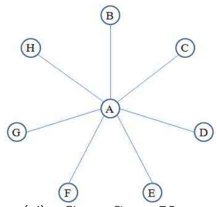
(a) $G_2 = K_{5,3}$



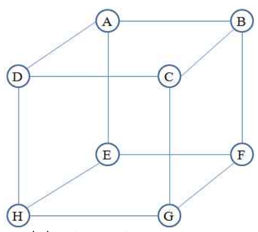
(b) $G_3 = L_6$



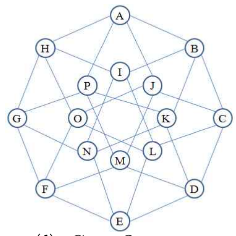
(c) $G_4 = P_4 \times P_6$



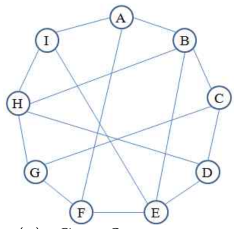
(d) $G_5 = S_7 = K_{7,1}$



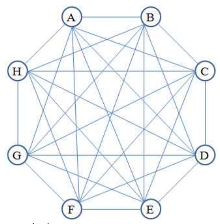
(e) $G_6 = Q_3$



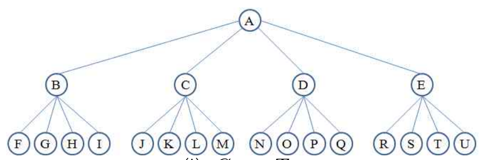
(f) $G_7 = Q_4$



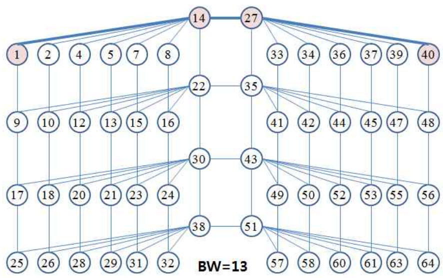
(g) $G_8 = Q_3$



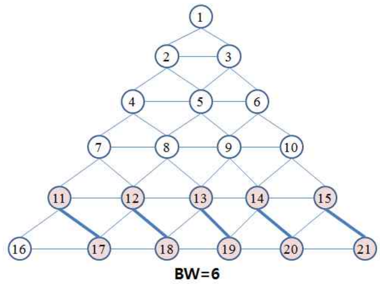
(h) $G_9 = K_8$



(i) $G_{10} = T_{4,3}$



(j) $G_{11} = G(3,6) \times P_4$

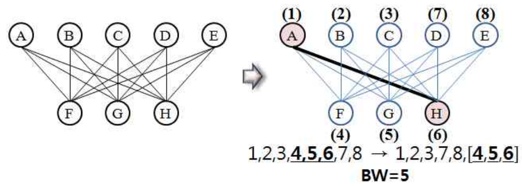


(k) $G_{12} = T_5$

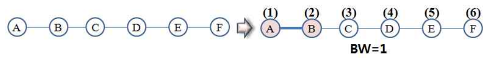
Fig. 4. Experimental data

G_{10}은 Chung[17]에서, G_{11}은 Kojima와 Ando[4]에서, G_{12}는 Miller와 Orlin[18]에서 인용되었다.

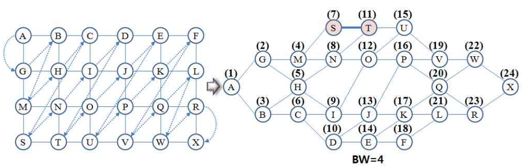
그림 4의 다양한 그래프들에 대해 제안된 MDVCLA를 적용하여 각 정점들에 번호를 부여하고 BW를 얻은 결과는 그림 5에 제시되어 있다.



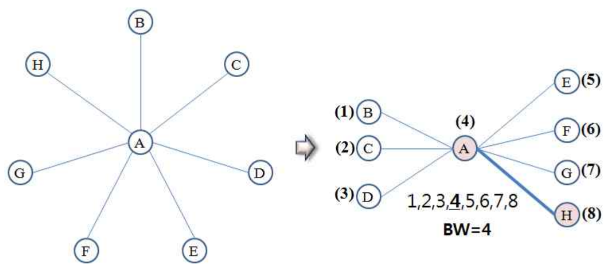
(a) $G_2 = K_{5,3}$



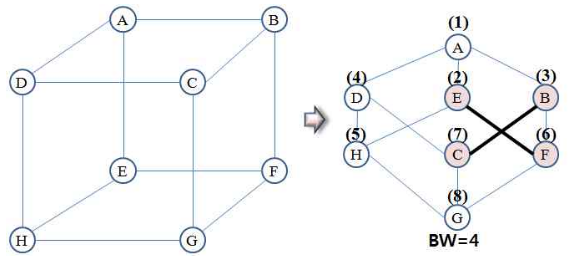
(b) $G_3 = L_6$



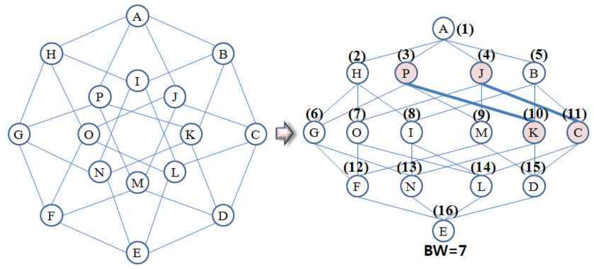
(c) $G_4 = P_4 \times P_6$



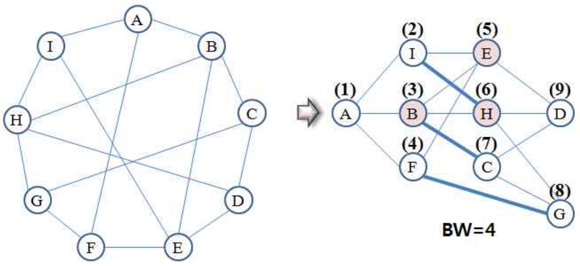
(d) $G_5 = S_7 = K_{7,1}$



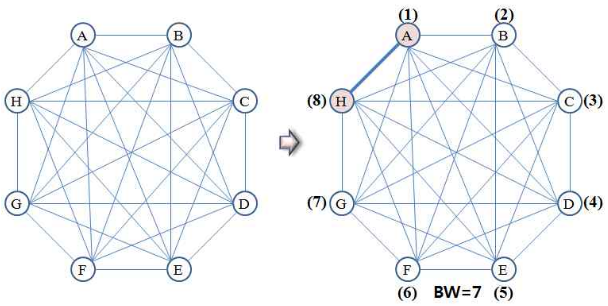
(e) $G_6 = Q_3$



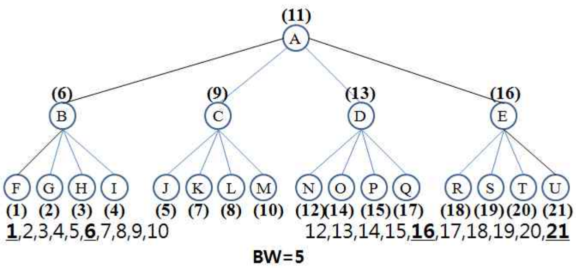
(f) $G_7 = Q_4$



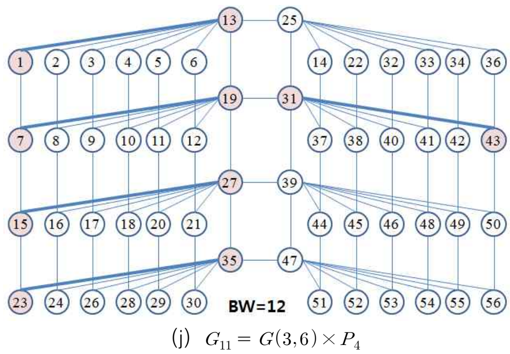
(g) $G_8 = Q_3$



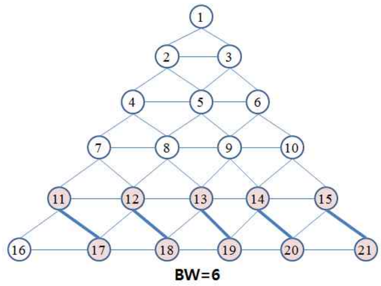
(h) $G_9 = K_8$



(i) $G_{10} = T_{4,3}$



(j) $G_{11} = G(3,6) \times P_4$



(k) $G_{12} = T_5$

Fig. 5. MDVCLA for experimental data

$G_4 = P_4 \times P_6$ 의 경우 대각선으로 번호를 부여하면 원하는 해를 쉽게 구할 수 있음을 보였다.

알고리즘의 성능은 주어진 실험 그래프들에 대해 기존 연구에서 알려진 BW와 MDVCLA로 얻은 BW를 비교하여 MDVCLA로 최적 해를 얻을 수 있는지 여부로 검증하였다. 본 논문에서 거론된 실험데이터에 대해 기존에 알려진 해인 대역폭과 MDVCLA로 얻은 대역폭의 결과를 비교하여 표 1에 제시하였다.

G_{11}에 대해 Kojima와 Ando[4]는 $m=56$에 대해 3,6,11, 19,38,46,54,59의 8개를 제외한 [1,64]의 번호를 부여하여 $BW=13$을 얻었다. 반면에, 제안된 알고리즘은 [1, 56]의 번호에 대해 누락없이 부여한 결과 $BW=12$를 얻을 수 있었다.

Table 1. Compare with Algorithm Performance

그래프	대역폭 $\phi(G)$	
	알려진 해	MDVCLA
G_1	**3**	**3**
G_2	**5**	**5**
G_3	**1**	**1**
G_4	**4**	**4**
G_5	**4**	**4**
G_6	**4**	**4**
G_7	**7**	**7**
G_8	**4**	**4**
G_9	**7**	**7**
G_{10}	**5**	**5**
G_{11}	13	**12**
G_{12}	**6**	**6**

V. Conclusions

본 논문은 NP-완전으로 분류되어 있어 최적 해를 다항시간으로 구하는 알고리즘이 제안되지 않고 있는 대역폭 최소화 문제에 대해 다항시간 알고리즘을 제안하였다.

제안된 알고리즘은 주어진 그래프 $G=(V,E)$에 대한 $\phi^* = {}_{\min}\phi(G)$, $\phi(G) = {}_{\max}\{|f(v_i)-f(v_j)| : v_i, v_j \in E|$를 찾기 위해 최대차수 정점 v_i를 전역 중심점 (GCP)로 설정하고, $[1,m]$ 범위의 중앙값 $\lceil m+1/2 \rceil$ 번호를 부여하는 방법을 적용하였다. 다음으로, 서브그룹으로

분할하여 서브그룹에서의 최대차수 정점을 지역 중심점 (LCP)으로 설정하고, GCP와의 거리가 가능한 최소가 되도록 서브그룹 내의 번호를 조정하였다. 이와 같이 수행 복잡도 $O(mn)$ 으로 정점들에 번호를 부여할 수 있었다.

제안된 알고리즘을 12개의 다양한 그래프들을 대상으로 실험한 결과, 11개 그래프에 대해서는 기존에 알려진 최적 해와 동일한 값을 얻었다.

G_{11} 에 대해서는 Kojima와 Ando[4]는 $m=56$ 에 대해 [1,64]의 번호를 부여하여 $BW=13$ 을 얻은 반면에, 제안된 알고리즘은 정점의 번호는 반드시 $[1,m]$ 으로 부여해야 한다는 제약조건을 충족시키면서도 $BW=12$ 로 감소시킬 수 있었다.

제안된 알고리즘은 $O(mn)$ 의 다항시간 수행 복잡도를 갖기 때문에 대역폭 최소화 문제를 적용할 수 있는 선형대수 방정식 정칙체계, 편미분방정식의 근사 해, 대규모 전력전송시스템, 회로 설계, 하이퍼텍스트 배치, 화학 반응속도, 수치화된 지구물리학, 데이터저장소, VLSI 설계와 망 생존성 등에 유용하게 사용될 수 있을 것이다.

본 논문에서 다수의 클러스터로 분할되는 경우는 G_{10} 과 G_{11} 의 2개 데이터로 제안된 알고리즘의 적용성을 충분히 검증하였다고 할 수 없다. 따라서 추후 보다 많은 다양한 그래프를 대상으로 제안된 알고리즘의 일반화된 적용성을 검증할 계획이다.

REFERENCES

[1] K. Varshney, "A New Evolutionary Algorithm with Level Node Swap for Bandwidth Minimization Problem," M. Sc., Mathematics, Dayalbagh Educational Institute, Agra, pp. 1-24, 2014.

[2] U. Feige, "Coping with the NP-Hardness of the Graph Bandwidth Problem," Lecture Notes in Computer Science, Vol. 1851, pp. 129-145, May 2000.

[3] C. Dubey, U. Feige, and W. Unger, "Hardness Results for Approximating the Bandwidth," Journal of Computer and System Sciences, Vol. 77, No. 1, pp. 62-90, Jan. 2010.

[4] T. Kojima and K. Ando, "Bandwidth of the Cartesian Product of Two Connected Graphs," Discrete Mathematics, Vol. 252, No. 1-3, pp. 227-235, May 2002.

[5] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, "The Bandwidth Problem for Graphs and Matrices-A Survey," Journal of Graph Theory, Vol. 6, No. 3, pp. 223-254, Sep. 1982.

[6] G. Hermann, "On Balanced Separators, Treewidth, and Cycle Rank," Journal of Combinatorics, Vol. 3, No. 4, pp. 669-682, Dec. 2012.

[7] K. Haim and S. Ron, "Pathwidth, Bandwidth, and Completion Problems to Proper Interval Graphs with Small Cliques," SIAM Journal on Computing, Vol. 25, No. 3, pp. 540-561, Jun. 1996.

[8] J. Díaz, J. Petit, and M. Serna, "A Survey of Graph Layout Problems," ACM Computing Surveys, Vol. 34, No. 3, pp. 313-356, Sep. 2002.

[9] I. Safro, D. Ron and A. Brandt, "Multilevel Algorithms for Linear Ordering Problems," ACM Journal of Experimental Algorithmics, Vol. 13, pp. 1.4-1.20, 2009.

[10] C. M. Pintea, G. C. Crişan, and C. Chira, "A Hybrid ACO Approach to the Matrix Bandwidth Minimization Problem," Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science, Vol. 6076, pp. 405-412, 2010.

[11] A. Esposito, M. S. Catalano, F. Malucelli, and L. Tarricone, "A New Matrix Bandwidth Reduction Algorithm," Operation Research Letters, Vol. 23, No. 3-5, pp. 99-107. Oct. 1998.

[12] A. Lim, B. Rodrigues, and F. Xiao, "Using an Evolutionary Algorithm for Bandwidth Minimization," Congress on Evolutionary Computation, Vol. 1, pp. 678-683, Dec. 2003.

[13] A. Lim, B. Rodrigues, and F. Xiao, "Ant Colony Optimization and Hill Climbing for the Bandwidth Minimization Problem," Applied Software Computing, Vol. 6, No. 2, pp. 180-188, Jan. 2006.

[14] A. Lim, J. Lin, and F. Xiao, "Particle Swarm Optimization and Hill Climbing to Solve the Bandwidth Minimization Problem," Applied Intelligence, Vol. 26, No. 3, pp. 175-182, Jun. 2007.

[15] R. Marti, M. Laguna, F. Glover, and V. Campos, "Reducing the Bandwidth of a Sparse Matrix with Tabu Search," European Journal of Operational Research, Vol. 135, No. 2, pp. 450-459, Dec. 2001.

[16] E. Pinana, I. Plana, V. Campos, and R. Marti, "GRASP and Path Relinking for the Matrix Bandwidth Minimization," European Journal of Operational Research, Vol. 153, No. 1, pp. 200-210, Feb. 2004.

[17] F. R. K., Chung, "Graph Theory, Chap. 7: Labelings of Graphs," pp. 151-168, Academic Press, 1988.

[18] Z. Miller and J. B Orlin, "NP-Completeness for Minimizing Maximum Edge Length in Grid Embeddings," Journal of Algorithms, Vol. 6 No. 1, pp. 10-16, Mar. 1985.

Authors


Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively. He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.