經營科學
第26卷 第1號
2009年 3月

스타이너 트리 문제를 위한 Max-Min Ant Colony Optimization

서민석* · †김대철**

A Max-Min Ant Colony Optimization for Undirected Steiner Tree Problem in Graphs

Minseok Seo* · †Daecheol Kim**

■ Abstract ■

The undirected Steiner tree problem in graphs is known to be NP-hard. The objective of this problem is to find a shortest tree containing a subset of nodes, called terminal nodes. This paper proposes a method based on a two-step procedure to solve this problem efficiently. In the first step, graph reduction rules eliminate useless nodes and edges which do not contribute to make an optimal solution. In the second step, a max-min ant colony optimization combined with Prim's algorithm is developed to solve the reduced problem. The proposed algorithm is tested in the sets of standard test problems. The results show that the algorithm efficiently presents very correct solutions to the benchmark problems.

Keywords : Steiner Tree Problem in Graphs, Ant Colony Optimization, NP-hardness

1. 서 론

방향성 없는 스타이너 트리 문제(Undirected Steiner Tree Problem : USTP)는 NP-hard(Garey and Johnson, 1979)로 알려져 있으며, 따라서 휴리스틱 알고리즘의 개발이 이루어졌다. 특히, 타부서치 방

논문접수일 : 2008년 07월 25일 논문게재확정일 : 2009년 01월 12일
* 삼성전자 Device solution부문(business) Test and Package 센터 TP 기술2팀
** 한양대학교 경영대학 경영학부
† 교신저자

법을 활용한 많은 연구들이 보고되었는데, 그 대표적인 예로써 Minoux(1990), Voss(1990), Xu et al.(1996, 1997), Duin and Voss(1994), 그리고 Gendreau et al.(1998)을 들 수 있다. 최근엔 Reibeiro and Souza(2000)가 기존 타부서치 방법을 수정 보완하여 좀 더 빠른 알고리즘으로의 변화를 시도하였다. 유전자 알고리즘을 이용한 대표적인 논문은 Esbensen(1995), Kapsalis et al.(1993)이고 simulated annealing의 경우 Dowsland(1991)의 사례가 대표적이다.

Ant colony optimization(ACO) 알고리즘은 외판원문제, 생산 스케쥴링 문제 등 다양한 NP-hard문제에 적용되었다(Dorigo and Stützle, 2004; Dorigo and Gambardella, 1997). 그래프상의 스타이너 트리 문제에 대한 ACO알고리즘을 활용한 사례는 최근에 발표된 Singh et al.(2004)과 서민석, 김대철(2008)의 연구에서 찾을 수 있다. Singh et al.은 각 에지(edge)에 페로몬이라는 값을 부여하여 매회마다(즉, 매번 새로운 해를 생성할 때마다) 이 값을 늘리거나 줄여주면서 좋은 해를 찾아가는 ACO 알고리즘의 방법을 활용하였는데, 페로몬이 급격히 줄거나 증가할 경우 다른 영역으로의 확대가 어려워질 수도 있다(Studetzle and Hoos, 2000).

서민석, 김대철(2008)은 일방향 스타이너 트리 문제에 대한 ACO 알고리즘을 개발하였다. 스타이너 트리가 최소신장트리 문제의 특수형태라는 것에 입각하여 이 문제의 최적화 알고리즘인 Prim(1957) 알고리즘을 활용하여 ACO 알고리즘과 접목함으로써 좋은 결과를 나타내는 알고리즘을 개발하였다. 그러나 이 문제에서도 페로몬이 급격히 줄거나 증가할 경우 다른 영역으로의 확대가 어려울 수가 있다. 특히 이러한 문제는 보다 복잡한 양방향 문제에 적용할 경우 좋은 결과를 얻지 못할 수도 있다.

이러한 문제를 개선하기 위해 Blum and Dorigo(2004)는 페로몬의 증감에 대한 한계를 제시하였다. 본 연구에서는 페로몬 값의 증감에 대한 새로운 방식과 최소신장트리 문제에 적용된 Prim(1957) 알고리즘을 활용하여, 기존에 개발된 Singh et al.의 알고리즘 보다 우수한 알고리즘을 개발하고자 한다. 또한, 알고리즘의 효율성을 높이기 위해 그래프 감소에 대한 Beasly(1984), Duin and Volgenant(1989), 그리고 Hwang et al.(1992)의 그래프 제거 법칙들 중 스타이너 문제에 유용한 일부를 사용하고자 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 스타이너 트리 문제에 대한 정의 및 이 문제에 대한 기존의 ACO 적용 연구들에 대하여 살펴본다. 제 3장에서는 본 연구에서 제시하고자 하는 알고리즘에 대한 것으로써, 그래프 감소 과정 및 Prim 알고리즘과 ACO 알고리즘을 접목하여 해를 구하는 전체 과정에 대한 단계별 설명을 포함한다. 제 4장에서는 제안한 알고리즘의 수행도 평가를 위해 28개의 테스트 문제에 대한 본 알고리즘 및 Singh et al.의 비교 수행결과를 제시한다. 제 5장에서는 연구의 의의 및 앞으로의 연구방향에 대한 제시로써 결론을 마치고자 한다.

2. 스타이너 트리 문제(Undirected Steiner Tree Problem : USTP)

그래프상의 USTP는 양방향 그래프 $G=(V, E)$와 모든 에지들 사이의 거리, 그리고 특정 정점(terminal node)들의 집합 $T(T \subseteq V)$가 주어졌을 때, T에 속한 모든 특정 정점들을 잇는 가장 최소의 트리를 찾는 문제라고 할 수 있다. 즉, USTP란 주어진 정점(node)들 중 특정 정점 들을 연결하는 최소신장 트리를 형성하는 문제이다. 스타이너 트리 문제의 대표적인 응용으로는 VLSI 디자인, 송유관 배설과 하수도 배관 문제 등 다양한 실제 네트워크 문제가 이에 해당되며, 보다 자세한 설명은 Hwang et al.(1992)에 잘 나타나 있다.

그래프상의 USTP에 대하여 ACO 알고리즘을 활용한 최근의 예는 Singh et al.(2004)과 서민석, 김대철(2008)에서 찾을 수 있다. 두 연구에서 적용한 ACO 알고리즘은 여러 회를 반복하면서 생성된 해들 중에서 가장 좋은 해를 최종해로 선택하는 것

인데 이러한 방식은 다른 메타휴리스틱 알고리즘들과 유사하다. 그러나 새로운 해를 생성할 때 기존 해에 바탕을 두고 그 중 일부분을 변경하는 형식이 아니라, 일단 한 해가 구해지면 다음 회에서는 기존에 구한 해의 에지에 부여된 페로몬이라는 값을 늘리거나 줄여준 후, 변경된 페로몬 값에 의거하여 전체 경로를 처음부터 새로이 생성하는 과정을 반복한다. 이 연구들의 알고리즘에서 이런 페로몬양을 변화시키는 과정에서 페로몬양을 급격하게 증가시키거나 줄어든 페로몬양과 증가된 페로몬양의 차가 클 경우에는 지역해로 빠지게 될 가능성이 높으며 이를 방지하기 위해 기존의 ACO 알고리즘을 기초로 페로몬의 증감을 제한시키는 방법이 개발되었다(Studetzle and Hoos, 2000; Blum and Dorigo, 2004).

Blum and Dorigo(2004)의 페로몬 증감 제한 방식과 일반적인 ACO 방식과의 차이는 첫째, 초기화 단계에서 일반적인 ACO 알고리즘은 초기 페로몬양을 계산하는 반면 새로운 방식은 페로몬양을 0.5로 고정한다. 둘째로, 일반적인 ACO 알고리즘은 페로몬양의 증감에 대한 한계선이 없는 반면, 새로운 방식은 페로몬값을 0과 1로 한정시켜 놓았다. 또한 새로운 방식에서는 연속적으로 같은 해를 반복 생성할 경우 페로몬양을 첫 페로몬양인 0.5로 다시 초기화시켜 새로운 해를 생성시키게 하였다. 셋째로, 일반적인 ACO 알고리즘에서는 매 회마다 선택된 해에 포함된 에지의 페로몬양을 줄이고 미리 지정된 회수만큼 해를 생성시킨 후 현재까지 가장 좋은 해에 포함된 에지의 페로몬양을 늘려주었다. 하지만 새로운 방식에서는 미리 지정된 수 만큼 해를 생성시킨 후 현재까지 가장 좋은 해에 포함된 에지의 페로몬양은 늘려주고 포함되지 않은 에지의 페로몬양은 줄여주었다. Blum and Dorigo(2004)의 연구에 따르면 이러한 새로운 방식은 일반적인 ACO 알고리즘에 비해 페로몬값을 보다 효과적으로 관리하여 지역해로 빠지는 것을 방지하는 효과가 있음을 알 수 있다. 따라서 본 논문에서는 스타이너 트리 문제를 풀기 위해 페로몬 증감에 대한 새로운 방식을 사용하고자 하며 기존에 스타이너 문제에 이러한 방법을 적용한 사례는 없는 것으로 알고 있다.

또한, 매 회마다 트리를 형성하기 위하여 앞서 논의되었던 모든 정점이 특정 정점일 경우 이 문제는 최소신장트리문제가 된다는 점에 착안하여, 최소 신장트리 문제에 대해 항상 최적해를 제공하는 Prim(1957) 알고리즘을 활용하고자 한다. 따라서 본 논문에서는 Prim 알고리즘을 활용하여 새로운 페로몬 증감방식을 적용한 ACO 알고리즘과 접목시킴으로써 스타이너 트리를 형성할 수 있도록 한다. 또한, 선행단계로 그래프 제거 법칙을 이용하여 문제의 크기를 줄임으로써 본 연구에서 제시하고자 하는 알고리즘의 효율을 증대시키도록 하고자 한다.

3. GPM 알고리즘

새로운 페로몬 증감 방식을 적용한 ACO 알고리즘을 Max-Min ACO(MMACO)라고 할 때, 본 연구에서 개발하고자 하는 알고리즘 즉, 그래프 제거법칙과 Prim 알고리즘을 활용하여 MMACO와 접목시킨 알고리즘을 GPM(Graph reduction, Prim algorithm, MMACO) 알고리즘이라고 하자. GPM 알고리즘은 <그림 1>과 같이 크게 두 부분으로 구성되어 있다. 첫 번째는 알고리즘의 효율성을 높이기 위한 그래프 감소단계로서 기존의 그래프 제거법칙들을 활용한 단계이다. 두 번째는 MMACO를 적용하여 최소의 스타이너 트리를 구하는 과정으로 필요한 변수들을 초기화하는 과정, 해의 생성과정, 에지의 페로몬 갱신과정, 그리고 에지에 부여된 페로몬 값의 수렴도 테스트 과정 등으로 구성되어 있다.

3.1 Graph Reduction

그래프상의 스타이너 트리 문제에서 최소 스타이너 트리를 구하고자 하는 GPM 알고리즘의 첫 단계로 그래프 제거 법칙을 사용함으로써 문제의 크기를 줄이고, 다음 단계에서 수행되는 MMACO

단계 효율을 증대시키도록 한다.

```
Procedure GPM 알고리즘(V', E')
  input SOL ; 매회마다 생성되는 스타이너 트리의 값
  input BS ; 현재까지 생성된 스타이너 트리중 가장 좋
          은 스타이너 트리의 값
  input r   ; 시작정점, r ∈ T
  input TM ; 시작정점으로 선택된 특정 정점들의 집합
  input n ; 페로몬 갱신에 필요한 해를 생성하기 위한
          반복회수
  input m ; 페로몬 갱신이 수행되는 회수

**Graph Reduction 단계 **

 Call Graph Reduction Procedure(V', E')
     New graph D(V, E) ; graph reduction 단계의 결
     과로 얻은 새로운 그래프

**MMACO 단계**

  Call 초기화 과정
       graph D(V, E) ; 모든 에지의 초기 페로몬 값(0.5)
                       이 부여된 그래프
  TM ← { },
  특정 정점들 중에서 임의로 특정정점 r을 선택
  TM ← TM ∪ {r}, j = 1
  for k = 1 to m do
     Call 시작정점선택과정
     if (j+1 ≤ |T|)
        then, 특정 정점들 중에서 TM에 포함되지 않은
        특정정점 r을 임의로 선택
          TM ← TM ∪ {r}
     else, TM ← { },
        특정 정점들 중에서 임의로 특정정점 r을 선택
        TM ← TM ∪ {r}, j = 1
     end-if
     for i = 1 to n do
        Call 생성과정
        SOL ← 생성과정에 의해 생성된 스타이너 트리
        의 값
        if i = 1, then BS ← SOL
        if (BS > SOL) then
           BS ← SOL
        end-if
        end-for
        Call 페로몬 갱신과정(BS)
        Call 수렴도테스트 과정
  end-for
end-procedure
```

〈그림 1〉 GPM 알고리즘의 수행과정

본 연구에서 사용된 그래프 제거의 방법들은 모두 여덟 개이며, 이를 설명하기 위해 필요한 용어에 대한 정의는 아래와 같다.

(x, y) : 정점 x와 정점 y를 잇는 에지,

N_x : 정점 x에 이웃한 정점들의 집합, $x \in V$,

$\deg(x)$: 정점 x를 잇는 총 에지의 개수, $x \in V$,

$c(x, y)$: 에지(x, y)의 길이,

$c(x, y)$: 정점 x와y를 잇는 최소경로, $x, y \in V$,

법칙 1 : 만약 $y | T$ and $\deg(y) = 1$, $x \in N_y$이면, y와 에지(x, y)는 제거되며, 제거된 $c(x, y)$는 최종 스타이너 트리에 포함된다. 또한 $x \not\in T$인 경우, x는 y가 제거된 후 특정 정점으로 변환된다(Beasly, 1984; Duin and Volgenant, 1989).

법칙 2 : 만약 $y \not\in T$ and $\deg(y) = 1$, $x \in N_y$이면, y와 에지는 제거된다(Beasly, 1984; Duin and Volgenant, 1989).

법칙 3 : 만약 $y \not\in T$ and $\deg(y) = 2$, $x, z \in N_y$이면, y와 에지(y, x) and (y, z)는 제거된다. 만약 에지(x, z)가 존재하고 $c(x, z) > c(y, x) + c(y, z)$이면, $c(x, z) = c(y, x) + c(y, z)$로 변환된다. 만약 에지$(x, z)$가 없다면, 에지$(x, z)$가 그래프에 추가되며 $c(x, z) = c(y, x) + c(y, z)$가 된다(Beasly, 1984; Duin and Volgenant, 1989),

법칙 4 : 만약 $c(x, y) > d(x, y)$이면, 에지(x, y)는 제거된다(Beasly, 1984; Duin and Volgenant, 1989).

법칙 5 : 만약 $x, y \not\in T$ and $z | T$이면서 조건 $c(x, y) > \max\{d(x, z), d(y, z)\}$를 만족하면, 에지$(x, y)$는 제거된다(Duin and Volgenant, 1989; Hwang et al., 1992).

법칙 6 : 두 정점 $x, y \in V$를 잇는 경로상의 중간 정점들이 모두 특정 정점일 경우, 이러한 경로를 특정 경로라고 하자. 특정 경로상의 가장 긴 에지의 길이를 특정 에지라고 할 때, 두 정점 x와 y를 잇는 가능한 모든 특정경로중 가장 짧은 특정 에지의 길이가 $c(x, y)$보다 작으면 에지(x, y)는 제

거된다(Duin and Volgenant, 1989).

법칙 7 : $x \not\in T$and $\deg(x)=3$이라고 할 때, $i, j, k \in N_x$ 만을 이용하여 따로 완전그래프를 만든다고 하자. 이 완전그래프를 잇는 각 에지의 길이는 원래 그래프에서 한 쌍의 정점 x에 이웃한 정점들을 잇는 최소경로의 길이와 같다고 하자. 만약 이 완전그래프의 최소신장트리의 값이 $c(x, i)+c(x, j)+c(x, k)$보다 작거나 같으면, x와 에지 (x, i), (x, j), (x, k)들은 제거된다. x가 제거되면, 이웃한 정점들을 서로 잇는 에지들의 길이, 예를 들어 $c(i, j)$는 $c(x, i)+c(x, j)$와 비교하여 짧은 길이로 치환된다(Duin and Volgenant, 1989),

법칙 8 : 완전그래프를 잇는 각 에지의 길이가 완전그래프가 아닌 원래 그래프에서 한 쌍의 정점 x에 이웃한 정점들을 잇는 법칙 6에서 정의되었던 특정에지 중 가장 짧은 특정에지의 길이와 같을 때도 법칙 7이 적용 가능하다.

3.2 MMACO

GPM의 두 번째 단계인 MMACO의 첫 과정인 초기화 과정에서는 초기 페로몬 값 (0.5)을 각 에지에 부여한다. 이 후 해의 생성과정에서는 Prim 알고리즘을 활용하여 각 에지의 페로몬 값과 거리를 고려하여 해당 정점을 하나씩 추가시켜 최종적으로 스타이너 트리를 형성시킨다. 이렇게 형성된 스타이너 트리에 포함된 에지들과 그렇지 않은 에지들의 페로몬 값의 증감을 페로몬 갱신과정에서 실행하고 다음 회를 진행시킨다. 정해진 수의 해들을 생성한 후 가장 좋은 해를 최종적으로 선택하는 종결과정을 통해 알고리즘을 마치게 된다. 각 과정에 대한 자세한 설명은 다음에 자세히 나타나 있다.

3.2.1 초기화

초기화 과정은 <그림 2>와 같이 MMACO에 필요한 파라미터들의 초기 값을 부여하는 단계이다. 서민석, 김대철(2008)에서 적용한 ACO의 경우, 초기 페로몬 값을 부여하기 위하여 Prim 알고리즘을 활용하여 초기해를 구한 뒤, 초기 스타이너 트리의 총거리의 합과 그 초기해에 포함된 총 노드의 수를 곱한 값의 역수를 취하여 얻을 수 있다. 본 연구에서는 이러한 초기해의 생성 과정을 거치지 않고 경험치를 바로 부여함으로써 초기해 생성에 따른 소요시간을 줄일 수 있다. 초기 값은 페로몬의 값이 너무 커지거나 작아지는 문제해결을 위한 페로몬의 상/하한의 범위로부터 구할 수 있는데, 본 연구에 사용된 상한값(= 1)과 하한값(= 0)은 실험에 의한 경험치이다. 초기값을 상/하한의 중간 값인 0.5로 설정함으로써 증가의 폭과 감소의 폭이 동일한 조건에서 시작하도록 배려하였다.

```
Procedure 초기화
  NewGraph D = (V, E)
    τ0 ; 초기 페로몬의 양(= 0.5)
    τ(x,y) ; 에지 (x, y)의 페로몬의 양
    v ; 정점들의 수 (= |V|)

  for x = 1 to x = v
    for y = 1 to y = v
      if (x, y)∈E, then τ(x,y) ← τ0
      end-if
    end-for
  end-for

 q0 ; 랜덤 수(= 0.8),
 ky ; 정점의 가중치 계수(= 7)
 ρ ; 페로몬 증발계수(= 0.9)
 α ; 페로몬 값의 가중치 지수(= 1),
 β ; 에지 길이의 가중치 지수(= 1),
 n ; 페로몬 갱신에 필요한 해를 생성하기 위한 반복회
     수(= 10)
 m ; 페로몬 갱신이 수행되는 회수(= 150)

End-procedure
```

〈그림 2〉 초기화 과정

3.2.2 생성

MMACO단계에서는 해를 생성하기 위해(<그림 3> 참조) Prim 알고리즘을 활용하여 시작정점을 시작으로 정점들을 추가하여 그래프안의 모든 정

점이 포함될 때까지 계속 진행된다.

Prim 알고리즘을 이용한 MMACO는 시작정점에 따라 탐색영역이 달라지게 되는데 이는 새로운 정점을 추가하는 매 스텝마다 선택된 정점에서 근접한 정점을 선택하게 되는 알고리즘의 특성에 기인한다. GPM 알고리즘에서는 시작정점을 정할 때 우선 임의로 하나의 시작정점을 정한다. 이후 n개의 해를 구한 뒤 다음 n개의 해를 구하기 위한 시작정점은 시작정점으로 정했던 특정정점을 제외한 나머지 특정 정점 중에서 하나를 임의로 정하게 된다.

```
procedure 생성
  input M    ; 선택된 정점들의 집합
  input F    ; 선택된 에지들의 집합
  input r    ; 시작정점, r ∈ T
  input N_x  ; 정점 x에 인접한 정점들의 집합
  input CAN ; 현재 선택이 가능한 정점들의 집합

  M ← {r}
  F ← NULL
  q ← RAN ; (0, 1)사이의 랜덤 수
  CAN ← {y : y ∉ N_x \ M, x ∈ M, (x, y) ∈ E \ F}
  do until ( T ⊂ M )
    q ← RAN
    if q ≤ q_0,
      then select a node y using deterministic rule in
      CAN
      y ∈ argmax {k_y · (τ_(x,y))^α (η_(x,y))^β :
                  x ∈ M, y ∈ N_x \ M, (x,y) ∈ E \ F}
      else select a node y using roulette wheel
      selection rule in CAN
      p_(x,y) = k_y · (τ_(x,y))^α · (η_(x,y))^β / Σ_{x∈M} Σ_{y∉M} k_y · (τ_(x,y))^α · (η_(x,y))^β ;
             x ∈ M, y ∈ N_x \ M, (x, y) ∈ E \ F
    end if
    M ← M ∪ {y}
    F ← F ∪ {(x, y)}
  end-do
end-procedure
```

〈그림 3〉 생성 과정

이렇게 시작정점이 결정되면 선택된 정점과 이웃한 정점들 중에서 추가할 정점을 선택하게 되는데, 기존의 서민석, 김대철(2008)의 경우와 달리 이웃한 정점들 중에서 특정정점에 가중치를 부여하여 알고리즘이 가장 가까운 정점보다 특정 정점을 우선적으로 선택하도록 한다. 스타이너 트리는 최소신장트리처럼 모든 정점을 포함하는 최소트리를 만드는 것이 아니기 때문에 가중치를 부여함으로써 스타이너 문제에 부합한 즉 특정 정점을 모두 포함한 최소신장트리를 만들게 된다.

해의 생성을 설명하기 위해 두 개의 집합이 필요한데, 추가되는 정점들을 포함할 집합 M과 추가되는 에지를 포함하는 집합 F이다. 알고리즘이 시작하기 전 시작정점(r)은 집합 M에 포함되게 되고, 집합 F의 초기값은 공집합이 된다.

이후 M에 포함된 정점(들)에서 인접한 정점들과 그를 잇는 에지가 다음에 추가될 정점의 후보가 되는데, 이때 해당 에지의 길이와 페로몬양을 고려하여 아래에 주어진 식 (1)에 의하여 정점을 선택하게 되고, 선택된 정점과 해당 에지는 각각 M과 F에 포함된다. 이 과정은 M에 주어진 모든 특정 정점들이 포함될 때까지 계속된다.

$$\cdot\ y \in \arg\max\left\{k_y \cdot (\tau_{(x,y)})^{\alpha}(\eta_{(x,y)})^{\beta} : x \in M,\ y \in N_x \setminus M, (x,y) \in E \setminus F\right\} \quad (1)$$

식 (1)에서 k_y는 앞서 논의되었던 정점의 가중치이다. 이 정점의 가중치는 MMACO의 해 생성과정에서 알고리즘이 특정 정점을 우선적으로 선택하게 도와준다.

스타이너 문제에서 정점을 선택할 때 식 (1)뿐만 아니라 아래의 식 (2)도 이용되는데, 식 (2)는 식 (1)에 의해서만 추가 정점을 선택하는 경우보다, 정점 선택의 영역을 좀 더 확대시켜주는 역할을 한다. 어떤 방식이 택해지는가는 확률적으로 결정되는데 우선 $q_0 \in (0,1)$을 임의로 정한 후, 새로운 정점을 추가할 때마다 새로운 $q_0 \in (0,1)$ 값을 생성시켜서, 만약 이 값이 $q \le q_0$이면, 식 (1)을 이용하고 그렇지 않으면 아래의 식 (2)를 이용한다.

$$p_{(x,y)} = \frac{k_y \cdot (\tau_{(x,y)})^{\alpha} \cdot (\eta_{(x,y)})^{\beta}}{\sum_{x \in M} \sum_{y \notin M} k_y \cdot (\tau_{(x,y)})^{\alpha} \cdot (\eta_{(x,y)})^{\beta}};$$
$$x \in M,\ y \in N_x \setminus M,\ (x,\ y) \in E \setminus F \tag{2}$$

3.2.3 페로몬 값의 갱신

기존의 ACO 알고리즘(서민석, 김대철, 2008)에서는 매회 해가 구해지면 해에 포함된 에지들의 페로몬 값을 줄여준다. 이런 이유는 ACO 알고리즘이 다음 번에 다른 해를 생성하게 도와줌으로써 같은 해를 생성하는 것을 막아주는 효과를 가져다 준다. 이 후 몇 회 동안 이런 과정 즉, 해를 생성하고 페로몬양을 줄여주는 과정을 n회(대략 10회 정도) 수행한 후 가장 좋은 해에 해당되는 에지의 페로몬양을 증가시켜주어 다음 n회 동안 이 증가된 페로몬양을 바탕으로 현재 제일 좋은 해를 보인 인근지역 다시 말해 더 좋은 해를 보일지 모르는 영역의 탐색을 도와준다. 그러나 이러한 ACO 알고리즘은 최적 해에 포함될지도 모르는 에지에 페로몬양을 n회 동안 반복적으로 줄여주거나 n회 동안 생성된 해중 가장 좋은 해가 지역해일지라도 지나치게 페로몬양을 증가시켜 반복적으로 알고리즘이 같은 지역해를 생성시킬 수도 있는 문제점이 있다(Studetzle and Hoos, 2000).

반면 MMACO 에서의 페로몬 갱신은 적용된 방식은 증가된 페로몬양을 제한된 범위 안에서 증감시키기 때문에 즉, $n(=10)$회 동안 생성된 해중에서 가장 좋은 해에 포함된 에지의 페로몬양을 제한된 범위 내에서 늘려주고 포함되지 않은 해는 제한된 범위 내에서 페로몬양을 줄여줌으로써 기존 ACO 알고리즘이 가지고 있는 문제점 즉, 지역해로 빠지는 것을 방지하도록 도와준다. MMACO에서는 식 (3)을 사용하여 페로몬양을 줄여주고, 식 (4)를 이용하여 페로몬양을 늘려준다(<그림 4> 참조).

$$\tau_{(x,y)(new)} = \tau_{(x,y)(old)} - \rho \times \tau_{(x,y)(old)},$$
$$(x,\ y) \in E \setminus F_{(best)}, \tag{3}$$
$$\tau_{(x,y)(new)} = \tau_{(x,y)(old)} + \rho \times (1 - \tau_{(x,y)(old)}),$$
$$(x,\ y) \in F_{(best)}, \tag{4}$$

식 (3)과 식 (4)의 $\rho \in (0, 1)$값은 페로몬의 증발 속도를 결정해주는 지표이다.

```
Procedure 페로몬 갱신
  NewGraph D = (V, E)
   F(best) ; 제일 좋은 해에 속한 에지들의 집합
   E' ; F(best)에 속하지 않는 에지들의 집합 (E\ F(best))
   v ; 그래프에서 총 정점의 개수(= |V| )
   τ(x,y)(old) ; 변화 전 에지 (x, y)의 페로몬의 양
   τ(x,y)(new) ; 변화 후 에지 (x, y)의 페로몬의 양
while ( E'≠ φ and F(best) ≠ φ) do
  For (x = 1 to v)
     If ( (x, y) ∈ E' )
        τ(x,y)(new) ← τ(x,y)(old) − ρ × τ(x,y)(old) ; E'에 속한 해
        당 에지의 페로몬양을 변경
        E' ← E' - {(x, y)}
     Else ((x, y) ∈ F(best))
        τ(x,y)(new)←τ(x,y)(old) + ρ × (1 − τ(x,y)(old)) ; F(best)에
        속한 해당 에지의 페로몬양을 변경
         F(best) ← F(best) - {(x, y)}
       End-if
   End-for
  end-while
end-procedure
```

〈그림 4〉 페로몬 갱신 과정

MMACO에서 증감하는 페로몬 양을 $0(=\tau_{\min})$과 $1(=\tau_{\max})$사이의 구간에 존재시키기 위해 다음과 같은 식으로 증감양을 변화시킨다. 식 (5)에서 x는 식 (3)과 식 (4)를 통해 변화된 페로몬 값을 의미한다.

$$f(x)\begin{cases}\tau_{\min} & if\ x < \tau_{\min}, \\ x & if\ \tau_{\min} \le x \le \tau_{\max}, \\ \tau_{\max} & if\ x > \tau_{\max}.\end{cases} \tag{5}$$

3.2.4 수렴도 테스트(Convergence Test)

수렴도 테스트는 페로몬양을 변화시킨 후 MMACO 단계에서 동일한 해만을 계속적으로 생성시키는 상황(즉, 정체상황이라 한다)에 도달했는지를 평가해 준다(Blum and Dorigo, 2004). 이는 페로몬 값이 어떤 에지에서는 0에 근접하게 되어 더 이상 해당 에지가 해에 포함되지 않거나 1에 근접하게 되어 해당 에지가 항상 해에 포함되는 상황을 평가해주는 과정이다.

$$CF = 2 \cdot \left(\left(\frac{\sum \max \{ \tau_{\mathrm{man}} - \tau_{(x,y)}, \tau_{(x,y)} - \tau_{\mathrm{min}} \}}{|E| \cdot (\tau_{\mathrm{man}} - \tau_{\mathrm{min}})} \right) - 0.5 \right) \tag{6}$$

식 (6)에서 *CF*(convergence factor)값이 0.001이하이거나 0.999이상 일 때를 정체상황이라고 보고 모든 에지의 페로몬값을 초기화 시킨다. 알고리즘에서 정체현상을 보인다는 것은 수치적으로 모든 에지나 에지의 페로몬 값이 0이나 1에 근접한다는 것을 의미한다.

3.2.5 종결

이 과정을 처음 정한 최대 허용 회수만큼 반복한다. 모든 메타휴리스틱과 마찬가지로 허용 회수가 높을수록 많은 해를 생성할 수 있기 때문에 좋은 해를 구할 확률이 높게 된다. 보통 문제의 크기에 따라 허용 회수는 많아지게 된다. 해의 최대 생성횟수는 $n \times m$(본 연구에서는 10×150)이다.

5. 실험결과

그래프상의 스타이너 문제를 위한 그래프 제거 법칙과 변형된 Prim 알고리즘과 MMACO를 접목시킨 GPM 알고리즘은 C/C++프로그램언어로 만들어졌으며, Pentium IV 3.4 GHz, RAM 1GB환경에서 테스트 되었다. GPM 알고리즘의 수행도 평가를 위하여 Beasley(1984)의 문제집합 B에 제시된 양방향 스타이너 트리 18개의 benchmark 문제가 사용되었으며, 주어진 문제들에 대한 Singh et al.(2004)의 알고리즘과 비교 평가하였다. 또한 Beasley의 문제집합 C에 대한 결과도 제시하였다.

<표 1>~<표 2>는 USTP에 대한 GPM 알고리즘의 결과를 나타내고 있다. <표 1>에서 첫 번째 열은 Beasley가 제시한 Set B의 문제 이름이고 두

〈표 1〉 알고리즘 수행도 평가 결과(Set B)

이름	Original problem			Reduced problem			GPM	Singh et al's	GPA	Best Known	Gap	Time (sec)
	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$						
b01	50	63	9	1	0	1	82	82	82	82	0	0
b02	50	63	13	1	0	1	83	83	83	83	0	0
b03	50	63	25	13	15	11	138	138	138	138	0	0.02
b04	50	100	9	20	33	8	59	59	59	59	0	0.04
b05	50	100	13	12	15	10	61	61	61	61	0	0.01
b06	50	100	25	35	52	23	122	-	122	122	-	0.13
b07	75	94	13	8	9	7	111	-	111	111	-	0.01
b08	75	94	19	19	24	14	104	104	104	104	0	0.03
b09	75	94	38	17	21	15	220	220	220	220	0	0.03
b10	75	150	13	46	95	13	86	86	86	86	0	0.27
b11	75	150	19	53	107	19	88	88	88	88	0	0.34
b12	75	150	38	55	85	31	174	-	174	174	-	0.27
b13	100	125	17	33	51	13	165	165	165	165	0	0.09
b14	100	125	25	39	59	20	235	235	235	235	0	0.15
b15	100	125	50	35	43	30	318	318	318	318	0	0.10
b16	100	200	17	72	149	17	127	132	130	127	5	0.61
b17	100	200	25	58	106	22	131	-	131	131	-	0.33
b18	100	200	50	61	97	41	218	223	219	218	5	0.38

번째부터 네 번째 열은 문제의 크기를 표시한다. 두 번째 열은 각 문제의 총 정점의 수를 나타낸다. 그리고 세 번째 열은 주어진 문제들의 총 에지의 수를 의미하며, 네 번째 열은 총 특정 정점의 개수를 보여주고 있다. 다섯 번째부터 일곱 번째 열은 그래프 제거 법칙으로 줄어든 문제의 크기를 보여준다. 여덟 번째 열은 최고 좋은 해로 알려진 스타이너 트리문제의 해와 GPM 알고리즘으로 구한 값이다. 아홉 번째 열은 Singh et al.이 개발한 알고리즘의 결과를 나타내고 열 번째 열에는 서민석, 김대철(2008)의 GPA 알고리즘의 결과가 나타나 있고, 다음 열에는 지금까지 알려진 각 문제에 대한 최우수해가 제시되어 있다. 열두 번째 열은 Singh et al.의 결과와 GPM 알고리즘의 결과와의 차이가 나타나 있다. 다음 열에는 CF 값의 초기화 횟수가 나타나 있다. 마지막 열은 각 문제에 대한 GPM 알고리즘의 평균 소요시간을 나타낸다.

GPM 알고리즘은 선행 실험결과 다음과 같은 파라미터 값을 이용할 때 가장 좋은 해를 보인다($\alpha=1$, $\beta=2$, $\rho_0=0.9$, $k_y=7$, $n=10$, $m=150$). GPM 알고리즘의 최대 허용 해 생성횟수($n \times m$)는 1500회이다. <표 1>에서 보듯이, 본 연구에서 개발한 GPM 알고리즘은 모든 benchmark 문제들에서 기존에 알려진 최적해와 동일한 해를 생성하였음을 알 수 있다. 또한 Singh et al.및 서민석, 김대철(2008)의 결과와 비교하였을 때, 모든 benchmark 문제들에서 동일하거나 보다 우수함을 알 수 있다. Singh et al.의 결과에 비해 우수한 해를 구한 이유 중의 하나는 최소신장트리 문제의 최적해를 제공하는 Prim 알고리즘을 응용하였다는 점이라고 할 수 있다. 스타이너 트리는 특정 정점으로 이루어진 트리라는 점에서 최소신장트리 문제와 매우 유사하다고 할 수 있다. 따라서 유사한 구조를 지닌 문제에 대한 최적해 생성 기법을 적용함으로써 우수한 해를 얻을 수 있었다. 또한, 서민석, 김대철(2008)의 GPA 알고리즘은 페로몬의 증감에 대한 제약이 없는 관계로 동일한 해가 반복되거나 유사한 해에 교착될 수 있는 문제점을 가지고 있는데, 페로몬의 변화 범위를 한정함으로써 이러한 상황을 해결할 수 있었고 결과적으로 보다 다양한 해를 생성시킬 수 있었던 것도 하나의 이유가 될 수 있다.

<표 2>에서는 Beasley가 제시한 Set C의 문제에 대한 GPM 알고리즘의 수행도 결과를 볼 수 있다. 결과에서 보듯이 테스트한 모든 문제에서 기존에 알려진 최우수 값과 일치하였다. 알고리즘이 문제를 해결하는데 소요되는 시간은 전반적으로 Set B에 비해 문제 크기가 늘어남으로써 증가되었는데, 전체 정점 및 에지의 수 보다는 특정 정점의

〈표 2〉 알고리즘 수행도 평가 결과 2(Set C)

이름	Original problem			Reduced problem			GPM	Best Known	Gap	CF	Time (sec)
	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$					
c01	500	625	5	142	255	5	85	85	0	3	2.50
c02	500	625	10	128	236	8	144	144	0	3	6.35
c03	500	625	83	120	223	37	754	754	0	3	93.55
c04	500	625	125	95	175	35	1079	1079	0	3	66.04
c05	500	625	250	1	0	1	1579	1579	0	-	0
c06	500	1000	5	369	847	5	55	55	0	3	6.74
c07	500	1000	10	382	869	9	102	102	0	3	24.09
c08	500	1000	83	353	825	56	509	509	0	3	432.05
c09	500	1000	125	350	817	80	707	707	0	3	767.92
c10	500	1000	250	52	91	24	1093	1093	0	3	27.08

수와 밀접한 연관이 있음을 알 수 있다. 테스트 한 두 세트 문제에서 CF 값이 0.9이상의 값을 가질 빈도수가 1500개의 해를 생성하는 동안 3회로 나타났다. 이것은 MMACO 알고리즘이 평균 약 400~500회 정도마다 더 이상 좋은 해를 발견하지 못하고 정체된다는 것을 의미하며, 이러한 정체상황을 페로몬값을 초기화 해줌으로써 MMACO 알고리즘이 더 좋은 해를 찾을 수 있도록 도와주게 된다. 이상의 결과를 종합할 때, GPM 알고리즘은 주어진 문제에 대하여 효율적이고 정확한 해를 생성한다고 할 수 있다.

6. 결 론

이 연구에서는 변형된 Prim 알고리즘을 MMACO 알고리즘과 접목시킴으로써 그래프상의 스타이너 문제를 효과적으로 해결할 수 있었다. 문제의 해를 구하기 전에 그래프 제거법칙을 이용하여 스타이너 문제의 크기를 줄이고, 이후 줄어든 그래프에서 ACO 알고리즘을 이용함으로써 최적해를 보다 빠르게 구하고자 했다. 양방향 그래프 경우 그래프제거법칙에 의해 문제의 크기가 줄어든 정점수를 기준으로 하면 크게 40% 이상 감소시킬 수 있었는데, 본 연구의 공헌도는 아래의 몇 가지 측면에서 살펴볼 수 있다.

첫째, ACO 메타휴릭스틱 알고리즘에 새로운 경로를 구축해 나가는 방법으로 Prim 알고리즘을 접목시켜 좋은 결과의 알고리즘을 개발하였다. ACO 메타휴리스틱은 개미의 길 찾기 특성을 반영한 것이다. 개미들은 자신들의 둥지에서 먹이가 있는 곳으로 이동을 하면서 길에 페로몬이라는 화학물질을 뿌려 놓는데, 여러 갈래의 길 중에서 가장 짧은 경로에 가장 많은 페로몬이 쌓이고 이를 활용하여 나중의 개미들은 자연스럽게 이 짧은 경로를 선택하게 되는 특성을 지닌다. 이러한 특성이 그래프상의 스타이너 트리 문제에 적용될 때, 그래프를 구성하고 있는 각 에지들 중에서 우수하다고 생각되는 경로에 페로몬 값이 많이 쌓이는 것으로 나타난다. 그러나 ACO 자체로는 각 에지의 중요도는 부여할 수 있지만 하나의 경로를 구축해 나가는 방법은 제시되지 않는다. 스타이너 트리는 모든 정점이 아닌 특정 정점을 포함한 최소신장트리를 구축하는 문제인데 일반적인 최소신장트리는 스타이너 트리의 한 특수한 형태라고 할 수 있다. 따라서 일반적인 최소신장 트리문제에 대한 최적해인 Prim 알고리즘을 스타이너 트리 문제에 적용하여 좋은 해를 가질 수 있었다.

둘째, 기존의 ACO 알고리즘에서 에지의 페로몬 값을 갱신하는 방식을 개선한 최신의 MMACO 알고리즘 방식을 도입함으로써 좋은 결과를 얻을 수 있었다. 기존의 ACO에서는 직전 회에 생성된 해에 포함된 에지의 페로몬 값을 줄여줌으로써 다음 회의 해 생성 시 가능하면 그 에지는 선택가능성이 낮도록 하고, 또한 미리 정해진 횟수만큼의 해를 구한 후 그 중에서 가장 우수한 해에 포함된 에지의 페로몬은 증가시켜 최적해가 있을 가능성이 높은 에지를 보유하도록 하고 있다. 그러나 페로몬을 증감시키는 과정에서 페로몬 값이 급격하게 증가하거나 또는 줄어든 페로몬의 값과 증가된 페로몬의 값의 차이가 클 경우에는 동일한 해가 반복해서 생성되는 정체상태에 빠져들 가능성이 높아진다. 이를 개선하기 위한 Blum and Dorigo(2004)의 최근 연구로부터 페로몬의 수렴도 테스트를 도입하여 알고리즘이 정체상태를 벗어나도록 하는 MMACO 알고리즘 방식을 적용함으로써 보다 다양한 해가 고려되도록 하였다.

셋째, 기존 연구에서 제시되었던 그래프의 감소법칙들을 그래프상의 스타이너 트리 문제에 적용하여 본래의 문제를 감소시켜 알고리즘의 효율성을 얻을 수 있었고, 끝으로, 위의 여러 시도를 통한 GPM 알고리즘을 개발함으로써 기존의 Singh et al. 의 알고리즘보다 신속하고 정확한 결과를 얻을 수 있었다. 경쟁적인 산업구조 속에서 최적해 또는 최적 해에 근사한 해를 얼마나 빨리 구하는가 하는 문제는 매우 중요하다. 본 연구에서 다루었던 스타이너 트리 문제는 현재 많은 산업현장에서 쓰이는

실제적인 문제로 적용이 가능하기 때문에 이 문제를 위한 빠른 알고리즘 개발은 산업현장에 도움을 줄 수 있다. 본 논문에서 제시한 GPM 방법을 이용하면 생산성 향상에 큰 도움이 될 것이다. 앞으로의 연구방향으로 보다 빠르고 정확한 알고리즘 개발을 위해 추가적인 그래프 제거법칙들의 개발 또는 ACO 알고리즘 자체의 성능을 보다 향상시키는 연구가 필요하다고 할 수 있다.

참 고 문 헌

[1] 서민석, 김대철, "스타이너 트리 문제를 위한 Ant Colony Optimization 알고리즘의 개발", 「한국경영과학회지」, 제33권, 제3호(2008), pp.17-28.

[2] Beasley, J.E., "An Algorithm for the Steiner Problem in Graphs," *Networks*, Vol.14(1984), pp.147-159.

[3] Blum, C. and M. Dorigo, "The Hyper-Cube Framework for Ant Colony Optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol.34(2004), pp.1161-1172.

[4] Dorigo, M. and T. Stützle, *Ant Colony Optimization*, The MIT press, London, England, 2004.

[5] Dorigo, M. and L.M. Gambardella, "Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, Vol.1(1997), pp.53-66.

[6] Dowsland, K.A. Hill-climing, "Simulated Annealing and the Steiner Problem in Graphs," *Eng. Optimization*, Vol.17(1991), pp.91-107.

[7] Duin, C.W and S. Voss, "Steiner Tree Heuristics-A Survey," Operations Research Proceedings, Springer-Verlag, Berlin, 1994, pp.485-496.

[8] Duin, C.W. and A, Volgenant, "Reduction Tests for the Steiner Problem in Graphs," *Networks*, Vol.19(1989), pp.549-567.

[9] Esbensen, H., "Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm," *Networks*, Vol.26(1995), pp.173-185.

[10] Kapsalis, A., V.J. Rayward-Smith, and G.D. Smith, "Solving the Graphical Steiner Tree Problem Using Genetic Algorithms," *Journal of Operation Research Society*, Vol.44(1993), pp.397-406.

[11] Koch, T., A. Martin, and S. Voss, "SteinLib: An Updated Library on Steiner Tree Problems in Graphs," Technical Report ZIB-Report 00-37, Konrad-Zuse-Zentrum fűr Informationstechnik Berlin, 2000.

[12] Garey, M.R. and D.S. Johnson, *Computers and Intractability : A Guide to Theory of NP-Completeness*, Freeman, New York, 1979.

[13] Hwang, F.K., D.S. Richard, and P. Winter, *The Steiner Tree Problem*. North-Holland, Publishing Company, Amsterdam, Netherlands, 1992.

[14] Maculan, N., P. Souza, and A.C. Vejar, "An Approach for the Steiner Problem in Directed Graphs," *Annals of Operations Research*, Vol.33(1991), pp.471-480.

[15] Minoux, M., "Efficient Greedy Heuristics for Steiner Tree problems Using Reoptimization and Supermodularity," *INFOR*, Vol.28(1990), pp.221-233.

[16] Prim, R.C., "Shortest Connection Networks and Some Generalisations," *Bell System Technical Journal*, Vol.36(1957), pp.1389-1401.

[17] Schiemangk, C., "Thermodynmically Motivated Simulation for Solving the Steiner Tree Problem and the Optimization of Inte-

racting Path Systems," *Optimization of Connection Structures in Graphs*, Iwainsky, A (ed.), CICIP, East Berlin, GDR, 1985, pp. 74-90.

[18] Singh, G., S. Das, S. Gosavi, and S. Pujar, *Ant Colony Algorithms for Steiner Trees: An Application to Routing in Sensor Networks*, In Recent Developments in Biologically Inspired Computing, L.N. de Castro and F.J. von Zuben, Eds. Idea Group Inc., Chapter 6. 2004

[19] Studetzle, Thomas and Hogler H. Hoos, "Max-Min Ant System," *Future Generation Computer Systems*, Vol.16(2000), pp.889-914.

[20] Voss, S., "Steiner's Problem in Graphs: Heuristic Methods," *Discrete Applied Mathematics*, Vol.40(1992), pp.45-72.

[21] Xu, J., S.Y. Chiu, and F. Glover, "Probabilistic Tabu Search for Telecommunications Network Design," *Combin Optim Theory Pract*, Vol.1(1997), pp.69-94.

[22] Xu, J., S.Y. Chiu, and F. Glover, "Using Tabu Search to Solve the Steiner Tree-Star Problem in Telecommunications Network Design," *Telecommun Syst*, Vol.6(1996), pp. 117-125.