

고객 대기 시간 최소화 차량 경로 문제의 발견적 해법

Heuristics for vehicle routing problem to minimize total delivery waiting time

강경환, 이영훈

(kang2000@yonsei.ac.kr, youngh@yonsei.ac.kr)

연세대학교 컴퓨터 과학/산업시스템공학과

서울시 서대문구 신촌동 134

Abstract

This paper is concerned with Vehicle Routing Problem to minimize the total delivery waiting time of customers. The delivery waiting time is the time taken to travel from the depot to the customer, which is important for the delivery of the perishing products or foods requiring freshness. We construct a mixed integer linear programming formulation of this problem, and CR (Clustering first Routing second) heuristic and SPH(set partitioning heuristic) are suggested. the results of computational experiments showed that these heuristics find successfully favorable solutions, in much shorter time than optimal.

1. 서론

차량 경로 문제(Vehicle Routing Problem : VRP)는 차고지(depot)에서 출발한 차량들이 서비스(배달, 수거)를 요구하는 수요지점들을 방문하고 다시 차고지로 복귀하는데 소요되는 최단시간(혹은 거리 : 이하 시간)의 경로를 결정하는 문제이다. 일반적인 차량 경로 문제의 목적함수는 총 운행 시간 최소화이며 제약 조건은 가용 차량의 대수, 가용 차량의 용량, 차량의 운행 시간에 대한 제한, 수요 지점의 수요량 등이 있다. 차량 경로 문제의 표준 형태(Standard Vehicle Routing Problem)는 Danzig와 Ramser[3](1959)가 최초로 제안한 것으로, NP-hard 문제로 알려져 있다.

전통적인 차량 경로 문제에 관한 연구중에서, Gillet and Miller[5](1974)는 극 좌표 상에서 한 지점을 무작위로 선택하여 그 지점을 씨앗지점(seed node)으로 선정하고 본점을 축으로 해서 시계방향, 혹은 반시계 방향으로 돌리면서 그 선이 휩쓸고 지나간 지점들을 하나씩 경로에 추가하는 스위핑 방법(Sweeping Method)을 제안하였다. 메타 휴리스틱을 이용한 연구도 활발히 진행되고 있으며, 최근에는 고객의 서비스 수준의 향상과 실제 문제의 적용이라는 차원에서 여러 가지 형태로 변화하고 있다.

첫 번째는 VRPTW(Vehicle Routing Problem

with time window)로써 고개의 서비스 요구 하한 시간(서비스가 이루어질 수 있는 가장 빠른 시간)과 상한시간(서비스가 이루어질 수 있는 가장 느린 시간)에 대한 제약 조건을 추가한 형태의 문제이다. Solomon et al.[7](1988)에 의하면 이 문제는 고전적인 차량 경로 문제보다 훨씬 어려운 문제로서, 최근에는 메타 휴리스틱을 이용한 연구가 활발히 진행되고 있다.

두 번째는 차량의 운행시간 제약의 범위 내에서 차고지에 대한 다회 방문이 허용되는 형태로서 VRPMT(Multi-Trip Vehicle Routing Problem)이다. Fagorholt[4]는 이러한 문제를 해결하기 위해 1, 2단계에서 모든 실행가능한 해를 찾고 3단계에서 partitioning problem을 푸는 최적화 알고리즘을 개발하였다. 유사한 형태로서 신해웅, 강맹규[2]는 수요 지점에 대한 다회방문이 가능한 형태인 분할 배달(Split Delivery)문제에 대해 분해해법을 제시하였으며, 이는 수요지점을 차량에 먼저 배정후 차량내 경로를 결정하는 알고리즘이다.

세 번째는 수요 지점에서 배달과 수거가 발생하는 형태인 PDP(Pick up Delivery Problem)형태로서 Mosheiov[6]는 tour partitioning heuristics을 제안하였다. 이외에도 차량의 용량이 서로 다른 문제인 HVRP(Heterogeneous Vehicle Routing Problem), 서비스의 상한 시간만 존재하는 Time deadline 문제, 차고지가 복수인 Multi-depot 차량 경로 문제 등이 제기되었다. 이와 같이 차량 경로 문제에 대한 연구추세는 고전적인 형태에서 현실적으로 추가되는 여러 가지 제약조건들과 고객 서비스 수준 향상이라는 목적 때문에 여러 가지 형태로 연구되어지고 있다.

강경환, 이영훈[1]은 차량 경로 문제의 새로운 형태로서 주문을 한 고객이 대기하는 시간을 최소화하기 위한 새로운 모델을 제시하였다. 고전적인 차량 경로 문제의 목적 함수가 총 차량 이동 시간의 최소화이며, 이는 서비스를 제공하는 공급자관점에서 가능한 신속히 모든 수요 지점의 서비스를 마치고 차고지로 복귀하는 것이다. 이에 반해 제안된 모델의 목적함수는 고객 대기 시간(total delivery waiting time)이며, 이는 각각의 고객들이 서비스를 받기 위해 기다리는 시간을 최소화하며, 서비스가 이루어지지 않은 고객의 수를 최소화한다는 개념으로서 고객의 입장을 고려한 모형으로 수리 모형과 발견적 해법을 제시하였다.

본 논문에서는 강경환, 이영훈[1]이 제안한 모

델에 대해 개선된 발견적 해법을 제안한다. 강경환, 이영훈[1]이 제시한 발견적 해법은 차고지에서 출발한 차량이 가까운 고객부터 방문하는 근접 이웃 방법(nearest neighborhood method)에 기준을 두고 제안된 해법이나, 다수의 차량이 사용될 경우 처음에 사용되는 차량의 목적함수 값은 우수하나, 나중에 사용되는 차량의 목적함수는 해가 좋지 않았다. 그 이유는 처음 사용되는 차량부터 근시안적으로 가까운 수요 지점만 방문하다 보니, 남겨진 수요 지점에 대해 방문하는 마지막 차량은 경로의 생성이 좋지 않을 수 밖에 없었다. 이에 따라 clustering을 먼저하고, 그 clustering내에서 경로를 생성하는 CR 해법과 경로를 먼저 생성하고, 최적의 clustering조합을 찾는 SPH를 제안한다.

본 논문의 구성은 다음과 같다. 1절의 서론에 이어 2절에서는 고객 대기 시간 최소화 차량 경로 문제의 최적해를 구하기 위한 혼합 정수 수리 모형을 제시하고, 3절에서는 발견적 해법인 CR해법과 SPH를 소개한다. 4절에서는 CR해법과 SPH의 성능을 평가하기 위한 실험결과가 최적해와의 비교를 통해 제시된다. 5절에서는 본 논문의 결론을 유도하고 추후 연구과제를 제시한다.

2. 수리모형

2.1 모형의 정의 및 가정

고객 대기 시간 최소화 차량 경로 문제의 모형은 단일의 차고지와 N개의 수요지점, 그리고 차량용량이 동일하게 U인 K대의 차량으로 구성되며 수요 지점의 수요량은 미리 알려져있으며, 차량의 운행시간 제한은 없는 것으로 가정한다. 기타 가정은 고전적인 차량 경로 문제의 가정과 동일하다.

2.2 기호의 정의

- N : 수요 지점의 수
- i : 수요 지점 ($i=1,2 \dots N; i=1$:차고지)
- D_i : 수요 지점의 수요량
- U : 차량의 용량
- K : 차량의 수
- T : 수요 지점의 부분집합
- V : 수요 지점의 전체집합
- d_{ij} : 수요 지점 i 와 j 의 시간
- x_{ijks} : 차량 k 에 의해 수요 지점 i 와 j 까지 마지막에서 s 번째 서비스가 이루어지면 1 그렇지 않으면 0

수리 모형은 다음과 같다.

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^K \sum_{s=1}^S sd_{ij} x_{ijks} \quad (2.1)$$

$$\text{s.t} \quad \sum_{j=2}^N x_{1jk1} \leq 1 \quad k=1, \dots, K \quad (2.2)$$

$$\sum_{i=1}^N \sum_{k=1}^K \sum_{s=1}^S x_{ijks} = 1 \quad j=2 \dots N \quad (2.3)$$

$$\sum_{j=1}^N \sum_{k=1}^K \sum_{s=1}^S x_{ijks} = 1 \quad i=2 \dots N \quad (2.4)$$

$$\sum_{i=1}^N x_{ikps} - \sum_{j=1}^N x_{ikps+1} = 0 \quad (2.5)$$

$$\begin{aligned} p &= 2, 3 \dots N \\ k &= 1, 2 \dots K \\ s &= 1, 2 \dots N-1 \end{aligned}$$

$$\sum_{i=1}^N D_i \left(\sum_{j=1}^N \sum_{s=1}^S x_{ijks} \right) \leq U \quad k=1, 2 \dots N \quad (2.6)$$

$$\sum_{i=1}^N \sum_{j=1}^N x_{ijks} \leq 1 \quad k=1, 2 \dots K, s=1, 2 \dots N \quad (2.7)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{s=1}^S x_{ijks} \leq |T| - 1 \quad k=1, 2 \dots K, \forall T \subset V, T \neq \emptyset \quad (2.8)$$

$$\begin{aligned} x_{iiks} &= 0, & k &= 1, 2 \dots K & (2.9) \\ & & i &= 1, 2 \dots K \\ & & s &= 1, 2 \dots S \end{aligned}$$

목적함수 (2.1)는 총 고객의 대기 시간을 최소화한다는 의미이다. 예를 들어 1대의 차량이 1-2-3-4-1의 순서대로 운행이 되고 차고지로 복귀하는 경우 목적 함수의 값은 $d_{12} + (d_{12} + d_{23}) + (d_{12} + d_{23} + d_{34}) + (d_{12} + d_{23} + d_{34} + d_{41})$ 가 되며 목적 함수의 값은 $4d_{12} + 3d_{23} + 2d_{34} + 1d_{41}$ 로 변환할 수 있으며 결정변수 x_{ijks} 의 값은 $x_{1411}=1, x_{4312}=1, x_{3213}=1, x_{2114}=1$ 이 된다. 식(2.2)은 차량이 운행이 될 경우 반드시 차고지로 복귀해야함을 의미한다. 식(2.3),(2.4)은 각 수요 지점은 한 대의 차량에 의해 한번만 방문되어야 한다는 것을 의미한다. 식(2.5)은 중간 경유지점 p 는 다음 서비스 지점으로 이동하기 위한 출발 지점이 됨을 의미한다. 식(2.6)은 각 경로에 포함된 수요 지점의 총 수요량은 차량의 적재 용량을 초과할 수 없음을 의미한다. 식(2.7)은 모든 서비스의 순서에 대해 각 수요 지점은 한번씩만 서비스를 받을 수 있음을 의미하며, 식(2.8)은 부분 경로(sub-tour)방지를 위한 조건으로 이 제약조건식은 $2^N - 1$ 개의 식으로 구성이 된다. 식(2.9)는 동일지점 이동은 불가함을 의미한다. 위 수리모형은 NP-hard 문제에 속해 문제의 규모가 커지면 최적해를 구하기가 매우 어려워진다.

3. 발견적 해법

3.1 CR 해법

고객 대기 시간 최소화 차량 경로 문제를 풀기 위한 발견적 해법을 개발하여 CR(Clustering first Routing second)해법이라 이름지었다. CR해법은 스위핑 방법에 의해 clustering을 먼저 하고 각각의 clustering에 대해 근접 이웃 방법을 사용한다. 이후 해를 개선하기 위한 과정을 진행하는데 이 진행 과정에 따라 CR-1해법과 CR-2해법으로 나누어 진행된다. 먼저 CR 해법의 개념을 요약하면 다음과 같다.

순서1) 본점과 각 수요 지점을 축으로 해서 시계방향, 혹은 반시계 방향으로 돌리면서 그 선이 휩쓸고 지나간 지점들을 차량의 용량이 허용하는 범위 내에서 포함시킨다. 차량의 용량을 초과할 경우 새로운 차량을 운행한다. 수요 지점이 n 일 경우 가능한 해의 경우의 수는 시계 방향으로 n 개, 반시계 방향으로 n 개, 총 $2n$ 개의 해가 경우의 수로 존재한다.

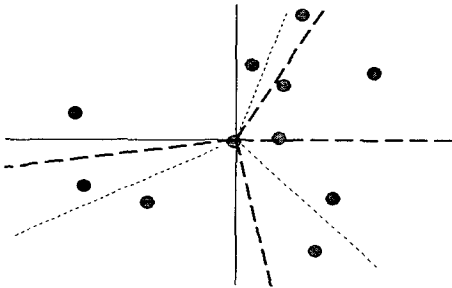
순서2) 순서 1과 같이 clustering을 한 다음, 각 clustering 내에서는 본점에서 출발하여 가장 가까운 수요 지점을 방문하고, 또 다시 그 수요지점에서 가장 가까운 수요지점을 방문하는 nearest neighborhood방법을 적용하여 해를 구한다.

이와 같이 순서1과 순서2를 통해서 해를 구했을 경우 스위핑 방법의 특성상 마지막 차량의 적재율은 상당히 낮아진다. 왜냐하면 각 차량은 차량의 용량이 허용하는 범위내에서 최대한 수요 지점을 포함시키기 때문에 마지막 차량은 아직 방문하지 않은 소수의 수요지점만을 방문하기 때문이다. 이는 목적함수 값에도 좋지 않은 영향을 미치게 된다. 이러한 문제점을 개선하기 위해 주어진 차량수의 범위내에서 적절히 적재율을 분배하는 것이 바람직하다. 예를들어 4대의 차량이 운행될 경우 1번 차량의 적재율이 95%, 2번 차량의 적재율이 80%, 3번 차량의 적재율이 90%, 마지막 4번 차량의 적재율이 30%일 경우 4대의 차량의 적재율을 가능한 동등히 하면 해가 개선이 된다. 왜냐하면, 목적 함수의 특성상 1대의 차량이 방문하는 수요지점이 증가할수록 목적 함수 값은 누적되어 증가하기 때문이다. 이렇게 차량의 적재율을 가능한 동등히 하기위해 개선 과정을 거치게 되는데 이 개선 과정에 따라 CR-1해법과 CR-2해법로 나누어 진다.

CR-1해법) 먼저 CR-1해법은 전체 차량의 평균 적재율과 마지막 차량을 제외한 나머지 차량의 평균 적재율의 중간값을 차량이 허용하는 제한 용량으로 설정한다.

즉 차량의 용량이 Q일 경우 CR-1해법은 $\beta \times Q$ 를 초과하기 전까지 수요지점을 방문하는데, β 값은 파라메타로서 순수한 스위핑 방법에서는 β 값은 1이다. CR-1 해법은 β 를 1보다 작은 값으로 설정하는데 $\beta = (\text{전체 차량의 평균 적재율} + \text{마지막 차량을 제외한 나머지 차량의 평균 적재율}) / 2$ 로서 앞선 예를 들면 $\beta = ((0.95 + 0.8 + 0.9 + 0.3) / 4) + ((0.95 + 0.8 + 0.9) / 3) / 2 = 0.81$ 로서 차량 용량의 81%까지만 적재를 한다. 이렇게 β 값을 구한다음 순서1과 순서2를 다시 반복한다.

CR-2해법) CR-2해법도 CR-1해법과 동일한 개념이나, $(1 - \alpha) \times Q$ 값을 초과하기 전까지 수요 지점을 방문하는데, $\alpha = (\text{최소 수요량} / \text{차량의 용량})$ 으로서 차량의 수가 동일한 범위내에서 α 값을 계속 뺀다.



<그림 1 CR-2 해법>

예를 들면 최소 수요량이 5이고 차량의 용량이 100일 경우 $\alpha = 0.05$ 가 되고, $0.95Q$, $0.9Q$, $0.85Q$ 와 같이 동일한 차량의 수 범위내에서 계속 α 값을 감소시키면서 해를 구하고, 이중의 가장 좋은 값이 찾고자 하는 해이다. 그림과 같이 처음 양의 x축을 기준으로 clustering했을 경우 최초에는 얇은 점선과 같이 4대의 차량으로 clustering되었으나, 개선 과정후에는 굵은 점선처럼 clustering되었다. 즉 동일한 차량수내에서 적절히 적재율을 배분함으로써 해의 개선이 있는 경우이다.

3.2 SPH(set partitioning heuristic)

또 다른 해법으로 SPH(set partitioning heuristic)이라 이름지었다. SPH는 경로를 먼저 생성한 다음 각각의 수요지점이 한번씩만 방문되도록 하는 set partitioning 문제를 풀어주면 된다. 수리 모형은 다음과 같다.

$$\text{Min} \sum_{r=1}^R c_r x_r \quad (3.1)$$

$$s.t. \sum_{r \in F_i} x_r = 1 \quad i=1, \dots, N \quad (3.2)$$

$$\sum_{r=1}^R x_r = K \quad (3.3)$$

$$x_r = 0 \text{ or } 1 \quad \forall r=1..R \quad (3.4)$$

목적함수 (3.1)은 최소의 값을 갖는 경로의 조합을 찾는 것을 의미한다. r 은 경로번호 index이며, c_r 은 route의 비용으로서 각 노드를 기준으로 차량의 용량이 허용하는 범위내에서 가장 가까운 수요 지점을 하나씩 포함시켜 나갈때의 경로비용이다. 즉 각 수요지점만을 방문하고 복귀하는 경로를 먼저 생성하고, 각 수요지점에서 가장 가까운 이웃 수요 지점을 방문하고 복귀하는 경로를 두 번째로 생성한다. 이와 같은 요령으로 차량의 용량이 허용하는 범위내에서 가장 가까운 수요 지점을 이어서 방문하는 절차를 반복한다. 이와 같이 각 수요 지점을 기준으로 차량의 용량이 허용하는 범위내에서 가장 가까운 수요 지점을 포함시켜 나갈 때의 가능한 모든 경로를 생성한다. x_r 은 생성되는 경로이다. 제약조건 (3.2)는 모든 수요지점 i 는 i 가 포함된(F_i) 모든 route의 합이 1임을 의미한다. 즉 수요 지점 i 는 한번만 방문해야함을 의미한다. 제약 조건 (3.3)은 가능한 경로의 수는 운행되는 차량의 수를 의미하므로 가능한 차량 수 K 이어야함을 의미한다. 제약 조건 (3.4)는 경로 x_r 이 선택되면 1, 그렇지 않으면 0이다.

4. 계산실험

발견적 해법의 일반적인 성능을 평가하기 위한 실험 디자인 5,10,15,20개 수요 지점에 대해 각 10개씩 40개의 데이터를 생성하였다. 수요량은 5 - 10까지 랜덤하게 생성하였으며, 차량 용량은 5,10개 수요 지점은 30, 15개 20개 수요 지점은 60으로 설정하였다. 수요 지점의 위치 또한 랜덤하게 생성하였다. 차량수는 미리 설정하였으며, 최적해는 OPL Studio 3.1로 구했으며, 알고리즘은 C로 구현되었다.

Instance	# node	# vehicle	CR-1			CR-2			SPH		
			value	P		value	P		value	P	
1	5	2	1.05	70.250	70.249	0.000	70.249	0.000	72.611	0.034	
2	5	2	3.19	79.550	79.554	0.000	79.554	0.000	79.553	0.000	
3	5	2	2.03	76.570	76.568	0.000	76.568	0.000	infeasible		
4	5	2	4.06	77.180	77.176	0.000	77.176	0.000	77.177	0.000	
5	5	3	7.41	83.830	83.830	0.000	83.830	0.000	87.085	0.039	
6	5	2	1.16	82.175	82.175	0.000	82.175	0.000	82.175	0.000	
7	5	2	1.37	117.392	122.704	0.045	122.704	0.045	117.394	0.000	
8	5	2	0.18	103.031	113.398	0.101	103.030	0.000	118.481	0.150	
9	5	2	3.95	151.125	162.832	0.077	162.832	0.077	infeasible		
10	5	3	2.20	99.458	100.660	0.012	100.660	0.012	100.660	0.012	
11	10	5	892.04	140.974	141.600	0.004	141.600	0.004	147.307	0.045	
12	10	4	886.22	147.210	142.726	0.011	142.726	0.011	infeasible		
13	10	5	975.04	147.250	147.246	0.000	147.246	0.000	158.583	0.077	
14	10	4	203.39	188.400	196.254	0.053	196.254	0.053	infeasible		
15	10	4	51.08	140.480	140.481	0.000	140.481	0.000	140.781	0.002	
16	10	4	183.95	225.474	225.474	0.000	225.474	0.000	infeasible		
17	10	5	11864.84	210.541	218.040	0.036	218.040	0.036	229.172	0.088	
18	10	3	202.48	190.077	192.006	0.010	192.006	0.010	infeasible		
19	10	5	7861.74	166.976	166.980	0.000	166.980	0.000	infeasible		
20	10	3	54.48	183.369	183.371	0.000	183.371	0.000	213.107	0.162	
21	15	3	732.87	237.820	255.476	0.074	255.476	0.074	infeasible		
22	15	3	723.04	275.725	284.082	0.030	284.082	0.030	296.678	0.076	
23	15	3	5467.47	311.133	326.522	0.049	326.522	0.049	infeasible		
24	15	4	2201.53	294.629	301.947	0.025	301.947	0.025	295.016	0.001	
25	15	3	397.93	265.420	291.252	0.097	291.252	0.097	330.671	0.246	
26	15	4	275.02	239.580	250.440	0.045	250.440	0.045	241.832	0.009	
27	15	4	243.26	256.481	295.522	0.152	282.034	0.022	279.599	0.090	
28	15	3	15689.79	321.626	321.630	0.000	321.630	0.000	infeasible		
29	15	3	1586.19	293.100	297.267	0.014	297.267	0.014	infeasible		
30	15	3	379.15	281.261	281.263	0.000	281.263	0.000	301.701	0.073	
31	20	4	1503.26	287.889	315.456	0.059	315.456	0.059	359.327	0.206	
32	20	4	10881.96	339.740	360.784	0.065	360.784	0.065	infeasible		
33	20	4	138.46	334.466	372.462	0.114	377.350	0.128	infeasible		
34	20	4	80593.95	354.391	396.601	0.119	396.601	0.119	infeasible		
35	20	4	18323.45	345.719	384.065	0.111	351.199	0.016	404.423	0.170	
36	20	4	78930.21	346.900	352.808	0.017	352.808	0.017	460.851	0.328	
37	20	4	83554.56	392.040	392.044	0.000	392.044	0.000	427.263	0.090	
38	20	4	137343.78	380.626	385.192	0.012	385.192	0.012	infeasible		
39	20	5	101479.41	342.073	370.127	0.082	355.527	0.039	381.347	0.056	
40	20	4	8310.78	325.731	332.979	0.022	332.979	0.022	infeasible		
						0.038		10.027		0.081	

<표 1 실험 결과>

*P :performance=(발견적 해법-최적해)/최적해

수요 지점이 20개를 초과할 경우는 최적해를 찾기 위한 계산 시간이 너무 오래 걸리고 컴퓨터의 메모리 초과로 실험이 어려워져 가장 큰 문제의 크기를 20개 수요지점, 5대 차량으로 제한을 하였다. 실험 결과 CR-1해법은 성능이 평균 0.035, CR-2해법은 0.026이 나왔으며, SPH는 0.081로 나왔다. 발견적 해법의 계산시간은 0에 가까웠고, 최적해와 비교해 우수한 해가 나왔음을 알 수 있다. SPH에서의 infeasible은 해당 차량의 수 내에서는 해를 찾지 못한 경우이다. 이는 SPH에서 초기 경로를 생성할 때 각 수요 지점을 기준으로 가까운 수요 지점을 방문할 때, 어느 지역에 밀집된 수요 지점의 경우, 반복해서 방문하는 중복된 경로를 생성하고, 이격된 수요지점은 방문하지 않으려는 특성 때문이다. 그러나 SPH는 충분히 개선의 가능성이 있는 발견적 해법으로 현재처럼 경로 생성을 하지 않고, 가능한 우수한 경로를 다양하게 포함하는 경로만 생성이 되면 우수한 해가 나올 것으로 판단된다. CR해법이 가지는 장점은 SPH에 비해 짧은 시간에 해가 나온다는 장점이 있다. SPH는 정수 계획법으로 해가 생성이 되므로 문제의 크기가 커지면 해를 구하는데 시간이 오래 걸린다는 단점이 있지만, 앞서 언급한 개선이 이루어지면 해는 우수할 것으로 예상된다.

5. 결론

본 논문에서는 고객 대기 시간 최소화 차량 경로 문제에 대해 개선된 발견적 해법을 제안하였으며, 제안된 발견적 기법에 대한 실험결과 최적해와 유사한 성능을 발휘함을 확인하였다. 또한 SPH에서 경로 생성에 대한 개선이 있다면 더 우수한

해가 나올 것으로 보이며, 추후 고객의 우선 순위가 고려된 문제에서의 고객 대기 시간 최소화라는 문제도 의미가 있을 것으로 판단된다.

참고 문헌

- [1] 강경환, 이영훈, "차량 경로 문제의 고객 대기 시간 최소화 해법에 관한 연구", 한국 과학회/대한 산업공학회 춘계 공동 학술대회, pp. 687-690, 2001.
- [2] 신해웅, 강맹규, "다회방문을 허용하는 차량 경로 문제의 발견적 해법", 한국공업경영학회지, 제 14 권, 제 24 집, pp. 141 - 147, 1991.
- [3] Dantzig, G.B. and J.H.Ramser, "The Truck Dispatching Problem", *Management Science*, Vol. 6, No. 1, pp.80-91, 1959.
- [4] Fagerholt, K., "Optimal fleet design in a ship routing problem", *International Transactions In Operational Research*, 6, pp. 453-464, 1999.
- [5] Gillett, B and L. Miller, "A Heuristic Algorithm for the Vehicle Dispatching Problem", *Operational Research*, Vol. 22, pp. 340-349, 1974.
- [6] Mosheiov, G., "Vehicle Routing with Pick-up and Delivery : Tour-Partitioning Heuristics", *Computers and Engng*, Vol. 34, No. 3, pp. 669-684, 1998.
- [7] Solomon, M.M., Baker, E. K. and Schaffer, J. R., "Vehicle Routing and scheduling problems with time window constraints: efficient implementations of solution improvement procedures", In *Vehicle Routing : Methods and Studies*, ed. B. L. Golden and A. A. Assad, pp 85-105, 1988.