

# 전기 마이크로-모빌리티 대리 배송을 고려한 차량경로 문제에 대한 발견적 해법

차형주<sup>1</sup> · 김동균<sup>1</sup> · 송보경<sup>2</sup> · 이충목<sup>2\*</sup>

<sup>1</sup>고려대학교 산업경영공학과 / <sup>2</sup>한국외국어대학교 산업경영공학과

## A Heuristic Algorithm for Vehicle Routing Problem with Electric Micro-mobility Delegations

Hyungjoo Cha<sup>1</sup> · DongKyun Kim<sup>1</sup> · Bokyeong Song<sup>2</sup> · Chungmok Lee<sup>2</sup>

<sup>1</sup>Department of Industrial and Management Engineering, Korea University

<sup>2</sup>Department of Industrial and Management Engineering, Hankuk University of Foreign Studies

This paper considers a novel vehicle routing problem with micro-mobility delegations. Recently, micro-mobility vehicles such as electric scooters and bicycles are gaining popularity thanks to advancements in battery and electric motor technologies. Most of the previous studies on the micro-mobility in the literature have been focusing on personal transportation applications. However, few studies investigate the use of the micro-mobility in the logistic context mainly because the micro-mobility has limited travel range and relatively low speed. We propose a combined approach of the micro-mobility and the conventional larger vehicles to mitigate drawbacks of the micro-mobility. Specifically, we assume that micro-mobility can be carried in the vehicles and deployed for the delivery as delegations. A mixed-integer programming formulation is presented, and recognizing the problem's difficulty, we develop an efficient heuristic algorithm. Computational study shows that the proposed approach and algorithm can provide reasonable quality solutions at relatively short times.

**Keywords:** Hierarchical Vehicle Routing Problem, Micro-Mobility, Vehicle Routing Problem with Multiple Depots

### 1. 서론

차량경로문제(Vehicle Routing Problem, VRP)는 물류 분야의 가장 대표적인 최적화 문제 중 하나이다. 일반적으로 하나의 디포(depot)와 여러 개의 고객 노드(customer node)들이 주어질 때, 디포에서 출발해서 모든 고객 노드를 방문하고 다시 디포로 돌아오는 경로들을 유효해(feasible solution)의 집합으로 가지며 이 중에 가장 효율적인 해를 찾는 것을 목표로 한다. VRP는 실제 환경(real-life)의 다양한 제약과 목표를 반영하기 위한 때

우 다양한 변종이 존재한다. 대표적인 예제로는 차량의 용량(capacity)이 주어지는 경우인 Capacitated VRP(CVRP), 고객 노드에 방문해야 할 시간대가 존재하는 경우인 VRP with Time Windows (VRPwTW) 등이 있다. 고객 방문 시간대가 방문 허용 마지막 시간대만 주어지는 경우에는 특별히 VRP with Deadline 라고 불리기도 한다. 일반적인 VRP는 차량의 종류가 하나임을 가정하나 차량의 종류가 여러 가지인 경우에는 Heterogeneous VRP라 불리기도 한다. 이때 차량의 특징은 거의 모든 경우 차량의 용량(capacity)과 비용의 차이로 표현되며 일반적으로 차

본 논문은 2021 춘계공동학술대회(제주, 대한산업공학회)에서 발표한 내용을 기반으로 함. 본 연구는 한국연구재단의 지원을 받아 수행된 연구임 (2021R1F1A1A1048540). 본 연구는 2021학년도 한국외국어대학교 교내학술 연구비의 지원에 의하여 이루어졌음.

\* 연락저자 : 이충목 교수, 17035 경기도 용인시 처인구 모현읍 외대로 81 공학관 530, Tel : 031-330-4378, Fax : 031-330-4120

E-mail : chungmok@hufs.ac.kr

2021년 12월 13일 접수; 2022년 1월 5일, 2022년 1월 6일 수정본 접수; 2022년 1월 10일 게재 확정.

량의 속도는 같다고 가정한다(Toth and Vigo, 2014).

위에 언급된 VRP문제의 다양한 변종은 대부분 트럭과 같이 화석연료를 사용한 짐차(cargo)를 사용함을 가정한다. 최근에 전기 배터리와 모터 기술이 빠르게 발전함에 따라 기존의 전통적인 운송수단과 다른 형태의 운송수단이 등장하고 있다. 특히 전기 자동차(electric vehicle)는 환경오염에 대한 경각심이 커진 점과 운행 비용이 저렴하다는 점에 있어 최근에 VRP 문제에 많이 적용되어 연구되고 있다(Lin *et al.*, 2014). 전기 자동차는 전통적인 차량과 운송 능력 면에서는 비슷하지만 운송거리가 짧고 충전시간이 매우 오래 걸리는 차이점이 있다. 따라서 전기 자동차를 이용한 Electric VRP(EVRP)는 주로 한정된 이동 거리와 충전시간을 고려하는 것이 주된 연구 주제이다. 이와 더불어 최근에 많이 연구되는 새로운 운송수단은 드론(drone)으로 대표되는 무인 항공기(Unmanned Aerial Vehicle, UAV)이다. 드론은 기존 차량과 달리 도로 사정에 영향을 받지 않고 빠르게 화물을 운송할 수 있으나 운행 거리가 매우 짧고 수송 능력도 매우 제한적이다(Jeong *et al.*, 2019). 따라서 드론을 이용한 물류 최적화 문제는 주로 차량과 결합한 형태로 많이 연구되고 있다. 예를 들어 Drone-Truck Routing Problem(Kang and Lee, 2021)은 트럭에 드론을 탑재하여 트럭이 장거리를 이동하고 배송지 인근에서 트럭이 드론의 운행 기지(operation station) 역할을 하게 된다. 따라서 드론의 단점인 배송 거리와 수송 용량의 한계를 기존 차량과의 결합 운영을 통해 해결하였다. 그러나 드론을 이용한 수송은 도심지와 같은 높은 건물이 많은 환경에서는 사용하기 어려우며(Jeong *et al.*, 2019), 일반적인 드론은 한 번에 하나의 비교적

가벼운 물건만 수송할 수 있으므로 드론의 출발지와 배송지 사이를 여러 번 왕복 비행할 필요가 있다. 따라서 드론의 경로는 종종 별 모양(star-shape)의 형태를 가지기 때문에 배송지를 순서대로 방문하는 것보다 더 긴 비행거리가 필요하다.

최근 대규모 아파트 단지 또는 신도시에서는 안전과 환경에 대한 우려로 지상에서의 차량 이동을 금지하고 있다. 또는 복잡한 전통 시장과 같은 지역은 배송 차량이 접근하기가 매우 어렵다. 이러한 지역들은 기존의 차량을 이용한 배송이 어렵고 드론을 통한 배송도 적절하지 않다. 예를 들어 서울의 남대문 시장 같이 매우 혼잡한 지역의 우편 배송은 실제로 우편 차량을 시장 인근에 주차하고 미리 확보한 모터바이크를 사용하여 이루어지고 있다. 이러한 사례는 배송지의 환경에 따라 더 효율적인 배송 수단을 선택적으로 사용할 필요가 있음을 시사한다. 마이크로-모빌리티(micro-mobility)는 전기 스쿠터(electric scooter) 또는 전기 자전거(electric bicycle) 등과 같이 1인이 탑승하며 주로 전기로 작동하는 개인 이동 수단을 의미한다. 마이크로-모빌리티는 일반적인 차량에 비해 비좁고 혼잡한 지역을 비교적 빠르게 이동할 수 있는 장점이 있다. 이 때문에 마이크로-모빌리티는 주로 도심지에서의 개인 이동 수단으로 많이 활용되고 있지만, 물류 수송 수단으로는 거의 사용되지 않고 있는데, 그 이유는 장거리 이동의 어려움과 작은 수송 용량 때문이다. 반면에 마이크로-모빌리티는 드론과 달리 배달원이 직접 운전해서 물품을 전달하기 때문에 신뢰성이 높고 매우 혼잡한 지역에서도 효율적으로 사용할 수 있다. <Table 1>은 차량, 드론, 마이크로-모빌리티를 물류 환경에서 사용할 경우의 특성을 비교한 표이다.

Table 1. Characteristics of Different Transportation Methods

	Conventional vehicle (e.g., truck)	Drone	Micro-mobility
Travel range	Very long	Medium	Short
Transport capacity	Very large	Very small	Small
Speed	Fast (sparse area) Low (dense area)	Fast	Low (sparse area) Fast (dense area)
Cost	High	Medium	Low
Route shape	Tour	Star-shape	Tour
Reliability	High	Low	High

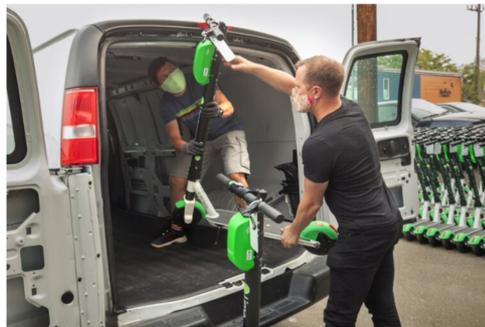


Figure 1. An example of a micro-mobility with storage (left, Way, 2017) and loading the micro-mobility on the vehicle(right, Banner, 2020)

<Table 1>에서 보이듯이 마이크로-모빌리티는 차량(트럭) 또는 드론과 뚜렷이 구별되는 특징을 가지고 있다. 하지만 드론과 마찬가지로 마이크로-모빌리티 단독으로는 거리의 제한과 수송 용량의 한계 때문에 실용적으로 사용하기 어렵다. 반면에 <Figure 1>과 같이 마이크로-모빌리티는 개인이 휴대가 가능할 정도로 크기가 작아 기존의 수송 차량에 탑재할 수 있다. 본 연구는 이에 착안하여 차량과 마이크로-모빌리티의 혼합 차량 경로 문제를 제시하고자 한다. 이때 마이크로-모빌리티는 차량에 실려 장거리를 이동하게 되고 마이크로-모빌리티가 차량보다 효율이 좋은 지역에서는 차량을 정지하고 마이크로-모빌리티를 이용한 대리 수송(delegation)을 수행한다. 드론을 이용한 드론-트럭 경로 문제는 드론의 경로가 트럭의 위치를 중심으로 왕복하는 형태(star-shape)를 가진다. 반면에 마이크로-모빌리티를 이용한 대리 수송은 정지한 차량을 출발/도착으로 하는 마이크로-모빌리티의 경로(tour)를 결정해야 한다. 즉, 마이크로-모빌리티는 차량의 경로를 대리 수행한다고 볼 수 있다. 따라서 본 논문에서는 이러한 문제를 VRP with micro-mobility delegations(VRPwMD)라 부르기로 한다. VRPwMD는 VRP의 일반 경우(general case)이므로 NP-hard에 속하는 문제이다.

<Figure 2>는 일반적인 VRP와 VRPwMD의 해를 비교한 것이다. 일반적인 VRP는 고객 노드들을 모두 차량을 이용해 방문한다. 반면에 VRPwMD는 일부 고객 노드(white nodes)들은 차량을 이용해 방문하고 나머지 노드(gray nodes)들은 마이크로-모빌리티를 이용해 방문한다. 차량과 마이크로-모빌리티의 경로에 모두 속하는 노드를 디커플링 노드(decoupling node)라고 정의하자. <Figure 2>에서는 두 개의 디커플링 노드(a와 b)를 확인할 수 있다. 본 연구의 VRPwMD는 다음과 같은 가정을 사용한다.

**가정**

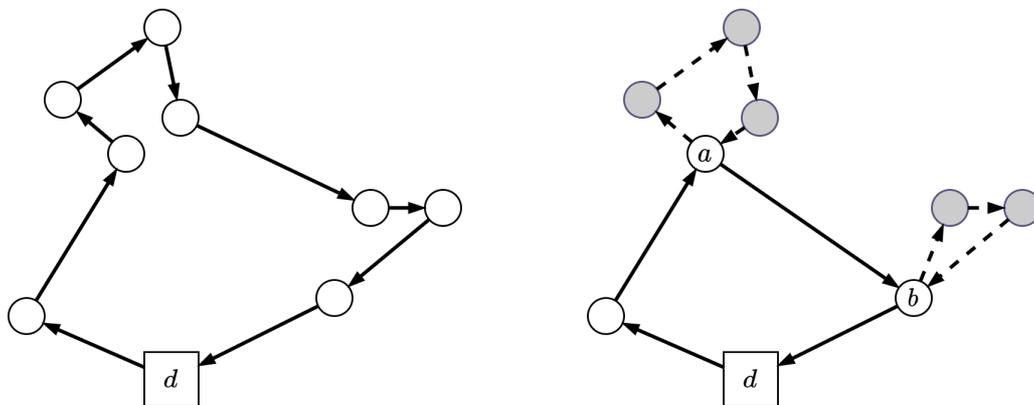
- (A1) 각각의 차량은 하나의 마이크로-모빌리티를 탑재한다.
- (A2) 디커플링 노드는 고객 노드 중에 하나이다.

- (A3) 디커플링 노드에서 차량은 마이크로 모빌리티가 돌아올 때까지 기다린다.
- (A4) 각각의 아크에 대해 차량과 마이크로-모빌리티의 운행 속도는 다를 수 있다.

가정 (A1)은 마이크로-모빌리티의 크기가 차량보다 매우 작기 때문에 가능하다. 또한, 차량에 탑재되어 이송하는 동안 마이크로-모빌리티를 충전하거나 배터리를 쉽게 교환할 수 있다. 가정 (A2)와 (A3)은 차량에서 마이크로-모빌리티를 분리하고 마이크로-모빌리티를 이용한 배송을 할 때 차량은 대기 상태로 있고 마이크로-모빌리티가 배송을 마치고 돌아오면 마이크로-모빌리티를 탑재하고 다음 배송지로 이동함을 의미한다. 가정 (A4)는 차량과 마이크로-모빌리티가 이동 구간에 따라 다른 이동 속도를 가질 수 있음을 의미한다. 예를 들어, <Table 1>에 보이듯이 도로의 밀집도에 따라 차량과 마이크로-모빌리티의 상대 속도는 달라질 수 있다. 즉, 차량과 마이크로-모빌리티는 각각 더 효율적인 지역이 다를 수 있고 VRPwMD는 이 점을 반영하여 총 배송시간을 단축하는 것이 목표이다. 본 논문의 주요 기여점은 다음과 같다.

- VRPwMD 문제를 처음 제시하고 수리 모형을 수립한다.
- VRPwMD를 빠른 시간에 좋은 해를 얻기 위한 메타-휴리스틱(meta-heuristic) 알고리즘을 제시한다.
- 실험을 통해 제안한 문제와 알고리즘이 기존 VRP보다 효율적인 해를 빠른 시간에 찾을 수 있음을 검증한다.

본 논문은 다음과 같은 구조를 가진다. 먼저 제2장에서는 VRPwMD와 관련된 기존 연구들을 살펴본다. 제3장은 VRPwMD를 정확하게 정의하고 이에 대한 혼합정수계획법(Mixed-integer programming) 수리 모형을 제시한다. 제4장은 VRPwMD를 풀기 위한 메타-휴리스틱 알고리즘을 제안하고, 제5장은 제안한 알고리즘의 성능을 검증한다. 마지막으로 제6장에서 결론을 제시한다.



**Figure 2.** Comparison of solutions between classical VRP (left) and VRPwMD (right). Solid arrows represent the route of truck while the dashed arrows show routes of micro-mobility

## 2. 기존 연구

VRPwMD는 두 가지 이상의 다른 운송수단을 사용한다는 점에서 차량과 드론을 동시에 고려하는 문제들과 공통점이 있다. Menglan *et al.*(2019)은 차량과 드론을 동시에 이용한 지역 감시 알고리즘(Vehicle-Assisted Multi-UAV)을 제시하였다. 이 논문의 목적은 주어진 지역을 차량과 드론을 이용하여 감시하는 것으로 드론은 차량을 떠나 주변 지역을 감시하고 다른 위치에서 차량과 합류한다. 이를 반복함으로써 주어진 지역을 모두 가장 작은 시간으로 감시하는 것이 목표이다. 이 문제는 기존에 연구되어 온 Travelling Salesman Problem with Flying Sidekicks(TSPwFS, Murray and Ritwik, 2020)의 연장선에 있는 연구이다. Kang and Lee (2021)는 차량이 정차하고 드론을 주변 배달지로 보내는 경우를 고려하였다. 이때 차량이 정차하는 위치를 대기 노드(waiting node)라 정의하고 대기 노드와 차량은 드론의 한정된 비행거리를 극복하는 역할을 한다. 이들 연구는 일반적으로 제한된 드론의 수송 용량을 고려하여 드론은 최대 하나의 고객을 방문하는 제약이 존재한다.

본 연구가 제시하는 디커플링 노드는 차량과 마이크로-모빌리티의 경로가 교차하는 하위 디포(intermediate depot)의 일종으로 볼 수 있다. Multi-depot VRP (MDVRP, Salhi *et al.*, 2014)는 여러 개의 디포를 이용한 배송문제를 다룬다. 이때 디포는 차량의 한정된 자원(수송능력 또는 연료 등)을 재공급(refurbishment)하는 역할을 한다. 그러나 MDVRP의 경우에는 디포의 위치가 주어지고 모든 차량의 속도도 동일하다고 가정한다. VRPwMD를 MDVRP에 비교하면 디포의 차량의 경로와 위치를 동시에 결정해야 하며 차량의 종류에 따라 다른 이동 속도를 고려해야 하기 때문에 VRPwMD는 MDVRP의 일반 경우(general case)로 볼 수 있다. MDVRP와 비슷한 문제로 Electric VRP(Lee, 2020)가 있다. 이 문제는 전기 자동차와 같이 주행거리가 제한된 차량의 경로를 결정하는 것으로 차량의 충전소(charging station)가 디포 역할을 한다. EVRP가 MDVRP와 다른 점은 전기 자동차를 충전할 때 필요한 긴 충전시간을 고려해야 하는 점이다. 특히 전기 자동차의 충전시간은 선형적이지 않고 비선형(nonlinear)이기 때문에 이 시간으로 고려하기 위해서 특별한 알고리즘이 필요하다.

VRPwMD는 차량을 이용한 경로 생성과 마이크로-모빌리티를 이용한 경로 생성 두 단계로 구성된다. 이렇게 다단계 경로 생성 문제를 주로 Multi-echelon VRP(MEVRP, Perboli *et al.*, 2011)이라고 한다. MEVRP는 공급망 사슬(supply chain network)과 같이 여러 단계가 있는 물류 문제에서 많이 발생한다. VRPwMD는 MEVRP의 관점으로 보면 두 개의 단계가 있는 Two-echelon VRP와 유사하다고 볼 수 있다. 그러나 MEVRP도 MDVRP와 마찬가지로 단계와 단계를 연결하는 노드가 고정되어 주어지고 차량의 속도(노드 사이의 이동 거리)도 모든 차량에 대해 일정하게 주어진다. 중간 디포에서 차량 사이에 화물의 교환을 고려해야 하는 경우에는 특별히 VRP with

cross-docking(VRPwCD, Lee *et al.*, 2006)이라 불린다. VRPwCD는 차량 사이의 교환 시기를 고려해야 하므로 차량 경로 문제와 디포에서의 차량 방문 순서에 대한 스케줄링 문제가 결합한 형태이다.

VRPwMD는 위치(디커플링 노드)와 경로를 동시에 결정해야 한다. 이런 유형의 문제를 Location-Routing Problem(LRP, Schneider *et al.*, 2017)이라고 부른다. 일반적으로 LRP는 디포의 위치와 차량의 경로를 동시에 결정하는 것이 목적이다. 그러나 LRP는 각각의 디포에서의 경로는 독립적으로 수립되며 각각의 차량 사이의 의존관계도 존재하지 않는다. 반면에 VRPwMD는 작은 차량(마이크로-모빌리티)이 큰 차량에 부속되어 사용된다. 예를 들어, VRPwMD에서는 차량이 방문하는 노드 중 디커플링 노드를 제외한 곳에서는 마이크로-모빌리티를 사용할 수 없지만, 일반적인 LRP는 디포의 위치가 정해지면 각 디포에서 사용되는 차량은 다른 디포에서 사용되는 차량에 관계없이 운영될 수 있다.

위에 소개된 연구들은 한 명의 계획 수립자가 모든 결정을 내리는 형태이다. 반면에 여러 명의 계획 수립자가 협력하거나 경쟁하는 경우도 존재한다. 이런 상황을 Cooperative VRP (Coop-VRP, Zibaei *et al.*, 2016)이라고 한다. Coop-VRP는 게임 이론(game theory)에 기반하며 각각의 차량은 각자의 목적에 따라 독립적으로 운행한다. 대부분은 하나의 차량이 하나의 에이전트(agent)가 되며 에이전트들은 협력하거나 경쟁할 수 있다. Otaki *et al.*(2019)은 차량에 해당하는 에이전트가 다른 작은 크기의 에이전트(예: 드론)와 협력하여 배송하는 경우를 다루었다.

VRP 관련 문제들을 푸는 알고리즘은 크게 두 가지로 구분할 수 있다. 첫 번째는 전역 최적해(global optimal solution)를 얻을 수 있는 최적해 알고리즘(exact algorithms)이다. VRP문제의 혼합정수계획법 수리 모형은 나쁜 선형 완화 한계값(LP relaxation bound)을 가지는 것으로 알려져 있다. 따라서 VRP에 대한 최적해 알고리즘은 선형 완화 한계값을 개선하는 방향을 중심으로 이루어졌다. 최적해 알고리즘 중에 가장 많이 사용되는 접근 방법은 분지평가법(branch-and-price)이다. 분지평가법은 Desrochers *et al.*(1992)가 처음 VRP에 대해서 적용하여 좋은 성능을 보였다. 분지평가법의 장점은 열 생성(column generation)으로 많이 사용되는 레이블링 알고리즘(labeling algorithm)을 수정하면 다양한 제약 조건을 고려하기가 비교적 수월하다는 것이다. 분지평가법은 차량의 경로를 하나의 결정 변수로 모형화하므로 차량의 경로를 유효하게 만들기 위한 제약식과 부가적인 결정 변수를 줄일 수 있고, 이는 결국 수리 모형의 선형 완화 한계값을 강하게 만드는 효과가 있다. 분지평가법은 CVRP(Chabrier, 2016), EVRP(Desaulniers *et al.*, 2016, Lee, 2020), MDVRP(Muter *et al.*, 2014), Robust VRP(Lee *et al.*, 2012) 등 다양한 연구에 많이 사용되었다.

최적해 알고리즘은 문제의 크기가 커질수록 최적해를 얻기 힘들어진다. VRP 문제들에 대한 두 번째 접근 방법은 빠른 시

간에 좋은 품질의 해를 얻을 수 있는 휴리스틱 알고리즘을 이용한 것이다. VRP 관련 문제가 다양한 만큼 다양한 휴리스틱이 제안되었다 (Cordeau *et al.*, 2002). 휴리스틱 알고리즘은 크게 두 가지 부류로 구분할 수 있다. 첫 번째는 차량의 경로를 생성하는 휴리스틱이고 두 번째는 이웃해를 통해 해를 개선하는 메타-휴리스틱(meta-heuristic) 알고리즘이다. 전통적인 차량 경로 생성 휴리스틱에는 saving heuristic (Clarke and Wright, 1964), sweep algorithm (Gillett and Miller, 1974) 등이 있다. 메타-휴리스틱 알고리즘은 타부 탐색(tabu-search, Gendreau *et al.*, 1994), 시물레이티드 어닐링(simulated annealing, Osman, 1993), 개미 군집 시스템(ant colony system, Bullnheimer *et al.*, 1999), 유전자 알고리즘(genetic algorithm, van Breedam, 1996), 신경망 알고리즘 (Ghaziri, 1996) 등이 있다.

### 3. 문제 정의 및 수리 모형

VRPwMD의 수리 모형은 차량과 마이크로-모빌리티의 경로를 가능해(feasible solutions)로 가져야 한다. 가능해가 되기 위해서는 차량의 경우 디포에서 출발하고 같은 디포로 돌아와야 하며 마이크로-모빌리티는 디커플링 노드에서 출발하여 같은 디커플링 노드로 돌아와야 한다. 이때 같은 디커플링 노드에서 여러 번의 마이크로-모빌리티의 경로가 출발/도착할 수 있고 수리 모형은 이것을 고려하여야 한다. 일반적인 TSP 또는 VRP의 수리 모형에서 경로는 모든 노드를 한 번만 방문하고 이것을 이용해서 차량의 경로가 부경로(subtour)를 가지지 않도록 하는 제약식을 추가한다. VRP 관련 수리 모형에서 가장 많이 사용되는 부경로 제거 제약식에는 MTZ(Miller *et al.*, 1960) 제약식이 있다. MTZ 제약식은 방문 노드마다 방문 순서의 의미를 가지는 결정 변수를 정의하여 부경로가 발생하지 않도록 한다. 그러나 디커플링 노드와 같이 하나의 노드가 여러 번 방문하는 경우에는 MTZ 제약식을 적용할 수 없다. Bard *et al.* (1998)은 MDVRP의 수리 모형을 위해 디포의 노드를 여러 개 복제하여 MTZ 제약식을 적용하였다. 본 연구에서도 이 방법을 응용하여 수리 모형을 제시한다. 단지 VRPwMD의 경우에는 디커플링 노드가 정해져 있지 않고 디커플링 노드에서 단 하나의 차량 경로와 여러 개의 마이크로-모빌리티 경로가 동시에 존재할 수 있는 상황을 고려하여야 한다. 또 한 가지 고려해야 할 것은 노드를 몇 개로 복제할 것인가이다. 모든 노드가 디커플링 노드가 될 수 있고 노드를 복제하는 수는 하나의 디커플링 노드에서의 마이크로-모빌리티의 경로의 수를 결정한다. 노드 복제 개수를 결정하는 파라미터를  $K$ 로 정의하자.  $K$ 는 최악의 경우 노드의 개수만큼의 값을 가져야 최적해를 보장할 수 있다.

디포 노드 역시 두 개로 복제하여 각각  $s, t$ 로 정의한다. 따라서 차량은 노드  $s$ 에서 출발하여 노드  $t$ 에 도착하여야 한다. 고객 노드의 집합을  $N$ 이라고 하고 임의의 고객 노드  $i \in N$ 을 복사한 노드의 집합을  $N_i$ 로 정의한다.

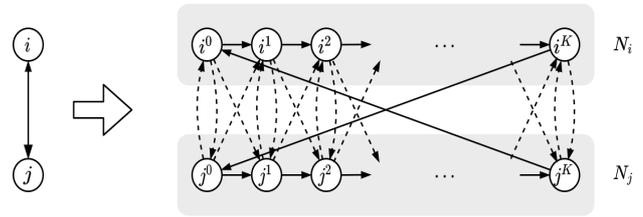


Figure 3. Duplication of Nodes for Possibility of Being Decoupling Nodes

<Figure 3>은 노드를  $K$ 개 만큼 복사하는 것을 나타낸다. 아크  $(i, j)$ 는 복사된 노드들의 집합  $N_i$ 와  $N_j$ 간의 여러 개의 아크로 대체된다. <Figure 3>에서 실선으로 되어 있는 아크들은 모두 차량이 이동 가능함을 나타낸다(차량 아크). 반면에 점선으로 표시된 아크들은 모두 마이크로-모빌리티의 이동 가능함을 나타낸다(마이크로-모빌리티 아크). <Figure 3>의 복잡한 구조는 차량과 마이크로-모빌리티가 모두 하나의 디커플링 노드를 방문할 수 있고 또한 마이크로-모빌리티는 디커플링 노드를 여러 번 방문할 수 있어야 한다는 조건 때문에 필요하다. 결과적으로 <Figure 3>과 같이 구성된 네트워크에서 차량과 마이크로-모빌리티의 경로를 길게 연결한 하나의 경로를 얻을 수 있고 이 하나의 긴 경로는 같은 노드를 두 번 이상 방문하지 않게 된다. 따라서 기존에 많이 사용하는 부경로 금지 제약식을 적용시킬 수 있다. 수리 모형에 필요한 집합들과 파라미터들은 다음과 같다.

#### 집합과 매개변수(Sets and Parameters)

$N$ : 노드들의 집합

$s, t$ : 복사된 디포 노드(출발, 도착)

$N_i := \{i^0, i^1, \dots, i^K\}$ . 노드  $i$ 에서 복사된 노드들의 집합

$\hat{N} := \bigcup_{i \in N} N_i$ . 모든 노드의 집합

$A_i^V := \{(i^k, i^{k+1}) | k = 0, 1, \dots, K-1\}$ . 복사된 노드들 사이를 연결하는 차량 아크 집합

$A^V := \{(i^K, j^0) | i \in N, j \in N, i \neq j\} \cup \{(s, j^0) | j \in N\} \cup \{(i^K, t) | i \in N\}$

두 개의 복사된 노드 집합을 연결하는 차량 아크의 집합

$A^M := \{(i^k, j^k) | i \in N, j \in N, i \neq j, k = 0, 1, \dots, K\} \cup \{(i^k, j^{k+1}) | i \in N, j \in N, i \neq j, k = 0, 1, \dots, K-1\}$ .

두 개의 복사된 노드 집합을 연결하는 마이크로-모빌리티 아크의 집합

$A := (\bigcup_{i \in N} A_i^V) \cup A^V \cup A^M$ . 모든 아크의 집합

$c_{ij}^V$ : 노드  $i$ 에서  $j$ 로 이동할 때 필요한 차량의 이동 시간

$c_{ij}^M$ : 노드  $i$ 에서  $j$ 로 이동할 때 필요한 마이크로-모빌리티의 이동 시간

$cp_i$ : 노드  $i$ 에서 제품의 부피

$wt_i$ : 노드  $i$ 에서 제품의 무게

$L$ : 마이크로-모빌리티로 한 번에 배송할 수 있는 최대 시간  
 $CP$ : 마이크로-모빌리티로 한 번에 배송할 수 있는 최대 부피  
 $WT$ : 마이크로-모빌리티로 한 번에 배송할 수 있는 최대 무게  
 $B$ : 충분히 큰 숫자(big-number)

### 결정 변수(Decision Variables)

$x_{ij}$ : 차량 또는 마이크로-모빌리티가 아크  $(i,j) \in A$ 를 이동하면 1 아니면 0의 값을 가지는 이진 변수  
 $y_i$ : 노드  $i \in N$ 이 디커플링 노드이면 1 아니면 0의 값을 가지는 이진 변수  
 $y'_i$ : 노드  $\hat{i} \in N_i$ ,  $i \in N$ 이 디커플링 노드이면 1 아니면 0의 값을 가지는 이진 변수  
 $u_i$ : 노드  $i \in \hat{N}$ 의 방문 순서를 나타내는 변수  
 $w_i$ : 노드  $i \in \hat{N}$ 를 마이크로-모빌리티를 이용하여 방문했을 때 마이크로-모빌리티로 이동한 총 시간을 나타내는 변수  
 $w_i^p$ : 노드  $i \in \hat{N}$ 를 마이크로-모빌리티를 이용하여 방문했을 때 총 부피를 나타내는 변수  
 $w_i^{wt}$ : 노드  $i \in \hat{N}$ 를 마이크로-모빌리티를 이용하여 방문했을 때 총 무게를 나타내는 변수

VRPwMD의 수리 모형 (P)를 다음과 같이 제시한다.

$$(P) \min \sum_{i,j \in A^V} c_{ij}^V x_{ij} + \sum_{i,j \in A^M} c_{ij}^M x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{(i,j) \in A^V: i=s} x_{ij} = 1 \quad (2)$$

$$\sum_{(i,j) \in A^V: j=t} x_{ij} = 1 \quad (3)$$

$$-y_i + \sum_{(j,\hat{i}) \in A^V \cup A_i^V} x_{j\hat{i}} \leq \sum_{(i,j) \in A^V \cup A_i^V} x_{ij} \quad (4)$$

$$\leq \sum_{(j,\hat{i}) \in A^V \cup A_i^V} x_{j\hat{i}} + y_i, \quad \forall \hat{i} \in N_i, i \in N$$

$$-y_i + \sum_{(j,\hat{i}) \in A^M \cup A_i^V} x_{j\hat{i}} \leq \sum_{(i,j) \in A^M \cup A_i^V} x_{ij} \quad (5)$$

$$\leq \sum_{(j,\hat{i}) \in A^M \cup A_i^V} x_{j\hat{i}} + y_i, \quad \forall \hat{i} \in N_i, i \in N$$

$$-(1-y_i) + \sum_{(j,\hat{i}) \in A} x_{j\hat{i}} \leq \sum_{(i,j) \in A} x_{ij} \quad (6)$$

$$\leq \sum_{(j,\hat{i}) \in A} x_{j\hat{i}} + (1-y_i), \quad \forall \hat{i} \in N_i, i \in N$$

$$\sum_{j \in N: i \neq j} x_{iK_j^0} + x_{iK_t} = \sum_{j \in N: i \neq j} x_{jK_i^0} + x_{st^0}, \quad (7)$$

$$\forall i \in N$$

$$\sum_{(i,j) \in A^V: \hat{i}=i^K} x_{ij} + \sum_{(i,j) \in A^M: \hat{i} \in N_i} x_{ij} \geq 1, \quad (8)$$

$$\forall i \in N$$

$$\sum_{(j,\hat{i}) \in A^V: \hat{i} \in N_i} x_{j\hat{i}} \geq y_i, \quad \forall i \in \hat{N} \quad (9)$$

$$u_i + 1 \leq u_j + B(1 - x_{ij}), \quad \forall (i,j) \in A \quad (10)$$

$$u_i^k \leq u_{i^{k+1}}, \quad \forall (i^k, i^{k+1}) \in A_i^V, i \in N \quad (11)$$

$$\sum_{i \in N_i} y'_i = Ky_i, \quad \forall i \in N \quad (12)$$

$$w_i + c_{ij}^M \leq w_j + B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (13)$$

$$w_i + c_{ij}^M \geq w_j - B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (14)$$

$$w_i + Ly'_i \leq L, \quad \forall i \in \hat{N} \quad (15)$$

$$w_i^p + cp_j \leq w_j^p + B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (16)$$

$$w_i^p + cp_j \geq w_j^p - B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (17)$$

$$w_i^p + CPy'_i \leq CP, \quad \forall i \in \hat{N} \quad (18)$$

$$w_i^{wt} + wt_j \leq w_j^{wt} + B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (19)$$

$$w_i^{wt} + wt_j \geq w_j^{wt} - B(1 - x_{ij} + y'_j), \quad \forall (i,j) \in A^M \quad (20)$$

$$w_i^{wt} + WT y'_i \leq WT, \quad \forall i \in \hat{N} \quad (21)$$

$$w_i = w_{\hat{i}}, \quad \forall i \in \hat{N}, \hat{i} \in N_i, \hat{i} \neq i \quad (22)$$

$$w_i^p = w_{\hat{i}}^p, \quad \forall i \in \hat{N}, \hat{i} \in N_i, \hat{i} \neq i \quad (23)$$

$$w_i^{wt} = w_{\hat{i}}^{wt}, \quad \forall i \in \hat{N}, \hat{i} \in N_i, \hat{i} \neq i \quad (24)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \quad (25)$$

$$y_i \in \{0, 1\}, \quad \forall i \in N \quad (26)$$

$$y_i \in \{0, 1\}, \quad \forall \hat{i} \in \hat{N} \quad (27)$$

$$u_i \geq 0, \quad \forall i \in \hat{N} \quad (28)$$

$$w_i \geq 0, \quad \forall i \in \hat{N} \quad (29)$$

$$w^{CP_i} \geq 0, \quad \forall i \in \hat{N} \quad (30)$$

$$w^{WT_i} \geq 0, \quad \forall i \in \hat{N} \quad (31)$$

목적 함수 (1)은 차량과 마이크로-모빌리티의 총 이동 시간의 합을 의미한다. 제약식 (2)와 (3)은 각각 출발 디포와 도착 디포 노드에서 출발/도착해야 함을 의미한다. 제약식 (4)-(6)은 노드가 디커플링 노드 여부에 따라 노드에서의 흐름 균형(flow balance)을 다르게 적용한다. 제약식 (4)와 (5)는 해당 노드가 디커플링 노드가 아니면 각각 차량과 마이크로-모빌리티에 대해서만 흐름 균형 제약식을 부여한다. 즉, 디커플링 노드가 아니면( $y_i = 0$ ) 해당 노드에서는 차량 또는 마이크로-모빌리티만이 방문해야 한다. 특히 제약식 (5)에서는 마이크로-모빌리티의 한 노드의 복사 노드 간 이동을 차량 아크를 사용하여 이동하게 된다. 만약 해당 노드가 디커플링 노드이면( $y_i = 1$ ) 제약식 (6)에 따라 차량과 마이크로-모빌리티 간의 전환이 가능하다. 제약식 (7)은 해당 노드가 차량과 마이크로-모빌리티가 동시에 방문하는 경우를 제한한다. 제약식 (8)은 모든 노드가 차량 또는 마이크로-모빌리티로 방문해야 하는 것을 표현한다. 제약식 (9)는 해당 노드의 차량 방문 여부에 따른 디커플링 노드 여부를 결정해 주는 기능을 한다. 제약식 (10)은 부경로(subtour)를 제거하는 MTZ 제약이다. 복제된 노드 사이의 부경

로를 방지하는 제약식은 제약식 (11)에서 표현된다. 제약식 (12)는 디커플링 노드를 결정하는 변수  $y_i$  값을 노드  $i$ 의 복사 노드들에 동일한 값을 할당하도록 결정 변수  $y'_i$ 의 값을 설정해 준다. 마이크로-모빌리티는 한 번에 정해진 시간  $L$ 보다 오래 배송을 할 수 없고, 이 조건은 제약식 (13)과 (14), (15)에 표현된다. 또한 마이크로-모빌리티를 통해서 한 번에 배송을 할 수 있는 최대 용량인 CP와 최대 무게인 WT가 존재한다. 각각 CP와 WT를 만족해야 하는 조건은 제약식 (16)~(18), (19)~(21)에 표현된다. 특히, 제약식 (13)~(21)은 디커플링 노드에서 마이크로-모빌리티가 한 번 나가는 경우 누적값이 0으로 재설정됨을 확인할 수 있다. 제약식 (22)~(24)는 복사 노드의 이동 시간, 누적 배송 용량 및 무게를 같게 해줌을 의미한다. 제약식 (25)~(27)은 결정 변수  $x_{ij}$ ,  $y_i$  그리고  $y'_i$ 의 이진 조건을 부여하며, 제약식(28)~(31)은 결정 변수  $u_i$ ,  $w_i$ ,  $w_i^{CP}$ , 그리고  $w_i^{WT}$ 의 비음 조건을 부여한다.

#### 4. 알고리즘

VRPwMD의 수리 모형 (P)는 이진 결정 변수가 많고 선형 완화 한계값이 좋지 않기 때문에 매우 작은 크기의 문제에 대해서만 최적해를 구할 수 있다. 수리 모형 (P)의 성능은 5장 전산 실험에서 더욱 자세히 보고한다. 본 연구에서는 비교적 좋은 해를 빠르게 얻기 위한 휴리스틱 알고리즘을 제시한다.

본 연구가 제시하는 알고리즘은 크게 두 가지 계층(phase)으로 구성된다. <Figure 4>는 제시하는 알고리즘의 대략적인 구조를 보여준다. 첫 번째 계층은 해를 생성하는 계층(solution construction phase)으로 3 단계의 휴리스틱으로 구성된다. 첫 번째 단계에서는 고객 노드 중에서 디커플링 노드의 목록을 선택한다. 두 번째 단계에서는 선택된 디커플링 노드를 중심으로 마이크로-모빌리티를 이용해서 배송할 노드의 집합을 할

당한다. 세 번째 단계에서는 각각의 디커플링 노드와 할당된 노드 집합에 대해 마이크로-모빌리티의 경로를 생성하고 디커플링 노드들을 모두 방문하는 차량의 경로를 생성한다. 이와 같은 과정을 거쳐 생성된 해는 해 개선 계층(solution improvement phase)에서 유전 알고리즘(genetic algorithm, GA)과 타부-탐색(Tabu-search, TS)을 이용하여 개선된다. 즉, 3단계 해 생성 계층과 GA, TS를 통한 해 개선 계층으로 구성된다. 본 연구에서는 두 메타 휴리스틱 방법론이 독립적으로 사용된 것이 아니라 GA의 연산을 통해 도출된 해를 TS가 추가 발전시키는 방법으로 연결되어 사용되었다. VRPwMD 문제의 경우 해가 디커플링 노드에 따라 매우 변동이 크기 때문에 해 개선 계층에서 GA와 TS가 연결되어 구성되는 것이 효과적이다. 즉, GA 연산으로 산출된 해를 TS에서 사용하는 교환(swap), 삽입(insertion), 삭제(deletion) 연산을 통해 개선을 효과적으로 할 수 있다.

#### 4.1 해 생성(Solution Construction Phase)

하나의 해를 생성하기 위해서는 디커플링 노드들을 선택한 후 차량과 마이크로-모빌리티가 방문할 노드들을 할당하고 각각의 경로를 생성해야 한다. 이 과정을 3단계 휴리스틱으로 정의하였다.

##### 4.1.1 디커플링 노드 선택

디커플링 노드를 결정하기 전에 먼저 디커플링 노드의 개수를 결정할 필요가 있다. 본 연구에서는 군집화(clustering) 알고리즘을 이용하여 먼저 디커플링 노드의 개수를 정하였다. 직관적으로 마이크로-모빌리티는 비교적 좁은 지역에 밀집해 있는 노드들을 방문하는데 효율적임을 알 수 있다. 본 연구에서는 이에 착안하여 밀도 기반 군집화(DBSCAN, Ester *et al.*, 1996) 방법과, K-중앙자 군집화 방법(K-medoids)을 적용하였다.

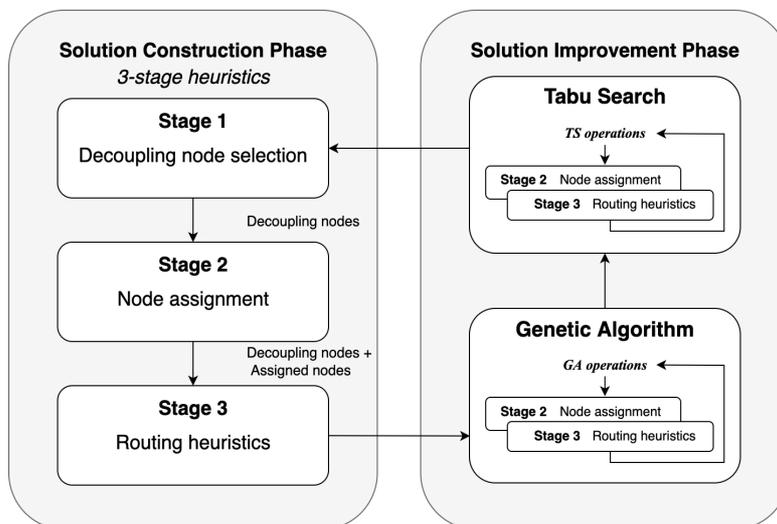


Figure 4. Overview of Proposed Algorithm

밀도 기반 군집화 방법은 군집 내 최소 요소 수(minimum sample)와 요소 간 거리를 나타내는  $\epsilon$ (epsilon) 값을 입력 파라미터로 주어야 한다는 단점이 있으나, 밀도 기반으로 군집을 구성함으로써 타 군집화 알고리즘과 달리 기하학적인 모양을 갖는 군집을 찾는 데 유리하다고 알려져 있다(Jiang *et al.*, 2011). DBSCAN은 하나의 군집으로 묶이기 위해 군집 안에 들어가야 하는 최소 데이터 포인트 수인 minimum sample과 데이터 포인트 간 최대 거리인  $\epsilon$ 를 설정해 주어야 한다. 본 연구에서는 뒤에서 후술할 Solomon, Augerat instance에 대해 상대적으로 데이터 포인트가 산재되어 있는 R-class와 P-class에 대해 minimum sample = 2,  $\epsilon = 0.8$ 을, 그 외에는 minimum sample = 2,  $\epsilon = 0.4$ 로 설정하였다. 밀도 기반 군집화 방법의 결과로 나온 군집의 개수를  $C$ 라고 정의하자. 각 군집별로 하나의 노드가 선택되어 총  $C$ 개의 노드 정보가 다음에 설명할 K-중앙자 군집화 방법론의 입력값으로 사용된다.

K-중앙자 군집화 방법은 기존의 군집화 알고리즘과 그 맥락을 같이한다. 대표적인 군집화 알고리즘으로 K-평균(K-means, MacQueen *et al.*, 1967) 방법론이 있는데, 해당 방법론은 K개의 군집 중심을 데이터 오브젝트 값의 평균값으로 찾게 된다. 그러나 K-평균 방법론은 극도로 크거나 작은 이상치에 매우 민감하고, 기존 데이터 값이 아닌 임의의 점을 군집의 중심으로 본다는 점에서 K-중앙자 군집화 방법론과 차이를 가진다. K-중앙자 군집화 방법은 각 군집에서 대표 객체(medoids)를 임의로 찾음으로써  $n$  개의 객체 중 K 개의 군집을 찾는 방법이다. 따라서 K-평균 군집화 방법보다 이상치에 대해 덜 민감하며, 군집의 중심이 객체 중 한 점으로 결정된다. K-중앙자 군집화 방법론은 앞서 설명한 밀도 기반 군집화 방법의 군집 개수  $C$ 와 해당 네트워크에서 지정 가능한 최대 디커플링 노드 수  $C_{max}$ 의 차이에 따라 다른 개수의 군집화 결과를 도출하게 된다.  $C_{max}$ 는 모든 디커플링 노드에서 마이크로-모빌리티가 한 번씩만 나가는 경우를 의미하며, 해당 네트워크 노드 수의 절반으로 계산한다.  $C$ 와  $C_{max}$ 의 차이가 3 이상이면  $C$ 와  $C_{max}$  사이의 2개의 값을 임의로 추가 선택하여  $C$ 와  $C_{max}$ , 추가로 선택된 2개의 값까지 총 4개의 군집 개수 후보에 대하여 군집화 결과를 도출하게 된다. 만약,  $C$ 와  $C_{max}$ 의 차이가 3보다 작으면  $[C, C_{max}]$ 의 범위에 속하는 모든 정수 값들이 군집 개수 후보가 되며 이들에 대하여 군집화 결과를 도출하게 된다. 여기서 디커플링 노드의 수를 늘리는 것은, 마이크로-모빌리티의 대리 배송이 일어나는 것이 해 개선의 관점에 있어 좋은 행위이며,  $C$ 개의 디커플링 노드를 상한값(upper bound)로 사용하기 위함이다. 이처럼 밀도 기반 군집화 방법과 K-중앙자 군집화 방법을 연속적으로 사용한 이유는 각 알고리즘이 다른 성향의 해를 찾는 경향성을 갖고 있기 때문이며, 이는 추후 단계에서 진행되는 메타 휴리스틱 알고리즘의 탐색 영역(search space)을 넓히는 역할을 수행한다.

위 과정을 거쳐 찾은 군집화 결과들은 후술할 유전 알고리즘으로 평가를 거치게 된다. 유전 알고리즘은 주어진 군집화

결과에 대해 100번의 세대 진화를 통한 해 탐색을 거쳐 각 군집화 결과에 대한 가능해(feasible solution)를 준다. 그 후 해당 결과를 비교하여 가장 좋은 해에 대해 최종적인 시작해(initial solution)로 간주하고, 이후 단계를 진행한다.

자세한 알고리즘 흐름은 다음과 같다.

- Step 1. 노드 집합  $N$ 의 모든 노드들의 수평, 수직 위치 좌표에 대해 최솟값은 0, 최댓값은 1로 설정한 후 이를 기준으로 모든 좌표값을 0과 1사이의 값으로 변환(scaling) 한다.
- Step 2. 변환된 노드들의 위치를 기준으로 DBSCAN 군집화를 시행한다.
- Step 3. 각 군집별로 임의의 한 노드를 선택한다.
- Step 4. DBSCAN 군집화 결과를 바탕으로 K-중앙자 군집화 방법을 시행한다. DBSCAN 군집화 결과 군집의 개수가 0이 나온 경우, 차량만을 사용한 경로를 주고 알고리즘을 마친다.  $C$ 와  $C_{max}$ 의 차이가 3 이상일 경우  $C$ 와  $C_{max}$ 를 포함한 총 4개 군집에 대한 해를 탐색한다. 만약  $C$ 와  $C_{max}$ 의 차이가 3보다 작은 경우, 범위 내의 가능한 군집 개수에 대한 해를 탐색한다. 해를 탐색할 때는 배송 노드 할당과 차량/마이크로-모빌리티 경로 생성의 과정을 거치며 해당 방법론은 아래에서 자세히 설명할 예정이다.
- Step 5. Step 4에서 구한 군집화 결과에 대해 유전 알고리즘의 해 탐색 과정을 거친다. 해당 과정으로 찾은 각 군집별 해 중 가장 좋은 값을 갖는 군집의 개수가 최종 군집 개수가 된다. 해당 군집 수에 대해 임의로 노드의 수를 추출하여 추후 단계를 수행한다.

위에 설명한 군집화 과정은 마이크로-모빌리티로 운송할 지역이 서로 인접하고 있다는 가정에 기반하고 있다. 하지만 경우에 따라 마이크로-모빌리티를 사용해야 하는 지역이 미리 주어지거나 밀집 지역이 아닌 지역에 마이크로-모빌리티를 사용해야 하는 경우가 있을 수 있다. 이러한 경우에는 군집화 과정에서 노드 간의 거리를 단순한 공간상의 위치를 통해 계산하는 것이 아니라 마이크로-모빌리티의 실제 이동시간으로 선처리(preprocessing)과정을 통해 대체하면 동일하게 적용할 수 있다. 즉, 군집화 과정에서 노드 간의 거리는 공간상의 거리가 아니라 마이크로-모빌리티로 인한 효율 증가가 반영된 논리적인 거리라고 생각할 수 있다. 단, 이러한 경우에 대한 고려는 배송 대상 지역의 특성에 따라 달라지기 때문에 본 논문에서는 가장 일반적인 경우로 가정하여 알고리즘을 기술한다.

#### 4.1.2 배송 노드 할당

주어진 디커플링 노드에 마이크로-모빌리티로 배송할 노드들을 할당하는 단계이다. 중복된 디커플링 노드가 있을 수 있으므로 편의상 (중복을 허용하는) 디커플링 노드의 집합을

$DN$ 으로 정의하자. 또한 중복을 제거한 디커플링 노드의 집합을  $\widehat{DN}$ 이라고 정의한다. 미할당 노드의 집합  $\widehat{N} := N \setminus \widehat{DN}$ 를 정의한다.

- Step 1. 주어진 디커플링 노드들 중에 아직 배송 노드가 할당되지 않는 노드  $dn \in DN$ 을 선택하고  $DN \leftarrow DN \setminus \{dn\}$ 로 업데이트한다. 디커플링 노드  $dn$ 에 할당된 노드 집합을 초기화한다( $N_{dn} \leftarrow \emptyset$ ).
- Step 2. 만약  $\widehat{N} = \emptyset$  이면 종료. 아니면  $\widehat{N}$ 의 노드들 중에  $dn$ 과 마이크로-모빌리티 이동시간 기준으로 가장 가까운 노드  $i$ 를 선택한다.
- Step 3. 만약  $N_{dn} \cup \{i\}$ 의 노드들의 총 부피와 무게가 모두 최대 부피  $C$ 와 최대 무게  $W$ 보다 작으면  $N_{dn} \leftarrow N_{dn} \cup \{i\}$ ,  $\widehat{N} \leftarrow \widehat{N} \setminus \{i\}$ 로 업데이트하고 Step 2로 이동한다.
- Step 4. 만약  $DN = \emptyset$  이면 종료하고 아니면 Step 1로 이동한다.

이 과정을 통해 각각의 디커플링 노드에 할당된 노드들은 모두 마이크로-모빌리티의 부피 제한  $C$ 와 무게 제한  $W$ 를 만족한다. 하지만 이동시간 제한  $L$ 은 만족하지 않을 수 있다. 해당 경우는 불가능 해로 단일 차량에 의한 경로 값을 반환한다.

#### 4.1.3 차량/마이크로-모빌리티 경로 생성

이 단계에서는 두 가지 경로를 생성한다. 첫 번째는 차량의 경로로 이 경로는 디포를 출발/도착으로 하고 디커플링 노드에 할당되지 않은 모든 노드들을 방문하는 경로이다. 이 경로는 결국 차량이 방문하는 노드들의 순서가 된다. 두 번째는 각각의 디커플링 노드와 그 노드에 할당된 고객 노드들에 대한 마이크로-모빌리티 배송 경로이다. 이때는 디커플링 노드가 출발/도착 노드가 되고 할당된 고객 노드들은 모두 방문하는 경로를 생성한다. 즉, 두 가지 경우 모두 출발/도착 노드가 정해지고 주어진 노드의 집합을 가장 빠른 시간대에 방문하는 외관원 문제(TSP)가 됨을 알 수 있다. 단지 차량과 마이크로-모빌리티의 이동시간이 다르기 때문에 차량의 경로는 임의의 아크  $(i, j)$ 에 대한 차량의 이동시간  $c_{ij}^V$ 을 기준으로 하고, 마이크로-모빌리티는 마이크로-모빌리티의 이동시간  $c_{ij}^M$ 을 기준으로 수립하여야 한다.

출발/도착 노드와 아크 간의 이동시간 정보가 주어지는 TSP문제를 생각해 보자. 본 연구에서는 주어진 TSP문제를 풀기 위해 Google ORTools(Perron and Furnon, 2019)를 사용하였다. ORTools는 google이 공개한 최적화 solver로 선형계획법, 정수계획법 등에 대한 공개 소프트웨어를 포함하고 있다. 또한 TSP문제에 대한 기존에 알려진 휴리스틱들도 구현해서 제공하고 있다. ORTools가 제공하는 경로 관련 휴리스틱에는 greedy, saving, guided local search, simulated annealing 등이 있다. 본 연구의 실험에서는 ORTools에서 제공하는 RoutingModel()을 사용하였으며 모든 휴리스틱 파라미터는 디폴트 값을 사용하였다.

마이크로-모빌리티 경로의 경우 경로 이동시간 제한  $L$ 을 넘는 경로가 생성될 수 있다. 이 경우에는 해당 디커플링 노드  $dn$ 에 할당된 노드의 집합  $N_{dn}$ 을 변경해야 한다.  $q := |N_{dn}|$ 로 정의하자. 다음 알고리즘은  $q$ 의 크기를 차례로 줄이면서 마이크로-모빌리티 경로 이동시간 제한  $L$ 보다 작은 경우를 찾는다.

- Step 1.  $N_{dn}$ 에서  $q-1$ 개의 노드를 뽑아 만들 수 있는 모든 조합에 대해 디커플링 노드에서 출발/도착하는 경로의 마이크로-모빌리티 이동시간을 구한다. 이렇게 구해진 모든 이동시간들 중에 최대 이동시간 제한값  $L$ 보다 작은 이동시간들의 집합을  $T_{q-1}$ 로 정의한다. 각각의 이동시간  $t_k \in T_{q-1}$ 에 대한 배송 노드들의 집합을  $N_{dn}^k$ 라고 정의한다.
- Step 2.  $|T_{q-1}| = 0$  이고  $q > 2$  라면  $q \leftarrow q-1$ 로 설정하고 Step 1로 이동한다.
- Step 3.  $|T_{q-1}| = 1$  이라면  $t_k \in T_{q-1}$ 에 대해  $N_{dn} \leftarrow N_{dn}^k$ 으로 변경하고 종료한다.
- Step 4.  $|T_{q-1}| > 1$  이면  $T_{q-1}$ 의 모든 원소들에 대해서 다음과 같이 확률값을 부여하여 임의로 선택한다.
- Step 4-1. 할당된 노드 집합이 추출될 확률  $p_k$ 를 다음과 같이 정의한다.

$$p_k = \frac{t_{\max} - t_k}{T_{diff}}, \quad \forall t_k \in T_{q-1}$$

$$\text{이 때, } T_{diff} = \sum_{k \in T_{q-1}} (t_{\max} - t_k) \text{ 그리고}$$

$$t_{\max} = \max_{t_k \in T_{q-1}} t_k \text{ 로 정의된다.}$$

- Step 4-2. 주어진 확률값에 따라서 뽑힌  $N_{dn}^k$ 으로  $N_{dn}$ 을 변경한다.

위의 과정을 통해 바뀐  $N_{dn}$ 이 마이크로-모빌리티의 최종 할당 그룹이고 여기서 구해진 경로가 최종 경로가 된다. 모든 디커플링 노드에 대해 위의 과정을 수행하고 나머지 노드들(디커플링 노드와 마이크로-모빌리티 배송 노드를 제외한 고객 노드)에 대해 차량 경로를 생성한다.

## 4.2 해 개선(Solution Improvement Phase)

해 생성에서 좋은 해를 얻기 위해서는 디커플링 노드를 잘 선택하는 것이 중요하다. 해 개선 과정은 다양한 디커플링 노드의 조합을 시도하여 더 좋은 해를 얻는 것이 목표이다. 디커플링 노드의 모든 조합을 시도하는 것은 조합의 수가 기하급수적으로 많기 때문에 효율적이지 못하다. 좋은 조합의 디커플링 노드들을 빠른 시간에 찾기 위하여 본 연구는 두 가지의 메타-휴리스틱 알고리즘을 적용하였다.

### 4.2.1 유전 알고리즘을 이용한 해 개선

유전 알고리즘(GA)은 생물의 진화를 모방하여 최적해를 구



Figure 5. Definition of a Chromosome

하는 메타 휴리스틱 기법으로 세대를 거듭할수록 좋은 해를 계속 만들어나가는 식으로 진행된다. 본 연구에서는 GA를 적용하기 위해서 다음과 같이 유전자를 정의하고 두 개의 유전자 사이의 교차(Crossover) 방법과 돌연변이(Mutation) 방법을 수립하였다(Whitley, 1994).

**염색체(Chromosome)**

유전 알고리즘에서 염색체(chromosome)는 다음 <Figure 5>와 같은 형태로 정의된다. 하나의 염색체에는 모두  $G$ 개의 유전자(Gene)가 있으며 각각의 유전자는 디커플링 노드를 의미한다.

**적합도(fitness)와 유전자 선택**

염색체의 적합도(fitness)는 염색체에 정의된 디커플링 노드들을 사용해 해 생성 휴리스틱(4.1장)으로 경로 생성을 완료한 해(solution)의 총 배송시간으로 정의하였다. 유전자 알고리즘은 많은 수(예: 100개 이상)의 염색체를 생성하고 염색체들 중에 적합도가 높은 염색체를 선택하여 교차시켜 새로운 염색체 집단(pool)을 생성하는 것을 반복한다. 본 연구에서는 현재의 염색체 집단에서 좋은 염색체들을 추출하기 위해 많이 사용되는 룰렛 선택(roulette selection)방법을 사용하였다. 모든 염색체의 집합을  $P$ 라고 하고, 염색체  $c \in P$ 의 적합도를  $f_c$ 로 정의하자. 염색체  $c$ 가 선택될 확률은 다음과 같이 정의한다.

$$p_c = \frac{f_{\max} - f_c}{F}, \quad \forall c \in P$$

이때  $F = \sum_{c \in P} (f_{\max} - f_c)$  이고  $f_{\max} = \max_{c \in P} f_c$ 이다. 염색체  $c$ 에 대한 확률  $p_c$ 를 모든 염색체의 집합  $P$ 에 더하면 1이 됨을 쉽게 알 수 있다. 만약 모든 염색체가 같은 적합도를 가지면  $p_c = \frac{1}{|P|}$ 로 정의한다.

**교차(crossover)**

유전 알고리즘에서 해의 다양성을 보장해 주는 방법 중 하

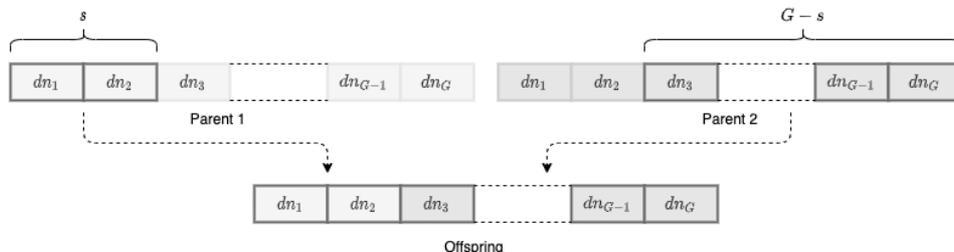


Figure 6. Crossover of Chromosomes

나인 교차(Crossover)는 단순 교차, 복수점 교차, 일정 교차 등 다양한 방법이 존재한다. 본 알고리즘에서는 단순 교차(Simple One Point Crossover)를 사용하였다, 두 개의 부모 염색체에 대해 교차는 다음과 같이 이루어진다. 먼저  $1 \leq s \leq G$ 인 임의의  $s$ 를 기준으로 첫 번째 부모 염색체에서 앞부분  $s$ 개의 유전자를 가져오고 두 번째 부모 염색체에서는 뒷부분  $G-s$ 개의 유전자를 가져와 결합한다(<Figure 6> 참조).

**변이(mutation)**

변이란, 유전자를 일정한 확률로 변화시키는 조작이다. 변이는 돌연변이가 생성이라고도 하며, 부모해가 가지지 못하는 성질을 부여하여 탐색 범위를 확보하는 역할을 담당하게 된다. 변이를 위해 해당 염색체의 임의의 유전자를 골라 이 유전자의 값을  $N$ 의 임의의 원소로 바꿔주었다. 이때 바뀐 유전자는 기존 디커플링 노드와 중복될 수도 있으며 이 경우에는 같은 디커플링 노드에서 여러 번의 마이크로-모빌리티 경로가 생성됨을 의미한다. <Figure 7>은 GA의 흐름도를 보여준다.

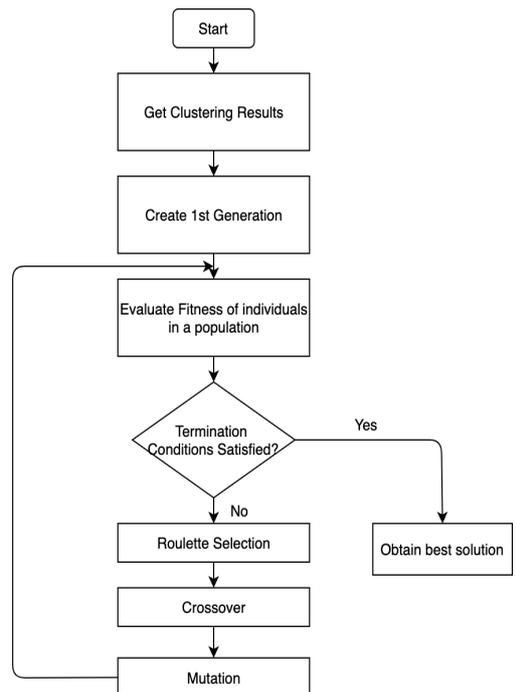


Figure 7. Flowchart of Genetic Algorithm for Decoupling Node Selection

4.2.2 타부-탐색 알고리즘을 이용한 해 개선

앞서 설명한 유전 알고리즘은 최적 경로에 대한 상한 값을 주게 되며, 변이 과정이 있지만 지역 최적점(local minimum)에 빠질 수 있다. 반면에, 타부-탐색은 인간의 기억 과정을 모방하여 개발된 메타-휴리스틱 기법으로, 타부 목록(Tabu list)을 통해 지역 최적점(local minimum)을 회피할 수 있는 것이 큰 장점이다(Glover, 1990). 타부-탐색은 새로운 해를 만들 때 타부 조건이라는 기억을 유지하여 이미 고려했던 열등한 해를 고려하지 않도록 하는 동시에 열망 조건(aspiration level)이라는 기억 장치를 통해 우수한 해의 등장 가능성을 높이며, 단기기억과 장기기억을 통해 해의 이동을 제한하거나 이동 방향을 결정한다. 단기기억은 일정 기간 동안 방문한 해의 이동정보를 기억하며 해의 순환을 막고 탐색 영역을 확장한다. 장기기억을 통해서 처음부터 현재까지의 모든 해의 이동에 관한 정보가 저장되며, 이를 통해 해의 탐색 빈도를 파악하여 탐색 빈도가 낮은 해 쪽으로 탐색을 유도한다. 탐색을 위한 이웃해(neighborhood solutions)는 다음과 같은 방법으로 생성하였다.

**이웃해 탐색 방법**

타부-탐색에서 이웃해 탐색 전략에는 교환 이동(swap), 삽입 이동(insertion), 인접 교환 이동(adjacent swap), 역순 이동(inversion), 삭제 이동(deletion) 등의 방법이 존재한다. 본 알고리즘에서는 이웃해를 탐색할 때 교환 이동(swap) 방식을 사용하였다. 추가로 삽입 이동(insertion), 삭제 이동(deletion) 방식을 확률적으로 사용되었다. 교환 이동 방식은 현재해(current solution)에서 임의로 선택한 두 노드의 위치를 서로 바꾸어주는 방식을 말한다. 본 연구에서 사용한 교환 이동 방식은 현재해에서 노드 집합  $N$  중 디포 노드와 디커플링 노드들을 제외한 노드 중에서 임의로 선택한 하나의 노드(노드1)와 디커플링 노드 중에서 임의로 선택한 하나의 노드(노드2)의 위치를 서로 교환하는 것이다. 이러한 과정을 통해서 노드1이 새로운 디커플링 노드가 되게 된다. 이렇게 디커플링 노드가 교환된 후에 디커플링 노드들에 대한 할당 작업을 진행하게 되므로 차량 노드 모빌리티 노드들 모두 변화가 가능하여 현재해보다 훨씬 더 다양한 이웃해를 탐색할 수 있게 된다. 삽입 이동 방식은 현재해에 임의의 노드를 추가하는 방식을 말하며, 삭제 이동 방식은 현재해 중 임의의 노드를 삭제하는 방식을 말한다. 본 연구에서 사용한 삽입 이동 방식은 현재해에 확률적으로 1) 디커플링 노드 혹은 2) 노드 집합  $N$  중 디포 노드와 디커플링 노드를 제외한 노드 중에서 임의의 노드를 선택하여 추가해주는 것이다. 1)의 경우, 같은 디커플링 노드에서 다른 모빌리티 그룹이 생성될 수 있는 기회를 제공하며, 2)의 경우가 발생하면, 기존과는 다른 새로운 디커플링 노드가 생기는 효과를 얻을 수 있다. 하지만 교환 이동 방식과 함께 삽입 이동 방식을 적용한다면, 디커플링 노드들이 무수히 많이 생기는 일이 발생할 수 있다. 이를 방지하기 위해서 삭제 이동 방식도 함께 적용되었다. 삭제 이동 방식은 디커플링 노드 중에서 임의의

노드를 선택하여 삭제하는 방식으로 사용하였다. 이웃해 탐색은 먼저 확률적으로 삽입 이동을 할지 삭제 이동을 할지 선택하여 해당 이동을 수행한다. 삽입 이동 혹은 삭제 이동을 수행한 결과에 교환 이동을 적용하게 된다. 아래에서 Step 1~3.은 삽입 이동과 삭제 이동의 수행 방식을, Step 4~5.은 교환 이동의 수행 방식을 설명하고 있다.

- Step 1. 50%의 확률로 삽입 이동을 할지, 삭제 이동을 할지 선택한다. 삽입 이동이 선택되었다면 Step 2로, 삭제 이동이 선택되었다면 Step 3으로 이동한다.
- Step 2. 노드 집합  $N$  중 디포 노드를 제외한 임의의 노드  $i$ 를 선택한다. 이때, 70%의 확률로 디커플링 노드들이 선택될 수 있도록 한다. 선택된 노드  $i$ 를 새로운 디커플링 노드로 추가하고 Step 4로 이동한다.

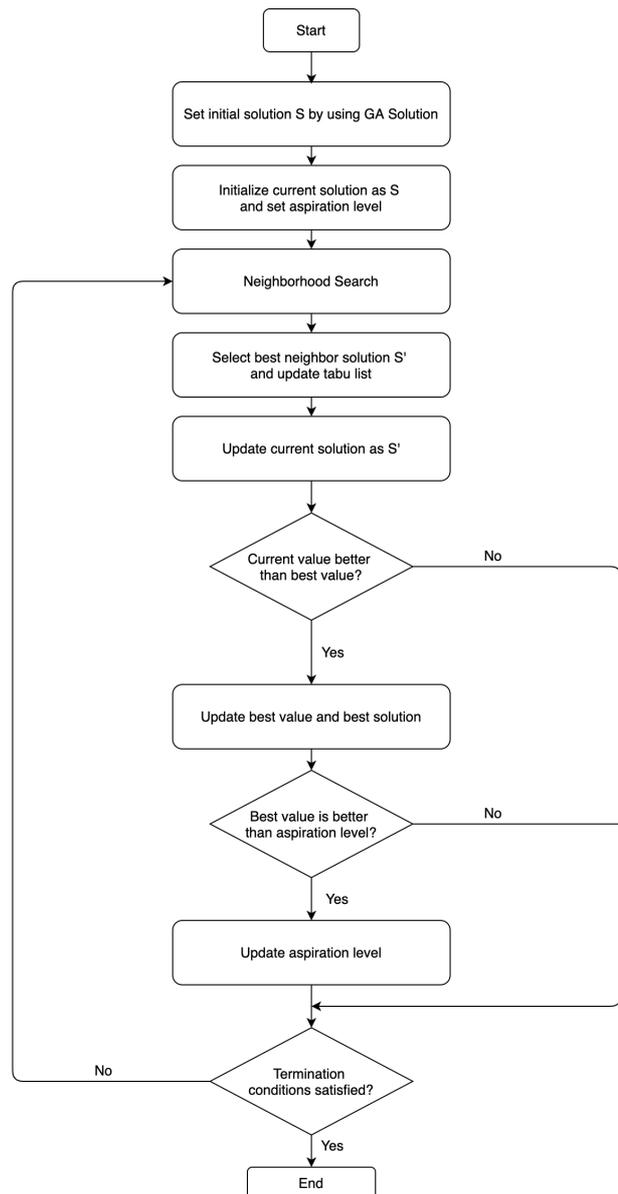


Figure 8. Flowchart of Tabu-search Algorithm

- Step 3. 디커플링 노드 중에서 임의의 노드  $i$ 를 선택한다. 선택된 노드  $i$ 를 디커플링 노드에서 제거하고 Step 4로 이동한다.
- Step 4. 노드 집합  $N$ 에 대하여, 디포 노드와 디커플링 노드들을 제외한 임의의 노드  $i$ , 디커플링 노드 중에서 임의의 노드  $j$ 를 선택한다.
- Step 5. 현재해에서 노드  $i$ 와  $j$ 의 위치를 교환한다. 결국 노드  $i$ 가 새로운 디커플링 노드가 되게 되고, 디커플링 노드였던 노드  $j$ 는 디커플링 노드가 아니게 된다. 이렇게 생성된 이웃해가 후보해 목록(candidate list)에 이미 존재한다면 Step 6으로 그렇지 않다면 Step 7로 이동한다.
- Step 6. 만약, Step 5의 결과가 후보해 목록에 이미 있는 경우, Step 1~5를 Step 5의 결과가 후보해 목록에 없을 때까지 반복한다.
- Step 7. Step 5의 결과가 후보해 목록에 없는 경우 해당 결과를 새로 후보해 목록에 추가하고 종료한다.

후보해 목록에 대해서 모빌리티 노드들을 할당하는 작업(4.1.2 배송 노드 할당)을 진행한다. 위의 단계를 거쳐서 새롭게 모빌리티 노드들을 할당하고 그에 해당하는 최적 경로 값을 도출하여 해당 후보해를 검증하는 과정을 거치게 된다. <Figure 8>은 타부-탐색 알고리즘의 전반적인 흐름을 보여준다.

## 5. 전산 실험

이 장은 앞 장에서 제안한 알고리즘에 대한 전산 실험 결과를 보고한다. 모든 알고리즘은 Python 3.7.4로 구현했으며, 모든 실험은 MacMini Apple M1 16-Core Neural Engine 16GB RAM에서 수행되었다. 혼합정수계획법 기반 수리 모형 (P)는 IBM Cplex를 이용하여 풀었다.

### 5.1 실험 대상 문제 및 파라미터

전산 실험은 총 두 가지 데이터 셋에 대해 진행하였다. TSP 문제를 위한 다양한 benchmarking 문제들이 존재하며, 이 중 Solomon instances(Solomon, 1987)와 Augerat VRP dataset (Augerat *et al.*, 1995)을 사용하여 실험을 진행하였다.

실험 진행에 앞서, 네트워크 상의 아크 비용을 설정해 주어야 하며, 비용은 유클리드 거리(euclidean distance)에 의한 이동 시간으로 설정하였다. VRPwMD에서의 네트워크는 두 고객 노드 간 두 개의 아크가 각 운송수단 별로 존재한다. 차량의 이동 시간과 마이크로-모빌리티의 이동시간은 운송수단의 이동 속도의 비로 산출 가능하다. 차량의 접근이 어려운 지역에 대해서는 차량의 평균 이동 속력과(최대 10km/h, Xing *et al.*, 2019) 마이크로-모빌리티의 평균 이동 속력은(최대 30km/h,

Campisi *et al.*, 2020) 약 3:1의 비를 가짐으로, 이를 이용하여 아크 비용을 설정할 수 있다. 하지만, 마이크로-모빌리티의 아크 비용을 일괄적으로 낮게 주면 마이크로-모빌리티가 고속화 도로와 같이 차량의 이동이 더 현실적이고 효율적인 곳에서 차량보다 시간적 이익을 주기 때문에 본 연구에서 목표하는 밀집 지역에 대한 마이크로-모빌리티 사용에 따른 효용이 나타날 수 없다. 따라서 네트워크에 대해 거리가 긴 상위 50%에 대해서는 마이크로-모빌리티의 아크 비용이 차량보다 2배 크도록 설정하였다. 즉, 아크의 길이가 상대적으로 짧은 하위 50%에 대해서만  $3 \cdot c_{ij}^M = c_{ij}^V$ 이 된다. 또한, 마이크로-모빌리티로 한 번에 배송할 수 있는 최대 시간  $L$ 은 차량의 최대 주정차 가능 시간인 600초(10분)로 산정하였으며, 마이크로 모빌리티로 한 번에 배송할 수 있는 최대 부피와 무게를 나타내는  $CP$ 와  $WT$ 는 각각 40ℓ, 10kg으로 설정해 주었다.

다음은 제안하는 휴리스틱 알고리즘에 대한 파라미터이다. 총 두 가지 알고리즘(i.e. 유전 알고리즘, 타부 탐색 알고리즘)은 다양한 파라미터를 입력받으며, 해당 파라미터들은 해의 품질에 큰 영향을 미치는 요소가 된다. 유전 알고리즘의 경우 한 세대의 크기는 100으로 설정하였으며, 총 100번의 진화를 통해 해를 산출한다. 타부 탐색 알고리즘의 경우에도 알고리즘의 반복을 총 100회 실시하게 된다.

### 5.2 실험 결과

본 장에서는 설정한 파라미터들에 대한 수리 모형(VRPwMD)과 메타-휴리스틱 알고리즘에 대한 성능을 비교한다. VRPwMD 문제의 경우 디커플링 노드의 수와 함께 마이크로 모빌리티의 용량, 무게 제한에 따라 해의 변화가 크다. 수리 모형의 경우 휴리스틱 알고리즘과 다르게 네트워크 내 노드를 파라미터  $K$ 개 만큼 복사해 주어야 한다. 문제 복잡도의 증가를 고려하여  $K$ 는 7로 설정해 주었으며, 이는 마이크로-모빌리티가 한 디커플링 노드에서 최대 7번 배송을 진행할 수 있음을 뜻한다. 노드를 더 많이 복사하여 마이크로 모빌리티의 동일 노드에서의 배송을 더 많이 허용할 수도 있으나, 노드의 수가 배로 늘어남으로써 생기는 연산 시간의 양이 기하급수적으로 증가하여 VRPwMD의 가능해를 찾지 못하는 경우가 매우 많게 된다.

<Table 2>에서는 수리 모형과 휴리스틱의 성능을 비교하도록 데이터 셋을 선택적으로 추출하여 해를 산출하였다. 추가적으로 마이크로-모빌리티를 사용하지 않는 단일 운송 수단으로 배송을 수행하는 경우의 가능해를 추가로 기재해 주었으며 이는 Google ORTools의 해이다. <Table 3>은 수리 모형이 풀지 못하는 사이즈의 문제에 대한 휴리스틱 알고리즘의 실험 결과를 보여준다. 각 테이블의 열(column)의 의미는 아래와 같다.

- $|N|$ : 네트워크 내 방문해야 하는 노드 수
- time(s): 해 도출을 위한 전체 시간(초), 수리 모형의 경우 3600초의 제한 설정

**Table 2.** Solution Comparison between VRPwMD and Heuristic Algorithm for Small-Size Instances

N	Problem	VRPwMD( $K=7$ )			Heuristic			Single-Mode	
		time(s)	sol	gap	time(s)	sol	diff_1	sol	diff_2
20	C101	3600	112.85	55.25	188.35	111.28	1.57 (1.41%)	117.09	5.81 (5.22%)
20	R101	3600	222.76	54.79	202.94	203.11	19.65 (9.67%)	252.89	49.78 (24.50%)
20	RC101	3600	207.41	47.39	278.53	205.99	1.42 (0.69%)	219.80	13.81 (6.70%)
20	A-n33-k5	3600	339.10	62.91	240.85	324.29	14.81 (4.57%)	363.10	38.81 (11.97%)
20	A-n34-k5	3600	306.26	41.54	367.32	302.82	3.44 (1.14%)	377.28	74.46 (24.59%)
20	A-n39-k5	3600	250.42	61.81	493.45	248.44	1.98 (0.80%)	316.22	67.78 (27.28%)
20	B-n38-k6	3600	402.73	73.14	354.87	299.10	103.63 (34.65%)	310.21	11.11 (3.71%)
20	B-n57-k7	3600	377.09	75.89	564.29	368.89	8.20 (2.22%)	384.43	15.54 (4.21%)
20	B-n50-k7	3600	311.32	62.70	323.86	310.42	0.90 (0.29%)	332.68	22.26 (7.17%)
20	P-n22-k8	21.40	264.71	0.00	340.41	264.71	0.00 (0.00%)	264.71	0.00 (0.00%)
20	P-n45-k5	3600	309.18	65.78	497.52	189.84	119.34 (62.86%)	257.24	67.40 (35.50%)
20	P-n70-k10	3600	239.45	64.65	516.08	202.62	36.83 (18.18%)	243.69	41.07 (20.27%)

- sol : 목적함수 값

- gap :  $(\frac{best\ obj.\ val - best\ lowerbound}{best\ obj.\ val}) \cdot 100(\%)$

- Single-Mode : 마이크로-모빌리티를 사용하지 않고 단일 운송 수단(차량)만을 사용한 Google ORTools RoutingModel()의 해

- diff\_1 :  $MILP\ obj.\ val - Heuristic\ obj.\ val$ , 최적해와 휴리스틱 알고리즘의 목적함수 값 차이. 양수의 경우 휴리스틱의 해가 더 좋음을 의미

- diff\_2 :  $Single\ Mode\ obj.\ val - Heuristic\ obj.\ val$ , 단일 운송수단 배송 경로의 목적함수 값과 휴리스틱 알고리즘의 목적함수 값의 차이. 양수의 경우 휴리스틱 알고리즘의 해가 더 좋음을 의미

VRPwMD에서 마이크로-모빌리티는 일반적인 차량이 접근하기 어려운 밀집 지역에 대해 매우 효율적이라는 가정을 갖는다. 따라서 차량과 모빌리티에 주어지는 배송지의 배치에 따라 해가 바뀔 수 있다. 즉, 주어지는 네트워크의 위상(network topology)에 따라 VRPwMD가 일반적인 VRP 보다 더 효율적인 경우가 있다. 본 실험에서 사용한 Solomon instance는 다양한 네트워크 위상을 나타낸다. 해당 데이터 셋의 경우 3가지 종류의 네트워크 형태로 구성되어 있다. 네트워크 형태는 고객의 군집화 정도로 나뉘어져 있는데, C 문제의 경우 고객들이 매우 강하게 군집을 이루고 있으며, R 문제의 경우 C와 반대로 산재되어 있다. RC 문제의 경우 C와 R의 중간 정도의 고객 분포를 지닌다.

<Table 2>는 수리 모형과 휴리스틱 알고리즘의 해를 비교한다. 모든 경우에 휴리스틱이 수리 모형보다 좋은 해를 주었으며, 해 도출 시간 또한 수리 모형 보다 더 짧은 시간에 좋은 유효해를 제공하였다. 특히, P-n22-k8의 경우, 고객에게 배송해야 하는 모든 물건의 무게와 용량이 각각  $WT$ ,  $CP$ 를 초과하

여 마이크로-모빌리티를 사용하지 않고 차량으로 배송해야 하는 경로에 대한 최적해를 주었다. 해당 경우에 대해, 휴리스틱 알고리즘은 마찬가지로 최적해를 제공하였다.

수리 모형의 경우 주어진 시간 내에 위 <Table 2>에 표기된 고객 수 이상의 네트워크에 대해서는 유효해를 제공해 주지 못하였다. 따라서 고객의 수가 더 많은 네트워크에 대해서는 휴리스틱 알고리즘만으로 해를 도출하였으며, 이는 아래 <Table 3>에 나타나있다.

<Table 3>은 Cplex가 주어진 시간 내에 유효해를 제공하지 못하는 크기의 네트워크에 대해 휴리스틱 알고리즘이 해를 제공함을 나타낸다. 휴리스틱 알고리즘의 경우, 수리 모형이 유효시간 내에 유효해를 제공하지 못하는 모든 경우에 대해 단일 운송수단에 의한 해 보다 좋은 해를 제공하였다. 휴리스틱 알고리즘은 노드의 수가 커짐에 따라 연산 시간이 증가하는 성향을 띠었으며, 100명의 고객에 대한 배송도 최대 60분 이내에 단일 운송수단의 해보다 좋은 값을 제공하였다.

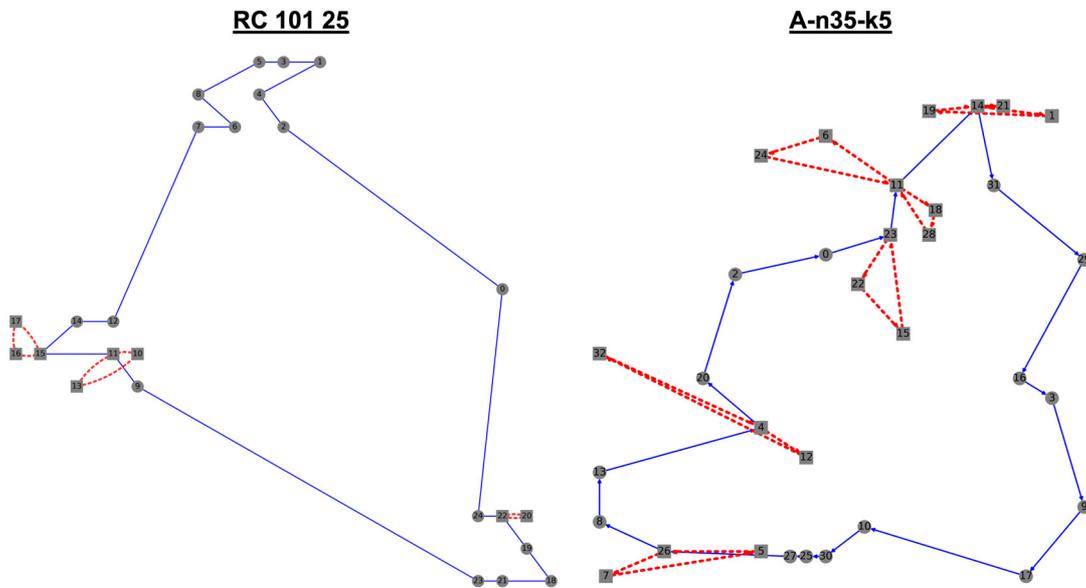
뿐만 아니라, 제안하는 휴리스틱 알고리즘과 Google ORTools를 사용하여 단일 운송수단으로(TSP) 배송을 수행한 경우에 대한 해 비교 또한 가능하다. 앞서 언급한 것과 같이 P-n22-k8의 고객은 모두  $WT$ ,  $CP$ 를 초과하는 값을 갖고 있어 마이크로-모빌리티를 사용하지 못하였으며, 이를 제외한 모든 경우에 대해 제안하는 휴리스틱이 더 좋은 값을 찾았다. 이는 곧 VRPwMD 문제가 기존의 배송 방법인 단일 운송수단에 의한 배송보다 항상 더 좋거나 같은 값을 제공할 수 있음을 시사한다.

아래 <Figure 9>는 Solomon, Augerat 데이터 셋 중 각각 하나의 instance들에 대한 휴리스틱 알고리즘의 해를 보여주고 있다. 붉은색 점선은 마이크로-모빌리티의 배송 경로를 나타내며, 청색 직선은 차량의 경로를 보여준다. 두 경우 모두 디커플링 노드에서 형성된 마이크로-모빌리티의 경로가 밀집 지역에 대해 차량의 배송을 대행하고 있음을 알 수 있다. 또한,

<Figure 9>의 우측 경로의 경우, 11번 노드에서 두 번 마이크로-모빌리티가 출발한 것을 확인할 수 있다. 즉, 휴리스틱 알고리즘에서도 한 디커플링 노드에서 여러 번 마이크로-모빌리티가 출발하는 것을 고려하고 있음을 알 수 있다.

**Table 3.** Heuristic Solutions for Large-Size Instances

N	Problem	Heuristic		Single-Mode	
		time(s)	sol	sol	diff_2
25	C101	240.58	121.96	138.15	16.19 (13.27%)
50	C101	626.38	228.21	245.09	16.88 (7.40%)
100	C101	3140.37	489.56	534.71	45.15 (9.22%)
25	R101	389.87	251.98	308.34	56.36 (22.37%)
50	R101	884.34	408.08	476.61	68.53 (16.79%)
100	R101	2867.06	621.59	671.28	49.69 (7.99%)
25	RC101	249.42	220.30	225.11	4.81 (2.18%)
50	RC101	744.84	362.99	369.03	6.04 (1.66%)
100	RC101	3193.26	628.72	677.95	49.24 (7.83%)
32	A-n32-k5	616.52	438.72	517.68	78.96 (18.00%)
33	A-n34-k5	967.45	443.89	492.44	48.55 (10.94%)
37	A-n37-k5	968.33	475.69	530.94	55.25 (11.61%)
39	A-n39-k5	772.99	511.62	561.52	49.90 (9.75%)
31	B-n31-k5	571.61	285.91	288.07	2.16 (0.76%)
34	B-n34-k5	627.99	312.22	317.09	4.87 (1.56%)
38	B-n38-k6	818.62	347.92	351.17	3.25 (0.93%)
39	B-n39-k5	824.57	311.57	318.95	7.38 (2.37%)
16	P-n16-k8	154.42	130.53	154.42	23.89 (18.30%)
19	P-n19-k2	171.67	169.38	171.67	2.29 (1.35%)
22	P-n22-k2	185.99	171.69	186.00	14.31 (8.33%)
22	P-n22-k8	500.84	278.44	278.44	0.00 (0.00%)



**Figure 9.** Heuristic Solution for a Solomon RC101 Instance with Node Size 25, and an Augerat A-n33-k5 Instance with Node Size 33. Red-dotted Lines and Blue Solid Lines Represent Micro-mobility and Truck Trajectories. Best Viewed in Color

### 5.3 민감도 분석

본 절에서는 다양한 상황에 대한 휴리스틱 알고리즘의 민감도 분석 결과를 보고한다. VRPwMD 문제의 경우 계층적인 경로가 형성되기 때문에 주어진 네트워크의 위상에 따라 효과성이 다르게 측정될 수 있다. 또한, 마이크로-모빌리티의 운행에 있어 마이크로-모빌리티의 배달 능력에 따라 해의 변화가 크다. 이를 위에서 사용한 두 데이터 셋을 통해 검증해 보고자 추가적인 민감도 분석을 실시하였다.

네트워크 위상에 따른 VRPwMD의 효용은 Solomon instance를 사용하여 측정해 볼 수 있다. 효용을 산출하는 식은 아래에서 확인할 수 있다.

$$Improvement (\%) = \frac{ORTools\ TSP\ Value - Heuristic\ Solution}{Heuristic\ Solution} \times 100$$

휴리스틱 알고리즘의 결과를 각 문제 종류와 크기에 대해 살펴보았을 때 아래 <Figure 10>과 같은 결과를 산출할 수 있었다. R class의 문제에서 가장 높은 해 개선율을 보였으며, 이는 단일 차량 사용 시 차량의 이동이 필연적으로 많은 네트워크에서 마이크로-모빌리티의 대리 배송이 효과적임을 의미한다. RC와 C class에 대해서는 상대적으로 낮은 해 개선율을 보였다. 이는, 마이크로-모빌리티의 능력이 제한된 상황에서 군집 내에 모든 노드를 한 번에 처리할 수 없기 때문으로 해석할 수 있다. 해당 결과는 배송이 필요한 고객 네트워크에 대한 높은 이해가 사용자로 하여금 VRPwMD를 더 잘 활용할 수 있도록 하는 요소가 될 수 있음을 시사한다. 단 본 결과는 차량이

모든 노드에 접근이 가능하다는 가정하였기에 R class 문제들에 대해 해 개선 효과가 크게 나타났음을 유의할 필요가 있다. 예를 들어 서론에 설명한 아파트 단지 또는 복잡한 시장과 같이 일부 노드들에 대해 차량이 접근이 아예 불가능하거나 매우 비효율적이라면 개선 효과는 매우 달라질 것이다. 본 실험에서는 마이크로-모빌리티를 전혀 사용하지 않는 알고리즘과의 비교를 위해 이러한 상황을 고려하지 않았으나, 실제 이러한 경우가 발생하는 상황에서는 본 연구의 알고리즘이 더 좋은 결과를 보일 것으로 쉽게 예측할 수 있다.

VRPwMD의 해 변화를 마이크로-모빌리티의 능력의 변화에 대해 알아보하고자 실험을 수행하였고 <Figure 11>이 그 결과를 보여준다. 해당 실험은 Augerat instance 중 A-n32-k5 데이터를 사용하여 진행하였으며, 마이크로-모빌리티의 능력을 의미하는 파라미터  $WT, CP, L$ 를 순차적으로 변화시켜 보며 해의 변화를 살펴보았다.

좌측부터 순서대로 마이크로-모빌리티의 탑재 가능 최대 무게, 용량, 그리고 1회 최대 배송 시간을 늘려가며 해를 구해보았다. 각 데이터는 경우별로 10회 반복 실시하여 목적함수의 평균값을 제시하고 있다. 각 파라미터가 0 값일 때에는 1대의 차량으로 운송하는 경우와 같은 해를 제공함을 알 수 있으며, 마이크로-모빌리티의 능력으로 대표되는 무게  $WT$ 를 늘려주면 해의 개선 정도가 뚜렷함을 <Figure 11>에서 확인할 수 있다. 하지만, 탑재 물품의 최대 용량  $CP$ 와 배송을 위한 1회 최대 운송 시간  $L$ 의 경우 무게와 용량과 같이 해의 개선 그래프가 우하향하는 그래프 형태가 아니었으며 특정 임계치 이상은 해의 변화가 크지 않음을 확인하였다. 이는 VRPwMD 문제에서 마이크로-모빌리티의 능력은 적재 물건의 무게에 의존적임

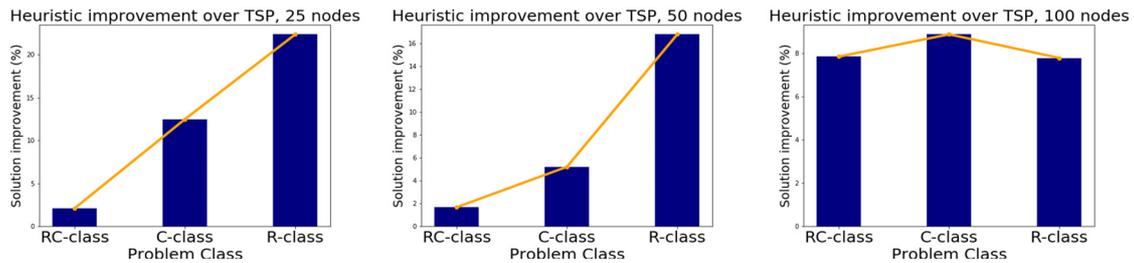


Figure 10. Solution Improvement Rate Compared to Single-Mode Route Solution

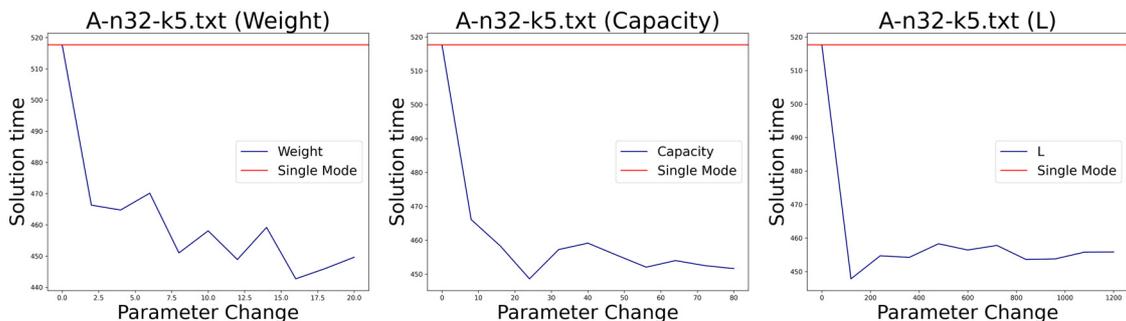


Figure 11. Sensitivity Analysis Results of Altering the Weight, Capacity and Possible tour time L, respectively

을 알 수 있으며, <Figure 10>의 RC, C Class와 같은 군집을 이루는 네트워크에 대해 마이크로-모빌리티의 능력이 향상하면 해 개선의 정도가 더 높아질 수 있음을 시사한다.

## 6. 결론

본 논문은 마이크로-모빌리티를 이용한 새로운 배송 방법을 제안하고 이를 VRPwMD로 정의하였다. VRPwMD는 기존에 많이 연구되어 온 차량 경로 문제의 새로운 유형으로 볼 수 있다. VRPwMD는 기존에 유사한 차량 경로 문제들과 비교하면 차량에서 마이크로-모빌리티로 전환하는 디커플링 노드와 경로를 동시에 결정하는 면에서 차이가 있다. 또한, 차량과 마이크로-모빌리티의 서로 크게 다른 특성(수송 용량, 속도 등)을 고려해야 하는 점이 특징이다.

VRPwMD를 풀기 위해 먼저 혼합정수계획법 기반 수리 모형을 제시하였다. 제시한 수리 모형은 문제의 크기가 매우 작은 경우에만 최적해를 구할 수 있었다. 따라서 더 큰 크기의 문제에 대해 빠른 시간에 좋은 해를 찾을 수 있는 휴리스틱 알고리즘을 개발하였다. 개발한 휴리스틱 알고리즘은 디커플링 노드 선택, 마이크로-모빌리티 배송 노드 할당, 경로 해 정의 단계로 이루어져 있다. 디커플링 노드 선택 시 좋은 해를 빠르게 탐색하기 위해서 두 가지 메타-휴리스틱 알고리즘을 적용하였다. 특히, 마지막 휴리스틱 단계인 경로 생성은 공개 소스 최적화 패키지인 Google ORTools를 사용하였고, 이로 인해 본 논문이 제안하는 알고리즘은 상용 소프트웨어 없이 구현할 수 있었다.

잘 알려진 차량 경로 문제에 대해 제안한 알고리즘을 적용하여 전산 실험을 수행하였다. 실험 결과는 제안한 휴리스틱 알고리즘이 수리 모형 상용 최적화 solver인 Cplex 보다 더 좋은 해를 빠른 시간에 찾을 수 있음을 보였다.

본 논문에서 다룬 VRPwMD는 마이크로-모빌리티의 수송 용량과 최대 이동 거리 제약만을 고려하였다. VRPwMD를 다른 제약 조건이 있는 경우로까지 확장하는 것은 간단하지 않다. 예를 들어, 고객 노드에 제약으로서 방문 가능 시간대가 존재한다면 디커플링 노드의 선택과 경로에 의해 고객 노드 방문 시간이 결정되게 된다. 방문 시점이 주어진 방문 시간에 속하도록 하기 위해서는 휴리스틱 알고리즘의 각 단계에서 시간에 대한 고려를 추가할 필요가 있으며 이는 적절한 향후 연구 주제가 될 수 있다.

## 참고문헌

- Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1995), Computational results with a branch and cut code for the capacitated vehicle routing problem. Tech. Rep. 949-M, Université Joseph Fourier, Grenoble, France.
- Banner, E. (2021), First Rental Scooters Hit Seattle Streets. *The Seattle Times*, <https://www.seattletimes.com/seattle-news/transportation/first-rental-scooters-hit-seattle-streets/>. (Accessed on 01/04/2022).
- Bard, J. F., Liu, H., Moshe, D., and Patrick, J. (1998), A Branch and Cut Algorithm for the VRP with Satellite Facilities, *IIE Transactions*, **30**(9), 821-834.
- Bullnheimer, B., Hartl, R. F., and Strauss, C. (1999), An Improved ant System Algorithm for the Vehicle Routing Problem, *Annals of Operations Research*, **89**, 319-328.
- Campisi, T., Nahiduzzaman, K. M., Ticali, D., and Tesoriere, G. (2020), Bivariate Analysis of the Influencing Factors of the Upcoming Personal Mobility Vehicles (PMVs) in Palermo, *International Conference on Computational Science and Its Applications*, Springer, Cham.
- Chabrier, A. (2006), Vehicle Routing Problem with Elementary Shortest Path Based Column Generation, *Computers & Operations Research*, **33**(10), 2972-2990.
- Clarke, G., and Wright, J. W. (1964), Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, *Operations Research*, **12**(4), 568-581.
- Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y., and Semet, F. (2002), A Guide to Vehicle Routing Heuristics, *Journal of the Operational Research Society*, **53**(5), 512-522.
- Desaulniers, G., Errico, F., Irnich, S., and Schneider, M. (2016), Exact Algorithms for Electric Vehicle-routing Problems with Time Windows, *Operations Research*, **64**(6), 1388-1405.
- Desrochers, M., Desrosiers, J., and Solomon, M. (1992), A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows, *Operations Research*, **40**(2), 342-354.
- Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996), A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In *Kdd*, **96**(34), 226-231.
- Gendreau, M., Hertz, A., and Laporte, G. (1994), A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science*, **40**(10), 1276-1290.
- Ghaziri, H. (1996), Supervision in the Self-organizing Feature Map: Application to the Vehicle Routing Problem, In *Meta-heuristics*, Boston, MA: Springer, 651-660.
- Gillett, B. E., and Miller, L. R. (1974), A Heuristic Algorithm for the Vehicle-dispatch Problem, *Operations Research*, **22**(2), 340-349.
- Glover, F. (1990), Tabu Search: A Tutorial, *Interfaces*, **20**(4), 74-94.
- Hu, M., Liu, W., Lu, J., Fu, R., Peng, K., Ma, X., and Liu, J. (2019), On the Joint Design of Routing and Scheduling for Vehicle-Assisted Multi-UAV Inspection, *Future Generation Computer Systems*, **94**, 214-223.
- Jeong, H. Y., Song, B. D., and Lee, S. (2019), Truck-Drone Hybrid Delivery Routing: Payload-Energy Dependency and No-Fly Zones, *International Journal of Production Economics*, **214**, 220-233.
- Jiang, H., Li, J., Yi, S., Wang, X., and Hu, X. (2011), A New Hybrid Method Based on Partitioning-based DBSCAN and ant Clustering, *Expert Systems with Applications*, **38**(8), 9373-9381.
- Kang, M. and Lee, C. (2021), An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem, *Transportation Science*, **55**(5), 1088-1112.
- Laurent Perron and Vincent Furnon, OR-Tools 7.2, <https://developers.google.com/optimization/>.
- Lee, C. (2020), An Exact Algorithm for the Electric-Vehicle Routing Problem with Nonlinear Charging Time Charging Time, *Journal of the Operational Research Society*, **72**(7), 1-24.

- Lee, Y. H., Jung, J. W., and Lee, K. M. (2006), Vehicle Routing Scheduling for Cross-docking in the Supply Chain, *Computers & Industrial Engineering*, **51**, 247-256.
- Lee, C., Lee, K., and Park, S. (2012), Robust Vehicle Routing Problem with Deadlines and Travel Time/demand uncertainty, *Journal of the Operational Research Society*, **63**(9), 1294-1306.
- Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., and Lam, H. Y. (2014), Survey of Green Vehicle Routing Problem: Past and Future Trends, *Expert Systems with Applications*, **41** (4 PART 1), 1118-38.
- Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960), Integer Programming Formulation of Traveling Salesman Problems, *Journal of the ACM (JACM)*, **7**(4), 326-329.
- Murray, C., and Raj, R. (2020), The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones, *Transportation Research Part C: Emerging Technologies*, **110**, 368-398.
- Muter, I., Cordeau, J., and Laporte, G. (2014), A Branch-and-Price Algorithm for the Multidepot Vehicle Routing Problem with Interdepot Routes, *Transportation Science*, **48**(3), 425-441.
- Osman, I. H. (1993), Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem, *Annals of Operations Research*, **41**(4), 421-451.
- Otaki, K., Koide, S., Hayakawa, K., Okoso, A., and Nishi, T. (2019), Multi-Agent Path Planning with Heterogeneous Cooperation, *Proceedings - International Conference on Tools with Artificial Intelligence*, ICTAI 2019-Novem, 93-100.
- Perboli, G., Tadei, R., and Vigo, D. (2011), The Two-echelon Capacitated Vehicle Routing Problem: Models and Math-based Heuristics, *Transportation Science*, **45**, 364-380.
- Salhi, S., Imran, A., and Wassan, N. A. (2014), The Multi-Depot Vehicle Routing Problem with Heterogeneous Vehicle Fleet: Formulation and a Variable Neighborhood Search Implementation, *Computers and Operations Research*, **52**, 315-325.
- Schneider, M., and Drexl, M. (2017), A Survey of the Standard Location-Routing Problem, *Annals of Operations Research*, **259**(1), 389-414.
- Toth, P., and Vigo, D. (2014), *Vehicle Routing: Problems, Methods, and Applications*, Philadelphia, PA: SIAM.
- van Breedam, A. (1996), *An Analysis of the Effect of Local Improvement Operators in Genetic Algorithms and Simulated Annealing for the Vehicle Routing Problem*, Belgium: RUCA.
- Way, Q. (2017), Onebot L1 and L2 Luggage Suitcase Scooters, Thepack. news-Electric motorcycle news, <https://thepack.news/onebot-l1-and-l2-luggage-suitcase-scooters/> (Accessed on 01/05/2022)/.
- Whitley, D. (1994), A Genetic Algorithm Tutorial, *Statistics and Computing*, **4**(2), 65-85.
- Xing, Y., Ban, X., Liu, X., and Shen, Q. (2019), Large-scale Traffic Congestion Prediction Based on the Symmetric Extreme Learning Machine Cluster Fast Learning Method, *Symmetry*, **11**(6), 730.
- Zibaei, S., Hafezalkotob, A., and Ghashami, S. S. (2016), Cooperative Vehicle Routing Problem: An Opportunity for Cost Saving, *Journal of Industrial Engineering International*, **12**(3), 271-286.

## 저자소개

**차형주:** 한국외국어대학교 산업경영공학과에서 2020년 학사학위를 취득하고 고려대학교 산업경영공학과 석사과정에 재학 중이다. 관심 연구 분야는 최적화 이론 및 활용, 강화학습 등이다.

**김동균:** 한국외국어대학교 산업경영공학과에서 2021년 학사학위를 취득하고 고려대학교 산업경영공학과 석사과정에 재학 중이다. 관심 연구 분야는 최적화 이론 및 활용, 데이터 분석 등이다.

**송보경:** 한국외국어대학교 산업경영공학과에 현재 재학 중이다. 관심 연구 분야는 최적화 이론 및 활용, 미량유해물질관리 등이다.

**이충목:** 고려대학교 물리학과에서 1997년 학사, 1999년 석사학위를 취득하고 KAIST에서 2009년 산업공학 박사학위를 취득하였다. ETRI 선임연구원, IBM Research Europe, Ireland에서 Research Staff Member 등을 역임하고 2015년부터 한국외국어대학교 산업경영공학과에서 교수로 재직 중이다. 관심 연구 분야는 최적화 이론, 정수계획법, 강건최적화, 데이터 마이닝 등이다.