방문판매원 문제에 적용한 개선된 개미 군락 시스템

김 인 겸† · 윤 민 영†

요 약

개미 군락 시스템 (Ant Colony System, ACS)은 조합 최적화 문제 중의 하나인 방문 판매원 문제에(Traveling Salesman Problem, TSP) 간단하게 응용할 수 있고 좋은 결과를 보여주었다. 그러나 ACS는 작은 TSP에서는 최적해를 구하였지만, 큰 TSP에 대해서는 최적해를 구하지 못하였다. 본 연구에서는 큰 TSP에 대한 최적해를 찾기 위하여 개미들에게 좀 더 많은 정보를 주는 기법의 하나로 한 도시에서 다음 도시로의 탐색 경로를 결정할 때, 인접한 도시들에(adjacent neighbor) 대한 거리로만 결정하지 않고 길이가 w인 부경로(subpath)에 대한 정보를 미리 수집한 후, 이 정보를 이용하여 경로를 생성하였다. 실험 결과 본 연구에서 제안한 기법이 기존의 ACS 기법보다 큰 그래프에서 최적해(Known Optimal)에 가까운 경로를 찾는 것을 볼 수 있다. 그래프에 따라서는 기존의 ACS 기법보다 최대 70% 이상의 성능이 개선되었으며 평균적으로 30%내외의 개선된 결과를 보여준다.

키워드 : 개미 군락 시스템, 조합 최적화, 방문 판매원 문제

Improved Ant Colony System for the Traveling Salesman Problem

Inkyeom Kim† · Minyoung Yun†

ABSTRACT

Ant Colony System (ACS) applied to the traveling salesman problem (TSP) has demonstrated a good performance on the small TSP. However, in case of the large TSP, ACS does not yield the optimum solution. In order to overcome the drawback of the ACS for the large TSP, the present study employs the idea of subpath to give more information to ants by computing the distance of subpath with length w. In dealing with the large TSP, the experimental results indicate that the proposed algorithm gives the solution much closer to the optimal solution than does the original ACS. In comparison with the original ACS, the present algorithm has substantially improved the performance. By utilizing the proposed algorithm, the solution performance has been enhanced up to 70% for some graphs and around at 30% for averaging over all graphs.

Key Words : Ant Colony System, Combinatorial Optimization, TSP

1. 서 론

메타 휴리스틱 탐색 방법 중의 하나인 개미 시스템(Ant System, AS)은 개미 집단이 먹이를 발견한 후 목적지까지 가장 빠른 경로를 찾아내어 운반하는 집단행동을 모방한 탐색 알고리즘이다. 개미들이 목적지까지 먹이를 운반할 때 그 경로를 따라서 페로몬(pheromone)을 남기게 되고 개미들은 페로몬의 흔적이 강한 곳을 선택하게 되므로 결국은 대부분의 개미들이 가장 빠른 경로를 찾게 된다는 논리이다. Dorigo[1] 등에 의해서 제안된 AS는 방문 판매원 문제(Traveling Salesman Problem, TSP)[2, 3], 라우팅 문제(Routing problem)[4], 배정 문제(Assignment problem), 스케줄링 문제(Scheduling problem) 등과 같은 조합 최적화 문제(Combinatorial Optimization Problem)에 응용되어 왔다. 본 연구의 기초가 되는 개미 군락 시스템(Ant Colony System, ACS)은 기존의 개미 시스템의 효율성을 개선하여 TSP에 응용되었다. 에이전트의 역할을 하는 개미들은 병렬 탐색을 수행하여 TSP를 해결하기 위한 좋은 해를 찾으며, 이때 개미들은 페로몬을 이용하여 서로 간접적이지만 전역적인 정보 교환을 할 수 있게 된다[5]. 각 개미들은 욕심쟁이 휴리스틱(Greedy heuristic)과 과거의 경험에 의한 정보를 서로 혼합하여 다음 탐색 도시를 선택하는 과정을 반복함으로써 TSP 해를 위한 경로를 구성하게 된다. TSP는 NP-hard 문제로써 최적의 해를 효율적으로 구하기 위하여 여러 종류의 메타 휴리스틱(Meta-heuristic) 탐색 방법들이 적용되었다.

방문 판매원 문제(TSP)는 완전 가중 그래프(complete weighted graph) $G=(V, E)$로 표현할 수 있다. V는 정점들의 집합으로 $|V|=n$이며 판매원이 방문해야 할 모든 도시들을 나타내고, E는 연결선들의 집합이다. 각 연결선

† 정 회 원 : 성결대학교 정보통신공학부 부교수
논문접수 : 2005년 3월 14일, 심사완료 : 2005년 10월 13일

$(i,j)\in E$에는 가중치 d_{ij}가 배정되어 있으며 이 가중치는 도시 i와 $j(i,j\in V)$ 사이의 거리를 나타낸다. 그러므로 TSP는 주어진 완전 가중 그래프에서 각 정점을 한 번씩 방문하여 다시 출발 정점으로 돌아오는 최단 경로를 찾는 해밀토니언 순회(Hamiltonian Circuit) 문제로 변환된다. TSP는 대칭(Symmetric) TSP와 비대칭(Asymmetric) TSP로 나눌 수 있는데, 비대칭 TSP는 한 쌍의 정점 (i,j) 사이의 거리인 d_{ij}가 순회 방향에 따라서 다를 수 있다. 즉 $d_{ij}\neq d_{ji}$를 만족하는 연결선이 최소한 하나라도 존재한다. 대칭 TSP에서는 E에 속한 모든 연결선에 대하여 $d_{ij}=d_{ji}$를 만족한다.

ACS는 작은 TSP에서는 최적해를 구하였지만, 큰 TSP에 대해서는 최적해를 구하지 못하였다. 본 연구에서는 큰 TSP에 대한 최적해를 찾기 위하여 개미들에게 좀 더 많은 정보를 주는 기법의 하나로 한 도시에서 다음 도시로의 탐색 경로를 결정할 때, 인접한 도시들에(adjacent neighbor) 대한 거리로만 결정하지 않고 길이가 w인 부경로(subpath)에 대한 정보를 미리 수집한 후, 이 정보를 이용하여 경로를 생성하였다. 실험 결과 본 연구에서 제안한 기법이 기존의 ACS 기법보다 큰 그래프에서 최적해(Known Optimal)에 가까운 경로를 찾는 것을 볼 수 있다. 그래프에 따라서는 기존의 ACS 기법보다 약 70% 가량의 성능을 개선할 수 있음을 보여주고 평균적으로 약 30% 정도의 성능 개선 효과가 있음을 알 수 있다. 본 연구는 대칭 TSP를 대상으로 하였으며 다음과 같이 구성되었다. 제2장은 기본 아이디어가 되는 ACS 알고리즘에 대하여 설명하고 3장에서는 기존의 ACS가 큰 그래프에서 최적해를 찾지 못하는 것을 개선하기 위하여 부경로에 대한 정보를 이용한 새로운 기법을 소개한다. 제4장은 기존의 ACS와 제안한 ACS를 실험하여 결과를 비교 분석하며 마지막으로 5장에서는 본 연구 결과를 요약하고 연구 방향을 제시한다.

2. 개미 군락 최적화(Ant Colony Optimization, ACO) 알고리즘

ACO 알고리즘은 TSP에 쉽게 적용할 수 있다[6]. 페로몬 경로(trail)는 연결선들로 구성되고 τ_{ij}는 연결선 (i,j)에 누적된 페로몬의 양을 나타내며, 도시 i에서 직접 도시 j로 이동할 수 있는 가능성의 정도를 나타낸다. 휴리스틱 정보는 $\eta_{ij}=1/d_{ij}$로 표현한다. τ_{ij}값들과 η_{ij}값들은 각각 페로몬 행렬과 휴리스틱 정보 행렬에 저장된다. 각 개미들은 무작위로 정해진 출발 도시에서 탐색을 시작하여 아직 방문하지 않은 도시들 중에서 τ_{ij}와 η_{ij}에 의하여 다음 방문할 도시를 선택하고 모든 도시들의 방문이 완료되면 다시 출발 도시로 돌아가서 하나의 경로를 완성하게 된다. 탐색이 끝나면 모든 개미들은 자신들이 구성한 경로에 대한 페로몬 양을 갱신하게 된다[7,8]. 개미 시스템(AS)은 맨 처음 개발된 ACO 알고리즘으로써 TSP 문제에 적용하기는 아주 간단하지만, 단순한 페로몬 갱신 방법으로 지역 최적화(local optima)에 빠지는 경향이 있어서 작은 문제들에서만 효율적인 해답을 줄 수 있었다. AS는 페로몬을 갱신하고(update) 페로몬 경로를 운용하는 방법에 의하여 elitist AS, rank-based AS와 MAX-MIN AS로 개선되었는데 AS보다는 TSP를 해결하는데 더 좋은 성능을 보여주었다[4]. 본 연구의 기초가 된 개미 군락 시스템(ACS)은 AS의 기본 아이디어를 사용한 ACO 알고리즘으로 AS의 문제점을 개선하여 효율성을 높였다. ACS는 NP-hard 조합 최적화 문제에 다양하게 응용되어 왔고, 또한 성능을 입증하였다.

```
algorithm: ACS_for_TSP {

Initialize Data;
while (not terminate) {
place m ants at m cities;
repeat (for each ant)
        apply tour construction rule to build a trail;
        apply local pheromone updating rule;
until (construct a solution)
apply local search;
apply global pheromone updating rule;
}
}
```

(그림 1) 알고리즘: ACS_for_TSP

그래프 $G=(V,E)$에서 $|V|=n$이라 가정할 때, ACS의 초기 단계에서는 m개의 개미들이 $(m\leq n)$ 무작위로 m개의 도시에 위치한다. 개미들은 경로 구성 규칙(Tour Construction Rule)에 의하여 다음 방문 도시를 반복적으로 선택하여 경로를 구성하게 된다. 이 과정에서 하나의 연결선이 경로에 추가될 때마다 지역 페로몬 갱신 규칙에(Local Pheromone Updating Rule)의하여 페로몬의 양을 수정하고 경로가 완성되면 지역 탐색(Local Search)을 적용시켜서 개미들에 의해서 구성된 경로들의 성능을 개선시킨다. 그 후, 현재까지 발견된 경로들 중에서 최소 경로인 전역 최적 경로에 대해서만 다시 페로몬을 갱신시키게 된다. 이 과정을 (그림 1)에 나타내었다.

2.1 경로 구성 규칙(Tour Construction Rule)

개미 k가 도시 i에서 다음 도시 j를 선택할 때에는 수식 (2.1)의 *pseudo-random proportional* 규칙에 따른다.

$$j=\begin{cases}\arg\max_{l\in N_i^k}\{\tau_{il}[\eta_{il}]^{\beta}\}, & \text{if } q\leq q_0;\\ J, & \text{otherwise;}\end{cases} \quad (2.1)$$

여기서 β는 연결선 (i,j)에 대한 페로몬의 양 τ_{ij}와 휴리스틱 정보 η_{ij}에 대한 상대적 중요도를 결정하는 변수이고 N_i^k는 개미 k가 방문할 수 있는 도시들 중에서 아직 방문하지 않은 도시들의 집합을 나타낸다. q는 [0, 1] 사이의 균일 분포를 갖는 무작위 변수이며 q_0는 $0\leq q_0\leq 1$ 을 만

족하는 변수이다. J는 다음 수식 (2.2)에서 계산된 확률 분포에 의하여 선택되는 무작위 변수이다.

$$p_{ij}^{k}=\frac{[\tau_{ij}][\eta_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{il}][\eta_{il}]^{\beta}},\quad \text{if } j\in N_i^k \tag{2.2}$$

수식 (2.2)에 의하면 연결선 (i,j)가 선택될 확률은 페로몬의 양 τ_{ij}와 휴리스틱 정보 η_{ij}의 값에 의해서 결정된다. 즉 각 개미는 페로몬의 양이 많으면서 거리가 짧은 연결선으로 연결된 도시 j를 다음 탐색 도시로 결정하게 된다. 변수 $\beta=0$이면 페로몬 정보에 의존해서만 다음 방문 도시가 결정되므로 일반적으로 $\beta>1$이며,[4]에 의하면 $2\leq\beta\leq5$일 때 좋은 성능을 보여주었다.

2.2 지역 페로몬 갱신 방법(Local Pheromone Trail Update)

ACS와 AS의 다른 점은 ACS는 지역 페로몬 갱신 방법을 사용한다는 것이다. 개미가 TSP의 경로를 구성하면서 연결선 (i,j)를 선택할 때마다 수식 (2.3)을 적용하여 해당 연결선의 페로몬 양을 갱신한다.

$$\tau_{ij}=(1-\xi)\tau_{ij}+\xi\tau_0 \tag{2.3}$$

변수 ξ는 $0<\xi<1$을 만족하며 실험적으로 $\xi=0.1$일 때 성능이 가장 좋은 것으로 나타났다[4]. τ_0은 초기 페로몬의 양을 나타내며 $\tau_0=1/(nC^{nn})$일 때 성능이 가장 좋다. 여기서 n은 TSP에서 도시의 개수를 나타내며 C^{nn}은 nearest neighbor 휴리스틱에 의해서 생성된 경로의 길이이다. 즉, 각 연결선의 페로몬 양은 욕심쟁이 기법(Greedy method)에 의해서 구성된 경로의 길이로 초기화 된다. 수식 (2.3)을 적용하면 개미가 연결선 (i,j)를 선택할 때마다 그 연결선에 해당된 페로몬의 양 τ_{ij}가 감소하게 되어서 다른 개미들이 그 연결선을 선택할 확률을 낮추어 준다. 이렇게 함으로써 개미들이 아직 선택되지 않은 다른 연결선들을 선택할 수 있는 확률을 높여서 모든 개미들이 같은 경로를 선택하는 현상인 stagnation을 막아준다. 본 연구에서는 대칭 TSP만을 고려하므로 $\tau_{ij}=\tau_{ji}$가 된다.

2.3 지역 탐색 방법(Local Search)

모든 개미들이 각각의 경로를 찾은 후, 그 경로를 구성하는 2개나 3개의 연결선을 교환하는 경로 개선 방법인 2-opt나 3-opt와 같은 기법을 적용하여 지역적으로 최적화된 해로 만들 수 있다. 이렇게 TSP의 경로 구성에 지역 탐색 기법을 조합하여 적용시키면 개미들에 의해서 구성된 경로의 성능을 크게 개선할 수 있다[10]. 본 연구에서는 3-opt 기법을 적용하여 수행하였다.

2.4 전역 페로몬 갱신 방법(Global Pheromone Trail Update)

전역 페로몬 갱신은 모든 개미들이 각각의 경로를 찾은 후, 지금까지 발견된 경로들 중에서 최적 경로 T^{bs}에 대하여만 다음 수식 (2.4)에 의하여 페로몬을 갱신하는 방법이다.

$$\tau_{ij}=(1-\rho)\tau_{ij}+\rho\triangle\tau_{ij}^{bs},\quad \forall(i,j)\in T^{bs} \tag{2.4}$$

$\triangle\tau_{ij}^{bs}$는 연결선 (i,j)가 최적 경로에 속하였을 경우 (i,j)에 더해지는 페로몬의 양이다. C^{bs}가 전역 최적 경로에 대한 길이를 나타낸다고 할 때, $\triangle\tau_{ij}^{bs}=1/C^{bs}$이다. ρ는 페로몬의 증발율(evaporation rates)로써 연결선에 저장되었던 페로몬은 ρ에 의해서 감소된다. 일반적으로 $\rho=0.1$일 때 가장 좋은 성능을 보여준다. 전역 페로몬 갱신이 지역 페로몬 갱신과 동일한 형태로 이루어짐에도 불구하고 전역 최적 경로를 이용하여 생성된 $\triangle\tau_{ij}^{bs}$의 값은 항상 이전의 τ_{ij}보다 큰 값을 나타내게 되는데 이것은 τ_{ij}가 τ_0로부터 시작하기 때문이다. ACS는 지역 페로몬 갱신에 의하여 stagnation에 빠지는 것을 방지하는 한편 전역 페로몬 갱신에 의하여, 최적의 경로로 탐색된 경로는 최적의 결과를 줄 수 있는 경로라는 것을 보장하고 있다.

3. 제안한 기법

ACS는 전역 페로몬 갱신 방법과 지역 페로몬 갱신 방법을 함께 사용한다. 전역 페로몬 갱신은 탐색을 시작하는 다른 개미에게 지금까지 발견된 경로들 중에서 최선의 경로에 대한 정보를 주기 위해서 사용되고, 지역 페로몬 갱신은 현재 방문한 연결선들에 대하여 페로몬의 양을 감소시켜 줌으로써 다른 개미들이 지역 최적화에 빠지는 것을 방지하고 탐색 범위를 넓혀 준다. 정점의 수가 많은 복잡한 그래프에서는 휴리스틱 기법만으로는 최적의 경로를 찾기가 힘들기 때문이다. ACS는 m개의 개미들을 무작위로 m개의 도시들에 배치하여 최적 경로를 탐색하게 한다. 각 개미들이 지역 페로몬 갱신 방법을 적용하며 탐색을 진행하는 동안 현재 방문한 도시가 최적 경로에 포함될 가능성이 높으면 그 도시를 기준으로 경로를 구성하는 것이 최적 경로를 발견할 확률을 높이게 된다. 본 논문에서는 한 도시에서 다음 도시로의 탐색 경로를 결정하는 경로 구성 규칙에서, 수식 (2.1)에서 사용한 η_{ij}의 값을 바로 인접한 도시들에(adjacent neighbor) 대한 거리로만 결정하지 않고 개미들에게 좀 더 많은 정보를 주기 위해서 길이가 w인 부경로(subpath) s_w를, $1\leq w\leq n$, 이용하여 경로를 생성하는 실험을 수행하였다. 여기서 길이 w는 부경로에 포함되는 연결선의 개수를 나타낸다. 부경로 길이 s_w에 대한 정보를 준다는 것은 현재 개미가 위치해 있는 도시 i로부터 연결선 w개 $(i,j)(j,k)\ldots(x,t)$로 구성할 수 있는 모든 부경로에 대한 거리를 미리 계산하여 그 중에서 가장 최소값을 주는 부경로에 소속되어 있는 인접도시를 다음 방문 도시로 선택하는 방법이다. 즉, 도시 i에서 다음 도시 j를 선택할 때에는 s_w의 값을 최소화 시킬 수 있는 도시를 선정하여 경로를 구성한다. 이 방법을 이용하여 수식 (2.1)을 다음과 같이 수정하였다.

$$j=\begin{cases}\arg\max_{l\in N_i^k}\{\tau_{il}[\eta_{il}^{sw}]^{\beta}\}, & \text{if } q\leq q_0;\\ J, & \textit{otherwise;}\end{cases} \tag{3.1}$$

이 기법을 수행하기 위하여 인접 이웃 리스트(nearest neighbor list)를 먼저 구한 후, 그 리스트에서 이웃하는 도시들 간의 최소 거리를 계산해내면 된다. 만일 부경로 길이 $w=3$이라고 가정한다면, 인접 이웃 리스트 l_i는 도시 i에 인접한 모든 도시 j에 대하여, $i \neq j$, 부경로 $(i,j)(j,k)(k,t)$에 대한 거리 $d_{ij}+d_{jk}+d_{kt}$를 계산한 값을 저장하여 그 값을 오름차순으로 정렬하여 저장하게 된다. 이 단계는 (그림 2)에서 보는 것처럼 preprocessing으로 처리하여 실제 알고리즘의 실행 시간에는 크게 영향을 미치지 않는다.

지역 페로몬 갱신 방법인 수식 (2.3)에 의하면 초기 페로몬 값인 τ_0는 경로 탐색 과정에 계속해서 영향을 미치게 되므로 매우 중요하다. 기존의 ACS에서 τ_0는 현재의 도시에서 가장 가까운 거리에 위치한 도시를 탐색 경로에 포함시키는 욕심쟁이 기법(Greedy method)에 의해서 생성된 경로의 길이 C^{nn}을 이용하여 $\tau_0 = 1/(nC^{nn})$로 초기화 된다. 본 연구에서 제안한 알고리즘은 τ_0 역시 s_w값을 이용하여 찾은 경로 C^{sw}를 이용하여 다음과 같이 초기화 된다.

$$\tau_0 = 1/(nC^{sw}) \tag{3.2}$$

C^{sw}는 C^{nn}보다는 작은 값을 가지므로 이 방법을 이용하면 초기 페로몬 양이 증가된 상태로 모든 경로가 초기화 된다. 또한 부경로 C^{sw}를 이용한 τ_0와 수식 (3.1)을 통하여 인접 이웃을 더욱 정확하게 검색할 수 있게 된다.

도시의 개수가 아주 많은 TSP의 경우에는 기존의 ACS 기법으로는 최적해를 찾는 것이 어렵다. 최적해를 찾기 위해서는 초기 탐색과정에서 올바른 최적의 경로를 선택함으로써 최단 경로를 찾거나 아니면, 수많은 반복 검색 과정을 거쳐서 점차적으로 생성되는 경로를 개선해 나가면서 최적해를 찾을 수 있다. 그러나 아주 큰 그래프에서는 모든 가능한 경로들을 반복 탐색 과정을 거쳐서 생성하여 최적 경로를 찾는 것은 불가능하므로, 초기 탐색 과정에서 최적 경로를 구성할 수 있는 가능성이 높은 도시들을 경로에 포함시키도록 한다. 본 연구에서는 페로몬의 초기값 τ_0를 크게 해 줌으로써 기존의 ACS 기법보다 최적해에 가까운 경로를 생성할 수 있다. 제안한 알고리즘은 부경로에 대한 정보 s_w를 이용하여 최근접 이웃을 결정함으로써 초기 경로 구성 과정에서 올바른 경로를 선택할 수 있는 확률을 높여주게 되는 것이다. 부경로의 길이 즉, w의 값이 너무 큰 경우 지역 최소화(local minima)에 빠질 수 있으며 너무 적은 경우에는 기존의 ACS와 거의 차이가 없는 경로를 선택하게 되므로 효율성이 떨어지게 된다. 그러므로 적절한 s_w의 선택이 매우 중요하다. (그림 2)는 본 논문에서 제안한 알고리즘을 보여준다. 밑줄로 표시된 부분이 기존의 ACS에서 개선된 부분이다.

```
algorithm: ACS_for_TSP {

preprocessing steps:
        construct a distance matrix;
        construct a nearest neighbor list by s_w;
Initialize Data;
while (not terminate) {
        compute τ_0 with s_w: τ_0 = 1/(nC^sw);
place m ants at m cities;
repeat (for each ant)
          apply tour construction rule to build a trail;
          apply local pheromone updating rule;
until (construct a solution)
apply 3-opt local search;
apply global pheromone updating rule;
}
}
```

(그림 2) 제안한 알고리즘

4. 실험 및 결과 분석

본 연구에서 제안한 기법을 Dorigo의 aco-code[11]에 적용하여 수행하였다. 프로그램 수행에 사용한 그래프들은 TSPLIB[12] 라이브러리에서 제공하는 그래프들로써 도시 개수 1000개 미만인 경우와 1000개 이상인 경우를 구분하여 실험하였고, 선택된 그래프들은 이미 최적 경로 값(Known Optimal)이 알려져 있어서 제안한 ACS 알고리즘의 효율성 분석이 용이한 것들이다. 실험 환경은 Enterprise RedHat 2.1(PentiumIV 1.7GHz, 768MB)을 사용하였으며 프로그램 수행을 위한 변수들은 지금까지의 연구 결과에서 최적 경로를 생성하는 것으로 알려진 값들로 다음과 같이 설정하였다 [7]: $\xi=0.1$, $\rho=0.1$, $\beta=2$, $q_0=0.9$, $m=10$. 페로몬의 초기값은 수식 (3.2)의 $\tau_0 = 1/(nC^{s3})$로 계산되었고 부경로에 대한 정보를 위하여 s_3를 이용하였다. 각 개미들에게 한 번의 탐색을 위해서 CPU time 100초씩을 할당하였고 10번씩 경로 탐색을 하도록 하였다. 이 프로그램을 각 그래프 별로 또한 10번씩 수행하여 최적의 값과 평균값을 산출하였다.

<표 1>은 기존의 Dorigo가 제안한 ACS와 부경로 길이 3에 대한 정보 s_3를 이용하여 다음 탐색 도시를 선택했을 경우의 실험한 비교 결과를 도시 개수가 1000개 미만인 경우에 대하여 보여준다. 표에서 Instance는 TSPLIB에서 추출한 그래프 이름을 나타내고 Known Optimal은 그 그래프에 대한 TSP의 최적 경로 길이를 말한다. Original ACS의 Best와 Average는 Dorigo가 제안한 방법인[9]를 수행하여 얻어진 최적 경로의 길이와 경로의 평균 길이를 나타내며 Proposed ACS Best와 Average는 본 연구에서 제안한 방법으로 생성된 최적 경로의 길이와 평균 길이를 나타낸다. 또한 맨 오른쪽 열의 NNChangeRate은 기존의 ACS 방법을 이용하였을 때 다음 탐색 도시로 선택하였던 도시들이 본 연구에서 제안한 기법인 s_3를 이용했을 때 바뀌는 비율을 보여준다. 실험 결과에서 보는 것과 같이 도시 개수가 적은 그래프의 경우에는 기존의 ACS와 제안한 ACS가 모두 Known Optimal 해를 실험 시간인 CPU time 100초 안에 찾음을 알 수 있다.

〈표 1〉 도시 개수 1000개 미만에 대한 실험결과

Instance	Known Optimal	Original ACS		Proposed ACS		NNChange-Rate(%)
		Best	Average	Best	Average	
att 532	27686	27686	27704.28	27686	27705.88	36.47
d 198	15780	15780	15780.19	15780	15780.1	23.23
lin 318	42029	42029	42086.48	42029	42087.58	19.18
pcb 442	50778	50778	50835.83	50778	50831.57	13.57
rat 783	8806	8806	8819.88	8806	8821.01	24.65
d 1291	50801	50801	50874.87	50801	50863.21	7.20

그래프 att 532에서는 s_3를 이용했을 경우 다음 탐색 도시가 기존의 ACS와 비교하여 30%이상 바뀌었음을 볼 수 있다. 이런 상황은 그래프가 복잡하여 연결선이 많을 경우에 나타날 수 있는데, 연결선이 많고 복잡하면 생성할 수 있는 부경로들의 개수도 많아져서 다음 탐색 도시가 바뀔 가능성이 커지기 때문이다. 그래프 d 1291는 도시의 개수가 1000개 이상으로 구성되어 있지만 도시들 사이의 연결성이 비교적 단순한 형태를 보여주어서 NNChangeRate이 약 7% 임을 보면 기존의 알고리즘과 비교하여 다음 탐색 도시가 변화할 확률이 적어짐을 알 수 있다.

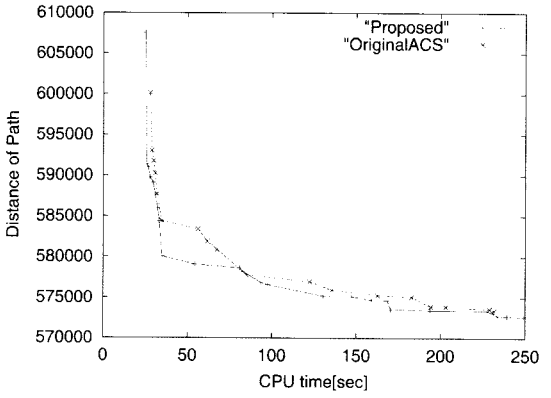
그러나 도시의 개수가 1000개 이상인 아주 큰 그래프에서는 기존의 ACS 알고리즘으로는 알려진 최적 경로를 찾을 수 있는 경우가 드물다. 표 2에서 볼 수 있는 것과 같이 아주 큰 그래프에서는 제안한 알고리즘이 기존의 알고리즘보다 known optimal 값에 가까운 해를 찾는 것을 알 수 있다. 제일 마지막 열에 계산된 Improved Rate은 기존의 ACS와 비교하여 제안한 알고리즘이 탐색한 경로가 개선된 비율을 100-{(③-①)/(②-①)*100}로 계산하여 나타내었다. 그래프 rl1889 같은 경우는 기존의 ACS 알고리즘보다 단지 0.5% 개선된 것으로 나타났으나 그래프 rl5915는 72.7%의 개선율을 보였다. 대부분의 큰 그래프에서 30% 이상의 개선율을 보임으로써 제안한 알고리즘이 기존의 알고리즘보다 훨씬 좋은 성능을 보이는 것으로 나타났다.

본 연구에서 제안한 알고리즘의 성능이 전체적으로 좋은 결과를 나타내면서도 표 2와 같이 각 그래프별로 성능의 차이를 보이고 있는 것은 각 그래프의 특성 때문이라고 생각할 수 있다. 기존의 ACS는 근본적으로 욕심쟁이 휴리스틱에 기초하여 인접 이웃을 검색하는 알고리즘이므로 인접 이웃과의 거리에 따라서 최적의 경로를 탐색하는 과정에 많은 영향을 받는 것을 알 수 있다. 그래프 u1432의 경우 인접 이웃 리스트의 변화를 알려주는 NNChangeRate가 1.47%로써 기존의 ACS와 거의 차이가 없으나 성능 면에서는 31% 개선된 것을 볼 수 있다. 이것은 다음 탐색 도시를 몇 개만 바꿔주어도 전체적인 성능을 개선할 수 있음을 보여준다. 그래프 fnl4461는 반대의 경우로서 인접 리스트의 데이터들이 30% 가까이 바뀌었음에도 불구하고 성능 향상이 3.4% 이루어 졌음을 볼 수 있다. 부경로에 대한 정보 s_3에 따라서 인접 이웃 리스트의 데이터들이 많이 바뀌는 그래프는 그렇지 않은 그래프보다 일정지역 혹은 전체적으로 매우 복잡한 형태의 그래프라고 생각할 수 있으며 따라서 이러한 그래프의 경우에는 TSP를 해결하는 다른 메타 휴리스틱 알고리즘들이 비슷한 결과를 낼 수밖에 없다고 생각할 수 있다.

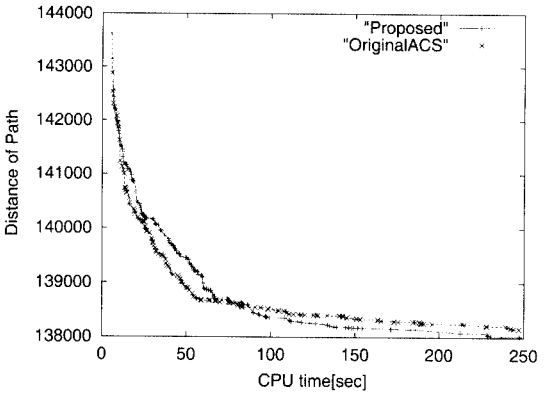
〈표 2〉 도시 개수 1000개 이상의 그래프에 대한 실험 결과

Instance	Known Optimal ①	Original ACS		Proposed ACS		NN Change Rate (%)	Improved Rate (%)
		Best ②	Average	Best ③	Average		
d 1655	62128	62153	62357.89	62147	62352.75	11.0	34.0
fnl 4461	182566	186492	186986.05	186361	187032.61	29.75	3.4
pcb 3038	137694	139098	139749.38	138933	139661.64	26.37	21.8
rl 1889	316536	317349	319232.81	317345	318849.68	20.12	0.5
rl 5915	565530	576654	581050.39	575837	581286.21	21.05	72.7
u 1432	152970	153204	153579.93	153131	153612.97	1.47	31.2
vm 1748	336556	336765	337531.19	336679	337641.23	25.69	41.2
pr 2392	378032	378838	380344.11	378654	380418.47	20.03	32.8

(그림 3)은 rl5915와 pcb3038에 대한 실험결과를 그래프로 나타낸 것으로써, CPU의 처리 시간에 따라서 발견되는 경로의 길이를 제안한 알고리즘과 원래의 ACS 알고리즘을 비교하여 보여준다. (그림 3)(a)의 그래프 rl5915에서는, 본 연구에서 제안한 알고리즘이 35초를 지나면서부터는 지속적으로 원래의 ACS보다 좋은 결과를 보여줌을 알 수 있다. rl5915 보다 단순한 그래프인 pcb3038 (그림 3)(b)은 CPU 처리 시간 80초가 지나면서 제안한 알고리즘이 더 좋은 결과를 보여준다.



(a) rl5915



(b) pcb3038

(그림 3) 그래프 rl5915와 pcb3038에 대한 제안한 알고리즘과 원래의 ACS의 비교

5. 결 론

본 연구에서는 기존의 ACS 기법의 성능을 향상시키는 알고리즘을 제시하였다. 기존 ACS 기법에 이미 알려진 지역 검색 기법인 opt-3를 적용하였고 제안한 알고리즘은 이러한 지역 검색 기법을 적용한 ACS 알고리즘보다 TSP에서 성능이 향상되었다. 기존의 ACS 알고리즘은 탐색 경로를 구성하기 위하여 인접 이웃을 검색한 후, 그 중 최소값을 주는 도시를 다음 탐색 도시로 선택하였지만 본 논문에서 제안한 부경로 ACS 알고리즘은 현재의 도시에서 시작하는 길이 w를 갖는 부경로들의 거리를 미리 계산한 후 그 정보를 이용하여 최적 경로 구성에 가장 가능성이 높은 도시를 다음 탐색 도시로 선택하였다. w의 크기가 커질 경우 즉, 부경로의 길이가 길어질 경우에는 stagnation에 빠질 수 있으므로 적당한 값의 w를 선택하는 것이 중요하다. 부경로에 대한 정보 s_w는 수식 (3.2)에서 보는 것처럼 페로몬의 초기값인 τ_0에도 큰 영향을 미친다. τ_0는 지역 페로몬 갱신에 계속해서 이용되므로 경로 생성 과정에 영향을 미치게 된다. 동일한 CPU time을 부여하여 실험한 결과 본 연구에서 제안한 기법이 매우 좋은 결과를 보여 주었다. 그래프상의 도시 개수가 적은 경우에는 두 알고리즘 모두 최적의 해를 찾음으로서 성능을 입증하였고 도시의 개수가 1000개 이상의 큰 그래프에서는 제안한 알고리즘이 기존의 ACS보다 최적 경로에 가까운 해를 찾음을 볼 수 있었다. ACS는 그래프의 크기나 연결선의 복잡 정도에 따라서 설정 변수 값을 조절함으로써 더 좋은 결과를 얻을 수 있을 것으로 보이며 제안한 기법 역시 그래프의 특성을 고려하여 w의 값을 조절해 줌으로써 더 좋은 결과를 얻을 수 있을 것으로 생각한다. 향후의 연구과제로 이러한 그래프별 특성에 적합하도록 변수들을 자동으로 설정할 수 있는 기법이 연구되어질 수 있다면 TSP에서 더욱 성능이 향상된 결과를 얻을 수 있을 것으로 예상된다.

참 고 문 헌

[1] Dorigo, M. and Maniezzo, V., "The Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol.26, No.1, pp.1-13, 1996.

[2] Dorigo, M., and Gambardella, L. M., "Ant Colonies for the Travelling Salesman Problem," BioSystems 43, pp.73-81, 1997.

[3] 이승관, 정태충, "Ant Colony System에서 효율적 경로 탐색을 위한 지역갱신과 전역갱신에서의 추가 강화에 관한 연구", 정보처리학회 논문지 B 제10-B, pp.237-242, 2003.

[4] Dorigo, M., Caro, G. D., and Gambardella, L. M., "Ant Algorithms for Discrete Optimization," Artificial Life, Vol.5, No.3, pp.137-172, 1999.

[5] Gomez, O. and Baran, B., "Reasons of ACO's Success in TSP," Proceedings of 4th International Workshop in Ant Colony Optimization and Swarm Intelligence, LNCS Vol.3172, pp.226-237, 2004.

[6] Gambardella, L. M. and Dorigo, M., "Solving symmetric and asymmetric TSPs by ant colonies," Proceedings of IEEE International Conference on Evolutionary Computation, IEEE-EC 96, IEEE Press, pp.622-627, 1996.

[7] Dorigo, M. and Stutzle T., Ant Colony Optimization, MIT Press, 2003.

[8] Bonabeau, E., Dorigo, M., and Theraulaz, G., "Swarm Intelligence: From Natural to Artificial Systems," Oxford University Press, 1999.

[9] Dorigo, M. and Gambardella, L. M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997.

[10] Bianchi, L., Knowles, J. and Bowler, N., "Local search the probabilistic salesman problem: correction to the 2-p-opt and 1-shift algorithms," Technical Reports, IDSIA-18-03, Dalle Molle Institute of Artificial Intelligent, Switzerland, 2003.

[11] http://www.aco-metaheuristic.org/aco-code.

[12] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp.

김 인 겸

e-mail : kik@sungkyul.edu

1990년 연세대학교 전자공학과(학사)

1992년 연세대학교 전자공학과(공학석사)

1997년 연세대학교 전자공학과(공학박사)

1996년~1998년 안양과학대학 전임강사

1998년~현재 성결대학교 정보통신공학부 부교수

관심분야 : 인공지능, 영상처리

윤 민 영

e-mail : alabama@sungkyul.edu

1989년 Univ. of Alabama in Huntsville, MS in Computer Science

1993년 Univ. of Alabama in Huntsville, Ph.D in Computer Science

1994년 Alabama A&M University, Assistant Professor

1995년~현재 성결대학교 정보통신공학부 부교수

관심분야 : Swarm Intelligence, Genetic Algorithms, Algorithm Analysis