Journal of Korean Society of Industrial and Systems Engineering
https://doi.org/10.11627/jksie.2023.46.1.015

Vol. 46, No. 1 : 15-22, March 2023
ISSN : 2005-0461 / EISSN : 2287-7975

Development of Fuzzy Logic Ant Colony Optimization Algorithm for Multivariate Traveling Salesman Problem

Byeong-Gil Lee* · Kyubeom Jeon** · Jonghwan Lee**†

*Graduate School of Consulting, Kumoh National Institute of Technology
**School of Industrial Engineering, Kumoh National Institute of Technology

다변수 순회 판매원 문제를 위한 퍼지 로직 개미집단 최적화 알고리즘

이병길* · 전규범** · 이종환**†

*금오공과대학교 컨설팅대학원, **금오공과대학교 산업공학부

An Ant Colony Optimization Algorithm(ACO) is one of the frequently used algorithms to solve the Traveling Salesman Problem(TSP). Since the ACO searches for the optimal value by updating the pheromone, it is difficult to consider the distance between the nodes and other variables other than the amount of the pheromone. In this study, fuzzy logic is added to ACO, which can help in making decision with multiple variables. The improved algorithm improves computation complexity and increases computation time when other variables besides distance and pheromone are added. Therefore, using the algorithm improved by the fuzzy logic, it is possible to solve TSP with many variables accurately and quickly. Existing ACO have been applied only to pheromone as a criterion for decision making, and other variables are excluded. However, when applying the fuzzy logic, it is possible to apply the algorithm to various situations because it is easy to judge which way is safe and fast by not only searching for the road but also adding other variables such as accident risk and road congestion. Adding a variable to an existing algorithm, it takes a long time to calculate each corresponding variable. However, when the improved algorithm is used, the result of calculating the fuzzy logic reduces the computation time to obtain the optimum value.

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

1. 연구배경 및 방법

Non-deterministic Polynomial-time hard(NP-hard) 문제의 조합 최적화 문제를 해결하기 위한 기법으로 유전자 알고리즘(Genetic Algorithm), 타부서치(Tabu Search), 시뮬레이티드어닐링(Simulated Annealing) 등 메타휴리스틱(Metaheuristic) 기법들이 있다[7, 8, 10]. 이 기법들은 문제들이 가지고 있는 특정한 정보에 제한되지 않고 다양한 문제에 적용 가능한 특성을 가진다.

순회 판매원 문제(Traveling Salesman Problem, TSP)는 방문해야 하는 모든 노드들을 거쳐 시작 노드로 다시 돌아오는 경로를 찾는 문제이다. 변수와 노드가 많아질수록 경로를 찾기가 어려워진다. 현재 순회 판매원 문제는 처음 이 문제가 제시되고 해결되기 전보다 도시의 크기와 수가 증가하면서 복잡해지고 있다. 또한, 도시 간의 경로도 복잡해지고 경로상 변수도 많아졌다. 도로의 혼잡도, 사고 위험, 제한 속도 등 많은 변수들을 고려하여

Received 21 November 2022; Finally Revised 25 December 2022; Accepted 27 December 2022
† Corresponding Author : shirjei@kumoh.ac.kr

순회 판매원 문제를 해결할 필요가 있다.

개미집단 최적화 알고리즘(Ant Colony Optimization Algorithm, ACO)는 개미들이 이동하면서 남긴 페로몬을 통해 최적 경로를 찾아가는 원리에서 착안된 알고리즘이다. 개미들이 노드를 이동하면서 남긴 페로몬이 경로에 남아 있는 양을 기준으로 갈림길에서 어떤 경로를 선택할지 결정한다. 이렇게 얻어진 페로몬 정보를 계속해서 업데이트하여 최적 경로를 찾아 문제를 해결한다.

본 연구에서는 순회 판매원 문제를 풀기 위한 메타휴리스틱 기법 중 개미집단 최적화 알고리즘을 설명하고 추가되는 변수에 대해 유연하고 정확한 의사결정을 할 수 있도록 개선하기 위해 퍼지 로직을 적용한다.

2. 순환 판매원 문제

순회 판매원 문제는 주어진 노드 간의 최적 경로를 찾는 문제이다. 순회 판매원 문제의 조건은 각 노드에 반드시 한번 방문하고, 처음 출발한 노드로 다시 돌아와야 한다. TSP의 일반적 형태에 대한 연구는 Karl Menger에 의해 1930년대에 처음 연구되기 시작했고, 나중에 프린스턴에서 Hassler Whitney와 Merrill Flood에 의해 더욱 발전되었다[9]. 순회 판매원 문제는 Asymmetric TSP와 Symmetric TSP로 나누어진다. 본 연구에서는 Symmetric TSP에 관한 알고리즘만 연구하도록 한다. Symmetric TSP가 가지는 제약 조건은 다음과 같다.

첫 번째, 경로를 반드시 이루어야 한다.

두 번째, 모든 노드를 꼭 한번 방문해야 한다.

세 번째, 노드 간의 거리는 순방향, 역방향이 같다.

순회 판매원 문제는 NP-hard 조합 최적화 문제로 문제를 풀기 위해서는 N의 제곱으로 시간이 증가한다. 이러한 문제를 해결하기 위해 유전자 알고리즘, 시뮬레이티드 어닐링, 개미집단 최적화 알고리즘과 같은 휴리스틱 기법이 사용된다[4].

순회 판매원 문제는 모든 노드들 간에 연결되어 있는 경로를 탐색하여 최적의 경로를 찾아내는 문제이다. 노드의 수가 많아질수록 탐색해야 할 경우의 수가 많아지고 결과를 찾는데 많은 시간이 걸리게 된다.

순회 판매원 문제를 해결하기 위해 제안되는 기법으로 유전 알고리즘이 많이 제안되고 있다. Lee[11]는 유전알고리즘을 이용하여 순회 판매원 문제를 해결하는 방법을 제안하였다. PM 방법에서 Union 연산자가 PMX 등 기존의 연산자보다 쉽게 최단경로를 찾아 주는 좋은 연산자였으나 PM 자체가 계산 시간이 많이 걸리는 알고리즘으로 문제가 있었다. 그래서 PM 방법의 Union 연산자를 기존의 n개의 도시에 대한 연산자로 바꾸어 PM을 사용하지 않고서도 쉽게 계산할 수 있게 하였다. Noh[16]은 경량화 된 유전알고리즘을 통해 기존 유전 알고리즘으로 해결할 경우 계산이 복잡함을 개선하는 알고리즘을 제안하였다. 실험결과 제안한 기법으로 계산의 복잡도는 줄일 수 있었지만 성능의 개선은 크게 이루어지지 않았다.

3. 메타 휴리스틱

3.1 개미집단 최적화 알고리즘

개미집단 최적화 알고리즘은 1992년 Marco Dorigo[3]의 논문을 통해 처음 발표되었다. 발표된 논문의 목적은 개미들이 각 목표지점 사이의 경로를 탐색하는 행동을 묘사하여 그래프에서 최적 경로를 찾는 것이었다. 그 후 NP-hard 문제를 해결하는 메타 휴리스틱 기법으로 활용되며 발전해왔다.

자연에서 개미의 행동을 분석하여 만들어진 알고리즘이 개미집단 최적화 알고리즘이다. 장애물이 처음 나타났을 때 장애물을 돌아가는 양쪽 길 중 하나를 같은 확률로 선택한다면 더 짧은 경로에 더 많은 개미가 통과하게 되고 더 많은 페르몬이 누적되게 된다. 시간의 흐름에 따라 긴 경로의 페로몬이 증발량이 누적량보다 많아지게 되면 점점 긴 경로를 선택하는 개미는 줄어들게 될 것이고 결국 짧은 경로만 선택되게 된다. 개미는 에이전트, 먹이는 노드, 페로몬은 경로 선택의 가중치로 활용되어 프로그래밍 된다. 에이전트가 노드들 사이의 경로를 이동하며 페로몬 흔적을 남기고 거리가 긴 경로의 가중치는 낮게 짧은 쪽은 높게 가져가게 된다. 가중치가 높다고 해서 무조건 선택되는 것이 아니라 자연에서와 같이 확률적 선택으로 프로그래밍 되기 때문에 국부최적으로 빠질 위험이 적어진다. 실제 개미 시스템에서 페로몬 증발의 영향은 불분명하지만 컴퓨터 프로그램에서는 위와 같은 이유로 매우 중요한 역할을 한다. 위와 같은 과정을 반복하여 최적의 경로를 찾게 된다. 이러한 개미집단 최적화 알고리즘을 코딩하기 위한 Pseudo code는 <Table 1>과 같다.

<Table 1> Ant Colony Optimization Algorithm's Pseudo Code

```
procedure ACO_MetaHeuristic
while(not_termination)
generateSolutions()
daemonActions()
pheromoneUpdate()
end while
end procedure
```

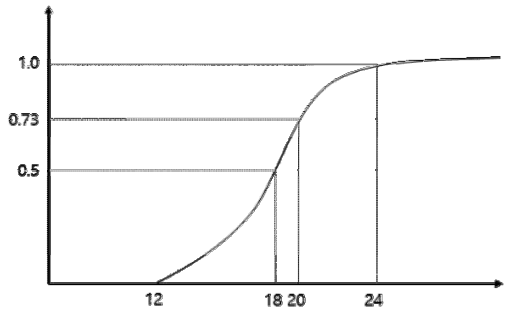
ACO 알고리즘의 기본 이론과 관련된 연구개발이 점차 활발해 지면서 여러 분야에 적용하는 연구들이 진행되었다. Lee and Han[15]은 개미집단 최적화 알고리즘을 이용하여 트러스구조물의 크기최적화 방안을 제안하였다. Park and Han[17]은 생산제품의 형상최적화에 개미집단 최적화 알고리즘을 사용하는 방법을 제안하였다.

3.2 퍼지 로직

퍼지 로직은 정성적 애매함을 정량적으로 표현하기 위해 1965년 L.A. 자데 교수에 의해 도입된 퍼지 집합의 사고방식을 기초로 하고 있다[6].

기존의 논리학에서 명제는 참이나 거짓 중에서 어느 하나를 나타내는 설명문이다. '2+4=5'라는 명제는 거짓이므로 0이 되고 '1+1=2'라는 명제는 참이므로 1이 된다. 하지만 '27도 이상의 기온은 덥다.'라는 명제는 사람에 따라 느끼는 바가 다를 수 있다. 이처럼 언어가 가지는 애매함 때문에 [0, 1] 사이 값으로 나타내는 무한치 논리의 퍼지 로직을 기반으로 다루어야 한다[1, 2, 5].

퍼지의 개념으로 '어른이다'라는 명제에 대해 그래프로 나타내어보면 <Figure 1>과 같이 표현할 수 있다. 어른의 시작을 13세경부터 완성을 24세경으로 보고 '어른이다'의 정도를 12세까지는 0으로 24세 이상은 1로 정한다.



<Figure 1> Fuzzy Graph About 'Be an adult'

18세 정도는 어떤 면에서는 어른이지만 어떤 면에서는 아직 어른이 아니라고 볼 수 있기 때문에 '어른이다'의 정도를 0.5로 정한다. 그러나 20세일 때의 높이는 그 사실의 정도를 의미하는 애매한 수치이다. <Figure 2>와 같은 그래프에서 나타내는 함수를 소속 함수라고 한다.

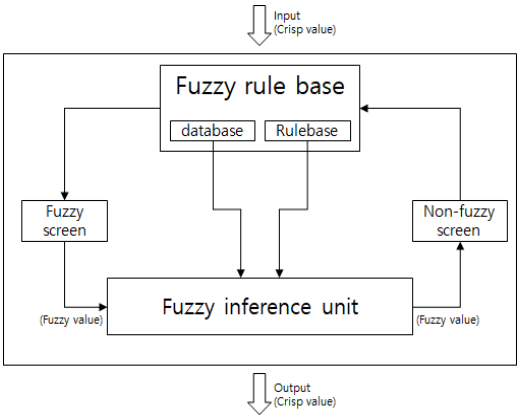
소속함수를 써서 퍼지 수는 0에서 1까지로 대응된다. 퍼지 집합은 이진법 논리가 아니라 각 대상이 모임에 속하는 정도를 소속함수로 나타내고 그 소속함수를 대응되는 대상과 함께 표기하는 집합이다[18].

퍼지 추론 시스템은 기존의 수학적인 시스템 모델링에 의해서는 잘 나타낼 수 없는 복잡하고 잘 정의되지 않는 그리고 불확실한 시스템을 if-then 형태의 규칙에 의해 잘 나타낼 수 있다고 알려져 있다.

본 연구에서는 퍼지 추론 방법 중 mamdani 추론법을 사용한다. 이 방법의 추론 과정은 <Table .2>와 같이 진행하고 이에 대한 순서도를 <Figure 2>처럼 도식화할 수 있다.

<Table 2> Mamdani's Inference Method Process

Step 1: Find the first half fit of each rule for a given input. Step 2: Obtain the inference result of each rule based on the fitness calculated in Step 1. Step 3: Obtain final inference results from inference results of each rule. Step 4: Obtain the real value through non-fuzzification.



<Figure 2> Fuzzy Inference System

4. 알고리즘 개발

4.1 기존의 알고리즘

개미집단 최적화 알고리즘은 개미의 습성을 모방한 메타 휴리스틱 기법이다. 이 알고리즘에서 중요한 것은 개미가 다음 노드를 경로상의 페로몬 양을 통해 확률적으로 선택한다는 것이다. 개미는 단순히 페로몬양이 많은 경로를 선택하기 때문에 다음과 같은 식을 이용하여 이동할 노드를 선택한다.

$$s = \begin{cases} \max \tau_{ij}^{\alpha} \eta_{ij}^{\beta} & \text{if } q \le q_0 \\ S & otherwise \end{cases} \quad (1)$$

τ_{ij}: 노드 i, j 사이 경로의 페로몬 양
η_{ij}: 노드 i, j 사이 경로의 중요도, 보통 1/거리
α, β: 페로몬과 경로 길이의 상대적인 중요도를 결정

하는 파라미터

q_0: 개미가 다음 노드를 선택할 확률. 0~1 사이의 값을 가지는 파라미터

q: 0~1 사이의 무작위 확률변수 파라미터

페르몬과 거리사이의 상관관계를 계산하여 가장 높은 수치가 나오는 곳을 다음 노드로 가는 경로로 선택하게 된다.

$q > q_0$가 되면 S는 식(2)를 따라 무작위 확률을 가지고 다음 노드를 선택하게 된다.

$$p_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} \quad (2)$$

p_{ij}: 개미가 노드 i에서 노드 j로 이동할 확률

위의 식(1), (2)에서 확인할 수 있듯이 페로몬의 양이 많을수록 거리가 짧을수록 그 경로를 선택할 확률이 높아진다.

단계 2의 내용에서 보이듯이 개미가 이동할 때 마다 페로몬을 업데이트함으로써 탐색중인 모든 개미들이 영향을 받게 된다. 이것을 지역 갱신이라고 한다. 지역 갱신은 다음과 같은 수식으로 진행한다.

$$\tau_{ij} = (1-\phi)\tau_{ij} + \phi\tau_0 \quad (3)$$

τ_0: 페로몬의 초기 값. 보통 1/(N*Nearest Neighbor의 값)

ϕ: 페로몬의 증발비율

지역 갱신은 노드와 노드 사이의 경로를 결정하는 단계라고 볼 수 있다. 이는 하나의 구간을 정하는 단계로 하나의 노드와 다음 노드 사이의 가장 짧은 경로를 찾아 갱신하여 페르몬을 업데이트해 저장해 놓는 것이다.

페로몬의 증발 비율이란 노드 간의 경로 길이가 길수록 개미가 다음 노드로 이동한 시간이 길어지고 휘발되어 사라지는 페로몬의 양을 컨트롤 하는 파라미터이다. 페로몬의 증발비율은 알고리즘 사용자가 알고리즘을 사용하는 환경에 따라 직접 수치를 부여할 수 있다. 일반적으로 페로몬 증발비율은 0.05의 수치를 사용한다. 이와 같이 개미들이 지나간 모든 경로에 같은 양의 페로몬 갱신 값을 부여한다. 지역 갱신까지의 반복이 완료되면 전역 갱신을 아래의 수식으로 진행한다.

$$\tau_{ij} = (1-\phi)\tau_{ij} + \phi\Delta\tau_{ij}^{best}$$

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/c_{best} & \text{if } best\ anttravels\ on \text{ arc } i,j \\ 0 & otherwise \end{cases} \quad (4)$$

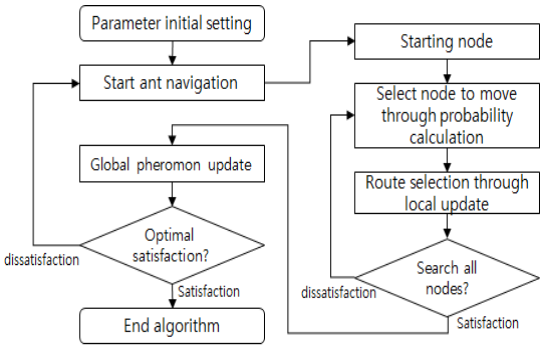
c_{best}: 가장 좋은 해를 찾은 개미의 경로

전역 갱신은 지역 갱신으로 생성된 모든 노드 간의 경로를 업데이트하는 단계라고 할 수 있다. 이 단계에서 출발 노드에서 모든 노드를 거쳐 출발 노드로 다시 돌아오는 최적 경로가 결정된다.

전역 갱신까지의 과정을 반복함으로써 다음 과정에서 더 좋은 경로를 선택할 확률을 높여 최적 해를 찾도록 유도한다. 이 개미집단 최적화 알고리즘의 진행 과정은 <Table 3>이고 이를 도식화한 순서도는 <Figure 3>이다 [12, 13, 14].

<Table 3> ACO Cycle Process

Step 1: Search to ants Step 1-1: Generate a random number between 0 and 1 and select it if less than q0. Step 1-2: If the random number is larger than q0, the next node is selected using probability distribution using equation (2). Step 1-3: Every time an ant moves through a route, it updates the area using equation (3). Step 2: Once the process up to the regional update is completed, it is updated globally with the pheromone of the ant that found the best solution($\Delta\tau_{ij}^{best}$). Step 3: The process up to the global update is repeated until a satisfactory solution is found.



<Figure 3> Existing Flowchart About ACO

4.2 개선 알고리즘

기존 알고리즘은 경로를 결정하는 중요한 요인들은 페로몬과 노드 사이의 거리이다. 하지만 순회 판매원 문제에 적용할 경우 도시사이 경로에는 많은 변수들이 존재한다. 그렇기 때문에 개미집단 최적화 알고리즘을 변수에 잘 대응할 수 있도록 퍼지 로직을 이용하여 개선을

한다. 개선된 알고리즘에서는 추가된 변수들을 각각 대응하여 계산하는 것이 아니라 퍼지 로직을 이용하여 변수들을 하나로 통합하여 표현한다.

$$s = \begin{cases} \max \tau_{ij}^{\alpha} \eta_{ij}^{\beta} \lambda_{i,j}^{\gamma} & \text{if } q \le q_0 \\ S & otherwise \end{cases} \quad (5)$$

τ_{ij}: 노드 i, j 사이 경로의 페로몬 양
η_{ij}: 노드 i, j 사이 경로의 중요도, 보통 1/거리
λ_{ij}^{γ}: 노드 i, j 사이 경로의 추가된 변수
α, β: 페로몬과 경로 길이의 상대적인 중요도를 결정하는 파라미터
q_0: 개미가 다음 노드를 선택할 확률. 0~1 사이의 값을 가지는 파라미터
q: 0~1 사이의 무작위 확률변수 파라미터

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta} \lambda_{ij}^{\gamma}}{\sum \tau_{ij}^{\alpha} \eta_{ij}^{\beta} \lambda_{ij}^{\gamma}} \quad (6)$$

p_{ij}: 개미가 노드 i에서 노드 j로 이동할 확률

초기 노드를 설정하는 수식에서만 변수들이 들어가고 뒤의 페로몬 업데이트에서는 이미 변수들을 이용하여 경로 선택을 마친 상태이기 때문에 변수들이 영향을 안줌으로 수식의 변화가 없다.

위의 식 (5), (6)과 같이 변수를 넣어 계산하게 되면 변수가 행렬로 입력이 되기 때문에 계산에 시간이 더 오래 걸리게 된다. 변수가 1개 이상 추가되면 '$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta} \lambda_{ij}^{\gamma} \lambda_{ij}^{'\gamma}}{\sum \tau_{ij}^{\alpha} \eta_{ij}^{\beta} \lambda_{ij}^{\gamma} \lambda_{ij}^{'\gamma}}$', 이 식과 같이 추가되는 변수 행렬을 뒤로 나열하는 식으로 수식이 만들어 진다. 변수가 늘어난 만큼 출발 노드의 선택이나 다음 이동할 노드의 선택 확률에 영향을 끼치게 되어 비슷한 경로를 자주 탐색하는 등의 국부 최적으로 빠질 위험성도 커지게 된다.

계속해서 갱신되는 페로몬을 제외하고 거리와 추가된 변수를 수식으로 계산되기 전에 퍼지 로직을 통해 먼저 좋은 경로를 판단하여 새로운 행렬을 만들어 위의 식에 적용한다면 기존 알고리즘에 변수를 적용하였을 때보다 정확하고 빠르게 최적 해를 도출할 수 있다.

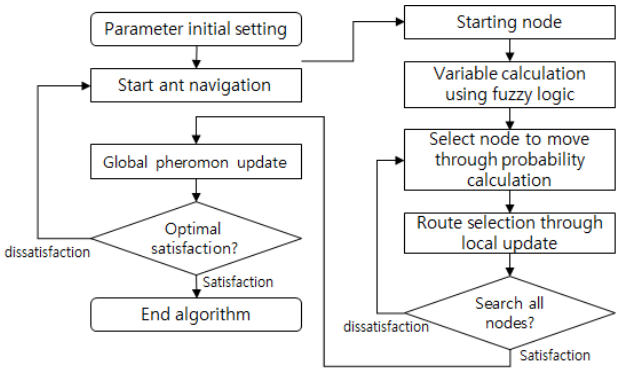
$$s = \begin{cases} \max \tau_{ij}^{\alpha} \eta_{ij}^{\beta} \omega_{i,j}^{\gamma} & \text{if } q \le q_0 \\ S & otherwise \end{cases} \quad (7)$$

τ_{ij}: 노드 i, j 사이 경로의 페로몬 양
η_{ij}: 노드 i, j 사이 경로의 중요도, 보통 1/거리
ω_{ij}^{γ}: 노드 i, j 사이 경로의 변수들의 퍼지 결과 값
α, β: 페로몬과 경로 길이의 상대적인 중요도를 결정하는 파라미터
q_0: 개미가 다음 노드를 선택할 확률. 0~1 사이의 값을 가지는 파라미터
q: 0~1 사이의 무작위 확률변수 파라미터

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta} \omega_{ij}^{\gamma}}{\sum \tau_{ij}^{\alpha} \eta_{ij}^{\beta} \omega_{ij}^{\gamma}} \quad (8)$$

p_{ij}: 개미가 노드 i에서 노드 j로 이동할 확률

퍼지 로직을 거치면 위의 식 (7), (8)과 같이 추가된 변수들 간의 상관관계를 통해 각 경로 사이의 적절한 값으로 새롭게 도출되어 하나의 변수 값처럼 도출된다. 여기서 도출된 값들은 하나의 변수 값으로 보이지만 이미 여러 변수 사이의 관계를 퍼지 로직을 통해 결정되어진 값들로 구성되어 있다. 개선된 개미집단 최적화 알고리즘의 순서도는 <Figure 4>처럼 표현할 수 있다.



<Figure 4> Improving Flowchart About ACO

5. 실 험

5.1 실험 설정

5.1.1 파라미터 설정

문제의 기본 환경의 파라미터는 <Table 4>와 같이 설정한다.

알고리즘의 최대 반복 횟수는 성능 비교를 하는 연구이기 때문에 500회로 하여 총 실행시간이 길어지지 않도록 하였다. 에이전트 수는 개미의 수로 20으로 설정하였다. 페로몬 가중치, 거리 가중치, 추가 변수 가중치는 모든 변수들이 같은 영향을 같게 하여 1로 동일하게 설정

하였다. 페로몬 증발률은 0.05로 설정하였다

<Table 4> Basic Parameters

Parameter	Size
Maximum number of iterations	500
Number of agents	20
Pheromone weight	1
Length weight	1
Added variable weight	1
Pheromone evaporation rate	0.05

5.1.2 퍼지 로직 설정

퍼지 로직은 Matlab R2017a의 어플리케이션인 Fuzzy Logic Designer를 이용하여 설정하여 주었다.

퍼지 로직을 통해 변수사이의 관계에 대해 계산을 하기 때문에 입력 변수는 설정되는 변수에 맞게 입력 값을 넣고 결과는 하나로 나와야 함으로 Total 하나로 설정한다.

변수의 관계를 계산하기 위해서 퍼지 룰이 필요하다. 퍼지 룰은 IF-THEN 룰을 사용하여 설정해 주었다. 설정된 룰은 <Table 5>와 같다.

<Table 5> IF-THEN Rule Setting

IF		THEN
Distance	Accident	Total
low	low	high
low	mid	high
low	high	mid
mid	low	high
mid	mid	mid
mid	high	low
high	low	mid
high	mid	low
high	high	low

5.2 실험 및 결과

5.2.1 변수 추가 전, 후 기존알고리즘 성능확인

기존 알고리즘에 변수가 추가되었을 때와 그렇지 않았을 때의 차이를 알아보기 위해 실험을 진행한다. 노드수는 20개이고 좌표는 1~100 사이의 난수를 사용하여 만들어졌고 좌표를 통하여 거리를 알 수 있다. 성능차이 비교를 위해 변수 추가 전과 후의 알고리즘을 10회 반복하여 도출한 최적 값과 최적 값을 찾는데 걸린 시간을 비교한다. <Table 6>에서 변수 추가 전의 최적 값 중 가장 큰 값과 후의 모든 최적 값들을 비교해보면 최대 16.43%, 최소 12.38%의 차이로 변수 추가 후 최적 값이 더 큰 것을 확인할 수 있다.

<Table 6> Comparison of Results before and after Adding Variables

	Before		After		Difference (%)
	Optimum value	Repeat	Optimum Value	Repeat	
1	362.03	176	424.42	169	14.01
2	364.95	357	422.52	193	13.62
3	362.03	258	423.37	114	13.79
4	362.03	198	434.12	463	15.93
5	363.10	413	418.63	214	12.82
6	362.03	383	436.73	475	16.43
7	362.03	201	435.97	147	16.28
8	362.03	84	433.42	292	15.79
9	363.89	403	436.45	26	16.38
10	362.03	347	416.52	23	12.38

5.2.2 기존 알고리즘과 개선 알고리즘 비교

변수 추가 후 노드수에 따른 성능차이를 비교하기 위하여 40개 60개로 실험을 진행한다. 노드가 60개인 경우 평균 최적값 감소비율은 -14.55%이고 노드 40개인 경우와 최적 값 감소비율은 -14.56%로 평균은 거의 변화가 없다.

<Table 7> Comparing Existing and Improving with 40 Nodes

	Existing		Improving		Difference (%)
	Optimum value	Repeat	Optimum Value	Repeat	
1	624.23	416	514.86	215	-15.06
2	618.04	393	507.56	242	-16.71
3	626.48	397	521.77	380	-13.54
4	597.66	439	507.52	436	-16.72
5	614.89	454	522.28	288	-13.42
6	592.42	493	516.05	428	-14.79
7	609.34	346	533.31	206	-11.08
8	620.39	317	507.89	351	-16.64
9	613.14	238	519.94	228	-13.94
10	609.24	430	521.10	429	-13.68

노드가 60개인 경우에 추가된 변수의 수에 따른 성능차이 비교하면 변수 2개 추가된 경우 변수 1개 추가된 문제들보다 확연하게 경로의 차이를 보이고, 감소율도의 차이도 높음을 알 수 있다. 기존 알고리즘은 변수가 1개일 때와 비교해도 최적값이 높게 도출되었고, 변수가 추가된만큼 한정되어 있는 반복수 안에서 이상적인 최적값을 도출하는데 어려움을 보여준다.

<Table 8> Comparing Existing and Improving with 60 Nodes

	Existing		Improving		Difference (%)
	Optimum value	Repeat	Optimum Value	Repeat	
1	811.60	339	685.70	391	-14.51
2	827.02	475	685.77	466	-14.50
3	808.95	436	689.80	369	-13.83
4	821.27	378	688.43	398	-14.06
5	789.55	329	689.39	493	-13.90
6	813.20	258	687.79	337	-14.16
7	808.74	200	697.09	137	-12.64
8	785.25	470	675.71	287	-16.21
9	838.40	366	676.22	348	-16.12
10	804.17	131	679.51	311	-15.56

<Table 9> Comparing Existing and Improving with 2 Variables

	Existing		Improving		Difference (%)
	Optimum value	Repeat	Optimum Value	Repeat	
1	956.84	398	693.99	449	-35.50
2	982.14	131	689.60	432	-36.36
3	963.46	462	682.62	303	-37.76
4	962.02	154	694.05	333	-35.49
5	989.66	105	676.45	415	-39.02
6	978.87	178	693.84	303	-35.53
7	972.34	366	707.23	227	-32.97
8	940.41	326	689.11	459	-36.46
9	998.12	377	688.24	371	-36.63
10	966.31	62	695.14	290	-35.28

6. 결론 및 향후 과제

6.1 결론

첫 번째 실험을 통해 변수가 추가되었을 때 변수가 추가되기 전보다 최적값이 높게 나오는 것을 확인할 수 있다. 개선된 알고리즘을 실행하였을 때 최적 값이 변수를 추가하기 전 기존 알고리즘의 최적 값과 같은 값으로 나왔다. 이는 추가된 변수가 첫 실험에서는 경로상에 큰 영향을 주지 못했다고 볼 수 있다. 그러나 변수가 추가된 후 기존 알고리즘의 최적 값이 높게 나온 것을 보면 변수가 추가되었을 때 제한된 알고리즘 반복 횟수 안에서 최적 값을 도출하지 못한 것을 알 수 있다. 이 실험으로 개선된 알고리즘이 성능에 영향을 줄 수 있다는 것을 확인하였다.

두 번째 실험으로는 변수가 추가된 기존 알고리즘과 개선 알고리즘의 성능을 비교하였다. 성능비교를 위해 노드의 수가 많아지는 경우와 변수가 많아지는 경우를 설정하여 실험을 진행하였다. 이 실험을 통해 노드의 수가 많아 질 경우 최적값이 높아지는 것을 확인할 수 있었다. 또, 기존 알고리즘과 개선 알고리즘의 성능 차이가 노드가 많은 문제에서 조금 더 많이 나는 것을 확인할 수 있었다. 변수가 많아지는 경우에서는 노드가 많아지는 경우보다 성능차이가 확실하게 많이 차이 나는 것을 확인할 수 있었다. 또, 기존 알고리즘의 최적 값을 비교해보면 변수가 추가된 경우에 더 높게 나오는 것을 확인하였다. 개선 알고리즘도 변수가 추가된 경우가 최적 값이 미세하게 높게 도출되는 것을 확인하였다. 이 실험을 통해 노드의 수가 증가하는 문제보다 변수가 추가되는 경우에 최적 값을 찾는데 어려움이 있는 것을 알 수 있었고 개선 알고리즘이 변수가 추가된 경우에 최적 값을 찾는데 도움이 되는 것을 확인하였다.

실험을 통해 도출된 값 중 실험 시간과 최적 값 최초 발견 반복 수는 실험 환경의 변화에 영향을 거의 받지 않는 것을 알 수 있었다. 실험 시간이 증가하는 경우는 노드의 수가 급격히 많아지는 경우를 제외하면 낮은 노드에서는 거의 비슷하게 나타났다. 최적 값 최초 발견 반복 수는 알고리즘 반복 횟수가 한정되어 있어 완전한 최적 값을 찾지 못한 이유도 있지만 개미집단 최적화 알고리즘이 확률을 통해 최적 값을 찾기 때문에 일정하지 않게 도출되었다.

6.2 향후 과제

본 연구에서는 퍼지 로직을 통해 변수들을 계산하여 최적 값을 도출하는데 도움을 줄 수 있도록 하였다. 실험의 설정들은 변수들의 가중치를 모두 같은 값으로 설정하여 실험을 진행하였다. 하지만 실제로 경로를 선택하는 경우에 있어서 모든 변수들이 같은 가중치를 가질 수 없다. 변수에 대한 가중치는 알고리즘 사용자의 주관으로 설정을 하여야 한다. 변수들의 가중치를 어떻게 줄 것인가에 대한 연구가 필요할 것이다. 개미집단 최적화 알고리즘의 특성상 반복 횟수가 많을수록 이상적인 최적 값을 도출할 수 있다. 본 연구에서는 시간적 제약으로 많은 반복 횟수로 실험을 진해하지 못하여 기존 알고리즘과 개선 알고리즘의 이상적인 최적 값을 도출하지 못하였기 때문에 반복 횟수를 증가시켜 보안할 필요가 있고 적당한 반복 횟수를 설정하는 방법에 대한 연구가 필요하다.

개미집단 최적화 알고리즘은 국부 최적 값에 빠져 많은 반복 횟수에도 이상적인 최적 값을 도출하지 못하는 단점이 있다. 본 연구의 실험 중에서도 최적 값 최초 반복 횟수가 매우 낮게 나온 경우가 이러한 경우이다. 이상

적인 최적 값을 도출하기 위해 국부 최적 값에 빠지지 않도록 보안할 필요가 있다.

Acknowledgement

This paper was supported by Kumoh National Institute of Technology

References

[1] Choe, I.C., Ha, S.H., Kim, S.J., and Jeon, H.T., A study on intelligent path searching and guide using RFID and fuzzy logic, *Journal of Korean Institute of Intelligent Systems*, 2009, Vol. 19, No. 1, pp. 139-144.

[2] Choi, W.K. and Jeon, H.T., Intelligent path guide system using fuzzy logic, *The Institute of Electronics Engineers of Korea - Computer and Information*, 2008, Vol. 45, No. 3, pp. 68-74.

[3] Colorni, A., Dorigo, M., and Maniezzo, V., Distributed Optimization by Ant Colonies, actes de la premiere conference europeenne sur la vie artificielle, pp. 134-142, 1991.

[4] Hong, S.M., Lee, Y.A., and Chung, T.C., Efficient path search method using ant colony system in traveling salesman problem, *Journal of KISS: Software and Application*, 2003, Vol. 30, No. 9·10, pp. 862-866.

[5] Jang, D.S., Cho, S.J., Tahk, M.J., Koo, H.J., and Kim, J.S., Fuzzy logic based collision avoidance for UAVs, *Journal of the Korean Society for Aeronautical & Space Sciences*, 2006, Vol. 34, No. 7, pp. 55-62.

[6] Jin, K.S., Ahn, H.G., and Yoon, T.S., An improved map construction for mobile robot using fuzzy logic and genetic algorithm, *Journal of Korean Institute of Intelligent Systems*, 2005, Vol. 15, No. 3, pp. 330-336.

[7] 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.

[8] Kim, S.J., Lee, G.S., and Choi, H.M., An efficient approach for the solution of traveling salesman problems based on self-organizing feature maps, *IEEK Summer Conference*, 1992, Vol. 15, No 1, pp. 722-726.

[9] Kim, Y.H., Kim, K.S., and Kwak, S.Y., A pedestrian collision warning system using a fuzzy logic, *Journal of Broadcast Engineering*, 2015, Vol. 20, No. 3, pp. 440-448.

[10] Lee, H.K. and Suh, S.M., A heuristic method for max(x, y), *Journal of the Korean Institute of Industrial Engineers*, 1993, Vol. 19, No. 3, pp. 37-49.

[11] 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*, 1995, Vol. 22, No. 4, pp. 559-566.

[12] Lee, S.G. and Choi, J.H., Balance between intensification and diversification in ant colony optimization, *Journal of The Korea Contents Association*, 2011, Vol. 11, No. 3, pp. 100-107.

[13] 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*, 2011, Vol. 16, No. 3, pp. 203-210.

[14] Lee, S.H. and Kim, Y.D., Ant colony system for vehicle routing problem with simultaneous delivery and pick-up under time windows, *Journal of the Korean Institute of Industrial Engineers*, 2009, Vol. 35, No. 2, pp. 160-170.

[15] Lee, S.J. and Han, Y.D., Truss size optimization using ant colony optimization algorithm, *Journal of The Architectural Institute of Korea Structure & Construction*, 2011, Vol. 27, No. 8, pp. 21-28.

[16] Noh, T.W., Oh, H.Y., and Kim, C.K., An light-weight genetic algorithm based scheme for traveling salesman problem, *Journal of Korea Information Science Society*, 2014, Vol. 41, No. 1, pp. 1222-1224.

[17] Park, M.C. and Han, S.Y., Shape optimization using the ant colony optimization algorithm, *Journal of The Korean Society of Manufacturing Technology Engineers*, 2014, pp. 100-100.

[18] Park, S.W., Park, J.W., Kim, D.W., Ohm, W.Y., Lee, C.H., and Ahn, C.H., A study on the synergy effect between neuro-fuzzy and genetic algorithms for non-linear system modeling, *The Journal of Korean Institute of Information Technology*, 2013, Vol. 11, No. 1, pp. 9-17.

ORCID

Byeonggil Lee | http://orcid.org/0000-0002-2673-178X
Kyubeom Jeon | https://orcid.org/0000-0001-8505-8620
Jonghwan Lee | http://orcid.org/0000-0001-9630-5236