GOSST 문제에 대한 전역적 배치와 지역적 배치 휴리스틱의 개선을 위한 *G-Node*와 단절에 관한 연구

정회원 김 인 범*, 김 재 각*

A Study on the *G-Node* and Disconnected Edges to Improve the Global and Local Locating Heuristic for GOSST Problem

Inbum Kim*, Chae-kak Kim* ***Regular Members***

요 약

GOSST 문제에 대한 이전 휴리스틱의 성능 개선을 위한 새로운 제안이 본 논문에서 제시된다. 이 문제는 다양한 등급의 서비스를 제공할 수 있는 통신 네트워크의 설계 등에 적용될 수 있다. GOSST 문제는 *G-Condition*을 만족하는 최소 구축비용의 네트워크를 찾는 것으로 NP-HARD 문제에 포함된다. 우리는 이전의 연구에서 이 문제와 관련된 두개의 휴리스틱을 발표하였다. 본 연구에서는 스타이너 트리 생성 시 이용되는 *G-Node*와 제거되는 에지의 선택 방법을 제안하고, 이를 기존의 휴리스틱에 접목한 새로운 휴리스틱을 구현한다. 실험 결과는 이 휴리스틱이 이전 것에 비해 우수함을 나타내는데, 새 지역적 배치 휴리스틱은 17%, 새 전역적 배치 휴리스틱은 14%의 네트워크 구축비용 절감비율의 증가를 보였다.

Key Words : GOSST problem, G-Node, G-Condition, Steiner point, Disconnection

ABSTRACT

This paper is on the enhancement of our heuristics for GOSST problem that could apply to the design of communication networks offering graduated services. This problem known as one of NP-Hard problems finds a network topology meeting the *G-Condition* with minimum construction cost. In our prior research, we proposed two heuristics. We suggest methods of selecting *G-Node* and disconnections for Global or Local locating heuristic in this research. The ameliorated Local locating heuristic retrenches 17% more network construction cost saving ratio and the reformed Global locating heuristic does 14% more than our primitives.

Ⅰ. 서 론

정보화 사회의 핵심 기반인 네트워크를 이용해 다양한 형태의 대용량 데이터와 유용한 정보가 이동되고 있다. 다양한 수요자들의 정보나 통신 서비스에 대한 요구는 여러 등급으로 분류할 수 있으며 이 때 공급자 중심의 필요 이상의 높은 수준 혹은 요구에 미달하는 수준의 서비스를 강요하기는 어려운 상황이다. 따라서 개인별 서비스 요구를 만족하면서 그 구축비용에 있어서 경제적인 네트워크가 설계되어야 할 것이다. 이 경우에 적용 가능한 것이 GOSST(Grade Of Services Steiner Minimum Tree)이다.

GOSST에 대한 대부분의 연구는 최적화에 집중되어왔다[1,2,3,4]. 이들 연구자들은 스타이너 포인트 문제와 관련된 많은 어려운 문제들을 해결하는 과

* 김포대학 컴퓨터계열 (ibkim@kimpo.ac.kr, ckkim@kimpo.ac.kr)
논문번호 : KICS2007-08-346, 접수일자 : 2007년 8월 2일, 최종논문접수일자 : 2007년 8월 31일

정에서 새로운 문제들을 도출해냈는데 그 중의 하나가 GOSST 문제이다[1,5]. 이 문제는 최적 해의 이론적 한계에 도달한 것으로 간주되고 있다[4]. 이 문제를 위한 대부분 최적화 알고리즘의 구현 목적은 이 문제가 PTAS(Polynomial Time Approximation Scheme)에 속한다는 것을 증명하기 위해서이다[4,5]. NP-Hard 문제에서 PTAS 문제는 매우 큰 다항식적 실행시간 내에 (1+ε)의 근사 해를 구할 수 있는 것으로 정의할 수 있다. 그러나 이 근사 알고리즘의 실행시간을 현실적인 수준으로 떨어뜨린다면 근사비율은 매우 나빠지게 된다. 대부분의 경우에 현실세계에 적용할 수 없을 무위미한 수준이 되는 것이다. 또한 많은 연구자들은 이 문제가 스타이너 트리와 관련된 이론적 문제라고 간주하기 때문에 이것을 현실 문제에 적용하려는 충분한 시도를 하지 않았다. 그러나 이 문제의 현실적인 해는 네트워크나 회로, 항로 및 도로개발, 물류 등에 적용 가능하기 때문에 실현 가능한 시간 복잡도를 가지는 휴리스틱의 개발은 반드시 필요하다. 우리는 이전의 연구를 통해서 이 문제에 대한 전역적 배치와 지역적 배치 휴리스틱을 제안했었다[12]. 본 논문에서는 이들을 개선하기 위해 *G-Node* 및 단절에 대한 연구를 통해 효과적인 선택방법을 제안하게 되었다.

Ⅱ. 연구 배경

2.1 스타이너 포인트와 스타이너 최소 트리

네트워크의 길이를 줄이는 문제는 최적화 분야에서 중요한 문제들 중의 하나이다. 어떤 포인트 S로부터 주어진 세 개의 포인트까지의 평면 거리의 합이 최소가 되게 하는 S를 찾는 문제가 있다. 이 문제는 다수의 터미널 포인트들을 대상으로 확장할 수 있는데 이 경우 별 모양을 형성하기 때문에 이를 *Steiner Star* 라고 부른다. 또한 이 문제는 터미널 포인트가 아닌 다수의 포인트들을 추가하여 모든 터미널 노드를 연결하는 최소 길이의 스타이너 최소 트리를 만들 수 있게 확장할 수 있다. 이 문제의 각 포인트를 이산적 데이터로 고려하는 경우, 이 문제는 NP-Hard 문제에 포함된다. 따라서 이 문제를 현실적 분야에 적용하기 위해서는 적절한 스타이너 최소 트리를 구하기 위한 휴리스틱이 필요하다. 그림 1에서는 간단한 스타이너 트리를 구하는 휴리스틱의 각 단계들을 보여준다. 즉 *G-Node*의 선택, 스타이너 포인트의 생성 및 위치 결정, 새로운 연결 생성 및 기존 연결 제거의 단계이다. *G-Node*

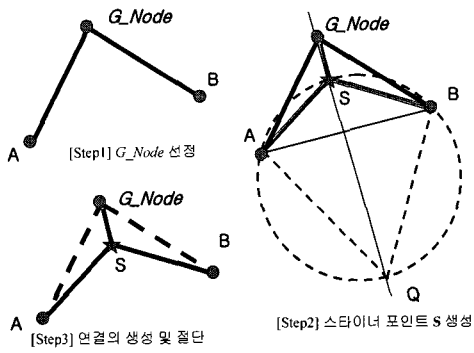


그림 1. 스타이너 트리를 생성하는 단계

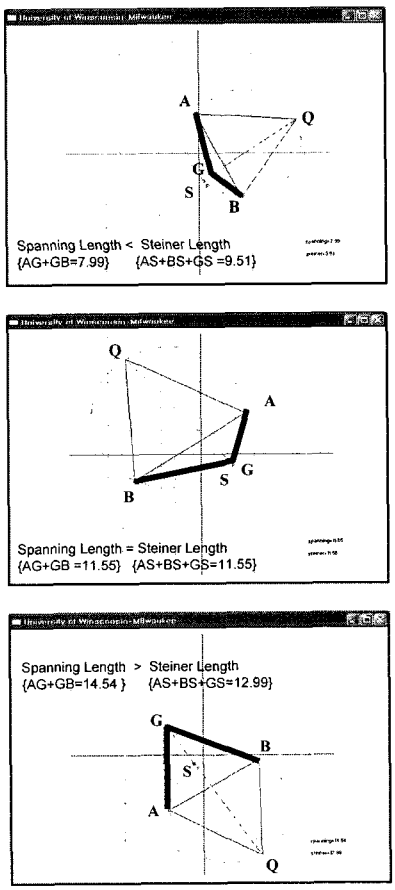


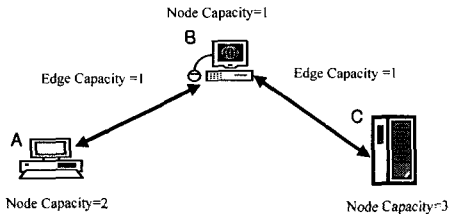
그림 2. 스타이너 길이와 스패닝 길이의 비교

는 스타이너 포인트를 구하기 위해 사용되는 중간 노드로 정의한다. 생성되는 스타이너 포인트의 위치 결정은 기존의 휴리스틱을 활용하였다[12]. 그러나 그림 2에서 보이는 것처럼 이 휴리스틱에 의한 스타

이너 트리가 항상 스타이너 최소 트리를 생성하는 것은 아니다. 스타이너 길이가 스패닝 길이보다 큰 경우에는 이 휴리스틱에 의해 생성되는 S는 버리고 *G-Node*위치에 스타이너 포인트를 생성한다.

2.2 GOSST 문제와 *G-Condition*

*G-Condition*이란 '*어떤 네트워크에 존재하는 임의의 두개의 노드 p_i와 p_j 사이에는 각 노드의 처리용량 가운데 작은 값보다 큰 처리용량을 가지는 경로가 한 개 이상 존재한다*'로 정의할 수 있다. GOSST는 모든 터미널 포인트들을 연결하는 네트워크들 중에서 *G-Condition*을 만족하면서 최소 구축비용이 소요되는 네트워크이다. 서비스 등급 u를 제공하는 에지의 비용은 에지의 유클리드 길이와 서비스 u를 제공하기 위해 소요되는 비용의 곱이다. GOSST 문제는 모든 터미널 포인트가 동일한 서비스 요청 등급을 가지는 ESMT(Euclidean Steiner Minimum Tree) 문제의 일반화이다[1,6,7,8,9,10,11]. 이 문제들은 NP-Hard 문제로 알려져 있으므로 작은 규모의 문제의 해를 얻기 위해 매우 큰 계산 량과 메모리 공간이 필요할 것이다. GOSST에 관한 이전의 많은 연구들은 기하학적 분석 및 근사 알고리즘에 관심을 보였다. 그러나 이를 현실 문제에 적용하기 위한 시도는 그리 활발하지 않았다.



(A) 주어진 네트워크

(B) 개선된 네트워크

그림 3. GOSST 휴리스틱의 *G-Condition* 위반 처리 방법

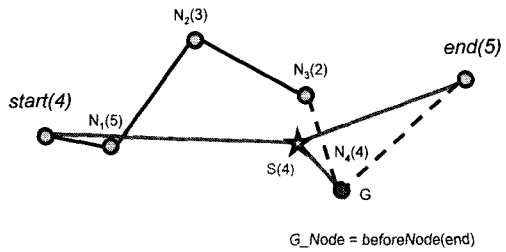


그림 4. 이전 전역적 배치 휴리스틱의 *G-Node* 및 단절 선택

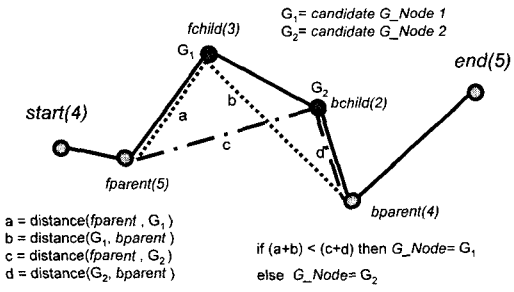


그림 5. 이전 지역적 배치 휴리스틱의 G_Node 선택방법

그림 3은 본 논문에서 GOSST 문제의 휴리스틱에서 *G-Condition* 위반을 처리하는 방법을 보이고 있다. 그림 3 (A)의 네트워크에서 다른 처리능력을 가지는 터미널 노드 A, B, C에서 B를 경유하는 경로 AC의 가용 용량은 1을 초과할 수 없다. 경로의 가용 용량의 낭비를 없애기 위해, 우리는 이전의 연구에서 그림 3의 (B)와 같이 스타이너 포인트라는 새로운 노드를 생성하여 터미널 노드 A, B, C에 연결을 시키고 불필요한 연결의 삭제 방법을 제안하였다[12]. 이 방법을 이용하면, A와 C를 연결하는 경로의 용량은 2로 증가하지만 네트워크의 길이는 증가할 가능성이 있다. 새 스타이너 포인트 위치가 네트워크 길이를 결정하는 중요한 요소가 되는 것이고 이는 네트워크 구축비용에 영향을 미치게 된다. *G-Condition*을 만족하는 네트워크의 구축비용 최소화는 GOSST 문제의 목표이다.

2.3 이전 전역적 배치와 지역적 배치 휴리스틱

참고문헌 [12]에는 우리의 이전 전역적 배치 휴리스틱과 지역적 배치 휴리스틱에 대한 내용이 있다. 그림 4에는 전역적 배치를 위한 이전 방법이 설명되어 있고, 그림 5에는 지역적 배치 방법이 보인다.

Ⅲ. 제안 방법

본 논문에서 제안하는 새 GOSST 휴리스틱은 각각의 후보 *G-Node*에 대해 선택 가능한 단절연결들을 조사해서 가장 경제적인 선택을 하도록 하는 전

략을 채택하였다. 그림 6은 *start*와 *end* 노드 사이의 경로에서 *G-Condition* 위반이 발생했을 때, 새 전역적 배치 전략을 사용하여 후보 스타이너 포인트를 생성하는 과정을 설명한다. 그림 6의 (A)에서 노드 G_1을 후보 *G-Node*로 간주하여 *start*와 *end* 노드와 함께 임시 후보 스타이너 포인트를 생성한다. 이때 가능한 단절연결의 경우가 그림 6의 (B), (C), (D), (E)와 같은 4개의 경우가 발생하는데, 이 중에서 가장 경제적인 것을 선택하고, G_1과 선택된 단절연결을 저장한다. 그림 6의 (F), (G), (H)는 G_2, G_3, G_4를 후보 *G-Node*로 하여 가능한 단절연결들의 경우들을 보이고 있는데 G_1의 경우와 같이 선택된 단절연결을 저장한다. 저장된 후보 *G-Node*와 단절연결들을 비교한 후, 가장 네트워크 구축비용을 절감할 수 있는 경우를 선택하여 *start* 와 *end* 노드 사이의 경로를 위한 후보 스타이너 포인트를 결정하게 된다. 그림 7은 본 논문의 새 지역적 배치 휴리스틱의 후보 스타이너 포인트를 생성하는 과정들을 보인다. *start*와 *end* 노드 사이의 경로에서 실제 *G-Condition* 위반이 발생하는 *fparent*와 *bparent* 노드 사이의 노드인 G_1, G_2가 후보 *G-Node*가 되어 각 각에 대해 가능한 단절 연결의 경우를 조사하고, 그 가운데에서 가장 비용을 절감할 수 있는 후보 *G-Node*와 단절 연결을 선택하게 된다. 새 전역적 배치전략과 지역적 배치전략에 대한 실행 과정은 다음과 같다.

3.1 전역적 배치 전략

step 1 Check *G-Condition* for all paths of a network.

step 2 If a *path* (**start**, **end**) violates *G-Condition* on subpart of the *path* (**fparent**, **bparent**),

1. The *Candidate G-node* **G** is one of the nodes on the *subpath*(*afterNode*(**start**), *beforeNode*(**end**)). For each **G**,
   [1] With **G**, **fparent** and **bparent**, *temporary Steiner point* **tSteiner** is created.
   [2] Calculate Steiner length *addedEdgeSum* as follows;
   addedEdgeSum=*len*(**start**,**tSteiner**) +*len*(**G**, **tSteiner**)+*len*(**end**, **tSteiner**)
   - Calculate length sum of deleted edge *deletedEdgeSum* with choosing two edges such as one edge $\mathbf{e}_1$ from *subpath*(**start**, **G**) and the other edge $\mathbf{e}_2$ from *subpath*(**G**, **end**) as follows;
   deletedEdgeSum=*len*(e1)+*len*(e2)
   [3] Find two edges $\mathbf{e}_3$ and $\mathbf{e}_4$ making *deletedEdgeSum* be most.
2. Calculate *diff* using the edges $\mathbf{e}_3$ and $\mathbf{e}_4$.
   diff = *addedEdgeSum* - *deletedEdgeSum*
3. Select **G** making *diff* be least and restore **G**, **tSteiner**, $\mathbf{e}_3$ and $\mathbf{e}_4$. Promote temporary Steiner point **tSteiner** to *candidate Steiner point.*
   [1] If *diff* is negative,
   - New *Steiner point(s)*is(are) created on the *subpath*(*afterNode*(**start**),*beforeNode*(**end**))
   - The capacity of *Steiner point(s)*is the value of *min*(*cap*(**start**), *cap*(**end**))

   [2] Else,
   - The *candidate Steiner point* becomes a *real Steiner point*
   - The capacity of *Steiner point*is the value of *min*(*cap*(**start**), *cap*(**end**)).
   - Edge $\mathbf{e}_3$ and $\mathbf{e}_4$ are deleted.
   - *edge*(**start**, **tSteiner**), *edge*(**G**, **tSteiner**) and *edge*(**end**, **tSteiner**) are created.

step 3 Else **Termination.**

3.2 지역적 배치 전략

step1 Check *G-Condition* for all paths of a network.

step2 If a *path* (**start**, **end**) violates *G-Condition* on subpart of the *path* (**fparent**, **bparent**),

1. The *Candidate G-node* **G** is one of the nodes on the *subpath*(*afterNode*(**fparent**), *beforeNode*(**bparent**)). For each **G**,
   [1] With **G**, **fparent** and **bparent**, *temporary Steiner point* **tSteiner** is created.
   [2] Calculate Steiner length *addedEdgeSum* as follows;
   addedEdgeSum=*len*(**fparent**, **tSteiner**) +*len*(**G**, **tSteiner**)+*len*(**bparent**, **tSteiner**)
   - Calculate each length sum of deleted edge *deletedEdgeSum* with choosing two edges such as one edge $\mathbf{e}_1$ from *subpath*(**fparent**,**G**) and the other edge $\mathbf{e}_2$ from *subpath*(**G**,**bparent**) as follows;
   deletedEdgeSum=*len*($\mathbf{e}_1$)+*len*($\mathbf{e}_2$)

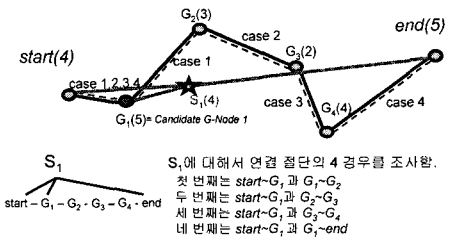
   [3] Find two edges $\mathbf{e}_3$ and $\mathbf{e}_4$ making *deletedEdgeSum*be most.
2. Calculate *diff* using the edges $\mathbf{e}_3$ and $\mathbf{e}_4$.
   diff = *addedEdgeSum* – *deletedEdgeSum*
3. Select G making diff be least and restore **G**, **tSteiner**, $\mathbf{e}_3$ and $\mathbf{e}_4$. Promote temporary Steiner point **tSteiner** to *candidate Steiner point.*
   [1] If *diff* is positive,
   - New *Steiner point(s)* is(are) created on the *subpath*(*afterNode*(**fparent**), *beforeNode*(**bparent**)).
   - The capacity of *Steiner point(s)* is the

value of *min*(*cap*(**start**), *cap*(**end**)).

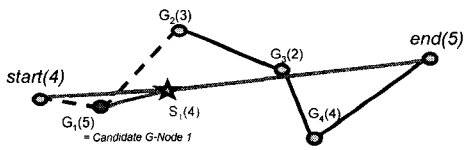
[2] Else,

- The *candidate Steiner point* becomes a *real Steiner point.*
- The capacity of *Steiner point* is the value of *min*(*cap*(**start**), *cap*(**end**)).
- edge $\mathbf{e}_3$ and $\mathbf{e}_4$ are deleted.
- *edge*(**fparent, tSteiner**), *edge*(**G,tSteiner**) and *edge*(**bparent, tSteiner**) are created.

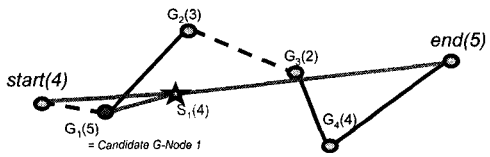
step3 Else **Termination.**



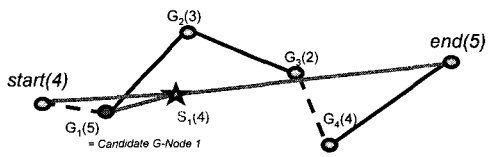
(A) G_1을 후보 *G-Node*로 간주하여 단절연결 경우 조사



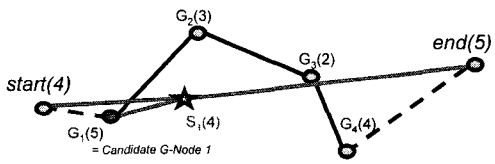
(B) 첫 번째 단절연결의 경우



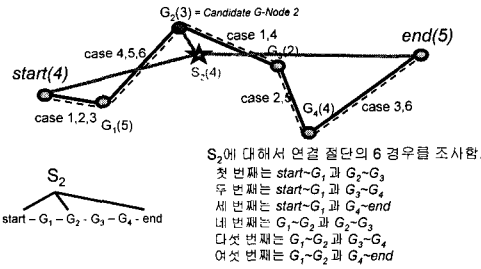
(C) 두 번째 단절연결의 경우



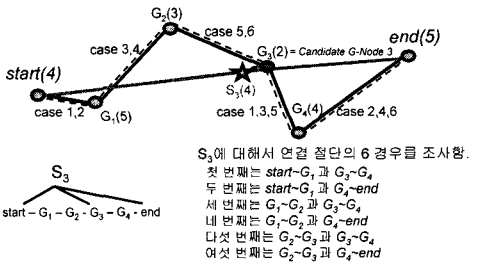
(D) 세 번째 단절연결의 경우



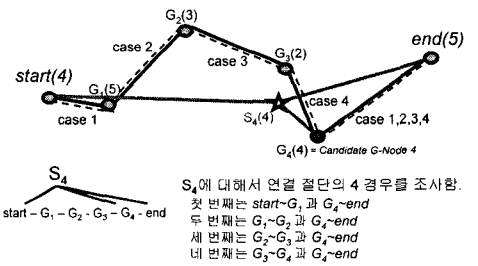
(E) 네 번째 단절연결의 경우



(F) G_2를 후보 *G-Node*로 간주하여 단절연결 경우 조사

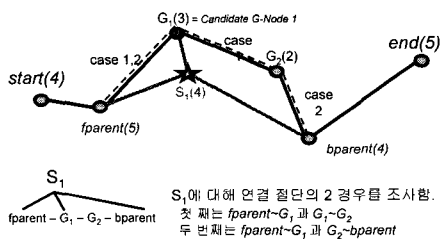


(G) G_3를 후보 *G-Node*로 간주하여 단절연결 경우 조사

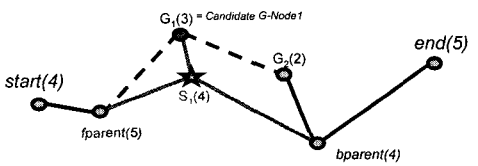


(H) G_4를 후보 *G-Node*로 간주하여 단절연결 경우 조사

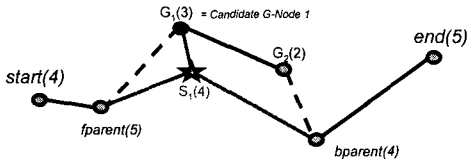
그림 6. 새 전역적 배치 휴리스틱의 *G-Node* 및 단절 선택



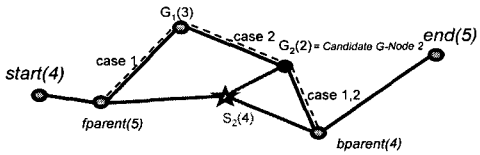
(A) G_1을 후보 *G-Node*로 간주하여 단절연결 경우 조사

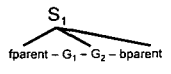


(B) 첫 번째 단절연결의 경우



(C) 두 번째 단절연결의 경우





S_2에 대해 연결 절단의 2 경우를 조사함.
첫 째는 *fparent*~G_1 과 G_2~*bparent*
두 번째는 G_1~G_2 과 G_2~*bparent*

(D) G_2를 후보 *G-Node*로 간주하여 단절연결 경우 조사

그림 7. 새 지역적 배치 휴리스틱의 *G-Node* 및 단절 선택

Ⅳ. 실험을 통한 제안하는 방법의 고찰

본 연구를 위한 실험의 인자는 터미널 노드의 수, 한 노드 당 가능한 최대 연결 수, 각 노드가 취할 수 있는 최대 요청 서비스 등급, 각 노드의 유클리드 평면상에서의 위치, 각 노드의 요청 서비스 등급 및 연결 정보이다. 실험에 사용된 터미널 노드의 수는 25, 50, 75, 100이고 각 노드의 위치는 평면상에서 중복을 허락하지 않는 무작위로 결정된다. 각 노드의 가능한 최대 연결의 수는 3, 5, 7, 9 으로 무작위로 선정된다. 각 노드가 할 당 받을 수 있는 최대 요청 서비스 등급의 크기는 3, 5, 7, 9 이고 각 노드는 1부터 이 크기까지의 수 가운데 무작위로 자신의 최대 가능한 요청 서비스 등급을 할당받게 하였다. *G-Node*와 단절 연결에 대한 새로운 선택방법을 채택한 GOSST 휴리스틱들인 거리우선 전역적 휴리스틱과 거리 우선 지역적 휴리스틱을 구현하여, 우리의 이전 GOSST 휴리스틱인 거리우선 전역적 휴리스틱과 거리 우선 지역적 휴리스틱, 그리고 본 실험의 컨트롤인 G-MST와 비교하였다[12]. G-MST는, 첫 번째 스테이지에서 생성된 최소 신장 트리를 이용해서 *G-Condition*을 만족시키기 위해 필요한 노드들의 처리용량을 변경시킨 트리들 중에서 가장 구축비용이 적은 트리 또는 네트워크이다. 우리의 두개의 이전 휴리스틱들과 새로운 두 개의 휴리스틱들에 대해 각각 64번의 실험을 시행하여 그 누적된 네트워크 구축비용, 비용의 절감 비율, 그리고 누적 생성스타이너 포인트의 수를 얻었다. 이 실험은 1.83 GHz 프로세서와 1기가 램을 장착한 랩탑 컴퓨터에서 실행하였고 각 휴리스틱들은 Microsoft의 Visual C++로 구현하였다.

그림 8은 각 GOSST 휴리스틱의 64번 실험에 대한 누적 네트워크 비용과 G-MST에 대한 비용 절감 비율의 변화에 대한 것이다. 본 논문에서 제안하는 새 거리 지역 배치 휴리스틱은 G-MST에 대해 10.3%의 네트워크 구축비용의 절감을 보인 것에 비해 이전의 거리 지역 배치 휴리스틱은 8.8%의 비용 절감을 보였다. 이전의 거리 전역 배치 휴리스틱은 8.9%의 비용 절감을 보였지만 개선된 거리 전역 배치 휴리스틱은 G-MST에 대해 10.1%의 구축비용의 절감을 보였다. 이것은 G-MST에 대한 누적 비용 절감비율에 대해 새 거리 지역 배치 휴리스틱은 이전 휴리스틱에 비해 17%의 개선을 보였고, 새 거리 전역 배치 휴리스틱은 14%의 개선을 나타냈다.

그림 9는 각 배치 휴리스틱에 대해, 터미널 포인트의 수의 변화에 따른 G-MST에 대한 누적 비용 절감비율의 변화를 나타내고 있다. 전역적 배치 혹은 지역적 배치 휴리스틱 모두, 이전 휴리스틱에서 작은 절감비율을 보인 터미널 노드의 수에서 새 휴리스틱은 절감비율의 가장 큰 개선율을 보였고, 이전 휴리스틱에서 높은 절감비율을 보였던 터미널 노드의 경우에는 새 휴리스틱에서 절감비율의 낮은 개선율을 보였다. 이것은 새 휴리스틱이 이전에 비해 터미널 포인트 수에 따른 절감비율의 편차가 작다는 것을 의미한다.

그림 10은 각 휴리스틱들의 누적 생성 스타이너 포인트의 수를 나타낸다. 새 휴리스틱들은 전체적으로 이전 버전의 휴리스틱들이 비해 적은 수의 스타이너 포인트를 필요로 함을 알 수 있다. 또한 새 휴리스틱들은 누적 생성 스타이너 포인트 수에 있어 이전 버전에 비해 작은 표준편차를 보이고 있다. 새로운 방법을 적용하면 전역적 배치와 지역적 배치에서 요구되는 스타이너 포인트 수의 차이가 이전 휴리스틱에 비해 작음을 알 수 있다. 즉, 전역적 배치 휴리스틱과 지역적 배치 휴리스틱의 스타이너 포인트 수가 좀 더 근접해짐을 의미한다.

Ⅴ. 결 론

본 연구는 GOSST 문제를 위한 이전의 휴리스틱의 성능을 개선하기 위한 방법에 관한 것이다[12]. GOSST 문제는 스타이너 포인트라는 임의의 노드들을 추가하여, *G-Condition*을 만족하면서 최소 구

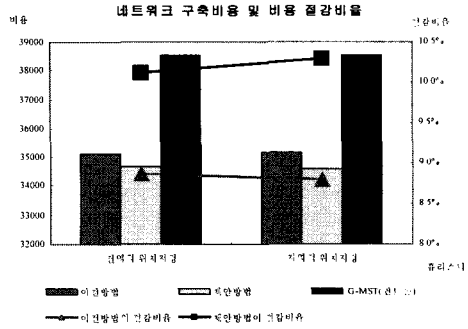


그림 8. 네트워크 구축비용과 절감비율의 비교

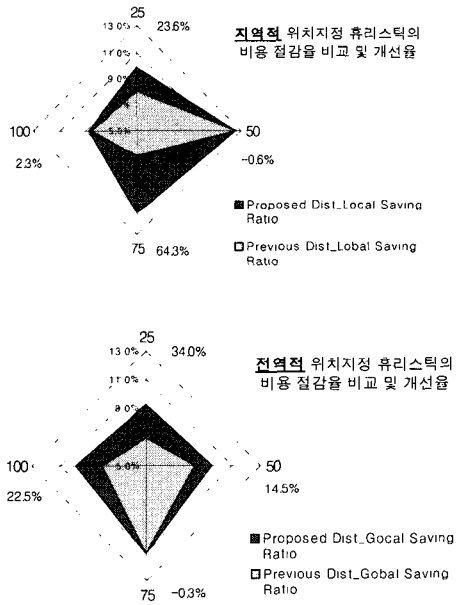


그림 9. 터미널 포인트 수에 따른 절감율 변화 및 개선율

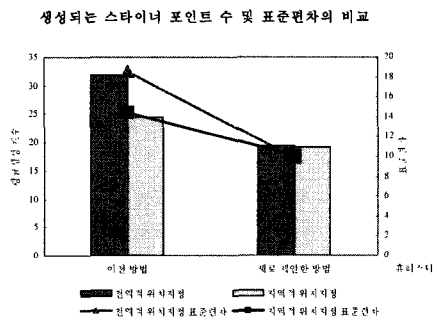


그림 10. 평균 스타이너 포인트 개수 및 표준편차의 비교

축비용으로 터미널 노드를 모두 연결하는 네트워크를 찾는 것이다. 제안된 방법을 반영한 새 GOSST 휴리스틱은 현실세계의 유용한 분야에서의 문제를 해결하는데 활용될 수 있을 것이다. 본 논문에서는 전역적 스타이너 포인트 배치 휴리스틱과 지역적 스타이너 포인트 배치 휴리스틱에 대해, *G-Node*와 제거되는 연결을 선택하는 방법을 제안하였다. 이상적인 스타이너 포인트의 결정을 위해, *G-Node*의 선택은 매우 중요하다. 제안된 방법은 가능한 모든 후보 *G-Node*에 대해 각 임시 스타이너 포인트를 생성하고 이를 이용해 가장 비용이 절감되는 후보 *G-Node*를 최종 *G-Node*로 선택하는 것이다. 단절되는 연결의 선택 또한 중요하다. 모든 단절 가능한 연결들을 조사하여 가장 비용절감이 큰 연결을 선택한다. 이전의 휴리스틱에 비해 새 지역 배치 GOSST 휴리스틱은 네트워크 구축비용의 17%를 개선할 수 있었고, 새 전역적 배치 GOSST 휴리스틱은 14%를 절감 할 수 있었다. 스타이너 포인트의 수에 대해, 전역적 배치 GOSST 휴리스틱에서는 평균 32개에서 평균 20으로, 지역적 배치 GOSST에서는 24개에서 19개로 감소하였다.

향후 과제는 GOSST 휴리스틱의 성능 개선을 위한 보다 많은 연구와 휴리스틱을 구성하는 각 모듈 및 인자들에 대한 분석 및 변형이 될 것이다. 또한 이들 휴리스틱을 적용할 수 있는 현실세계에서의 유용한 분야의 발견 및 개발이 필요하다.

참 고 문 헌

[1] D.Z. Du and F.K. Hwang, "An Approach for Providing Lower Bounds; Solution of Gilbert-Pollak Conjecture on Steiner Ratio", *Proceedings of IEEE 31sy FOCS*, pp.76-85, 1990

[2] G.L. Xue, G.H. Lin and D.Z. Du, "Grade of Service Steiner Minimum Trees in Euclidean Plane", *Algorithmica*, Vol.31, pp.479-500, 2001

[3] J. Kim and I. Kim, "Approximation Ratio 2 for the Minimum Number of Steiner Points", *Journal of KISS*, pp.387-396, 2003

[4] J. Kim, M. Cardei, I. Cardei and X. Jia, "A Polynomial Time Approximation Scheme for the Grade of Service Steiner Minimum Tree Problem", *Algorithmica*, Vol.42, pp.109-120, 2005

[5] S. Arora, "Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems", *Proceeding of 37th IEEE Symposium on Foundations of Computer Science*, pp.2-12, 1996.

[6] E.J. Cockayne and D.E. Hewgrill, "Exact Computation of Steiner Minimal Trees in the Plane", *Information Processing Letters*, Vol.22, pp.151-156, 1986

[7] E.J. Cockayne and D.E. Hewgrill, "Improved Computation of Plane Steiner Minimal Tree", *Algorithmica*, Vol.7, pp.219-229, 1992

[8] F.K. Hwang, "A Primer of the Euclidean Steiner problem", *Annals of Operations Research*, Vol.33, pp.73-84, 1991

[9] F.K. Hwang, D.S. Richards and P. Winter, "The Steiner Tree Problem", *Annals of Discrete Mathematics*, Vol.53, North-Holland, 1992

[10] M.J. Smith and B. Toppur, "Euclidean Steiner Minimal Trees, Minimum Energy Configurations and the Embedding Problem of Weighted Graph in E3", *Discrete Applied Mathematics*, Vol.71, pp.187-215, 1997

[11] P. Winter, "An Algorithm for the Steiner Problem in the Euclidean Plane", *Networks*, Vol.15, pp.323-345, 1985

[12] I. Kim, C. Kim and S.H. Hosseini, "A Heuristic Using GOSST with 2 Connecting Strategies for Minimum Construction Cost of Network", *International Journal of Computer Science and Network Security*, Vol.6, No.12, pp.60-72, 2006

[13] A. Balakrishnan, T.L. Nagnanti, P. Mirchandani, "Modeling and Heuristic Worst Case Performance Analysis of the Two Level Network Design Problem", *Management Science*, Vol.40, pp.846-867, 1994

[14] C. Duin and A. Volgenant, "The Multi-weighted Steiner Tree Problem", *Annals of Operations Research*, Vol.33, pp.451-469, 1991

[15] G.H. Lin and G.L. Xue, "Steiner Tree Problem with Minimum Number of Steiner Points & Bounded Edge-length", *Information Processing Letters*, pp.53-57, 1999

[16] J.R. Current, C.S. Revelle and J.L. Cohon, "The Hierarchical Network Design Problem ", *European Journal of Operational Research*, Vol.27, pp.57-66, 1986

김 인 범 (Inbum Kim) 정회원

1989년 2월 서울대학교 컴퓨터공학과 졸업

1991년 2월 서울대학교 컴퓨터공학과 석사

1991년~1995년 대우통신, 한국오라클 근무

1996년~현재 김포대학 컴퓨터계열 부교수

<관심분야> 컴퓨터이론, 네트워크, 데이터베이스

김 재 각 (Chae-kak Kim) 정회원

1981년 2월 숭실대학교 전자계산학과졸업

1985년 2월 연세대학교 산업대학원 전자계산전공(공학석사)

2002년 2월 숭실대학교 컴퓨터공학과(공학박사)

1985년~1994년 LG전자, 삼보컴퓨터 근무

1996년~현재 김포대학 컴퓨터 계열 부교수

<관심분야> 암호이론, 시스템보안, 인터넷응용