

# Symmetric Traveling Salesman Problem을 해결하기 위해 Ant Colony System에서의 효과적인 최적화 방법에 관한 연구

정태웅\*, 이승관, 정태충  
경희대학교 전자계산공학과  
e-mail : jung@iislab.kyunghee.ac.kr

## An Effective Ant Colony System Optimization for Symmetric Traveling Salesman Problem

Tae-Ung Jung\*, Sung-Gwan Lee, Tae-Chung Jung  
Dept of Computer Engineering, Kyung-Hee University

### 요약

조합 최적화 문제인 Traveling Salesman problems(TSP)을 Genetic Algorithm(GA)[3]과 Local Search Heuristic Algorithm[8]을 이용하여 접근하는 것은 최적해를 구하기 위해 널리 알려진 방법이다. 본 논문에서는 TSP문제를 해결하기 위한 또 다른 접근법으로, 다수의 Ant들이 Tour들을 찾는 ACS(Ant Colony System) Algorithms[4][6][7]을 소개하고, ACS에서 Global Optima를 찾는 과정에서, 이미 이루어져 있는 Ant들의 Tour결과들을 서로 비교한다. Global Updating Rule에 의해 Global Best Tour에 속해 있는 각 Ant Tour의 edge들을 update하는 ACS Algorithm에, 각 루프마다 Ant Tour들을 우성과 열성 인자들로 구분하고, 각각의 우성과 열성 인자들에 대해서 Global Updating Rule에 기반한 가중치를 적용(Weight Updating Rule)하므로써 기존의 ACS Algorithm보다 효율적으로 최적 해를 찾아내는 방법에 대해서 논하고자 한다.

### 1. 서론

의미론상의 순회 판매원 문제(Traveling Salesman Problem, 이하 TSP)는 여러 개의 도시에 각 도시간의 거리가 주어지고, 그 도시간의 최단거리를 찾는 문제이다. 여기서 각각의 도시는 정확히 한번 방문을 해야 하며, 시작한 도시로 다시 돌아와야 한다. 여기서는 Symmetric TSP[5]문제만을 다루겠다.

#### <STSP 제약 조건>

1. Tour를 이루어야 한다.
2. 모든 도시를 꼭 한번 들러야 한다.
3. 도시 A, B에서  $Distance(A,B) = Distance(B,A)$

위와 같은 TSP문제는 널리 알려져 있는 NP-hard combination optimization 문제이다. 이런 문제를 풀기 위해서 측정되는 시간은 N승의 제곱으로 시간이 증가한다. 이 문제를 해결하기 위해 많은 휴리스틱 기술(Genetic Algorithm[3], Simulated Annealing, Tabu Search)이 제안되었다.

본 논문에서는 ACS Algorithm[6]에서 최적해를 구하는 방법에 대해 소개를 하고, Global Optima를 찾는 알고리즘 개선에 대해서 논하고자 한다.

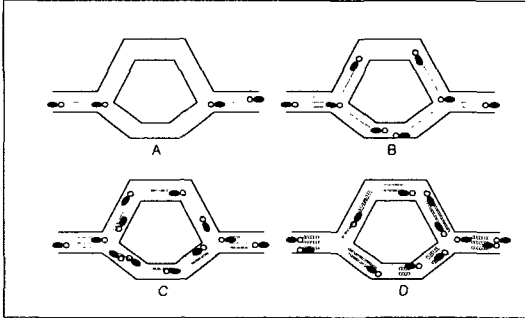
일반적으로 ACS Algorithm의 개선에 관한 방법들은 각 Ant들이 Local Optima를 선택하기 위해 edge들을 선택할 때, 보다 넓고 다양하게 검색하고, 효율적으로 해를 찾는 방법들을 제시하고 있다. 여기서도 마찬가지로 Ant들의 edge선택에 있어서 보다 좋은 유전 형질을 갖는 edge들을 선택할 수 있는 방법을 제시하겠다.

### 2. 기존의 ACS Algorithm

#### 2.1 Ant System

ACS는 기존의 Ant System[1][2]에서 출발한 Heuristic Search Algorithm이다. Ant System은 Ant들이 목적지를 향하여 지나가는 경로에 Ant들의 분비물인 Pheromone이 쌓이고, 이후에 지나가는 Ant들은 그 경로에 쌓여있는 Pheromone을 정보로 이용하여 다음 경로를 선택하는 원리를 Heuristic Search에 적용시킨 System이다[9]. [그림 1]을 살펴보면, 먼저 A과정에서 각 Ant들은 Left, 또는 Right를 선택해야 하는 decision point에 도달하게 된다. 그들은 다음 경로를 random하게 선택하고, 결국

평균적으로 Ant들의 절반정도는 Left를 나머지는 Right를 선택하게 된다. B, C과정은 계속해서 같은 속도로 Ant들이 움직이는 과정을 보여준다.



[그림 1] Real Ant System

여기서 각 Ant들이 선택하는 길은, 대략, Ant들이 지나간 경로에 쌓이게 되는 Pheromone의 양에 비례하게 된다. Lower Path는 Upper Path보다 짧기 때문에, 평균적으로 더 많은 Ant들이 Lower Path를 방문하게 된다. 결국 Pheromone의 양은 Upper Path보다 더 빠르게 쌓이게 된다. 그 후 시간이 흐르면, 이 Pheromone의 양은 새로운 Ant들이 경로를 선택할 시에 영향을 줄 정도로 각 경로에서 커다란 차이를 보이게 된다.(D과정) 이런 과정들이 지나면 Ant들은 Pheromone의 양을 비교해, Lower Path를 선택할 확률이 더 높아지게 된다. 그러나, 이 Ant System은 Ant들이 짧은 edges가 있으면 그것만을 선택하고자 하는 성질로 인하여 Local Optima에 빠질 확률이 높기 때문에, 다음과 같은 ACS(Ant Colony System Algorithm)가 새롭게 대두되게 된다 [4][7]. ACS Algorithm은 Ant들이 지나가는 경로에 Pheromone을 기탁하는 과정 외에, Local Updating Rule을 통해 Pheromone을 조금씩 상쇄시킴으로서 State Transition Rule을 통해 다음 경로를 선택할 때, 현재 지나온 경로가 선택될 확률을 더 적게 만든다. 반대로, edge가 Global Best Tour에 속해 있으면 그 Tour의 경로를 다시 update하는 과정이 추가되었다.

## 2.2 Ant Colony System

이 System은 State Transition Rule, Local Updating Rule, Global Updating Rule을 각 Ant의 Behavior에 적용시키므로써 가능한 한 Local Optima에 빠지지 않고, 탐색 범위를 보다 넓고 다양하게 검색하고 더욱 효율적으로 최적 해를 찾도록 구성되어 있다. 먼저 각 도시에 초기 Ant들이 random하게 location되고, 각 edges들의 초기 Pheromone이 setting된다. 그 후, 각 Ant들은 State Transition Rule을 이용해서 Pheromone의 양과 edge길이를 참조해서, 또는 Random으로(Local Optima에 빠지는 것을 방지) 경로를 선택하고, 선택할 때마다 Local Updating Rule을 적용시키므로써, 선택한 각 edge의 Pheromone의

Initialize

Loop

Each ant is positioned on a starting node

Loop

Each ant applies a state transition rule to incrementally build a solution and a local pheromone updating rule

Until all ants have built a complete solution

A global pheromone updating rule is applied

Until End condition

[그림 2] The ACS Algorithm.

양을 update시킨다.(decay) 이 과정이 도시의 수만큼 반복되면, 각 Ant들의 완성된 Tour들에 대해 Global Best Tour를 찾고, Best Tour에 속해 있는 edge들에 대해 Global Updating Rule을 적용시켜서 그들의 Pheromone을 update시킨다.

## 3. 제안한 방법(Weight Updating Rule)

기존의 ACS Algorithm의 경우, Local Search는 Local Updating Rule과 Global Updating Rule에 의해 다음 edge들의 Pheromone의 정보를 update하므로써 그것을 토대로 Ant들이 다음 edge로의 탐색을 하게 된다. Pheromone의 일반적인 Global updating 방법은 다음과 같이 적용된다.

$$\tau(r, s) \leftarrow (1 - \alpha) \times \tau(r, s) + \alpha \times \Delta \tau(r, s)$$

$$\text{where } \Delta \tau(r, s) = \begin{cases} (L_{gb})^{-1}, & \text{if } (r, s) \in \text{global\_best\_tour} \\ 0, & \text{otherwise} \end{cases}$$

$0 < \alpha < 1$  : pheromone decay parameter  
 $L_{gb}$  : 시작으로부터 globally best tour의 length

[그림 3] Global Updating Rule

$\alpha$ 는 Pheromone decay parameter이고,  $L_{gb}$ 는 현재까지의 Global Best Tour Length이다.  $\tau(r, s)$ 는 city r 과 city s 사이의 Pheromone 양이고,  $\Delta \tau(r, s)$ 는 edge r, s가 global\_best\_tour에 속해 있으면  $(1/L_{gb})$ , 아니면 0으로 주어지고, 결국 decay parameter  $\alpha$ 에 의해 Pheromone의 양은 줄어들게 된다. 본 논문에서 제안된 방법(Weight Updating Rule)은, 각 Ant들이 한 Cycle의 Tour를 완성한 후, 각 Ant들의 Tour들 중 Global Optima를 찾는 과정에서, 이미 이루어져 있는 각 Ant들의 Tour결과들을 서로 비교해서, Ant들의 전체 Tour가 이루어지는 각 루프마다, Ant Tour들을 우성과 열성 인자들로 구분한다. 그리고, 각각의 우성과 열성 인자들에 대해서 Global Updating Rule에 기반한 가중치를 부여하므로써, 이후 새로운 Ant들이 Tour를 구성할 때, 보다 나은 edge를 선택할 확률을 높게 만든다. [그림 5]는 우성, 열성 인자 선택과 가중치를 부여하는 과정이다. 이것은 Ant들의 Tour들 중 Global Best Tour가 있는지 조사하는 과정에

추가로 [그림 5]의 Procedure가 항상 수행되게 된다. 비록 'Ant Number×City Number' 만큼의 Loop가 더 수행되지만, Ant들이 더 나은 edge들을 선택할 확률을 높임으로서 더 빠른 시간 안에 최적의 해에 접근해 나갈 수가 있다.

```

Procedure Weight Updating Rule
begin
  For k:=0 to m do
    if (Tour(Ant[k]) is priori set) then
       $T(r_k, s_k) = (1 - \alpha) \times T(r_k, s_k) + \alpha \times (1 / \text{Tour}(\text{Ant}[k]))$ 
    end if
     $T(r_k, s_k) = (1 - \alpha) \times T(r_k, s_k) -$ 
       $\alpha \times (1 / \text{Tour}(\text{Ant}[\text{symmetrical}(k)]))$ 
  End-for;
  Routine Lbest Perform
end;
    
```

[그림 5] 우성, 열성 인자에 따른 가중치 Pseudo Code

[그림 5]의 Procedure Weight Updating Rule은 [그림 3]에 있는 Global Updating Rule에 기반한 것이다. 본 논문의 Weight Updating Rule에서는, 우성인자의 경우엔, 가중치로서 자신의 Tour Length의 역수를 적용시켜서, 우성이 높으면 높을수록 더 큰 가중치가 더해지도록 한다.

```

procedure Global Updating Rule
begin
  For k:=1 to m do
    Compute Lk /* Lk : Ant k에 의한
      Tour의 length */
  End-for;
  Begin /* Procedure Weight Updating Rule 수행 */
    Perform procedure Weight Updating Rule
    Compute Lbest
  end;
  /* Global Updating Rule 을 사용해서 Lbest에 속해
    있는 edge들을 update */
  For each edge (r,s)
     $T(r_k, s_k) = (1 - \alpha) T(r_k, s_k) + \alpha (L_{best})^{-1}$ 
  End-for;
end;
    
```

[그림 6] 전체 Global Updating 수행도 및 제안 방법

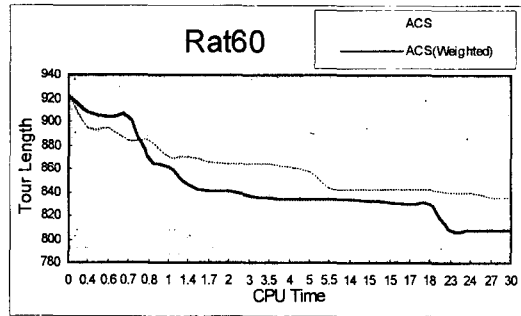
그리고, 열성인자의 경우엔, 서로 대칭되는 인자의 Tour Length의 역수를 적용시켜서 열성이 강하면 강할수록 더 큰 가중치로 상쇄되도록 하고 있다. 결국, 각 인자는, Pheromone을 updating하는 방법으로, 기존의 ACS Algorithm에서 적용하고 있는 Local Updating Rule과 Global Updating Rule 외에, 본 논문의 Weight Updating Rule을 적용시키므로써 자신의 우성, 열성 비율에 따라 정해지는 가중치를 추가로 updating하는 효과를 볼 수 있다. [그림 6]은 Global Updating 부분에서 Procedure Weight Updating Rule이 추가된 부분을 보여주고 있다.

#### 4. 실험 결과

본 논문에서 제안하고 있는 방법을 실험하기 위해 서 사용된 환경은 Windows NT(PentiumIII 550MHz 256Mb)이고, 프로그램은 C-language로 구현되었다. 도시들의 위치는 TSP 예제로 널리 알려진 TSPLIB에서 추출하여 실험을 하였다.

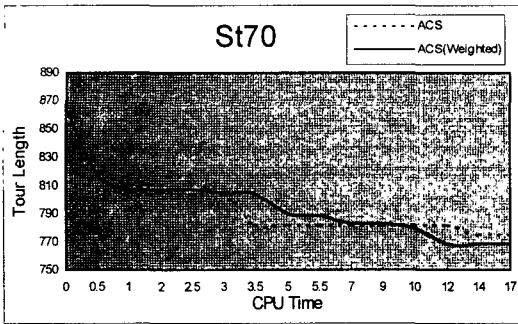
아래의 [그림 7], [그림 8], [그림 9]는 각각 Rat60, St70, Pr152의 도시들을 가지고, 기존의 ACS Algorithm과 본 논문에서 제안된 방법에 대해, 서로 비교한 결과를 그래프로 도식한 것이다. 여기서 접선은 기존의 ACS Algorithm의 기울기, 실선은 제안된 방법의 기울기, y축은 각 Ant들이 만들어 내는 Best Tour Length들의 변화, x축은 시간의 변화를 나타내고 있다.

실험에 사용된 Ant들의 숫자는 기존의 ACS 논문 [4][6]에서 계산되고 널리 사용되고 있는 10마리로 정하고 있고, 우성과 열성을 구분하는 기준으로는, 우성은 Tour들의 상위 20%를 열성은 나머지 80%를 채택했다. 이 값은 여러 경우의 시뮬레이션의 결과를 비교한 후, 그 중 좋은 결과 값을 보이는 경우를 선정한 것이다. 이 수치의 적합성은 차후의 연구에



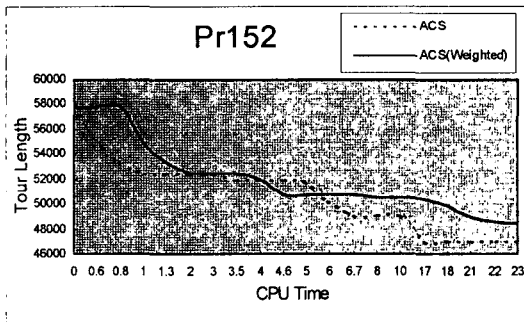
[그림 7] Comparison ACS with ACS(Weighted) for Rat60

서 수식으로서 연구를 하고, 각 도시들의 edge의 특성과 각 Ant들의 Tour Length를 적용하므로써 위 보다 더 좋은 결과를 보이는 수치를 구할 수 있어야겠다. [그림 7]과 [그림 8]의 경우는 제안된 방법이 기존의 ACS Algorithm보다 더 빠르게 수렴하는 모습을 보이고 있다. 이것은 제안된 방법에 의거해 그만큼의 가중치가 가감된 edge들의 정보를 참고로 해서 다음 edge들을 선택하기 때문이다. 결국 가중치가 부여된 우성의 edge들을 선택해 나갈 수 있는 확률은 전보다 더 높아지게 된다. 그러나, [그림 9]의 경우엔, 오히려 기존의 ACS Algorithm보다 좋지 않은 결과를 보이고 있다. 이것은 많은 도시들을 탐색할 때는, 그 만큼 도시 edge들의 특성이 강화되므로, 결국 이 특성을 적용시키지 않은 고정된 우성, 열성 기준 설정과 가중치 부여가 원인으로 추정된다. 시뮬레이션 결과, 얻은 결론은 도시의 수가 적은



[그림 8] Comparison ACS with ACS(Weighted) for St70

경우[그림 7],[그림 8]엔 모두 기존의 ACS Algorithm 보다 더 좋은 결과를 보이고 있지만, 도시의 수가 많아지면 많아질수록 항상 좋지는 않다는 것이다. 이것은 제안된 방법이 도시들의 edge의 특성에 대한 고려가 부족하다는 것과, 그에 기인하여 Ant들의 개수와 우성과 열성의 구분 기준이 고정되어 있다는 것이다. 그리고, 각 우성과 열성에 대해 부여하는 가중치는 Global Updating Rule에 기반해서 각 Tour Length를 적용시키지만, 이것 또한 각 edge들의 특성을 부여하는 것에 그다지 유연하지 못하다는 것이다. 결국, 도시의 개수가 적은 경우엔 영향이 적지만, 개수가 많은 경우엔 각 도시 edge의 특성에 따



[그림 9] Comparison ACS with ACS(Weighted) for Pr152

라 좋은 결과를 보이는 경우도 있고, 그렇지 않은 경우[그림 9]도 있었다.

## 5. 결론과 앞으로의 연구 방향

본 논문은, 기존의 ACS Algorithm의 Global Updating 부분에, 모든 경로를 방문한 각 Ant들의 Tour값을 비교해 Global Updating Rule에 기반한 Weight Updating Rule을 적용시키므로써, 각 Ant들이 가능한 한 우성의 성질을 갖는 경로를 선택할 수 있는 방법을 제시하고 있다. 결국, 기존의 ACS

Algorithm과 본 논문에서 제안한 방식을 같은 환경에서 실험을 했을 때, 도시의 갯수가 적은 경우, 속도 면에서 기존 보다 더 빠르게 수렴하게 되는 결과가 나왔다. 그러나, 도시의 갯수가 많아지는 경우엔, 꼭 그렇지만은 않은 결과를 보이고 있었다. 이것은 도시의 갯수가 많아지면 많아질 수록 도시 edge들의 특성이 더 강화되므로, 결국 경로의 특성을 파악하지 않은 상태에서 정적으로 미리 정해놓은 Ant들의 개수와 우성, 열성 구분 기준, 좀더 유연하지 못한 가중치 부여 방법이 원인으로 추정된다.

따라서, 도시 edge들의 특성을 연구하고 수학적으로 정의해서, 그것에 근거한 Ant들의 개수와 우성, 열성 구분 기준의 설정, 가중치 부여 방법의 개선에 관한 연구가 이루어진다면 현재보다 더 좋은 결과를 보일 수 있을 것이다.

## [Reference]

- [1]. A. Colomi, M. Dorigo, and V. Maniezzo, "An investigation of some properties of an ant algorithm," pp. 509-520.
- [2]. A.Colomi, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," pp. 134-144.
- [3]. B. Freisleben and P. Merz, "Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems," pp. 616-621
- [4]. L. M. Gambardella, E. Taillard, and M. Dorigo, "Ant Colonies for QAP," IDSIA, Lugano, Switzerland, Tech. Rep. IDSIA 97-4, 1997.
- [5]. L.M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," Proceedings of IEEE International Conference of Evolutionary Computation, Press,1996, pp. 622-627.
- [6] Marco Dorigo, and Luca Maria Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem.
- [7]. Marco Drigo, V.Maniezzo, and A.Colomi, "The ant system: optimization by a colony of cooperation agents," IEEE Transactions of Systems, Man, and Cybernetics-Part B, vol. 26, No. 2, pp. 29-41, 1996.
- [8]. S. Lin and B.W. Kernighan, "An effective Heuristic algorithm for the traveling salesman problem," Operations Research, vol.21, pp. 498-516, 1973.
- [9]. T. Stutzle and H. Hoos, "The ant system and local search for the traveling salesman problem,"