



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

스마트도시공학석사 학위논문

융합 유전자 알고리즘 기반의 공공자 전거 재배치 방안에 관한 연구

- 서울시 '따릉이'를 중심으로 -

Optimizing the Bike-sharing Rebalancing Scheme
using A Novel Fused Genetic-Ant Colony
Optimization Algorithm: Focusing on Seoul Bike Case

2022년 1월

서울대학교 대학원

건설환경공학부 스마트도시공학전공

쥬 통

융합 유전자 알고리즘 기반의 공공자전거 재배치 방안에 관한 연구

- 서울시 '따릉이'를 중심으로 -

지도 교수 황 준 석

이 논문을 스마트도시공학석사 학위논문으로
제출함

2022년 1월

서울대학교 대학원

건설환경공학부 스마트도시공학전공

쥬 통

쥬통의 스마트도시공학학 석사 학위논문을 인준함

2022년 1월

위 원 장 _____ 구 윤 모 _____ (인)

부위원장 _____ 윤 현 영 _____ (인)

위 원 _____ 황 준 석 _____ (인)

국 문 초 록

더욱 합리적이고 인간적인 공공자전거 서비스를 제공하기 위해서 공공자전거가 시스템 더 효율적으로 운영되고 서비스 이용자 만족도를 높이도록 한 가지 자전거 재배치 경로 최적화 목적으로 공공자전거 재배치 최적화 모델을 제시하였다. 기존 재배치 모델들의 속도 느림, 정확도 낮음 등 한계점을 개선하기 위해서, 본 논문에서 GA와 ACO 알고리즘을 조합해서 GAACO-BSP(a Genetic Hybrid Ant Colony Optimization Algorithm for Solving Bike-sharing Scheduling Problem) 알고리즘을 개발하였다. 그리고 성능 향상시키기 위하여 GA 수행횟수 제어 함수를 수립하여 두 알고리즘을 동적으로 연결하였다. 우선 GA가 스케줄링 가능한 초기해를 구하고, 그 다음으로 GA 수행횟수 제어 함수를 통해 최적 전환 시기를 파악해서 동적으로 ACO으로 전환한다. ACO가 GA에게서 초기화 필요한 페로몬을 얻고 최종 최적해를 찾는 것이다. 서울시 공공자전거 따릉이 사례로 결과를 검증하여, GAACO-BSP은 전통 단일 알고리즘보다 뛰어난 성능 우세로 대규모 자전거 시스템에 적용하고 더 짧은 시간 만에 재배치 거리를 더 많이 줄였다. 실험을 통해 GAACO-BSP가 실제 도시 공공자전거 시스템에서 적용할 수 있다는 것을 알 수 있다.

주요어 : 따릉이, 공공자전거 재배치 최적화, 클러스터링, 유전 알고리즘, 융합모델, 최단 경로

학 번 : 2020-25957

목 차

제 1 장 서 론	1
1.1. 연구의 배경	1
1.2. 연구의 내용	2
제 2 장 선행 연구	3
2.1. 기존 공공자전거 재배치에 관한 연구.....	3
2.2. 기존 GA-ACO 융합 알고리즘.....	5
제 3 장 모델 구축 방법론	8
3.1. BSP 문제의 수학적 해석	8
3.2. BSP 해결을 위한 GAACO-BSP	11
3.2.1. 기본 생각	11
3.2.2. 전체 프레임워크.....	11
제 4 장 GAACO-BSP 알고리즘	13
4.1. GA 부분의 규칙.....	14
4.1.1. 인코딩 방식 및 초기화.....	14
4.1.2. 선택.....	15
4.1.3. 교차 및 변이.....	15
4.1.4. 정지 조건 및 전환.....	16
4.2. ACO 부분의 규칙	17
4.2.1. ACO 초기화	17
4.2.2. 경로 선택 규칙.....	18
4.2.3. Pheromone 농도 조절	18
4.3. 알고리즘 흐름도	20
제 5 장 실험 및 결과	21
5.1. 데이터 전처리.....	21

5.2. 지역센터(배송팀) 재구분.....	26
5.3. 재배치 전략방안 도출.....	29
5.3.1. 수요현황 분석.....	29
5.3.2. 재배치 최적화 방안 도출.....	32
제 6 장 결 론.....	38
참고 문헌.....	41

표 목차

[표 2-1]	4
[표 2-2]	6
[표 2-3]	7
[표 4-1]	20
[표 5-1]	24
[표 5-2]	28
[표 5-3]	29
[표 5-4]	30
[표 5-5]	33
[표 5-6]	34
[표 5-7]	36

그림 목차

[그림 1]	14
[그림 2]	22
[그림 3]	23
[그림 4]	24
[그림 5]	25
[그림 6]	26
[그림 7]	27
[그림 8]	29
[그림 9]	31

제 1 장 서 론

1.1. 연구의 배경

최근 몇 년 사이 저렴한 비용으로 쉽게 대여가 가능한 공공자전거가 늘어나면서 사람들의 건강지킴이와 환경지킴이 역할까지 톡톡히 하고 있다. 공공자전거는 이미 많은 나라, 지역에서 보급됐으며, 서울의 경우는 2015년 10월에 ‘따릉이’란 공공자전거 서비스가 등장하고 지난 5년간 지구 3,500바퀴에 달하는 1억 4,000만km를 달려 이제 서울시 대표적인 교통수단이 되었다.

현재 공공자전거를 서울 시내 곳곳에 설치 운영하고 있으나 어떤 곳을 1년 동안에 이용자 몇 번도 없고 방치된 상태로 있는 곳도 볼 수 있는 상태이고, 반면에 어떤 장소 경우는 이용자가 많아서 상대적으로 공급이 부족한 상태로 되고 있다. 이것은 바로 제대로 자전거 재배치를 못 하여 배정된 자전거 수량과 수요가 일치하지 않아서 나타난 현상이다.

대여소별 특성에 따라 대여소 자전거 이용률이 낮은 곳도 있으며 대여소 자전거 이용률이 높은 곳도 있습니다. 이러한 대여소별 공공자전거 이용률 차이를 해소하고자 적절하게 공공자전거를 배분하여 배송해야 하지만, 재배치문제는 수요예측, 운송원가, 배차 간격 선정 등 여러 요소를 포함한 복잡한 문제이므로 미흡한 부분이 계속 존재한다. 따라서 실제 상황에 적용된 최적화 공공자전거 재배치 전략이 항상 요구된다.

이런 요구로 인해 세계적으로 공공자전거 재배치에 관련된 연구가 많이 진행하였지만, Search Efficiency와 Solution Quality를 동시에 최적화하는 것은 항상 보장되지 못한다. 공공자전거 재배치문제(Bike-sharing Scheduling Problem, BSP)는 배송 차량이 배송센터에서 출발하고 일정 지역 안의 각 대여소에 가서 이미 설정된 수량으로 자전거 재배치하고 나서 다시 배송센터로 돌아가는 과정이라고 할 수 있다. 배송경로 계획, 이용자 만족도, 시간대별 모델 최적화, 대여소 연계성 등 다양한 부분이 관련되어 있어서 매우 복잡하고 어려운 이슈이다.

그러므로 기존의 연구들 방향성이 매우 다양하고 가장 대표적인 한 가지는 최단 경로로 목적은 스케줄링 모델이다. 모델 성능을 개선하기 위하여 서로의 장단점을 보완해 줄 수 있는 두 가지 이상 알고리즘을 융합해서 Fusion Algorithm을 개발하는 경우가 많으나, Multiple Object Decision Model에 스케줄링 효과가 좋지 않음, Search Efficiency 부족, Iterative Convergence 낮음 등 약점들이 흔히 존재한다. 게다가 해외에서 완비된 연구성과가 있기는 하지만 대부분이 개별적인 사례를 연구하는 것이므로 한국 국내에서 적용하는 것이 어려울 수도 있다.

1.2. 연구의 내용

본 연구는 서울시 공공자전거 '따릉이'의 대여소 분포와 대여 특징 및 패턴을 고려하여 배송경로 최적화를 목적으로 하고 서울시 맞춤형 재배치 전략을 도출해보자고 한다.

먼저 거리를 기반으로 대여소 전체를 적당한 담당 지역센터로 나누고 대여 이력 정보를 이용하여 대여 특징을 분석해서 첨두시(Peak Time)와 비 첨두시(Non-peak Time)로 구분한다. 그 다음 Genetic Algorithm과 Ant Colony System이 조합된 Genetic Hybrid Algorithm을 개발해서 각 지역센터 재배치 최적 이동 경로를 수립하고 서울시 공공자전거 서비스를 더 효율적으로 운영할 수 있도록 맞춤형 재배치 전략방안을 제시해보려고 한다.

경로 생성 부분은 Genetic Algorithm으로 초기값을 구하고 이 값으로 Ant Colony System을 초기화해서 최단 경로를 도출하는 것이며, 목적인 이동 경로 최적화된 위에 단일 알고리즘보다 Solution Speed를 동시에 향상해보자고 한다.

본 논문은 연구 소개, 선행연구의 고찰, 수학적인 문제 정의 및 모델구조, 논리적인 알고리즘 해석, 실험결과, 결론 등 부분으로 나누어 구성되어 있다.

제 2 장 선행 연구

2.1. 기존 공공자전거 재배치에 관한 연구

공공자전거 세계적으로 큰 주목을 받으면서 인기가 많은 연구 분야가 되어 있다. 국내외의 재배치 관련 연구는 크게 배송경로 계획,

이용자 만족도 높임, 시간대별 모델 최적화, 대여소 연계성 최적화 등 4가지로 나눌 수 있다. 그중 일부 대표적인 연구를 정리해봤다(표 2-1).

[표 2-1] List of some bike-sharing scheduling models

Previous Studies	Originality
Vehicle Scheduling under Stochastic Trip Times: An Approximate Dynamic Programming Approach (2018)	It's Stochastic dynamic vehicle scheduling and Approximate dynamic programming.
A Clonal Selection Algorithm for Urban Bus Vehicle Scheduling (2015)	A clonal selection algorithm-based bus vehicle scheduling approach is proposed.
Solving Vehicle Routing and scheduling problems using hybrid genetic algorithm (2011)	Vehicle Routing Problem with Time windows (VRPTW) scheduling problems is solved in various scenarios.
A Hybrid Metaheuristic Algorithm for the Integrated Vehicle Routing and Three-Dimensional Container-Loading Problem (2009)	This problem considers customer demand to be formed by 3-D rectangular items.
A Fuzzy Genetic Algorithm for Driver Scheduling (2003)	It solves the bi-objective scheduling problem. Individual shifts and the schedule as a whole have to be evaluated.
A Time- Space Network Based Exact Optimization Model for Multi-Depot Bus Scheduling (2006)	Time-space-based instead of connection-based networks are used for multi-vehicle-type bus scheduling problem (MDVSP) modeling.

2.2. 기존 GA-ACO 융합 알고리즘

Genetic Algorithm과 Ant Colony System이 모두 개발된 지 오래됐으며, 다양한 영역에서 흔히 사용되고 있다.

Genetic Algorithm은 자연 세계의 진화과정에 기초한 계산 모델로서 John Holland에 의해서 1975년에 개발된 전역 최적화 기법으로, 최적화 문제를 해결하는 기법의 하나이다. 주어진 문제에 대한 Solution을 무작위로 생성한 뒤 이 해답 집단을 진화 시켜 좋은 해답을 찾는다. 생물의 진화를 모방한 진화 연산의 대표적인 기법으로 주요 ‘선택 (selection)’, ‘교차 (crossover)’, ‘변이 (mutation)’로 구성되어 있다. Solution은 고정된 길이의 문자열로 표현되며, NP-Complete Problems 풀이 과정에서 광범위하게 사용된다.

Ant Colony System은 1992년에 Marco Dorigo 박사의 논문에서 처음으로 제안되었으며, 먹이를 찾는 개미의 행동을 보고 만들어졌다. 개미가 밖에서 먹이를 발견하면 Pheromone을 길가에서 분비하면서 집으로 돌아간다. 그 후에 다음 개미도 앞선 개미가 뿌린 Pheromone을 참조하며 비슷하게 Pheromone을 분비하면서 먹이 찾으러 길을 떠난다. 따라서 더 좋은 길일수록 더 많은 개미가 그 길을 선택하여 더 많은 Pheromone을 남긴다. 이런 Positive Feedback Mechanism 속성을 이용하여 주로 경로 최적화 문제에 자주 활용된다.

그러나 재배치문제를 해결할 때 GA와 ACO는 각각 한계점이 있다(표 2-2).

[표 2-2] GA와 ACO의 한계점

GA의 한계점 :
1. Global Searching Ability가 강한 편이지만 확정해야 하는 변수가 매우 많아서 쉽게 Local Optimal Solution에 빠진다. 2. Feedback 정보를 이용하지 않아서 Exact Solution을 구하는 것이 효율적 없다. 3. 특히 일정 범위까지 좁힌 후 방향성과 목적성이 부족하므로 쓸데없이 Redundancy Iteration가 많다.
ACO 한계점 :
1. 비교적 Local Searching Ability가 좋으나 초기에 Initial Pheromone이 모자라서 연산 속도가 느리다. 2. 개미가 배치되는 객체의 수는 최종 결과에 결정적인 영향을 미치므로 적절한 설정이 안 되면 성능 크게 떨어진다.

둘이 서로 보완해 줄 수 있는 특징을 가지고 있으므로 GA와 ACO를 조합해서 Fusion Algorithm 개발의 경우도 많다. 그중 일부 융합 알고리즘을 제시한 논문을 정리해 보았다(표 2-3).

그러나 아직은 성능과 연산 속도가 모두 우수한 조합방법을 못 찾아내는 상황이다. 최적 전환 시기, 초기해 이용 방식, 파라미터 전달 등 개선하는 데 아직 잠재력이 매우 크다.

대부분 기존의 융합 알고리즘은 미리 주어진 최대 Iteration 수에 의하여 GA가 중지되는 시기를 판단한다. 알고리즘이 이미 Convergence 해임에도 불구하고 계속 고정된 최대 Iteration까지 수행하면 시간이 오래 걸리고 얻은 초기해도 좋아질 수 없다. 이런

문제점을 개선하기 위해 'GA 수행횟수 제어 함수' C_G 를 제시하고 GA가 수행하는 횟수를 동적 제어한다.

[표 2-3] List of typical genetic fusion algorithms

Previous Studies	Originality
Inverse identification of elastic properties of composite materials using hybrid GA-ACO-PSO algorithm (2017)	The effect of random initialization of GA is subdued by passing the products of GA through the ACO operator.
A hybrid intelligent algorithm for the vehicle scheduling problems with time windows (2014)	A hybrid intelligent algorithm based on the max-min ACS and a GA is put forward to solve VRP.
A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing (2014)	Take the advantage of strong positive feedback of ACO on convergence rate of the GA into account.
QoS constraints task scheduling based on genetic algorithm and ant colony algorithm under cloud computing environment (2014)	It made use of the predict cost of time to definite fitness function combining ant operator to improve the accuracy of solution.
Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment (2008)	A novel algorithm of genetic algorithm with ant colony optimization for multiple sequence alignment.
A hybrid approach of GA and ACO for TSP (2004)	Every chromosome of GA is at the same time an ant of ACO in this hybrid approach.

제 3 장 모델 구축 방법론

3.1. BSP 문제의 수학적 해석

그래프 이론의 수학 영역(the mathematical field of graph theory) 정의에 의하면 BSP 문제는 complete graph로 표시할 수 있다. 즉, $G = (V, E)$ 이다. 여기서 V 는 배송센터와 대여소의 집합이고, V' 는 배송센터 없이 대여소의 집합이다. 또한, E 는 모두 대여소 간의 거리 집합이다.

더 명확하게 논술하기 위해서 본 논문에서는 아래와 같은 파라미터를 정하였다.

M : 재배치 배송 차량 총수

Q : 배송 차량의 최대 자전거 운반 수

d_i : 대여소 i 에서 회수/보충한 자전거 수

($d_i > 0$ 회수, $d_i < 0$ 보충)

$Cost_{ij}$: 대여소 i 와 대여소 j 사이에서의 배송 코스트

θ_i : 대여소 i 에서 배치 후 남은 자전거 수

x_{ij} : 결정변수 ($x_{ij} = 1$: 거리 (i, j) 방문, $x_{ij} = 0$: 거리 (i, j) 미방문)

그리고 연산량을 줄이고 더 효율적으로 연구 주제에 잘 맞는 연구 결과를 도출하기 위하여 몇 가지 가정했다. 우선 공공자전거 이용량에 영향을 미친 요인에 대한 가정이다.

첫째, 연구 주요 목적은 수요예측이 아닌 재배치 경로 최적화라서 더 목표성이 있게 연구를 수행하기 위해서 자전거 수요가 시간대에만 영향을 받고 다른 요인(요일, 날씨 등)과 모두 상관없다고 가정했다. 둘째, 해당 시간대가 시작되면 해당 시간대 안의 자전거 요구량을 분초까지 계산하지 않고 총수로 연산합니다. 즉, 대여소 i에 배송 · 회수해야 할 자전거 대수는:

$$d_i = \text{재배치 기간내의 총 대여수} - \text{기간내의 총 반납수}$$

($d_i > 0$ 때, 배송; $d_i < 0$ 때 회수)

셋째, 한 대여소에서 다른 대여소까지 항상 직선으로 갈 수 있고 배송 단위속도가 동일하다고 가정했다. 또한, 연산량을 줄이기 위해서 경위도로 인해 생기는 차이를 간과했다. 이론적으로 경위도와 평면 거리의 차이를 고려하면 경위도 좌표(Lon i, Lat i)인 대여소i와 경위도 좌표(Lon j, Lat j)인 대여소j 간의 거리는:

$$Cost_{ij}' = R * \arccos(K) * \frac{\pi}{180}$$

그중, $K = \sin(90 - \text{Lat } i) * \sin(90 - \text{Lat } j) * \cos(\text{Lon } i - \text{Lon } j) + \cos(90 - \text{Lat } i) * \cos(90 - \text{Lat } j)$ (북위, 동경의 경우만)

본 연구는 위에서 언급된 가정을 기반으로 데이터셋에서 제공된 경위도 좌표 데이터를 이용하여 아래와 같은 자전거 배송거리(Cost)의

연산 방식을 정했다:

$$Cost_{ij} = \text{sqrt}(\text{pow}(i, 2) + \text{pow}(j, 2))$$

그중, $i = \text{abs}(\text{Lat } i - \text{Lat } j), j = \text{abs}(\text{Lon } i - \text{Lon } j)$.

따라서 수학적으로 BSP 모델을 표시:

$$\min F(x) = \sum_{i=1}^n \sum_{j=1}^n Cost_{ij} x_{ij} \quad (1)$$

$$\begin{aligned} & s. t. \\ & \sum_{i=1}^n x_{ij} = 1, j \in V' \end{aligned} \quad (2)$$

$$\sum_{i=1}^n x_{ji} = 1, j \in V' \quad (3)$$

$$\sum_{j=1}^n x_{1j} \leq M \quad (4)$$

$$\sum_{j=2}^n x_{1j} = \sum_{j=2}^n x_{j1} \quad (5)$$

$$\theta_j \geq (\theta_i + d_i) x_{ij}, i \in V, j \in V' \quad (6)$$

$$\theta_i \geq (\theta_j - d_j) x_{ij}, i \in V', j \in V \quad (7)$$

$$\max(0, d_j) \leq \theta_j \leq \min(Q, Q + d_j), j \in V \quad (8)$$

여기서 수식(1)은 목표 함수이고, 식(2)~(3)은 모든 대여소를 한번만 방문하는 것을 확보하며, 식(4)~(5)는 배송 차들이 배송 완료 후에 모두 센터까지 돌아가는 것을 확보한다. 또한, 식(6)~(8)은 배송 차량 최대 운송수 초과를 예방한다.

3.2. BSP 해결을 위한 GAACO-BSP

3.2.1. 기본 생각

이런 VRP보다 더 복잡한 BSP 문제를 효율적으로 해결하기 위하여 GAACO-BSP 알고리즘을 제시하고 한 최단경로로 목적인 공공자전거 재배치 모델을 수립하였다. GAACO-BSP 기반한 모델구조의 기본 생각은 GA와 ACO가 조합되어 같이 최적해를 구하는 것이다. 처음에 GA가 랜덤으로 초기값을 구하여 만들어진 초기해를 ACO로 보내고 ACO가 이를 Initial Pheromone으로 이용해서 최종 최적해를 도출한다.

3.2.2. 전체 프레임워크

모델 수행을 위해 필요한 정보는 두 가지 있고, 하나는 대여소 정보(대여소 아이디, 위치 좌표)이고, 다른 하나는 대여이력 정보 (자전거 대여/반납의 대여소 아이디, 시간, 수량 등)이다. 모델의 전체 프레임워크는 크게 4 부분으로 나눌 수 있다:

첫번째는 데이터 전처리(Data Preprocessing)이다. 필요한 이력 정보는 시간대별로 통계되고 더 중요한 것은 전체 대여소를 적당한 개수의 군집으로 나누는 것이다. 여기서 K-means 군집화 알고리즘(K-means Clustering Algorithm)을 사용하였다.

K-means 군집화 알고리즘은 거리 기반한 군집화 알고리즘이며, 속도가 빠르고 Convergence 보장되고 다양한 군집 모양에 잘 적용할 수 있는 특성이 있어서 흔히 쓰는 것이다.

군집화 과정을 보면, 먼저 랜덤으로 초기화 군집 중심 $c_j, j \in (1, K)$ 을 선택하고, $\min \sqrt{(x_{il} - c_{il})^2}$ 라는 규칙에 따라서 전체 대여소를 분배한다.

군집 생성한 후 새로운 군집 중심 c_j' 을 계산한다:

$$c_j' = \frac{1}{N_j} \sum_{x_i \in T_j} x_i, j \in (1, K) \quad (9)$$

여기서 N_j 는 j 번째 군집 T_j 중 대여소 개수이다. 계속 반복하여 군집 중심이 다시 변하지 않게 되면 군집화 완성된다.

그리고 군집 보수는 같은 군집 안에 제공하거나 배치해야 할 자전거 수가 많이 크면 배차하기 복잡해질 수 있어 더 작은 군집으로 나누는 과정이다.

두번째는 유전자 알고리즘(Genetic Algorithm) 부분이다. 처음에 GA 알고리즘의 global search 능력, rapidity, randomness 등 특성을 이용해서 몇 최초해를 생성한다. 이 부분은 ACO 수행하기 위해 Initial Pheromone을 준비하는 과정이라고 생각할 수 있다. 이때 얻은 해답들이 모두 수행 가능한 재배치 경로뿐이다. 쓸데없는 헛걸음이 많이 있을 수도 있다.

세번째는 GA 수행횟수 제어 함수이다. GA가 중지하고 ACO로 전환되는 적당한 시기를 판단하려고 해서 자전거 스케줄링 특성을 고려하여 아래와 같은 'GA 수행횟수 제어 함수' C_G 를 설치하였다:

$$C_G^n = \frac{\overline{Cost}_{ij}^n}{(Cost_{ij}^{n-1})^p} (1 \leq n \leq N_G) \quad (10)$$

여기서 \overline{Cost}_n^G 는 GA가 n번째 (수행 횟수 Iteration=n)반복 수행한 후에 얻은 스케줄링 거리 평균값. N_G 는 GA가 수행 가능한 최대 Iteration 수. p 는 상수.

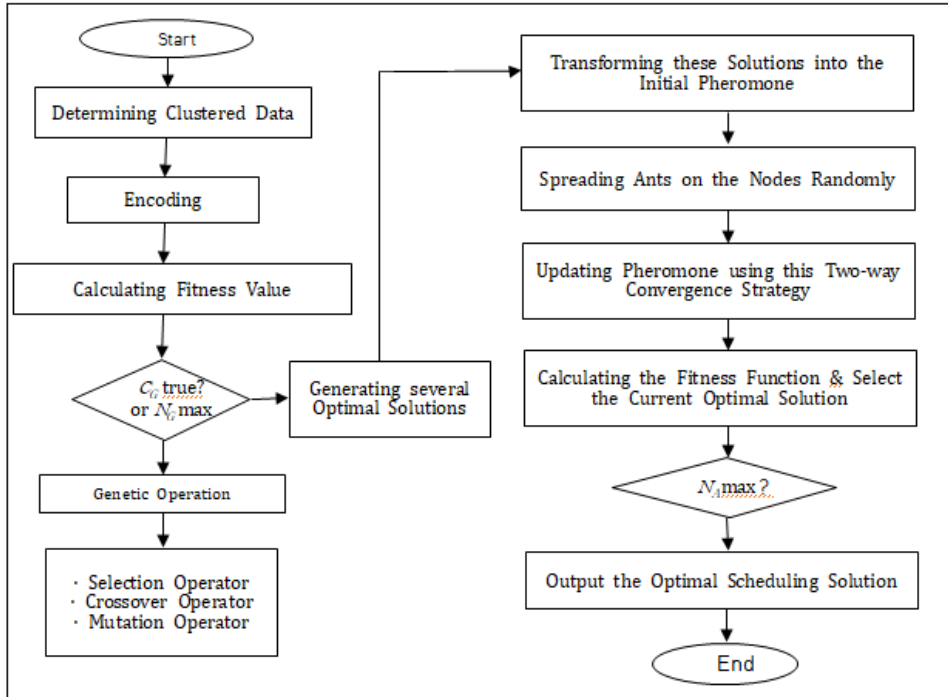
GA가 돌리는 동시에 C_G 도 모니터링하고 있다. C_G 의 값이 몇 Iteration만에 연속적으로 일정하게 나오면 최대 수행 횟수가 안 되어도 GA 부분이 중지된다. C_G 값 변화를 통해 GA가 ACO로 전환하는 최적 시기가 동적 파악되며, 알고리즘 수행 시간도 잘 줄일 수 있다.

마지막 네번째 부분은 개미 군집 최적화 알고리즘(Ant Colony Optimization Algorithm)이다.

ACO가 가진 Positive Feedback Mechanism, Fast Convergence 등 특성을 고려하여 최종적으로 이를 이용해서 최적화 스케줄링 방안을 구한다.

제 4 장 GAACO-BSP 알고리즘

GAACO-BSP 알고리즘을 제시하려고 해서 우선 알고리즘의 논리 구조도를 그려 봤다(그림 1). 필요한 데이터 전처리와 대여소 군집화 완성된 후에 GA로 초기해 찾기 시작된다. 또한, 수행횟수 제어 함수 C_G 나 최대 수행 Iteration N_G 중 하나만 부합하면 ACO로 전환된다. 그리고 ACO 정지 조건은 최대 수행 Iteration N_A 하나만 있으므로 예외 없이 N_A 부합될 때까지 수행하고 최적화 결과를 도출한다.



[그림 1] GAACO-BSP 알고리즘의 논리 구조

4.1. GA 부분의 규칙

4.1.1. 인코딩 방식 및 초기화

경로 코딩 방식으로 시점부터 종점까지의 노드 서열이 한 염색체로 만들어진다. 즉, 길이와 상관없이 한 경로는 한 염색체로 정의되고 모든 유전자자리(genetic locus)에 대여소 번호, 염색체에 차례, 배치 가능한 용량의 상한선, 하한선이 포함된다.

그 다음으로 목적 함수와 Fitness Function의 선택이다. 연구 목적 배송경로 최적화에 따라 목적 함수는 $F = \min Cost_{all}$ 로 정하였다. 또한,

선택 부분에서 쓰는 Roulette Wheel Selection이 수행할 때 항상 $Fitness Function > 0$ 요구한다. 여기서 Fitness Function은 $f = e^F$ 로 설치한 것이다.

경로 생성 과정에서 같은 대여소에 반복 방문하는 것을 예방하기 위해 ‘방문 마크’ 배열도 설치해 놓았다. 초기화 값이 항상 0이고 방문한 후에 1로 바뀌서 중복 방문을 막을 수 있다.

4.1.2. 선택

가장 많이 사용하는 Roulette Wheel Selection으로 선택 (Selection) 방법이 선정되었다. 선택 확률은 Fitness Function과 직접 관련되며, Fitness 좋을수록 선택 가능성이 높아진다:

$$p_j(i) = \frac{f(i)}{\sum_{n=1}^k f(n)} \quad (11)$$

여기서 $p_j(i)$ 는 유전자 i 가 선택되는 확률, $f(i)$ 는 i 의 Fitness Function. k 는 유전자 총수량.

4.1.3. 교차 및 변이

Self-Adaptive 방법은 교차(Crossover) 및 변이(Mutation) 확률을 동적 조절하는 데 효과가 좋기는 하지만 비교적으로 Local Optimal Solution에 쉽게 빠질 수 있어서 본 연구에서 한 가지 보완된

Self-Adaptive Crossover Probability (P_c)하고 Mutation Probability (P_m)를 사용하였다:

$$P_c = \begin{cases} \frac{k_1(f_{max} - f')}{f_{max} - f_{avg}} & , \quad f' \geq f_{avg} \\ k_2 & , \quad f' < f_{avg} \end{cases} \quad (12)$$

$$P_m = \begin{cases} \frac{k_3(f_{max} - f')}{f_{max} - f_{avg}} & , \quad f \geq f_{avg} \\ k_4 & , \quad f < f_{avg} \end{cases} \quad (13)$$

여기서 k_1, k_2, k_3, k_4 는 조절 파라미터, f_{max}, f_{avg} 는 Fitness Function의 최대값과 평균값, 그리고 f', f 는 각각 교차와 변이의 Individual Fitness.

4.1.4. 정지 조건 및 전환

GA가 중지되는 조건은 본격적으로 'GA 수행횟수 제어 함수' C_G 의 값을 판단하는 것이다. GA가 실제 수행하는 횟수가 N_G 로 가정하면 $N_{Gmin} \leq N_G \leq N_{Gmax}$ 범위에서 GA가 반복 수행하면서 제어 함수 C_G 값도 동시에 통계한다. 연속한 Iteration 3개에서 $C_G^{n+1} = C_G^n$ 값이 성립되면, 즉 C_G 값이 계속 일치하게 나오면 GA가 중지하는 적당 시기로 판단하고 ACO로 전환된다.

4.2. ACO 부분의 규칙

4.2.1. ACO 초기화

먼저 ACO 부분의 목적 함수와 Fitness Function은 GA와 일치하게 설치되었다. 개미 개수는 노드 개수와 동일하게 설정되었다.

그 다음으로 ACO 초기화에 가장 중요한 Initial Pheromone 정하는 것이다. 전통적으로 ACO의 Initial Pheromone이 무작위로 생성된 정보로 생성된다. 이와 달리, 본 연구는 앞에서 GA 생성한 초기해들을 이용해서 ACO Initial Pheromone인 τ_s 를 만들어진다.

Pheromone 초기화 값 τ_s 는 아래와 같다:

$$\begin{aligned}\tau_{ij}^s &= \tau_{ij}^c + \tau_{ij}^g \\ \tau_{ij}^g &= \sum_{k=1}^{\psi} \Delta\tau_{ij}^k, \Delta\tau_{ij}^k = Qr_{ij}\end{aligned}\tag{14}$$

여기서 τ_{ij}^c 는 자전거 시스템 규모에 따라 주어진 Pheromone 상수. τ_{ij}^g 는 GA가 구한 Solution 결과에 의해 전환된 Pheromone 값. 그중 ψ 는 GA가 생성된 초기해 개수, τ_{ij}^k 는 k번째 해에 경로(i, j)의 Pheromone 농도, Q 는 Adjustment Coefficient.

무작위로 생성된 Initial Pheromone 보다는 이처럼 GA가 이미 구한 초기해로 ASC를 초기화하여 최적해 구하는 것이 목적성 더 높고 효율성이 있다.

4.2.2. 경로 선택 규칙

대여소 (노드) i 에 위치한 k 번째 개미는 아래 수식과 같은 방식으로 다음 목적 노드 j 를 찾는다.

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{l \in U} [\tau(i, l)]^\alpha [\eta(i, l)]^\beta}, & j \in U \\ 0 & , j \notin U \end{cases} \quad (15)$$

여기서: $P_k(i, j)$ 는 k 번째 개미가 노드 i 에서 다음 노드 j 를 선택하는 확률; $\eta(i, j)$ 는 경로 (i, j) 의 Visibility로서 $\frac{1}{Cost_{ij}}$ 로 설치하였다. U 는 도착 가능한 노드 집합이다.

α, β 는 모두 ASC의 수행 Adjustment Coefficient며, α 는 경로 궤적의 상대적 중요성이고 β 는 경로 Visibility 상대적 중요성이다. 파라미터 클수록 Convergence Speed 빨라지지만, Local Optimal Solution에 빠질 확률도 높아진다. 따라서 시스템의 실제 규모 및 특성에 따라 값을 심중하게 정해야 한다.

4.2.3. Pheromone 농도 조절

- Pheromone 갱신 규칙

Residual pheromone이 많아 휴리스틱 정보 잠기는 것을 막기 위해

Residual pheromone 분포를 항상 실시간 갱신해야 한다. 구체적으로 말하자면 Global 갱신과 Local 갱신으로 나눌 수 있다.

Global 갱신은 Iteration 끝날 때마다 개미들이 모두 경로 탐색 완성한 후에 진행된다. 노드 i 와 j 가 모두 이번 순환에서 생성된 경로에 포함되면 경로 (i, j) 에서 분포하는 pheromone 농도가 아래 수식과 같다:

$$\begin{aligned}\tau_{ij}(i, j) &= (1 - \rho) \cdot \tau_{ij}(i, j) + \Delta\tau_{ih_i} \\ \Delta\tau &= \sum_{k=1}^m \Delta\tau_k(i, j), \rho \in (0, 1)\end{aligned}\quad (16)$$

또한, 경로 구축하는 과정에서 모든 개미가 한 경로 (i, j) 에 도착하면 이 개미가 이 경로에서 분포하는 Pheromone을 Local 갱신한다:

$$\begin{aligned}\tau_{ih_i}(i, j) &= (1 - \xi) \cdot \tau_{ih_i}(i, j) + \tau_0 \\ \Delta\tau_0 &= Q / L_g, \xi \in (0, 1)\end{aligned}\quad (17)$$

ρ, ξ 는 Pheromone Global와 Local 휘발성 (비중계수)이고, Q 는 Initial Pheromone 초기화 때와 같은 Adjustment Coefficient이고, L_g 는 개미 집단이 이때까지 찾아낸 최적 경로의 총 길이이다.

- 관련 파라미터 갱신 규칙

ACO에서는 Adjustment Coefficient 총 α, β, Q 3가지로 나와 있다. 초기값은 시스템 규모와 특징을 결합해서 주어지지만 실제 상황에 맞추려고 수행 기간에서도 새로운 순환이 시작되기 전에 항상 갱신해야 한다. 본 연구에서는 가중 평균법으로 파라미터 α, β, Q 를 갱신하며,

가중치가 지난 순환에서 개미 집단이 찾은 경로 거리로 정한다. 자세한 방법은 아래와 같다:

$$\alpha_g = \frac{\alpha_{k_1} L_{k_1} + \alpha_{k_2} L_{k_2}}{L_{k_1} L_{k_2}}, \quad (18)$$

$$\beta_g = \frac{\beta_{k_1} L_{k_1} + \beta_{k_2} L_{k_2}}{L_{k_1} L_{k_2}}, \quad (19)$$

$$Q_g = \frac{Q_{k_1} L_{k_1} + Q_{k_2} L_{k_2}}{L_{k_1} L_{k_2}}, \quad (20)$$

4.3. 알고리즘 흐름도

GAACO-BSP 알고리즘 흐름도이다(표 4-1). 핵심 과정이 단계별로 제시되어 있으며, 대응한 수식이 뒤에서 표시되어 있다. 전반적으로 모델 구성과 작업 순서를 알 수 있다.

[표 4-1] GAACO-BSP 알고리즘 흐름도

GAACO-BSP Algorithm	
1	Data Preprocessing
2	대여소 K-means 군집화; [수식(9)]
3	<i>gen</i> = 1, <i>C</i> = <i>false</i> ;
4	for(<i>gen</i> = 1; <i>gen</i> < <i>G_{max}</i> ; <i>gen</i> ++)
5	{
6	선택(Roulette Wheel Selection); [수식(11)]
7	교차(Crossover); [수식(12)]
8	변이(Mutation); [수식(13)]
9	if(<i>C</i> = <i>true</i>) break;

10	else C 갱신; [수식(10)]
11	}
12	Pheromone 초기화; [수식(14)]
13	ACO의 제어 파라미터 설치
14	for($iter = 1; iter < A_{max}; iter ++$)
15	{
15	모든 개미 랜덤으로 시점 선정
16	for($n=1; n < \text{대여소 수량}; n++$)
17	{
18	경로 선택; [수식(15)]
19	local 갱신; [수식(17)]
20	}
21	global 갱신; [수식(16)]
22	파라미터 갱신; [수식(18), (19), (20)]
23	}
24	최종 최적해 도출
25	end

제 5 장 실험 및 결과

5.1. 데이터 전처리

이번 연구는 서울시 열린데이터 광장^①에서 제공되는 따릉이 대여소 정보를 사용했으며, 데이터 클렌징(Data Cleansing)을 통해 그중 대여 정보 있는 대여소 1,806개에 한해서 대여소 아이디와 좌표를 취득하여

^① 서울시 열린데이터 광장에서 제공되는 따릉이 대여소 정보:
<https://data.seoul.go.kr/dataList/OA-13252/F/1/datasetView.do>

실험을 수행하였다. 열린데이터 광장은 국가에서 보유하고 공개 가능한 서울시에 관련된 데이터를 시민들이 용이하게 활용할 수 있도록 다양한 공공데이터(Dataset)와 Open API를 제공하고 있다. 본 연구는 2021년 6월에 가장 새로운 업데이트된 따릉이 대여소 정보를 사용했습니다 (그림 2).

서울특별시 공공자전거 대여소 정보

서울특별시 공공자전거 대여소(따릉이) 현황정보입니다.
대여소의 이름, 관리번호, 위치정보와, 거치대수 정보를 제공합니다.

파일내려받기 * 파일에 이상이 있는 경우 '오류신고'를 통해 운영자에게 알려주세요. [오류신고](#)

NO	항목	파일명	용량 (MB)	수정일	내려받기
1	데이터	공공자전거 대여소 정보(21.06월 기준).csv	0.2	2021.07.27	
2	데이터	공공자전거 대여소 정보(21.01.31 기준).csv	0.2	2021.02.22	

```

In [23]: amount_inner=pd.merge(dataSites,dataCount1, on='ID')
         print('all_dataframe: {}'.format(amount_inner.shape))

all_dataframe:(1806, 5)

```

[그림 2] 따릉이 대여소 정보(서울시 열린데이터 광장 제공)

그리고 본 연구 기본데이터 중 하나인 따릉이 대여이력 정보도 동일하게 서울시 열린데이터 광장에서 제공된 것이고 2020년 12월부터 2021년 6월까지 총 6개월간의 대여정보를 사용하였다(그림 3). 선택한 구간안에의 유효 대여이력 데이터는 총 6,894,866개 있다.

서울특별시 공공자전거 대여이력 정보

서울특별시 공공자전거 대여이력 정보입니다. 자전거 이동경로에 대한 데이터 분석이 가능하도록 년도별, 대여소별, 자전거별 대여이력 원천 데이터를 제공합니다.

파일용량이 큰 관계로 LTFviewer과 같이 대용량파일 프로그램을 이용하셔야 모든 내용을 확인하실 수 있습니다.

파일내려받기

* 파일에 이상이 있는 경우 '오류신고'를 통해 운영자에게 알려주세요.

오류신고

NO	항목	파일명	용량 (MB)	수정일	내려받기
1	데이터	공공자전거 대여이력 정보_2021.06.csv	461.1	2021.08.20	
2	데이터	공공자전거 대여이력 정보_2021.05.csv	393.0	2021.08.20	
3	데이터	공공자전거 대여이력 정보_2021.04.csv	405.1	2021.08.20	
4	데이터	공공자전거 대여이력 정보_2021.03.csv	290.6	2021.08.20	
5	데이터	공공자전거 대여이력 정보_2021.02.csv	164.2	2021.08.20	
9	데이터	공공자전거 대여이력 정보_2021.01.csv	107.1	2021.02.22	
10	데이터	공공자전거 대여이력 정보_2020.12.csv	155.2	2021.02.22	



```
In [16]: hour_df = pd.read_csv('./대여정보.csv', encoding='GBK')
print('Data form: {}'.format(hour_df.shape))
Data form: (6894866, 12)
```

[그림 3] 따릉이 대여 이력 정보(서울시 열린데이터 광장 제공)

또한, 인터넷을 통해서 현재 사용중인 따릉이 운영체제에 관련 자료를 얻었다. 기존의 공공자전거 재배치 운영체제는 관리소 2개, 자전거대기소 2개, 지역센터 6개 기준으로 구분하여 운영하고 있으며, 각각 중랑센터, 4대문센터, 영남센터, 천호센터, 잠실센터, 사당센터, 여의도자전거대기소, 도봉자전거대기소, 강북자전거관리소, 강남자전거관리소이다(그림 4).



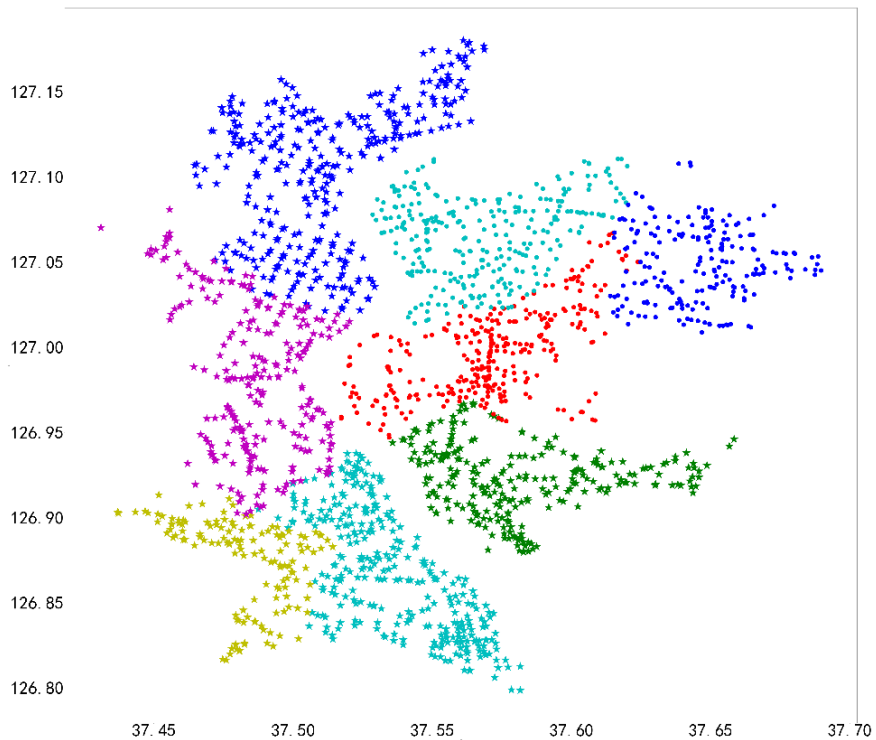
[그림 4] 기존의 따릉이 배송팀 구성

[표 5-1] 기존체제: 배송차량 및 대여소 분배

연 번	지역센터명	배송차량 대수	대여소 수
1	영남	6	245
2	천호	4	271
3	서남	3	207
4	사당	4	240
5	상암	10	218
6	4대문	10	235
7	중랑	9	232
8	도봉	3	270
9	기타	7	165
합 계	-	56	2083 (*무효 대여소 포함)

그 다음으로 이런 관할체제를 기반으로 현재 따릉이 배송팀은 총 56대 배송 차량을 보유하고 팀 9개로 나뉘서 팀별로 담당하는 대여소를 구분하여 자전거 배송·회수 업무를 수행하고 있다. 우선 기존 배송팀의 분포 및 각 구역 대여소 수를 정리해 놓았다(표 5-1).

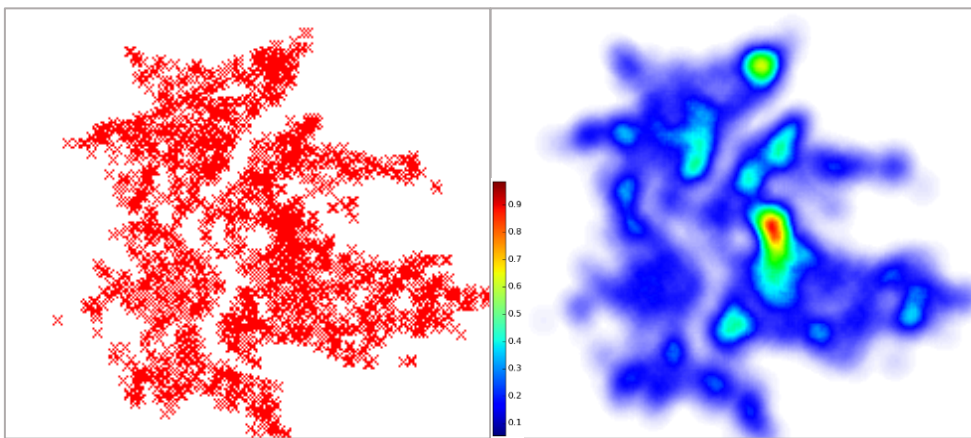
더 직관적으로 보기 위해서 위에서 언급된 데이터 정보를 이용하여 각 배송팀 관할구역의 대여소 위치를 표시하였습니다(그림 5).



[그림 5] 기존 배송팀 관할구역 대여소 위치 표시

전반적으로 보면, 대여소 전체가 지리적 위치에 따라 균일하게 각 지역센터로 나뉘서 합리적으로 구분되어 있다. 그러나 현실 상황은 각

대여소의 대여량 많이 달라서 재배치 문제를 많이 일으킬 수 있다. 게다가 기존의 분배 방식은 각 구역 포함된 대여소가 너무 많으므로 배송경로가 복잡해질 수 있다는 단점도 존재하다. 전체 대여이력 정보로 열지도를 그려보면 대여소 간의 자전거 대여 수량이 몇 십배 차이가 날 수도 있다는 것을 알 수 있다(그림 6).



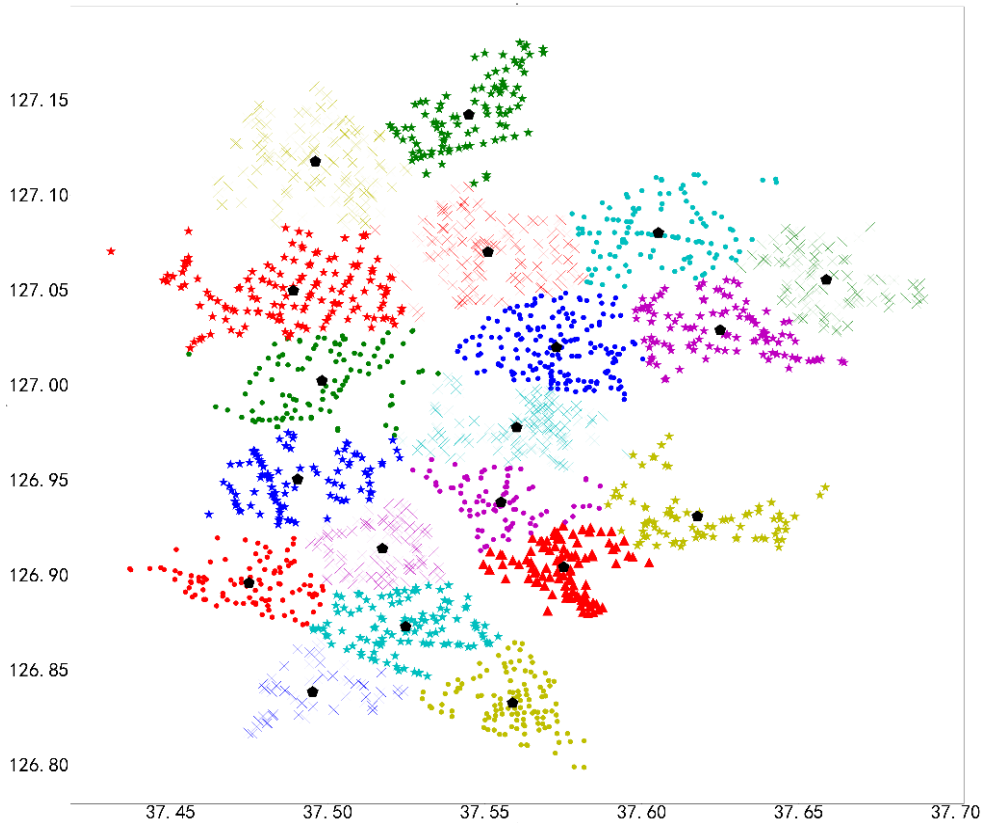
[그림 6] 따릉이 전체 대여정보 열지도(Heatmap)

5.2. 지역센터(배송팀) 재구분

이런 현황을 더 효율적으로 개선하기 위해서 재배치 수행하기 전에 우선 지역센터와 배송팀을 재구분할 필요가 있다. 실제 대여이력 정보 밀도를 참조해서 제4장에서 언급된 Weighted Kernel K-means 군집화 방법으로 기존의 배송팀과 배송차량을 합리적으로 재분배하였다.

실험결과에 의하면 배송팀 19개로 나누는 것이 더욱 적당하다고 한다.

따라서 아래 그림과 같이 새로운 배송팀 구성하였으며, 지역센터는 배송팀과 동일하게 설치하였다(그림 7).



[그림 7] 제시된 개선방안: 지역센터 관할구역 및 위치 표시
(검은색 점은 지역센터 위치)

그리고 기존 대여 이력 데이터를 기반으로 각 지역센터 정확한 위치 좌표와 보유하는 최적 배송차량의 수량을 선정하였다. 결론을 말하자면 기존의 56대 기초 위에 2대 추가하여 총 배송차량 58대로 팀 19개로 구분해서 자전거 배송업무를 수행하는 것을 추천한다(표 5-2).

[표 5-2] 개선방안: 배송차량 및 대여소 분배

연 번	차량대수	대여소 수	연 번	차량대수	대여소 수
1	1	83	11	3	107
2	1	94	12	3	97
3	1	57	13	3	133
4	1	95	14	3	115
5	2	103	15	4	140
6	3	106	16	5	117
7	2	94	17	5	139
8	2	80	18	3	106
9	6	142	19	7	165
10	3	110			
합 계	배송차량: 58 대여소: 2083				

개선된 운영방식 성능을 검증하기 위하여 두가지 운영방식으로 실제 재배치 문제를 해결하고 성능을 대비해보았다. 따릉이 실제 대여정보를 이용해서 기존 운영 방식(배송팀 9개)와 개선된 운영 방식(배송팀 19개)로 재배치 실행하였다.

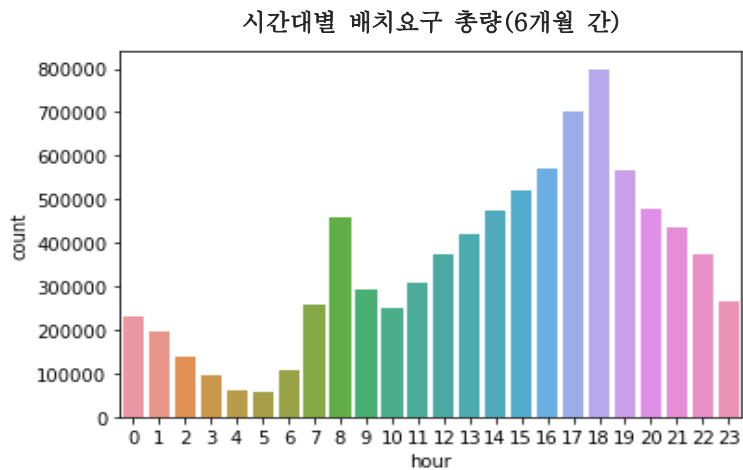
기존 체계로 재배치 알고리즘 수행하는 결과는 수행시간이 12198.562초이고 총거리 11.12726로 나왔다. 개선 방안 같은 경우, 그룹 19개로 나눈 후에 연산시간 7401.054초까지 크게 줄고 배송경로 총거리가 9.71553으로 나타났다. 기존 운영방식에 비해 제시된 새로운 개선방안은 재배치 총시간과 총거리 각각 로 크게 감소시켰다(표 5-3). 이런 결과는 배송팀 재구분의 필요성을 충분히 증명할 수 있다.

[표 5-3] 배송팀 구분 방안 성능 대비

	기존체계	본 연구	성능 향상
총 연산시간	12198.562	7401.054	39.39%
총 거리	11.12726	9.71553	12.69%

5.3. 재배치 전략방안 도출

5.3.1. 수요현황 분석



[그림 8] 각 시간대 전 대여소 배치요구 총량

모델 설명 부분에서 정해진 가정과 규칙에 따라서 먼저 각 시간대에 수요 현황을 파악하기 위해서 시간대별 대여 정보를 통계했다. 시간대에 따라서 재배치 필요한 자전거 수량이 얼마나 차이 나는지 전반적으로 관찰하려고 해서 먼저 대여이력 데이터셋 기간인 6개월 간에 모든

데이터를 통합해서 시간대별 배치 요구 총량 그래프를 제작했다(그림 8). 이 부분에서 통계한 수량은 각 대여소가 매일 해당 시간(단위: 1시간당)의 대여량과 반납량 차이값의 합이다. 그래프를 통해 출퇴근 시간에 공공자전거 이용량이 대폭 증가한다는 것을 알 수 있다. 이럴 때 수요 급증으로 인해 공급 부족한다는 문제가 더 쉽게 발생된다. 따라서 이용량이 많이 높은 시간대는 다른 시간대와 동일하게 처리하면 안되고 수요를 보장하려고 해서 재배치 경로 최적화하는 목적을 적당하게 약화해야 한다는 결론을 얻었다.

재배치 경로 최적화하는 목적을 약화하는 것은 스케줄링 알고리즘의 파라미터 조절을 통해 이루어진 것이다. 이를 위해서 수요량 추세만 대충 관찰하면 안 된다. 배송경로 최적화의 동시에 이용자의 만족도 높이기 위해 모든 시간대를 명확하게 침두시와 비침두시로 구분하는 것이 필요하다.

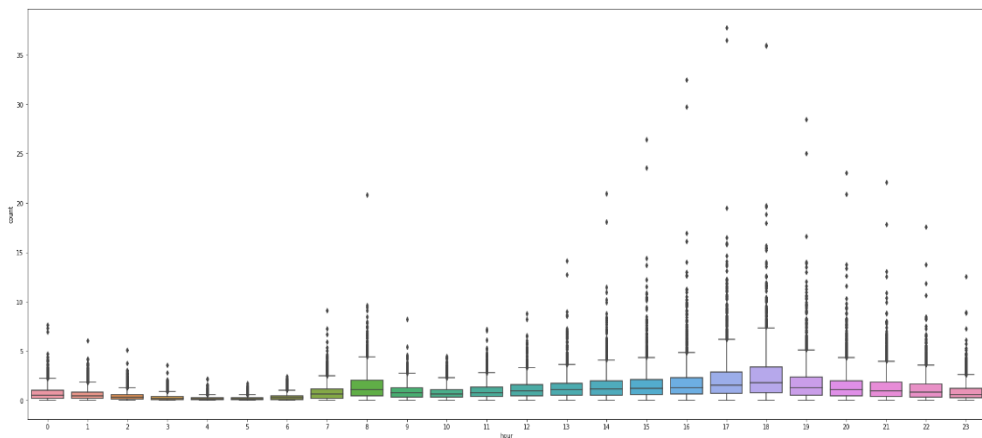
[표 5-4] 일평균 시간대별 재배치 수요량(전 대여소)

Hour	Average Amount	Hour	Average Amount
0	1267.049451	12	2042.373626
1	1068.478022	13	2317.010989
2	752.082418	14	2604.923077
3	517.873626	15	2853.708791
4	329.038462	16	3136.681319
5	310.983516	17	3844.967033
6	582.032967	18	4391.516484
7	1412.923077	19	3113.186813
8	2514.236264	20	2621.307692
9	1601.252747	21	2397.225275
10	1374.532967	22	2050.637363
11	1690.945055	23	1462.241758

대여소 전체의 일평균 시간당 재배치 수요량을 표로 정리했다(표 5-4). 표를 통해 시간별에 따라 수요량이 크게 차이 있는 것을 알 수 있다. 최초 시간당 수요량은 오전 5시에 나타난 평균 310.98대이고 최대 시간당 수요량은 퇴근 시간인 오후 6시(18시)에 나타난 평균 4391.51대이므로 차이가 14배로 나왔다.

그리고 실제 배송 조건을 보면, 배송차량이 총 58대이고 차량당 한번에 자전거 약 40~50대를 운반 가능하다. 종합적으로 고려해서 첨두시와 비첨두시 시간대의 경계선은 ‘배치수요량 2500개’로 선정됐다. 그림 8과 표 5-4를 보면, 아침 8시와 오후 2시부터 8시까지는 한 시간당 배치 수요량이 평균 2500개 넘었다. 따라서 이들은 첨두시로 정의하고 나머지 시간대는 비첨두시로 정했다.

그 다음으로 지역이 다르기 때문에 각 대여소 각 시간대의 자전거 수요량도 서로 많이 다르다. 이를 확인하려고 해서 Boxplot으로 각 대여소 시간당 수요량을 더 세부적으로 분석하였다(그림 9).



[그림 9] 각 대여소 시간당 대여량 분포

상자 그림을 통해 데이터 분포에서 심하게 벗어난 극단 데이터를 확인할 수 있다. 이런 IQR(Inter Quartile Range)보다 많이 높은 점이 해당 대여소가 해당 시간대에 수요 급증한다는 의미하며, 이런 경우를 응대하려면 비슷하게 스케줄링 할 때 수요 보장하기 위하여 배송경로 최적화하는 목적을 적당하게 약화해야 한다. 본 연구에서 상위경계(Upperfence)보다 더 높은 아웃라이어들 모두 극단 데이터로 정의하고 알고리즘 파라미터 조절을 통해

대여 및 수요 현황 분석의 결론으로서, 오전 8시, 오후 2시~ 8시 사이는 대여 침두시라고 할 수 있고 각 시간대에도 수요량이 보다 아주 높은 특별한 대여소가 존재한다. 예를 들면, 대여 총량 가장 높은 대여소는 502번 ‘뚝섬유원지역 1번출구 앞’이고 6개월 동안 총 대여 횟수는 58491회나 초과했다. 그리고 수요 보장하기 가장 어려운 대여소는 207번‘여의나루역 1번출구 앞’이다. 오후 5시(17시)의 시간당 평균 수요량이 37.74에 달하였다. 뒤에서 GAACO_BSP 알고리즘을 수행할 때 이런 부분에 신경을 더 많이 썼으며, 가장 적당히 파라미터를 따로 찾았다.

5.3.2. 재배치 최적화 방안 도출

대여 특징을 분석해본 다음에 제안된 모델로 재배치를 수행했습니다. 먼저, GAACO_BSP가 기존 대여 정보로 생성한 각 지역센터 배송경로 거리 및 연산시간을 정리하였다(표 5-5). 그리고 GA와 ACO가 같은 시간 안에 똑같은 대여 데이터로 생성한 경로 길이를 오른쪽에서

대응하여 표시하였다. 그중 ‘성능 향상’은 GAACO_BSP가 ACO의 Cost와 비교해서 얻은 것이다.

[표 5-5] 제안된 GAACO_BSP와 단일 GA/ACO의 대비

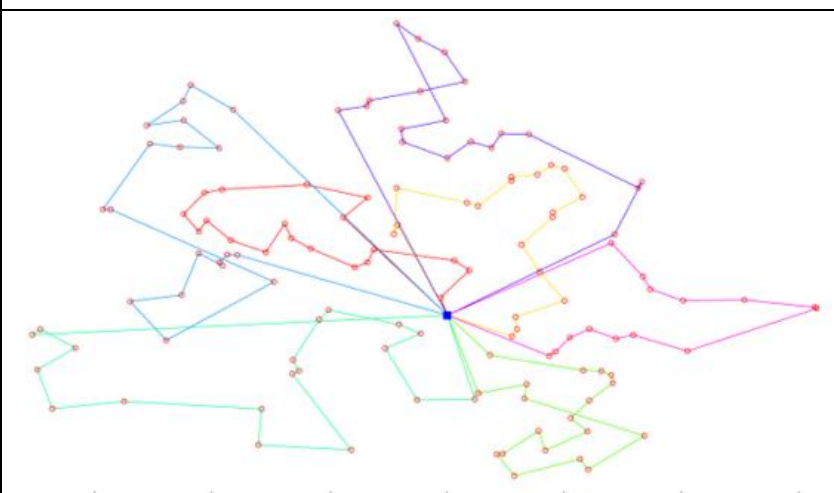
GAACO_BSP ($N_{G_{max}}=80, N_{A_{max}}=100$)				GA	ACO	
연 번	차량 대수	Cost	Solution Time	Cost	Cost	성능 향상
1	1	0.52081	479.623501	0.9582904	0.5670223	8.15%
2	1	0.61178	494.234942	1.101204	0.6751793	9.39%
3	1	0.47476	375.776363	0.759616	0.5177882	8.31%
4	1	0.42855	314.145724	0.6813945	0.4696953	8.76%
5	2	0.59859	569.255298	0.9397863	0.665988	10.12%
6	3	0.61096	723.770098	0.8920016	0.7255195	15.79%
7	2	0.48319	423.799957	0.7344488	0.54145	10.76%
8	2	0.55763	374.126900	0.8587502	0.6291662	11.37%
9	6	0.58674	487.747760	0.7979664	0.7536802	22.15%
10	3	0.65467	389.104747	0.8052441	0.9109086	28.13%
11	3	0.5934	460.971442	0.783288	0.696152	14.76%
12	3	0.39672	285.029512	0.6043776	0.49721	20.21%
13	3	0.45454	267.552078	0.749991	0.5465192	16.83%
14	3	0.40326	206.832362	0.5726292	0.4473211	9.85%
15	4	0.54393	398.472551	0.8485308	0.6609916	17.71%
16	5	0.6049	401.325984	0.804517	0.7382231	18.06%
17	5	0.58551	378.380811	0.878265	0.6465437	9.44%
18	3	0.362	168.078406	0.64074	0.4341569	16.62%
19	7	0.7644	682.448907	1.0286192	0.99441	23.17%
평균	3.05	0.538755	414.7725	0.812614	0.637786	14.7147%
합계	58	9.71553	7401.054	14.48137	11.5509	—

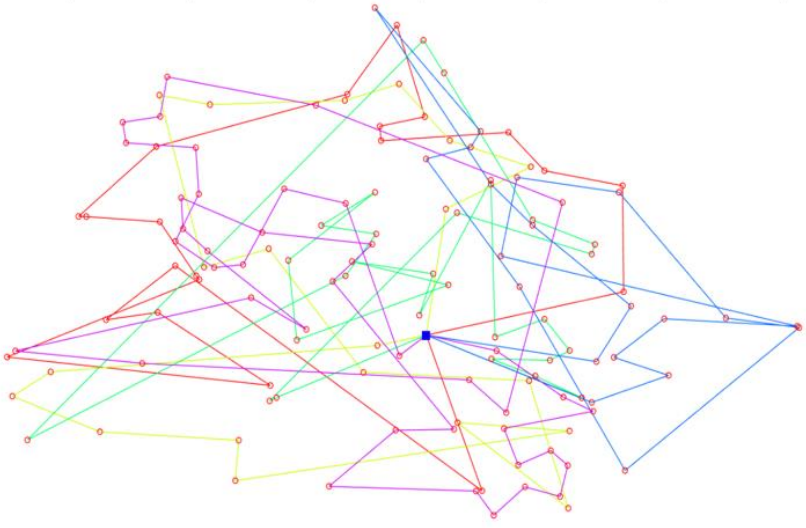
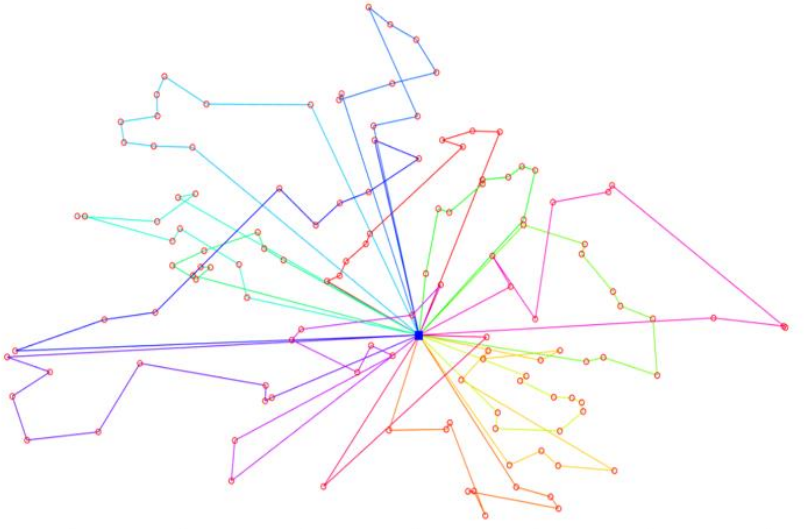
차량 자원 합리적으로 분배하기 위해서 수요 급증에 대응하는

방법은 첨두시에 배송 차량 대수를 증가하는 것이 아니, 배치 간격을 조절해서 완성하는 것이다. 첨두시에 배치 간격 줄일 필요가 있는 지역은 10번째, 12번째, 19번째 있으며, 10번 구역과 19번 구역은 1.5배로 증량 배송하고 12번 구역 2배로 증량하여 자전거 배송업무 수행한다.

한 예시로 구역 19의 알고리즘 수행 결과를 구체적으로 살펴본다(표 5-6). 구역 19는 가장 복잡하고 수요 제일 높은 곳이다. 제시된 알고리즘이 이런 복잡한 재배치 문제 해결할 때도 효과를 잘 나왔다. 시간만에 단일 GA나 ACO 보다 모두 20% 넘어 경로 최적화했다, 노선도도 상대적으로 더 깔끔하게 나왔다.

[표 5-6] 가장 복잡한 구역 19의 재배치 효과 비교

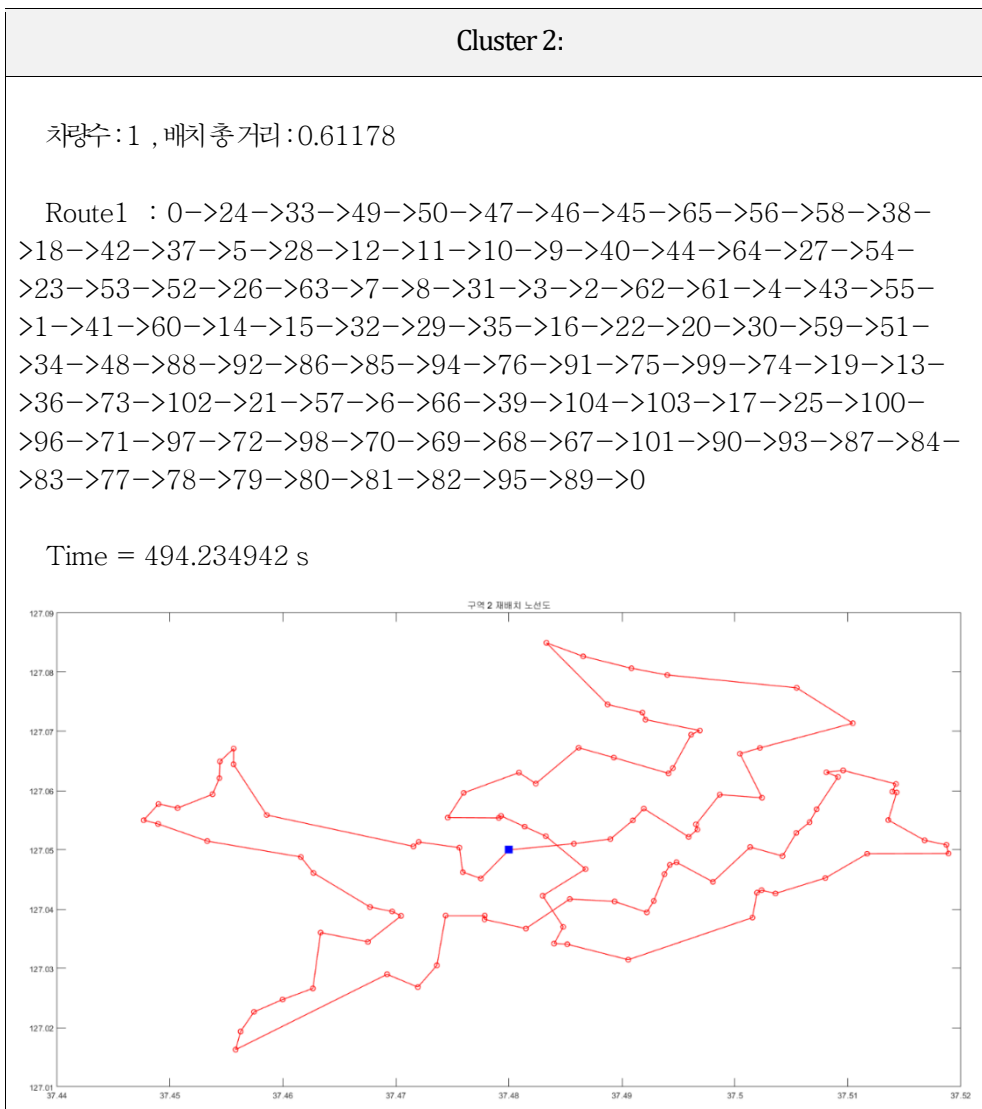
노선도 및 총 거리(Solution Time=682.448907s)	
GAACO _BSP	
Cost=0.7644	

GA	
<p>Cost=1.0286192 (GA 보다 GAACO_BSP의 성능 25.67% 상승)</p>	
ACO	
<p>Cost=0.99441 (ACO 보다 GAACO_BSP의 성능 23.17% 상승)</p>	

결론을 말하자면, 단일 알고리즘보다 본 연구에서 제시한 GAACO_

BSP가 재배치 모델의 성능을 크게 향상시키고 생성된 경로 평균 15.5605%로 최적화하였다. 일부 GAACO_BSP가 생성한 재배치경로도 첨부하였다(표 5-7).

[표 5-7] 일부 생성한 재배치 경로 및 노선도



Cluster 12:

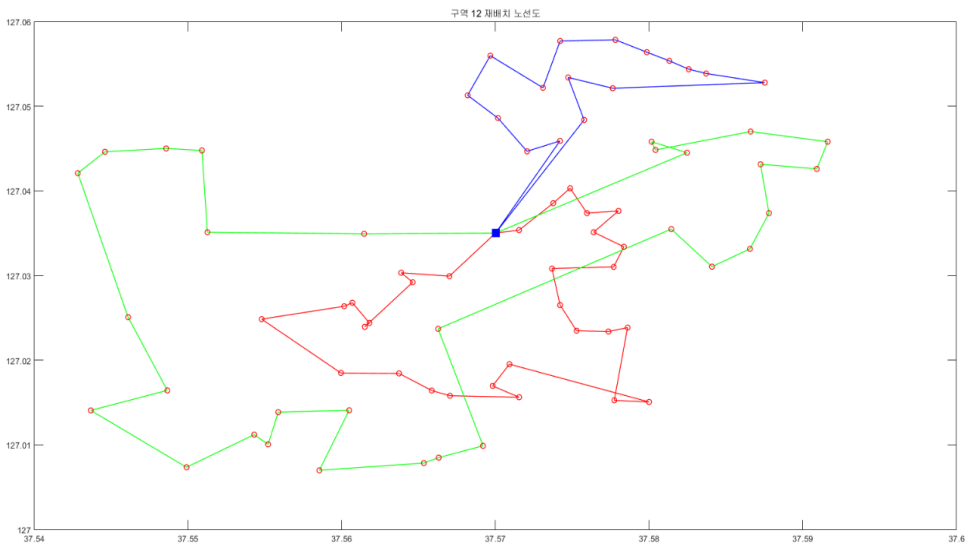
차량수 : 3 , 배치 총 거리 : 0.39672

Route1 : 0->51->24->17->1->29->10->28->9->7->3->8->32->33->58->59->71->66->57->74->69->65->75->46->45->47->44->72->40->53->52->0

Route2 : 0->54->39->48->50->49->42->43->55->56->70->62->68->63->64->67->61->73->60->41->36->34->35->5->26->21->22->30->19->37->23->0

Route3 : 0->16->31->76->18->2->6->12->14->38->4->11->15->20->25->13->27->0

Time = 285.029512 s



Cluster 19:

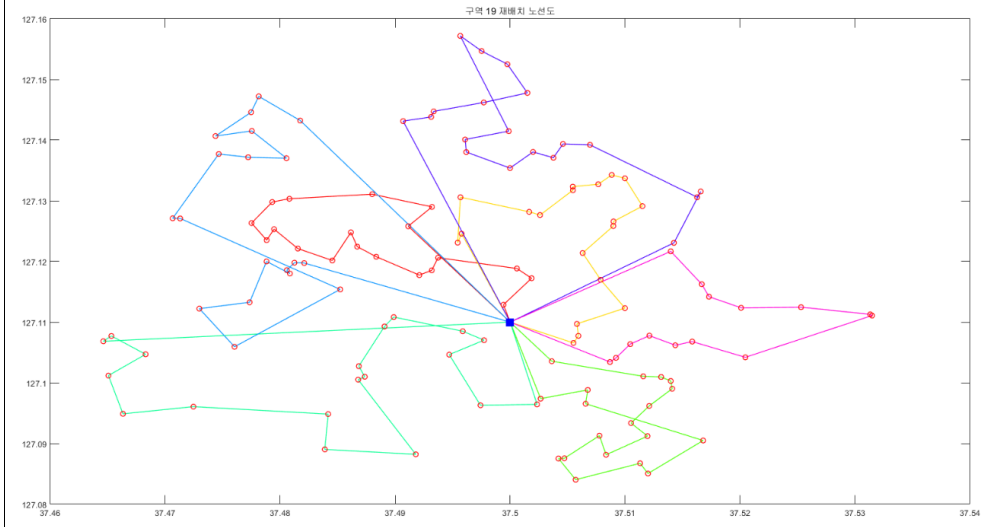
차량수 : 7 , 배치 총 거리 : 0.7644

Route1 : 0->23->93->126->58->107->99->59->52->113->102->80->81->90->22->89->67->79->21->119->0

Route2 : 0->37->38->39->63->118->40->30->97->29->112->31->92->78->83->109->98->108->120->114->0

Route3 : 0->103->51->91->6->4->5->13->3->12->9->7->2->15->1->14->11->62->42->41->0
 Route4 : 0->43->44->46->47->49->48->86->88->27->84->61->105->94->45->8->10->106->55->16->0
 Route5 : 0->17->104->122->54->56->36->96->95->70->117->66->65->82->123->125->24->121->71->25->0
 Route6 : 0->53->64->33->18->35->34->115->19->68->69->87->73->77->111->60->74->72->110->75->0
 Route7 : 0->28->57->124->101->26->50->128->100->127->32->116->85->76->20->0

Time = 682.448907 s



제 6 장 결 론

이번 연구는 BSP 특징을 고려하고 GA와 ACO의 장단점이 서로 보완해 줄 수 있는 특징을 이용하여 GA와 ACO 알고리즘을 조합해서 GAACO_BSP라는 융합 알고리즘을 제시하고 공공자전거 재배치 모델을

구축하였다. 기존 팀을 쓰는 게 아니라, K-means Clustering Algorithm을 이용해서 전체 대여소를 더 합리적으로 재분배하였다.

또한, 전환 최적 시기를 파악하기 위하여 GA 제어함수를 수립하여 두 알고리즘을 동적 조합하였다. GA가 얻은 초기해로 ACO의 Initial Pheromone을 초기화해서 Convergence 속도를 향상시켰다. GA 수행횟수 제어 함수를 수립하는 것을 통해 GA가 ACO로 전환되는 시기 동적 판단하고 GA 단계의 필요 없는 탐색 감소 등 직접적인 성과를 얻었다. 이 외에도 간접적으로 GA와 ACO의 융합 알고리즘 연구 추진하고 재배치 성능을 향상시켜 공공자전거 사업 발전에 이바지를 했다.

실험 결국 보니 전통 단일 GA나 ACO 보다, 제안된 GAACO_BSP 방법은 규모 자전거 시스템에 적용하고 평균적으로 14.7147%의 재배치 경로 품질(quality)을 향상시켰다.

그러나 시간 제한하므로 이번에 단순화된 재배치 모델만 수행해봤다. 더 현실적인 의미 있는 모델을 개발하기 위해서 향후 연구 방향을 정하였다. 우선, 연구 결과를 보니 각 관련 파라미터 값을 선정하는 것은 모델 성능에 큰 영향이 있으므로 향후 파라미터 설치에 관해 연구할 예정이다.

또한, 실제로 공공자전거 재배치 문제를 더 잘 해결하기 위하여 자전거 수요 예측에 대해 더 공부하고 다양한 요소(날씨, 기온, 요일 등)의 영향이 반영된 수요로 재배치 방안 도출할 계획이다.

마지막으로 나중에 모델 일반화 작업을 해볼 계획도 있다. 지금 서울 따릉이 실제 대여 데이터를 기반으로 연구를 수행한 것이며,

파라미터를 수동 조절하는 부분이 많이 존재하다. 때문에 더 넓게 사용할 수 있게 일반화 작업하려고 해서 API를 만들어볼 생각을 들었다. 앞으로 꾸준히 연구하고 더 재미있고 의미 있는 연구결과를 내도록 하기로 했다.

참고 문헌

[국내논문]

엽가혜, and 김인재. "중국에서 공공자전거 사용에 영향을 미치는 요인 분석." 인터넷전자상거래연구 19.4 (2019): 181-198.

정현영, 김규미, and 김현희. "소셜 네트워크 분석과 기계 학습을 적용한 공공자전거 스테이션 활용도 예측 모델." 한국정보과학회 학술발표논문집 (2019): 1254-1256.

이은탁, and 손봉수. "이용수요 기반의 서울시 공공자전거 재배치전략 도출." 대한교통학회지 37.1 (2019): 27-38.

송병준, et al. "공공 빅데이터를 사용한 자전거 교통 시스템 설계 및 활용." 한국컴퓨터정보학회 학술발표논문집 27.1 (2019): 433-434.

임희중, and 정광현. "서울시 공유자전거의 수요 예측 모델 개발." 한국콘텐츠학회논문지 19.1 (2019): 132-140.

구영은. 유전 알고리즘을 이용한 공공 자전거 정적 재배치 문제에 관한 연구. Diss. 한양대학교, 2018.

서영현, et al. "시간대별 확률적 수요 패턴을 고려한 공공자전거 동적 재배치 문제." 대한교통학회 학술대회지 (2017): 570-575.

문현수, and 이영석. "대전시 공공 자전거 (타슈) 공개 데이터 시각화 및 분석." 정보과학회 컴퓨팅의 실제 논문지 22.6 (2016): 253-267.

윤영채, and 조병용. "대전광역시 공공자전거 '타슈' 발전 방안에 관한 연구." 사회과학연구 25.2 (2014): 591-608.

권영훈. "공공자전거 운영을 통한 도시 재생: 창원시 공공자전거 시스템을 중심으로." 서비스경영학회지 15.4 (2014): 171-189.

나호영, and 이상헌. "유전자 알고리즘을 이용한 도시공간형태별 입지선정 모델." 한국시물레이션학회논문지 17.3 (2008): 35-44.

[해외논문]

ZUO X Q, CHEN C, TAN W, et al. Vehicle Scheduling of an Urban Bus Line via an Improved Multiobjective Genetic Algorithm[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 16(2): 1030-1041.

GONG Y C, HUANG R G, YANG X R, et al. Improved Ant Colony Algorithm for Vehicle Scheduling Problems of Military Logistics Distribution[C]// 2010 International Conference on Logistics Systems and Intelligent Management. Harbin: IEEE, 2010: 669-673.

MINOCHA B, TRIPATHI S, MOHAN C. Solving Vehicle Routing and Scheduling Problems Using Hybrid Genetic Algorithm[C]// 2011 3rd International Conference on Electronics Computer Technology. Kanyakumari: IEEE, 2011.

SHUI X G, ZUO X Q, CHEN C, et al. A Clonal Selection Algorithm for Urban Bus Vehicle Scheduling[J]. Applied Soft Computing, 2015,

36: 36–44.

TARANTILIS C D, ZACHARIADIS E E, KIRANOUDIS C T. A Hybrid Metaheuristic Algorithm for the Integrated Vehicle Routing and Three-Dimensional Container-Loading Problem[J]. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(2): 255–271.

LI J P, KWAN R S K. A Fuzzy Genetic Algorithm for Driver Scheduling[J]. European Journal of Operational Research, 2003, 147(2): 334–344.

STEINZEN I, BECKER M, SUHL L. A Hybrid Evolutionary Algorithm for the Vehicle and Crew Scheduling Problem in Public Transit[C]// 2007 IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007.

JOZEFOWIEZ N, SEMET F, TALBI E G. An Evolutionary Algorithm for the Vehicle Routing Problem with Route Balancing[J]. European Journal of Operational Research, 2009, 195(3): 761–769.

FAIZ T I, VOGIATZIS C, NOOR- E- ALAM M, et al. A Column G Algorithm for Vehicle Scheduling and Routing Problems[J]. Computers and Industrial Engineering, 2019, 130: 222–236.

BOYER V, IBARRA-ROJAS O J, RIOS-SOLOLV Y A, et al. Vehicle and Crew Scheduling for Flexible Bus Transportation Systems[J]. Transportation Research Part B: Methodological, 2018, 112: 216–229.

HE F, YANG J, LI M. Vehicle Scheduling under Stochastic Trip Times: An Approximate Dynamic Programming Approach[J]. *Transportation Research Part C: Emerging Technologies*, 2018, 96:144–159.

KLIEWER N, MELLOULI T, SUHL L. A Time– Space Network Based Exact Optimization Model for Multi–Depot Bus Scheduling[J]. *European Journal of Operational Research*, 2006, 175(3): 1616–1627.

ZHENG L J, DONG D C, WANG D Y. A Hybrid Intelligent Algorithm for the Vehicle Scheduling Problems with Time Windows[C]// 17th IEEE International Conference on Intelligent Transportation Systems. Qingdao: IEEE, 2014.

DORIGO M, GAMBARDELLA L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 53–66.

Zheng, L. J., Dong, D. C., & Wang, D. Y. A hybrid intelligent algorithm for the vehicle scheduling problems with time windows. In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 2756–2761). (2014, October). IEEE.

Daun, Weijun, et al. "QoS constraints task scheduling based on genetic algorithm and ant colony algorithm under cloud computing environment." *Journal of Computer Applications* 34.S2 (2014): 66–

69.

Faiz, Tasnim Ibn, Chrysafis Vogiatzis, and Md Noor–E–Alam. "A column generation algorithm for vehicle scheduling and routing problems." *Computers & Industrial Engineering* 130 (2019): 222–236.

Liu, Chun–Yan, Cheng–Ming Zou, and Pei Wu. "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing." *2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*. IEEE, 2014.

Cheng, C. H. E. N., and X. I. N. G. Li–ning. "GA–ACO for solving flexible job shop scheduling problem [J]." *Computer Integrated Manufacturing Systems* 3 (2011).

Deng, Guang–Feng, and Woo–Tsong Lin. "Ant colony optimization–based algorithm for airline crew scheduling problem." *Expert Systems with Applications* 38.5 (2011): 5787–5793.

Wu, Zhangjun, et al. "A market–oriented hierarchical scheduling strategy in cloud workflow systems." *The Journal of Supercomputing* 63.1 (2013): 256–293.

Almeder, Christian, and Lars Mönch. "Metaheuristics for scheduling jobs with incompatible families on parallel batching machines." *Journal of the Operational Research Society* 62.12 (2011): 2083–2096.

Gutjahr, Walter J. "A graph-based ant system and its convergence." *Future generation computer systems* 16.8 (2000): 873–888.

Voss, Stefan. "Book Review: Morco Dorigo and Thomas Stützle: *Ant colony optimization* (2004) ISBN 0–262–04219–3, MIT Press, Cambridge." *Mathematical Methods of Operations Research* 63.1 (2006): 191.

Ergu, Daji, et al. "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment." *The Journal of Supercomputing* 64.3 (2013): 835–848.

Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." *Future generation computer systems* 28.5 (2012): 755–768.

Raviv, Tal, Michal Tzur, and Iris A. Forma. "Static repositioning in a bike-sharing system: models and solution approaches." *EURO Journal on Transportation and Logistics* 2.3 (2013): 187–229.

Ho, Sin C., and W. Y. Szeto. "Solving a static repositioning problem in bike-sharing systems using iterated tabu search." *Transportation Research Part E: Logistics and Transportation Review* 69 (2014): 180–198.

Kim, Gitae, et al. "Solving the dynamic vehicle routing problem under traffic congestion." *IEEE Transactions on Intelligent*

Transportation Systems 17.8 (2016): 2367–2380.

Adham, Manal T., and Peter J. Bentley. "An ecosystem algorithm for the dynamic redistribution of bicycles in London." International Conference on Information Processing in Cells and Tissues. Springer, Cham, 2015.

Bopardikar, Shaunak D., and Vaibhav Srivastava. "Dynamic vehicle routing in presence of random recalls." IEEE Control Systems Letters 4.1 (2019): 37–42.

Xu, Haitao, Feng Duan, and Pan Pu. "Dynamic bicycle scheduling problem based on short-term demand prediction." Applied Intelligence 49.5 (2019): 1968–1981.

Liu, Zhenheng, and Lei Ren. "A Sharing Bike Scheduling Optimization Algorithm Based on Two-Dimensional Dynamic Model and Improved Genetic Algorithm." 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 2018.

Abstract

Optimizing the Bike-sharing
Rebalancing Scheme using A
Novel Fused Genetic-Ant Colony
Optimization Algorithm:
Focusing on Seoul Bike Case

Tong ZUO

Smart City Engineering Major

Department of Civil & Environmental Engineering

The Graduate School

Seoul National University

To improve the service efficiency and customer satisfaction degree of public bicycle, a bike-sharing scheduling model is proposed, which aims to get the shortest length of the bicycle scheduling. To address the slow solution speed of the existing algorithms, which is not conducive to real-time scheduling optimization, this paper designed

a Genetic Hybrid Ant Colony System Algorithm for Solving Bike-sharing Scheduling Problem (GAACS-BSP).

Genetic algorithm was used to search initial feasible scheme , which was used to initialize pheromone distribution of ant colony algorithm. It solved problem of lack initial pheromone, to improve the efficiency of bike-sharing scheduling tasks.

There also proposed a genetic algorithm control function to control the appropriate combination opportunity of the two algorithms. Finally, the results show that compared with GA or ACS, it is more suitable for solving the problem of large-scale bike-sharing scheduling tasks, which shortens the scheduling distance in a short period.

Keywords : Public Bike-sharing Scheduling, Genetic Algorithm, Fusion Model, Shortest Path, Clustering, GAACO-BSP

Student Number : 2020-25957