

Improvement of Ant Colony Optimization Algorithm to Solve Traveling Salesman Problem

Juyoung Jang · Minje Kim · Jonghwan Lee[†]

Department of Industrial Engineering, Kumoh National Institute of Technology

순회 판매원 문제 해결을 위한 개미집단 최적화 알고리즘 개선

장주영 · 김민제 · 이종환[†]

금오공과대학교 산업공학부

It is one of the known methods to obtain the optimal solution using the Ant Colony Optimization Algorithm for the Traveling Salesman Problem (TSP), which is a combination optimization problem. In this paper, we solve the TSP problem by proposing an improved new ant colony optimization algorithm that combines genetic algorithm mutations in existing ant colony optimization algorithms to solve TSP problems in many cities. The new ant colony optimization algorithm provides the opportunity to move easily fall on the issue of developing local optimum values of the existing ant colony optimization algorithm to global optimum value through a new path through mutation. The new path will update the pheromone through an ant colony optimization algorithm. The renewed new pheromone serves to derive the global optimal value from what could have fallen to the local optimal value. Experimental results show that the existing algorithms and the new algorithms are superior to those of existing algorithms in the search for optimum values of newly improved algorithms.

Keywords : Ant Colony Optimization Algorithm, Traveling Salesman Problem, Optimization

1. 연구배경 및 방법

개미집단 최적화 알고리즘(Ant Colony Optimization Algorithm, ACO)[1, 2, 3, 4, 6, 10]은 복잡한 문제를 해결하기 위한 메타 휴리스틱 방법이다. 식량을 찾아 이동하는 개미들의 행동을 기반으로 하여 만들어진 이 알고리즘은 개미들이 이동을 하면서 페로몬(Pheromone)을 남기게 된다. 이동경로에 쌓여있는 페로몬의 정보를 이용해 다음 경로를 선택하게 하는 원리를 휴리스틱 탐색기법에 적용시킨 방법이다. 개미는 갈림길에서 서로 다른 페로몬의 양을 기준으로 길을 선택하게 되고, 이러한 페로몬의 정보가

업데이트 되면서 문제를 해결하게 된다. 이러한 알고리즘은 여러 문제에서 사용되고 있다. 대표적으로 작업 상점 스케줄링 문제(Job-shop Scheduling Problem, JSP), 차량 경로 문제(Vehicle Routing Problem, VRP), 순회 판매원 문제(Traveling Salesman Problem, TSP)[3, 4, 6, 8, 9, 10] 등이 있다.

개미집단 최적화 알고리즘은 처음 고안된 것이 1992년이며[9], 처음 고안된 시기에 비해 시간이 많이 지났다. TSP 문제는 시간이 지날수록 더욱 복잡해졌고, 도시의 수는 많이 증가하게 되었다. 본 논문에서는 기존의 개미집단 최적화 알고리즘이 가지는 한계점을 설명하고 한계점을 극복하여 개선한 것을 보여주고자 한다.

본 논문에서는 많은 도시 수의 TSP 문제를 해결하기 기존의 개미집단 최적화 알고리즘에서 유전 알고리즘(Genetic Algorithm, GA)[5, 8, 9]의 돌연변이를 더한 개선된

새로운 개미집단 최적화 알고리즘을 제시하여 문제를 해결하는 방법을 제시한다. 이 새로운 개미집단 최적화 알고리즘은 기존의 개미집단 최적화 알고리즘에서의 문제점인 국부 최적 값에 쉽게 빠지는 현상을 돌연변이를 통해 변경된 경로들을 통해서 전역 최적 값으로 넘어갈 수 있는 기회를 제공한다. 개선된 경로는 개미집단 최적화 알고리즘을 통해서 페로몬이 갱신되게 된다. 갱신된 새로운 페로몬은 자칫 국부 최적 값에 빠질 수 있었던 것을 전역 최적 값으로 유도하는 역할을 하게 된다.

2. 순환 판매원 문제

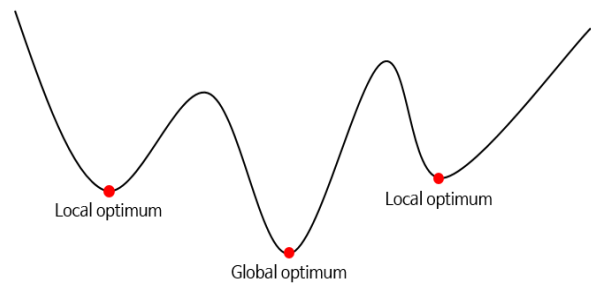
순환 판매원 문제(Traveling Salesman Problem, TSP)[3, 8, 9, 10]는 조합 최적화 문제의 하나이다. 이 문제는 NP-Hard 문제에 속하는 문제이다. 순환 판매원 문제는 주어진 여러 개의 도시가 있는 경우에 무작위로 선택된 한 도시에서 출발하여 모든 주어진 도시들을 한번씩 방문하고 다시 선택된 도시로 되돌아오는 문제이다. 이 문제는 모든 도시들 간의 이동시간이나 비용 등이 존재하며, 모든 도시들을 방문하고 돌아왔을 때의 최소 비용 혹은 최소 시간 등을 구하는 문제이다. 순환 판매원 문제는 대칭적인 거리를 가지는 문제와 비대칭적인 거리를 가지는 문제로 나뉘어진다. 순환 판매원 문제가 NP-Hard 문제에 속하는 문제라는 것은 이미 알려져 있는 사실이다. n 개의 도시를 방문해야 하는 순환 판매원 문제를 풀기 위해서는 n 개의 경우의 수를 다 확인하여 비교해야 한다는 문제점이 있다. 이러한 문제점을 해결하기 위해서 여러 가지의 솔루션들이 제시 되었다. 가장 대표적인 것으로는 시뮬레이티드 어닐링(Simulated Annealing), 터부 검색(Tabu Search), 유전 알고리즘(Genetic Algorithm), 개미집단 최적화(Ant Colony Optimization) 등이 존재한다.

3. 메타 휴리스틱

3.1 메타 휴리스틱

휴리스틱(Heuristic)이라는 개념이 등장하면서 대다수의 알고리즘은 다양한 최적화 문제를 해결하기 위해 개발되고 발전되는 방향으로 나아가고 있다. 이러한 알고리즘의 대부분은 기술기나 수식의 정보를 필요로 한다. 보통의 알고리즘은 시작점(Starting Point)에서 시작하여 시작점보다 개선된 해를 찾는 수학적 선형 혹은 비선형 수식을 구하여 그 수식을 해결하여 보다 나은 해를 선택하는 방향으로 진행된다. 수학적 최적화 알고리즘은 해

결하고자 하는 모델이 단순하고 이상적일수록 전역 최적 값(Global Optimum)을 찾는 데 유용하다. 그러나 많은 모델들의 모형이 매우 복잡하고 이상적이지 않은 모델이 많아서 이러한 알고리즘을 이용하여 전역 최적 값을 구하는 것은 매우 어려운 일이다. 또한 전역 최적 값을 구한다고 하더라도 많은 시간이 소요되어서 하나의 문제를 푸는데 오랜 시간이 걸릴 수도 있다. 이러한 문제를 해결하기 위해 국부 최적 값(Local Optimum)이라는 개념을 가지고 오게 된다. 이 국부 최적 값은 전역 최적 값이라고 불리는 완벽한 답은 아니지만 전역 최적 값에 근접하는 값을 찾고, 문제를 해결하는데 필요한 시간을 보다 줄이게 된다. 그리하여 어떠한 문제에 하나 이상의 국부 최적 값이 있다면 이 결과는 시작점의 선택에 의해 결정이 된다. 그러나 이렇게 구해진 최적의 값은 반드시 전역 최적 값이라 할 수 없다.



<Figure 1> Global Optimum and Local Optimum

<Figure 1>과 같은 모형일 경우에 국부 최적 값의 근처에서 시작점이 생성된다면 수학적 알고리즘에 의해 최적의 값은 국부 최적 값으로 수렴하게 될 것이다. 그러나 최적의 값은 국부 최적 값이 아닌 전역 최적 값이다. 휴리스틱 기법은 수학적 알고리즘에 의해서 값이 결정된다. 그리고 휴리스틱 기법은 문제마다에 맞는 수학적 알고리즘을 찾아야 한다. 문제에 마다 맞는 수학적 알고리즘을 찾기 위해서는 그 문제에 맞는 경험이나 직관 등을 이용하여 해결해야 한다. 이러한 문제를 해결하기 위해 메타 휴리스틱(Meta Heuristic)이라는 개념이 등장하게 되었다. 메타 휴리스틱 기법에는 여러 가지의 방법이 존재한다. 가장 대표적인 것으로 유전 알고리즘(Genetic Algorithm : GA), 시뮬레이티드 어닐링(Simulated Annealing : SA), 터부 탐색(Tabu Search : TS) 등이 있다. 그리고 개미집단 최적화(Ant Colony Optimization : ACO)가 있다.

3.2 개미집단 최적화

개미집단 최적화는 1992년 Marco Dorigo의 박사 학위 논문에서 가장 처음 등장하였다[3]. 이 당시의 개미집단

최적화 알고리즘은 집단과 각 식량이 존재하는 곳 사이의 경로를 탐색하는 개미들의 행동을 보고 그래프에서 최적 경로를 찾는 것을 목표를 두었다.

개미들은 이동하면서 페로몬을 뿌리고 다니게 된다. 이 페로몬은 개미들에게는 대화의 수단이 되고, 이동경로가 된다. 페로몬을 통해서 먹이의 위치를 공유하고, 집으로 가는 길을 공유하게 되면서 개미들은 이동하게 된다. 또 다른 개미들이 이동하게 되면서 페로몬을 뿌리게 되고 페로몬은 중첩이 되어서 기존의 페로몬보다 보더 더 진한 페로몬이 된다. 뒤따라오는 개미들은 기존의 페로몬보다 더 진하고 강한 페로몬에 이끌리게 되고 그렇게 보더 강해진 페로몬으로 남게 되는 것이다. 진하고 강한 페로몬이 끝에 있는 먹이를 개미들은 운반하게 된다. 그러나 먹이를 다 옮기게 되면 개미들은 더 이상 그 곳으로 가지 않게 되면서 점차 페로몬의 길이 줄어들게 된다. 그렇게 페로몬을 통해서 개미들은 먹이가 있는 가장 짧은 길을 자연스럽게 찾게 되고, 그것을 공유하게 된다.

이러한 개미들의 특성을 활용해 빠르게 최적의 값을 찾을 수 있도록 만들어진 알고리즘이 개미집단 최적화 알고리즘이다. 개미집단 최적화 알고리즘은 순환 판매원 문제를 해결하는데 가장 최적화 되어있는 것으로 유명하다. 그 이유는 우선 수많은 경우의 경로를 수학적 알고리즘에 의해 여러 번 이동하면서 정해지는 수많은 경로들 중에 겹치게 되는 경로들이 존재할 것이다. 겹치게 되는 경로들은 그렇지 않은 경로들보다 페로몬이 쌓이게 되면서 보더 강력한 페로몬으로 남게 된다. 겹치게 되는 경로들을 개미집단 최적화 알고리즘으로 적용하여서 보더 나은 경로를 찾게 되는 것이다.

4. 알고리즘 개발

4.1 기존의 알고리즘

기존의 개미집단 최적화 알고리즘에서 가장 중요한 요소는 개미이다. 개미들은 문제를 해결하기 위해서 스스로 길을 찾아 움직이게 된다. 개미들이 움직이면서 이동하는 경로를 따라서 페로몬을 남기면서 움직이게 된다. 후발주자로 따라오게 되는 개미들은 페로몬을 따라서 움직이게 된다. 따라오는 개미들 역시 페로몬을 남기면서 움직이게 된다. 페로몬은 시간이 지남에 따라서 증발하게 된다. 따라오는 개미들이 많은 길은 페로몬이 증발하는 것 보더 남는 것이 많아서 강력한 페로몬이 남게 될 것이다. 이러한 것을 응용한 것이 바로 개미집단 최적화 알고리즘이다[1, 2, 3, 4, 6, 10].

개미집단 최적화 알고리즘에서 가장 큰 장점은 최적

의 값을 찾아가는 수렴속도가 매우 빠르다는 것과 전역 최적 값을 찾는 탐색능력이 우수하다는 것이다. 그러나 국부 최적 값에 수렴할 확률이 높다는 것이다. 국부 최적 값에 수렴했을 경우에 전역 최적 값으로 넘어갈 수 있는 어떠한 수식이나 조건이 필요한데 개미집단 최적화 알고리즘에서는 그러한 능력이 부족하다. 개미집단 최적화 알고리즘에서는 많은 개미들이 한 번에 움직이기 때문에 메모리를 많이 차지한다는 단점이 존재한다.

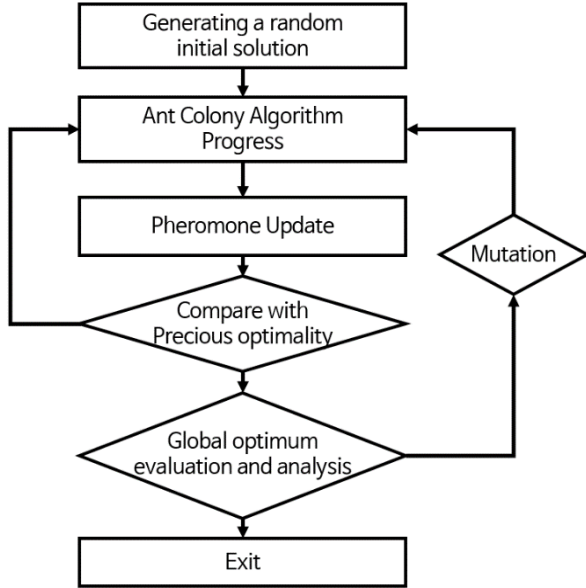
이러한 한계점을 해결하기 위해서 기존의 개미집단 최적화 알고리즘에서 일정 수식을 더하여서 전역 최적 값으로 보더 쉽게 넘어갈 수 있는 방법을 생각해 보았다. 가장 대표적으로 전역 최적 값으로 가장 많이 수렴할 수 있는 방법인 유전 알고리즘(GA)에서의 돌연변이, 회음 탐색 법(HS)에서의 새로운 하모니 생성 등 여러 가지의 방법들이 존재한다. 여기에서 유전자 알고리즘에서의 돌연변이라는 과정은 기존의 국부 최적 값에 빠질 수도 있는 알고리즘을 새로운 값으로 변경하여 문제를 해결함으로써 국부 최적 값에 빠지지 않고 전역 최적 값으로 유도하는 과정이다[5, 9]. 그리하여 개미집단 최적화 알고리즘에서의 가장 큰 문제인 국부 최적 값에 쉽게 수렴할 수 있다는 문제점을 해결하기 위해서 돌연변이라는 개념을 가져 새로운 개미집단 최적화 알고리즘을 만들어보고자 한다.

4.2 새로운 알고리즘

기존의 개미집단 최적화 알고리즘에서는 국부 최적 값에 빠지기 쉽다는 단점을 가지고 있으나, 새롭게 개선된 개미집단 최적화 알고리즘에서는 국부 최적 값 일수도 있는 최적 값을 돌연변이라는 과정을 통해서 새롭게 개선된 알고리즘을 계획해 보았다. 기존의 개미집단 최적화 알고리즘에서 개선된 알고리즘의 문제점인 국부 최적 값에 빠지기 쉽다는 점을 개선하여 새로운 알고리즘을 도식화했다.

처음에는 기존의 개미집단 최적화와 같이 임의의 시작점을 잡고 시작하게 된다. 임의의 시작점을 시작으로 하여 개미집단 최적화 알고리즘을 시작하게 되고 개미집단 최적화 알고리즘은 최적의 값을 찾게 된다[3]. 최적의 값을 찾게 되었을 때 일정한 값으로 수렴하게 될 것이다. 이때 기존의 개미집단 최적화 알고리즘이라면 이때의 값이 전역 최적 값인지 국부 최적 값인지 확인 할 수 있는 방법은 없다. 이때 일정확률을 통해서 돌연변이를 진행해 본다. 이때 돌연변이는 개미집단 최적화 알고리즘에서 나온 경로에서 한 숫자를 뽑아 변형을 주게 된다. 돌연변이를 통해 생성된 경로를 다시 한 번 개미집단 최적화 알고리즘에 적용하여 기존의 경로와 돌연변이를 통해

생성된 경로를 비교하게 된다. 이때 돌연변이를 통해 나온 값은 전혀 새로운 경로를 제시하기 때문에 만약 국부 최적 값으로 수렴하고 있다면 이러한 돌연변이를 통해서 전역 최적 값으로 수렴할 수 있는 기회를 얻게 될 것이다.



<Figure 2> New Algorithm Schematic

4.3 알고리즘에서 기호 및 변수 정의

ACO 알고리즘에서는 노드(r)에 있는 개미(k)가 노드(s)로 이동할 때 식 (1)을 사용해 다음 노드로 이동을 하게 된다. 이것을 상태전이 규칙이라 부른다. 여기서 $\tau(r, u)$ 는 노드(r)과 노드(u)사이의 페로몬의 양이다. 또한 $\eta(r, u) = 1/\delta(r, u)$ 로서 $\delta(r, u)$ 는 노드(r)과 노드(u)사이의 거리이며, $J_k(r)$ 은 노드(r)에 있는 개미(k)가 방문할 남아있는 노드들의 집합이다[11].

$$s = \begin{cases} \arg \max \{ [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (1)$$

q 는 $[0, 1]$ 사이에 분포된 무작위 파라미터이며, q_0 는 $[0, 1]$ 사이의 값을 가진다. S 는 식 (2)에서 주어진 함수식에 따라 선택된 무작위의 파라미터이다. 페로몬과 간선 사이의 길이에 의한 연산으로 선택된 것이 아니라 확률분포를 이용하였다.

$$P_k(r, s) = \begin{cases} \frac{[\tau(r, u)] \cdot [\eta(r, u)]^\beta}{\sum_c [\tau(r, u)] \cdot [\eta(r, u)]^\beta} & \text{if } s \in J_k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

식 (3)은 지역 갱신을 하게 되는 규칙을 설명하고 있다. 지역갱신은 개미들이 방문한 각 간선들을 아래의 지역 갱신 규칙을 적용하여 페로몬의 양을 갱신시킨다.

$$\gamma(r, s) = (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\gamma(r, s) \quad (3)$$

식 (4)은 전역갱신을 하게 되는 규칙을 설명하고 있다. 여기서 은 nearest neighbor heuristic에 의해 생성된 경로 길이이고 n 은 노드 수이다.

$$\gamma(r, s) = (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\gamma(r, s) \quad (4)$$

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

5. 연구방법

5.1 기존의 알고리즘

5.1절에서는 기존의 개미집단 최적화 알고리즘에서 개선된 개미집단 최적화 알고리즘을 구현한다. 문제의 정의를 위해 <Table 1>과 같은 파라미터를 설정하였다[3, 6].

개미집단 최적화 알고리즘에서 최대 반복횟수(Maximum Number of Iterations)은 1000이고 페로몬이 가지는 가중치(Pheromone Weight)는 1이다. 또한 휴리스틱이 가지는 가중치(Heuristic Weight) 또한 1이다. 페로몬의 가중치와 휴리스틱의 가중치는 기존의 값을 사용하였다. 페로몬의 증발률(Evaporation Rate)은 0.05로 적용하였고, 개미집단의 크기(Population of Ant)는 실험에 적용되는 도시들의 수를 따라 적용되도록 하였다[3, 6].

<Table 1> Parameters for ant Colony Optimization Algorithm Implementation

Parameter	
Maximum Number of Iterations	1000
Pheromone Weight	1
Heuristic Weight	1
Evaporation Rate	0.05
Population of Ant	(city number)/4

5.2 돌연변이의 적용

유전 알고리즘에 차용한 돌연변이라는 개념을 이용하여 기존의 개미집단 알고리즘에 쉽게 빠질 수 있는 국부 최적 값에서 벗어나 전역 최적 값으로 수렴할 수 있도록 한다. <Table 2>에서 돌연변이를 적용하기 위한 파라미터를 설정하였다.

<Table 2> Parameters for Applying Mutations

Parameter	
MutateProb	0.15

돌연변이를 할 수 있는 확률(MutateProb)은 0.15로 정하였다. 이는 많은 유전자 알고리즘에서 사용하는 하는 값을 사용하였다[5, 8]. 만일 최적 값이 하나의 값으로 일정하게 머무르게 된다면 기존의 경로에서 돌연변이를 할 수 있는 것을 찾아서 돌연변이를 진행한다. 돌연변이를 통해 순서를 변경된 경로는 새로운 경로가 되어서 개미 집단 최적화 알고리즘에 다시 적용되어서 기존의 값과 비교하여 진행을 하게 된다.

5.3 연구 프로세스

기존의 개미집단 최적화 알고리즘과 개미집단 최적화 알고리즘에 돌연변이를 적용한 코드를 제작하였다. 언어는 Mathworks사의 MATLAB 언어를 사용하였다. 본 논문에서는 TSP Library에서 가져온 대표적인 데이터를 사용하여 분석을 진행했다. 또한 데이터 별로 다른 도시의 수 때문에 적절한 개미집단의 크기를 정하는 것이 중요하다고 생각했다. 개미집단의 크기가 커지면 최적의 값에 수렴하는 속도는 증가하지만 문제를 해결하는데 걸리는 시간은 크게 증가하게 된다. 이러한 이유 때문에 적정수의 개미집단의 크기가 중요하다. 또한, 반복 수 역시 커지게 되면 최적의 값에 수렴할 확률이 높아지지만 문제를 해결하는데 걸리는 시간 역시 크게 증가하게 된다. 따라서 본 논문에서는 개미집단의 크기를 순환 판매원 문제에서 도시 수의 1/4로 설정하였다.

6. 실험 및 결과분석

본 연구에서는 개선된 개미집단 최적화 알고리즘을 이용해서 도시의 수가 많은 대표적인 문제들을 풀어보았다. 위와 같은 알고리즘을 MATLAB을 통해 구현해보았다. 반복횟수(Iteration)를 1000번으로 진행하였고, 같은 실험을 10회 반복하여 데이터를 구하였다. 각 도시 사이의 거리는 두 점 사이의 거리를 구하는 공식을 사용하여 거리를 구하였다. 거리는 A도시에서 B도시를 가는 거리와 B도시에서 A도시를 가는 거리는 같다고 가정하였다. 기존의 개미집단 최적화 알고리즘과 개선된 개미집단 최적화 알고리즘과의 비교를 진행하면서 데이터를 모았다. <Table 3>에서 각 문제에서 설정된 도시의 수와 각 문제의 최적 값을 표를 통해서 정리하였다.

그리고 <Table 4>, <Table 5>, <Table 6>에서는 기존의 알고리즘과 개선된 알고리즘 각각에서 나온 데이터들을 비교하였는데, 비교는 상대 오차율을 통해서 비교하였다.

<Table 3> The Number of Cities and Optimal Values for Each Problem

	No. of Cities	Optimum Length
kro100	100	21282
Ch150	150	6528
Tsp225	225	3919

<Table 4> kro100 Problem Data

	Optimum length	Conventional Algorithm	Relative Error	Improved Algorithm	Relative Error
1	21282	25540.117	20.00%	22489.0519	5.67%
2		25120.343.	18.04%	23155.0222	8.80%
3		26580.415	24.90%	23054.5956	8.33%
4		25385.406	19.28%	22775.317	7.02%
5		25822.2576	21.33%	23313.2777	9.54%
6		25931.7636	21.85%	23225.1509	9.13%
7		25802.7474	21.24%	22791.8942	7.09%
8		25503.9053	19.84%	22824.5236	7.25%
9		25701.2072	20.77%	22922.9612	7.71%
10		25256.3926	18.67%	22500.7716	5.73%
Best		25120.343	18.04%	22489.0519	5.67%

<Table 5> ch150 Problem Data

	Optimum length	Conventional Algorithm	Relative Error	Improved Algorithm	Relative Error
1	6528	7962.4715	21.97%	6784.9176	3.94%
2		8135.0692	24.62%	6857.3392	5.05%
3		8168.3811	25.13%	6967.1759	6.73%
4		7943.1866	21.68%	6886.1793	5.49%
5		7794.3565	19.40%	6941.4413	6.33%
6		7913.2815	21.22%	6912.5907	5.89%
7		7585.1152	16.19%	6993.101	7.12%
8		8089.9436	23.93%	6847.7074	4.90%
9		7974.6736	22.16%	2884.4726	5.46%
10		7874.4401	20.63%	6921.9707	6.04%
Best		7585.1152	16.19%	6784.9176	3.94%

<Table 6> tsp225 Problem Data

	Optimum length	Conventional Algorithm	Relative Error	Improved Algorithm	Relative Error
1	3919	5323.9048	35.85%	4450.6171	13.57%
2		5291.4826	35.02%	4346.1847	10.90%
3		5226.3559	33.36%	4370.9603	11.53%
4		5311.7255	35.54%	4393.5516	12.11%
5		5123.5602	30.74%	4288.6834	9.43%
6		5295.2658	35.12%	4425.2559	12.92%
7		5324.6624	35.87%	4478.61	14.28%
8		5344.4208	36.37%	4433.9518	13.14%
9		5255.6737	34.11%	4455.4529	13.69%
10		5264.7122	34.34%	4411.8534	12.58%
Best		5123.5602	30.74%	4288.6834	9.43%

$$\text{상대 오차(\%)} = ((\text{RL}-\text{OL})/\text{OL}) \times 100 \quad (5)$$

식 (5)에서의 RL은 기존의 알고리즘 혹은 개선된 알고리즘을 통해 나온 값이다. OL은 해당 문제에서의 최적 값을 나타낸다. 상대 오차율을 통해서 알고리즘을 통해서 나온 두 가지의 값이 어떤 것이 더 오차율이 적다는 것을 확인할 수 있을 것이다.

7. 결론 및 향후 과제

7.1 결론

본 논문에서는 기존의 개미집단 최적화 알고리즘에서 국부 최적 값에서 벗어 날 수 있는 방법을 추가한 새로운 알고리즘을 제안하였다. 이 방법은 유전 알고리즘에서 돌연변이라는 방법을 추가하여서 국부 최적 값에서 벗어나 전역 최적 값으로 갈 수 있는 기회를 제공한다.

돌연변이라는 것을 이용하여 기존의 개미집단 최적화 알고리즘이 쉽게 빠질 수 있는 국부 최적 값에서 전역 최적 값으로 갈 수 있도록 하였다. 알고리즘에서 값에 영향을 줄 수 있는 변수들은 일정한 값으로 고정하여 실험을 진행하였다. 실험을 통해서 나온 결과 값을 정리하여 <Table 7>에 정리하였다.

<Table 7> tsp225 Problem Data

	Conventional Algorithm	Relative Error	Improved Algorithm	Relative Error
kro100	25120.34	18.04%	22489.0519	5.67%
ch150	7585.115	16.19%	6784.9176	3.94%
tsp225	5123.56	30.74%	4288.4834	9.43%

<Table 7>을 통해서 확인 할 수 있듯이 기존의 알고리즘에서의 최적 값에 대한 접근도 보다 개선된 알고리즘에서 최적 값에 대한 접근도가 더 높다는 것을 확인 할 수 있다. 기존의 알고리즘에서 걸리는 시간과 개선된 알고리즘의 시간은 비슷한 편이었다. 문제의 크기가 커지면 커질수록 시간이 오래 걸리는 것을 확인할 수 있었는데 이것은 도시의 개수가 증가하면서 고려해야 하는 것과 비교해야 하는 것들이 기하급수 적으로 증가하게 되면서 시간이 증가하는 것을 확인할 수 있었다. 도시의 수가 150개의 경우에는 개선된 개미집단 알고리즘에서 최적 값에 가장 근접하였고, 도시의 수가 225개의 경우에는 기존의 알고리즘에서 보다 크게 최적 값에 가까워진 것을 확인할 수 있었다.

또한, 이러한 실험 결과를 통해서 알 수 있었던 것은 결과 값에 대한 분산 값이 다르다는 것이었다. 도시의 수

가 100, 150, 225개로 늘어나게 되면서 오차율이 점차 증가되는 것이 아니라, 들쭉날쭉 한 것을 확인할 수 있었다. 이것은 완전한 정답을 찾는 것이 아니라 최적 값에 근접한 답을 찾는 것을 목표로 하는데, 개미집단 알고리즘이 갱신하게 되는 페로몬에 따라서 값이 달라질 수 있다는 것을 알 수 있다.

마지막으로 TSP 문제에서 도시의 수가 증가하는데 반해 최적의 값은 점차 줄어드는 것을 확인할 수 있는데, 이것은 각 문제에서 각 도시들의 거리에 대한 값이므로 약간의 차이가 있음을 확인할 수 있었다.

7.2 향후 과제

이번 연구에서 개선된 개미집단 최적화 알고리즘이 좋은 값을 얻을 수 있었다. 순환 판매원 문제에서 도시의 수가 급증하게 되면 기존의 개미집단 최적화 알고리즘에서는 국부 최적 값에 쉽게 빠지게 된다. 이렇게 한번 빠지게 된 국부 최적 값에서 페로몬의 중첩으로 쉽게 빠져나올 수 없었다. 그러나 개선된 개미집단 최적화 알고리즘에서는 돌연변이라는 것을 통해서 새로운 경로를 풀어보게 되면서 국부 최적 값에서 벗어나 전역 최적 값으로 갈 수 있는 기회를 얻게 된 것이다. 이러한 것을 반복적으로 진행하면서 보다 나은 값으로 데이터들이 이동을 할 수 있게 되었고 기존의 해 보다 더 나은 값을 찾을 수 있게 되었다.

알고리즘을 통해 같은 데이터로 문제를 해결하여도 반드시 최적 값에 수렴한다고 할 수는 없다. 물론 반복의 수를 증가시키고 여러 번 반복을 하게 된다면 전역 최적 값은 나오지 않더라도 전역 최적 값에 근접한 답을 얻을 수 있을 것이다. 본 연구에서는 시간적 제약으로 많은 반복을 진행하지 못하였다. 연구결과는 많은 데이터가 있게 되면 보다 많은 정보를 얻을 수 있기 때문에 보다 많은 실험 결과를 통해서 데이터를 보완하는 점이 필요할 것이다. 또한 해결해본 3가지의 문제에서는 차이를 확인 할 수 있었지만, 도시의 수가 더 많아지게 된다면 반복의 수를 증가시켜서 보다 나은 값을 얻도록 해야 할 것이다.

Acknowledgement

This paper is the result of sabbatical year support from Kumoh National Institute of Technology.

References

- [1] Aziz, Z.A., Ant Colony Hyper-heuristics for Travelling Salesman Problem, *Procedia Computer Science*, Vol. 76, 2015, pp. 534-538.

- [2] Cho, S.Y., Chang, H.Y., and Choe, K.G., Ant colony Optimization Heuristic and Graph Optimization Algorithm of an Aisle-Based Order Picking System, *Journal of the Korea Society of Supply Chain Management*, Vol. 11, No. 2, 2011, pp. 13-19.
- [3] Colomi, A., Dorigo, M., and Maniezzo, V., Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, *Elsevier Publishing*, 1992, pp. 134-142.
- [4] Hong, S.M., Lee, Y.A., and Chung, T.C., Efficient Path Method using Ant Colony System in Traveling Salesman Problem, *Journal of KISS : Software and Application*, Vol. 30, No. 9-10, 2003, pp. 862-866.
- [5] Kim, I.K. and Youn, M.Y., Improved Ant Colony System for the Traveling Salesman Problem, *The KIPS transactions. Part B*, Vol. 12, No. 7, 2005, pp. 823-828.
- [6] Kim, J.S., Jeong, J.Y., and Lee, J.H., Optimizing Work-In-Process Parameter using Genetic Algorithm, *Journal of Society of Korea Industrial and Systems Engineering*, 2017, Vol. 40, No. 1, pp. 79-86.
- [7] Kim, Y.J. and Cho S.B., An Improved Cultural Algorithm with Local Search for Traveling Salesman Problem, *Korea Information Science Society*, 2008, pp. 267-271.
- [8] Lee, K.K., Han, S.K., and Lee, S.W., A Genetic Algorithm for the Traveling Salesman Problem, *Journal of The Korea Information Science Society*, Vol. 22, No. 4, 1995, pp. 559-566.
- [9] Lee, K.M. and Lee K.M., A Genetic Algorithm for Traveling Salesman Problem with Precedence Constraints, *Journal of KISS(A) : Computer System and Theory*, Vol. 25, No. 4, 1998, pp. 362-368.
- [10] Lee, S.G. and Kang, M.J., Ant Colony System for solving the traveling Salesman Problem Considering the Overlapping Edge of Global Best Path, *Journal of the Korea Society of Computer and Information*, Vol. 16, No. 3, 2011, pp. 203-210.
- [11] Lee, S.U., Polynomial Time Algorithm of a Traveling Salesman Problem, *Journal of the Korea Society of Computer and Information*, Vol. 18, No. 12, 2013, pp. 75-82.
- [12] Yan, Y.Z. and Sohn H.S., German Reyes, A modified ant system to achieve better balance between intensification and diversification for the traveling salesman problem, *Applied Soft Computing*, Vol. 60, 2017, pp. 256-267.

ORCID

- Juyoung Jang | <http://orcid.org/0000-0003-3952-5800>
- Jonghwan Lee | <http://orcid.org/0000-0001-9630-5236>
- Minje Kim | <http://orcid.org/0000-0001-5794-3921>