

시간제약을 가진 다회방문 차량경로문제에 대한 휴리스틱 알고리즘

A heuristic algorithm for the multi-trip vehicle routing problem with time windows

김미이, 이영훈

연세대학교 정보산업공학과 서울시 서대문군 신촌동 134
(miyikim@yonsei.ac.kr, youngh@yonsei.ac.kr)

Abstract

This paper is concerned with a novel heuristic algorithm for the multi-trip vehicle routing problem with time windows. The objective function is the minimization of total vehicle operating time, fixed cost of vehicle and the minimization of total lateness of customer. A mixed integer programming formulation and a heuristic algorithm for a practical use are suggested. A heuristic algorithm is constructed two phases such as clustering and routing. Clustering is progressed in order to assign appropriate vehicle to customer, and then vehicle trip and route are decided considering traveling distance and time window.

It is shown that the suggested heuristic algorithm gives good solutions within a short computation time by experimental result.

1. 서론

일반적인 차량경로문제(Vehicle Routing Problem)는 차고지(depot)에서 출발한 차량이 배달(delivery)이나 수집(pick-up) 등의 서비스를 요구하는 고객(customer)을 모두 방문하고 다시 차고지로 돌아오는 최소 비용의 경로를 결정하는 문제이다. 이는 현재 기업들이 직면하고 있는 현실의 조건들을 가정하여 간단한 문제로 정의한 것이다. 따라서 본 논문에서는 현실적인 제약조건을 추가한 시간제약을 가진

다회방문 차량경로문제(Multi-Trip Vehicle Routing Problem with Time Windows)에 대한 최적화 수리모델과 발견적 해법을 제시한다. MDVRP는 고객에 대한 서비스를 마친 차량이 또 다시 차고지를 출발하여 다른 고객을 서비스할 수 있으며 고객이 요구하는 서비스 시간이 존재한다는 새로운 제약조건을 포함시켰다. 즉, 고객만족도를 최대화하면서 이동하는데 소요되는 시간(비용)을 최소화하고자 한다. 하지만 Lenstra와 Rinnooy Kan(1981)에 의하면 시간제약을 지닌 차량경로문제는 NP-hard 조합최적화문제에 해당된다. 따라서 기존의 연구를 보면 휴리스틱기법을 통해 초기해를 구하고 메타 휴리스틱을 이용하여 초기해를 개선하는 방식으로 최적해에 근사한 해를 구하고자 하였다.

지금까지 시간제약을 지닌 차량경로문제에 대한 연구는 많이 진행되었다. Barrie Baker와 Ayechev(2003)는 고객서비스 제공 시간제약과 총 이동거리의 제약을 지닌 문제를 고객의 Polar angle에 따라 고객을 정렬하여 차량을 할당하는 방식의 Gillet와 Miller(1974)가 제시한 휩쓸음방법과 차량을 임의로 할당하는 무작위방식으로 초기해를 작성하여 이를 GA(Genetic Algorithm)로 개선하는 형태의 알고리즘을 제시하였다. 또한 Hoong Chuin Lau, Melvyn Sim과 Kwong Meng Teo(2003)는 차량의 수를 제한하여 최소한의 차량으로 시간제약을 지닌 고객들을 가장 많이 서비스할 수 있는 경로를 형성하도록 2단계 해법과 유사한 방법으로 1번째 단계에서는 차량의 수를 최소화하기 위해 기존의 Clarke과 Wright(1964)가

제시한 절약방법을 통해 얻어진 초기해를 바탕으로 삼입방법(Or(1976)), 2-Opt(Potvin과 Rousseau(1995))와 교환방법(Osman(1993)) 등을 이용하여 이웃해를 구하고 2번째 단계에서는 서비스된 고객의 수를 최대화하기 위해 타부서치를 이용하여 해를 개선하였다. 이외에도 시뮬레이티드 어닐링, 뉴럴 네트워크나 Ant colony 등의 메타휴리스틱 기법을 이용한 연구도 많이 진행되었다. 그리고 다회방문을 허용하는 연구의 경우, Fleischmann(1990)는 절약방법을 이용하여 68에서 361개의 고객이 존재하는 문제의 해를 제공하였고, Taillard *et al.*(1996)는 bin packing 휴리스틱을 응용하고 Brandao와 Mercer(1997)는 타부서치를 이용하는 알고리즘을 제시하였다. 그리고 Christian Prins(2004)는 고객방문의 시간제약을 추가하여 유전자 알고리즘을 이용한 해법을 제시하였다.

본 연구에서는 시간제약을 지닌 다회방문 차량경로문제에 대하여 시간제약을 최대한 지키면서 차량의 총 이동거리도 최소화할 수 있는 간단한 발견적 해법을 제시하고자 한다.

2. 수리 모형

Notation

X_{kijr} : 차량 k 가 고객 i 에서 j 로 r 번째 trip에서 방문할 경우 1, 그렇지 않으면 0(결정변수).

t_{ij} : 고객 i 에서 j 로의 이동시간

g_k : 차량 k 의 시간당 변동비용

f_k : 차량 k 의 고정비용

P : 벌금비용

L_{ki} : k 차량의 i 고객에 대한 tardiness {tardiness= $\max(0, \text{차량의 도착시간}-\text{고객의 서비스요구시간 상한})$ }

E_{ki} : k 차량의 i 고객에 대한 earliness {earliness= $\max(0, \text{고객의 서비스요구시간 하한}-\text{차량의 도착시간})$ }

α, β : 적재오차율 및 패킹오차율

V_k : 차량 k 의 적재용량

A_{kj} : 차량 k 가 고객 j 에 서비스 가능하면 1, 아니면 0(진입도로 제한의 특성반영)

s_j : 고객 j 에 대한 서비스 시간

C_{kjr} : 차량 k 가 고객 j 에 r 번째 trip에 도착한 시간

T_{ji} : 고객 j 의 서비스 요구시간 하한

T_{ju} : 고객 j 의 서비스 요구시간 상한

RD_z : Region z 의 속하는 고객들의 수요의 합

최적화 수리모델

$$\text{Minimize } w_1 \left(\sum_{k=1}^K \sum_{r=1}^{R-1} \sum_{i=0}^N \sum_{j=0}^N X_{kijr} t_{ij} g_k \right) + \sum_{k=1}^K \sum_{j=0}^N X_{k0j} f_k + w_2 \sum_{k=1}^K \sum_{i=0}^N (P_1 L_{ki} + P_2 E_{ki})$$

s.t

$$\sum_{k=1}^K \sum_{r=1}^{R-1} \sum_{j=0}^N X_{kijr} \leq 1 \quad \forall i(\text{단}, i \neq 0) \quad (2.1)$$

$$\sum_{k=1}^K \sum_{r=1}^{R-1} \sum_{i=0}^N X_{kijr} \leq 1 \quad \forall j(\text{단}, j \neq 0) \quad (2.2)$$

$$\sum_{i=0}^N X_{kijr} - \sum_{j=0}^N X_{kijr} = 0 \quad (2.3)$$

$$\sum_{k=1}^K \sum_{r=1}^{R-1} \sum_{i=0}^N X_{kijr} = 0 \quad (2.4)$$

$$\sum_{i=0}^N \sum_{j=0}^N D_j X_{kijr} \leq (1-\alpha)(1-\beta) V_k \quad \forall k, r(\text{단}, r=1, \dots, R-1) \quad (2.5)$$

$$\sum_{k=1}^K \sum_{r=1}^{R-1} \sum_{i=0}^N \sum_{j=0}^N D_j X_{kijr} = \sum_{i=1}^N D_i \quad (2.6)$$

$$\sum_{r=1}^{R-1} \sum_{i=0}^N X_{kijr} \leq A_{kj} \quad \forall k, j(\text{단}, j \neq 0) \quad (2.7)$$

$$\sum_{i \in J_p} \sum_{j \in J_p} X_{kijr} = 0 \quad \forall k, r(\text{단}, r=1, \dots, R-1), p \quad (2.8)$$

$$\sum_{i \in J_p} \sum_{j \in J_p} X_{kijr} = 0 \quad \forall k, r(\text{단}, r=1, \dots, R-1), p \quad (2.9)$$

$$\sum_{i=1}^N X_{k0ir} \geq \sum_{i=1}^N X_{k0(r+1)i} \quad \forall k, r(\text{단}, r=1, \dots, R-1) \quad (2.10)$$

$$E_{k0} = \max(0, T_{0l} - \sum_{r=1}^{R-1} \sum_{i=0}^N \sum_{j=0}^N X_{kijr} (t_{ij} + s_j)) \quad (2.11)$$

$$L_{k0} = \max(0, \sum_{r=1}^{R-1} \sum_{i=0}^N \sum_{j=0}^N X_{kijr} (t_{ij} + s_j) - T_{0u}) \quad (2.12)$$

$$C_{k01} = 0 \quad \forall k \quad (2.13)$$

$$\sum_{i=1}^N X_{k0ir} (C_{kir} + s_i + t_{i0}) = C_{k0(r+1)} \quad (2.14)$$

$$\sum_{i=1}^N X_{kijr} (C_{kir} + s_i + t_{ij}) = C_{kjr} \quad \forall j(\text{단}, j \neq 0) \quad (2.15)$$

$$E_{kj} = \max(0, \sum_{r=1}^{R-1} \sum_{i=0}^N X_{kijr} (T_{ji} - C_{kjr})) \quad \forall k, j(\text{단}, j \neq 0) \quad (2.16)$$

$$L_{kj} = \max(0, \sum_{r=1}^{R-1} \sum_{i=0}^N X_{kijr} (C_{kjr} - T_{ju})) \quad \forall k, j(\text{단}, j \neq 0) \quad (2.17)$$

최적화 수리모델의 목적함수는 각 차량의 변동비와 고정비, 벌금비용의 합을 최소화한다는 의미이다. 업체의 특성에 따라 유동성 있게 적용 가능하도록 각 비용에는 가중치가 부여된다. 가중치의 조정을 통해 의사결정의 유연성이 보장된다. 식 (2.1)과 식(2.2)는 각각의 고객은 하나의 차량만이 방문할 수 있다는 것으로, 하나의 수요지를 여러 대의 차량이 동시에 들어갈 수 없다는 제약식이다. 식 (2.3)은 수요지에 방문한 차량은 반드시 다른 곳을 방문해야지 바로 차고지로 돌아오면 안 된다는 식이다. 식 (2.4)는 모든 차량은 같은 곳으로 이동할 수 없다는 식이고, 식 (2.5)는 동일한 차량에 적재되는 수요량은 차량의 적재용량을 넘어서는 안 된다는 뜻이다. 여기서 D_i 는 고객 i 의 수요량을 나타내며, a 는 적재오차율, β 는 패킹오차율, V_k 는 차량 k 의 적재용량을 나타낸다. 식 (2.6)은 실제 배송된 양은 고객들의 수요량의 총합과 동일하다는 뜻이며, 식 (2.7)은 진입 차량의 제약에 관련된 제약식이다. J_p 는 같은 타입의 수요를 가지고 있는 고객의 집합을 나타낸다. 예를 들면 $J_p = \{1,2,3\}$, $J_{p'} = \{4,5\}$ 일 경우, 식 (2.8)과 식 (2.9)에 의해 J_p 와 $J_{p'}$ 가 혼재 불가한 품목이라면 고객 1번과 4번의 경우에 같은 차량에 할당을 하는 것은 불가능하게 한다. 식 (2.10)은 차량의 trip순서는 순차적으로 일어나야 한다는 뜻이다. 식 (2.11), (2.12), (2.16)과 (2.17)은 각 고객에 대한 earliness와 tardiness를 계산하기 위한 식이다. 식 (2.13)은 배송을 시작하는 시간의 초기화를 나타내고, 식 (2.14)는 배송하고 차고지로 돌아오는 도착시간을 나타내며, 식 (2.15)는 차고지를 제외한 수요지의 도착시간을 나타낸다.

3. 알고리즘

알고리즘은 선분할-후경로의 방법과 유사하다. 우선, 차고지를 중심으로 고객들의 Polar angle에 따라 Region을 동일한 각도로 나누어 차량을 할당하게 되고, 할당된 차량을 바탕으로 차량의 경로를 구하는 방식으로 진행된다.

3.1 차량할당

기존의 휩쓸음 방법의 경우에는 고객을 차량에 할당하다가 차량의 적재용량이 넘으면 다음 차량에 고객을 할당하는 방식으로 진행되었다. 이 방법에 경우에는 실행 가능한 차량할당 해가 구해지지만 이것은 차량의 총 이동거리 측면에서는 그렇게 좋은 해를 주지는 않는다. 예를 들어, 휩쓸음 방법을 이용하여 차량을 할당할 경우 고객 i 와 j 가 Polar angle의 차이와 서로간의 거리가 멀 경우에도 같은 차량에 할당될 가능성이 존재하여 이동거리와 시간제약 측면에서 좋은 않은 해를 줄 가능성이 높다. 따라서 이러한 문제점을 개선하기 위해 차고지를 중심으로 위치한 고객들의 각도를 이용하여 동일한 크기의 Region으로 나누어 해당 Region의 고객들의 수요량의 합에 따라 차량을 할당하도록 하였다. 동일한 크기의 Region으로 나누는 것은 지역별 수요에 맞는 적합한 차량을 구해 할당된 고객들간의 거리를 최소화하고자 한 것이다.

단계 1) 고객들은 Polar angle, 차량은 적재용량의 오름차순으로 번호를 할당한다. $jump$ 는 다양한 차량할당의 해를 구하기 위해 Region을 나누는 기준이 되는 각도로 모든 단계가 끝나면 15° 가 증가하게 된다. $tripnum$ 은 주어진 문제의 평균 다회방문횟수를 의미한다. 그리고 a 는 나뉘지는 Region의 수, $\angle x$ 는 각 Region이 이루는 각도를 의미한다.

$$jump=0, \quad tripnum = \left\lceil \frac{\sum_{i=1}^n D_i}{\sum_{v=1}^k V_k} \right\rceil,$$

$$a = \left\lceil k \times \frac{\sum_{i=1}^n D_i}{\sum_{v=1}^k V_k} \right\rceil, \quad \angle x = 360/a$$

구한다.

단계 2) 차고지와 $jump$ 의 각도가 이루는 반직선을 중심으로 반시계방향으로 각 Region의 각이 x° 가 되도록 Region을 할당하고 1부터 Region의 번호를 부여한다. 각 Region에 해당하는 Polar angle을 지나는 고객들의 수요의

합 RD_z 을 구한다. 여기서 z 은 Region번호를 의미한다.

단계 3) 차량 1번부터 $V_{k-1} < RD_r \leq V_k$ (단, $V_0=0$)를 만족하는 Region을 찾아 차량을 할당한다. 이는 $trip$ 개의 Region이 할당될 때까지 반복하는데 만약 모든 Region을 검사하여도 $trip$ 만큼 할당되지 않을 경우에는 그만한다. 그리고 모든 차량에 대해서 단계 3을 실시한다. 모든 차량을 실시한 후 얻어진 결과, 모든 Region이 차량에 할당된 경우에는 차량 경로설정단계로 넘어가고 하나의 Region이라도 차량에 할당되지 않는 경우에는 단계 4를 진행한다.

단계 4) $RD_r \neq 0$ 이지만 차량이 할당되지 않은 Region r' 을 찾아 현재 사용가능(차량이 $trip$ 번만큼 Region에 할당되지 않음)한 차량 중 가장 인덱스가 높은 차량의 V_k 보다 $RD_{r'}$ 이 큰 경우 Region에 해당되는 고객 중 $RD_{r'} - V_k$ 보다 큰 수요량을 가지는 고객 i 리스트를 작성한다. 그리고 Region $r'+1$ 과 Region $r'-1$ 을 탐색하여 고객 i 리스트에 있는 것과 Region교환이 일어나도 해가 실행 가능할 경우 고객 j 리스트에 추가한다. 마지막으로 i 와 j 가 교환 가능한 경우 중 교환하고자 하는 고객들의 거리가 가장 작은 경우를 선택하여 Region을 변경한다.

단계 5) $RD_r=0$ 이어서 할당되지 않은 Region의 경우에는 사용가능한 차량을 임의로 할당한다.

3.2 차량경로설정

차량경로설정을 위해 시간제약을 고려하여 고객별로 할당된 차량의 몇 번째 다회방문(Trip)에 경로를 설정할 것인지를 정하기 위해 시간상한과 하한의 등수의 평균값을 사용하였다. 여기서 다회방문(Trip)이란 차량이 차고지를 떠나는 횟수로 Trip번호가 작을수록 먼저 방문이 일어난다는 것을 의미한다. Trip을 정한 이후에는 이동거리측면을 고려하여 경로가 동그랗게 형성되도록 하기 위해 다음과 같은 단계를 진행한다.

단계 1) 각 차량마다 할당된 Region의 속하는 고객들을 하나의 집합으로 만든다. 차량에 할당된 고객들의 시간하한의 등수와 시간상한의

등수를 매겨 이의 평균값(DT_i)이 가장 작은 고객부터 빠른 Trip에 방문이 일어날 수 있도록 한다. DT_i 가 가장 작은 고객의 수요량부터 합하여 차량적용량을 넘으면 Trip수도 1만큼 증가시켜 해당 차량에 할당된 고객들의 Trip을 정해준다.

단계 2) 그림 1처럼 Trip = 1에 속하는 고객들 중 DT_i 가 가장 작은 $E1$ 고객을 방문하고 차고지를 연결한다(그림1(a)). 이 직선을 $L1$ 이라고 하고 경로가 형성되지 않은 고객이 존재하면 그림1(b)처럼 직선 $L1$ 과 직교하는 거리가 가장 짧은 고객 $E2$ 를 찾아 방문하고 $E1$ 과 연결($L2$)한다. 그리고 그림1(c)처럼 두 직선 $L1$ 나 $L2$ 와 직교하는 거리가 가장 짧은 고객 $E3$ 을 찾아 방문하고 연결($L3$)한다. 이런 방식으로 할당된 모든 고객들을 방문할 때까지 반복한다. 밑의 그림1의 경우에는 $E1, E2, E3$ 그리고 $E4$ 의 고객을 방문하는 경로가 생성된다. 단계 3) 모든 Trip을 조사하였으면 마치고 그렇지 않으면 단계 2)로 돌아간다.

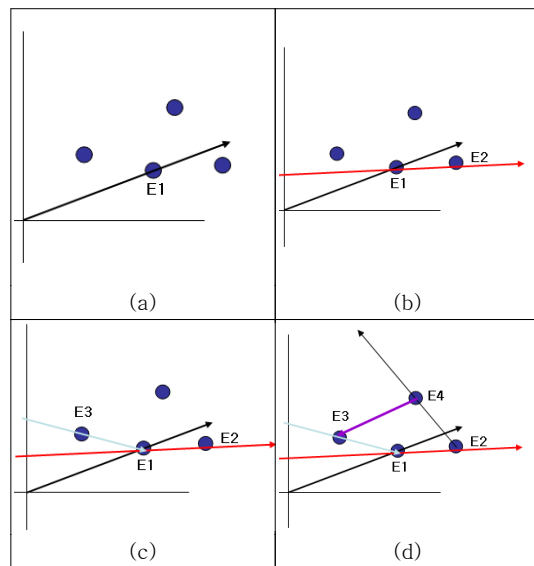


그림 1. 차량경로설정

4. 실험결과 및 분석

실험은 앞에 언급한 수리모델을 ILOG사의 OPL Studio를 이용해 얻은 최적해와 제시한 알고리즘의 해를 비교하고자 한다. 하지만 OPL Studio를 이용해 1시간 안에 최적해가 나

오지 않으면 실험을 중단하고 그때까지 도출된 해 중 가장 좋은 해를 최적해로 가정하였다. 실험은 다회방문을 가지는 임의로 생성된 데이터를 바탕으로 고객이 6곳에서 12곳으로 존재하고, 2대에서 5대의 차량이 존재하는 경우를 임의로 생성하여 실험하였다. 여기서 w_1 은 이동거리를, w_2 는 고객서비스 시간 위반정도를 목적함수에 포함할지를 결정하는 수로 포함되면 1, 그렇지 않으면 0의 값을 지닌다. p_1 은 시간하한보다 고객에 차량이 빨리 도착한 경우, p_2 는 시간상한보다 고객에 차량이 늦게 도착한 경우를 각각 목적함수에 포함할지를 결정하는 수로 포함되면 1, 그렇지 않으면 0의 값을 지닌다.

표1의 경우는 이동비용의 최소화를 목적으로 실험한 결과이다.

표1. 성과비교($w_1=1, w_2=0$)

instance	최적해 ($w_1=1, w_2=0$)			제안된 알고리즘 ($w_1=1, w_2=0$)			
	value	소요 차량	계산 시간	value	소요 차량	계산 시간	GAP
1	476*	2	77.88	641	2	0.1	34.66
2	635*	2	22.42	660	2	0.1	3.94
3	790*	1	436.5	1104	2	0.1	39.75
4	1306	1	1시간	1409	2	0.1	7.89
5	1270*	2	2476	1428	2	0.1	12.44
6	1307	3	1시간	1473	3	0.1	12.70
7	1320	2	1시간	1041	2	0.1	-21.14
8	1458	4	1시간	874	2	0.1	-40.05

(*는 최적해 값을 나타냄, 계산시간단위 : 초)

$$GAP(\%) = \frac{\text{suggested algorithm value} - \text{optimal value}}{\text{optimal value}} \times 100$$

실험결과 최적해가 제공되지 않은 경우에 제안된 알고리즘의 성과가 10.15% 높은 것으로 나타났다.

다음으로 표2는 이동비용과 시간제약을 동시에 고려하여 이동비용과 시간제약에 대한 벌금비용을 최소화하려는 목적을 가지고 실험한 결과이다.

표2. 성과비교($w_1=1.0, w_2=1.0$)

instance	최적해 ($w_1=1.0, w_2=1.0$)			제안된 알고리즘 ($w_1=1.0, w_2=1.0$)			
	value	소요 차량	계산 시간	value	소요 차량	계산 시간	GAP
1	631*	2	113.4	798	2	0.1	26.47
2	691*	2	33.73	766	2	0.1	10.85
3	1267*	2	644.6	1649	2	0.1	30.15
4	2302	1	1시간	1921	2	0.1	-16.55
5	1646	2	1시간	2260	2	0.1	37.30
6	1359	3	1시간	1592	3	0.1	17.14
7	1862	2	1시간	1304	2	0.1	-29.97
8	1421	4	1시간	1344	2	0.1	-5.42

표3의 경우에는 시간제약의 만족을 최대화하기 위한 목적으로 실험한 결과이다. 최적해가 존재하지 않을 경우 제안된 알고리즘의 성과가 0.5% 안 좋은 것으로 나타났다.

표3. 성과비교($w_1=0, w_2=1$)

instance	최적해 ($w_1=0, w_2=1$)			제안된 알고리즘 ($w_1=0, w_2=1$)			
	value	소요 차량	계산 시간	value	소요 차량	계산 시간	GAP
1	30*	2	61.05	157	2	0.1	423.33
2	39*	2	17.00	50	2	0.1	28.21
3	10*	2	166.4	295	2	0.1	2850
4	920	1	1시간	485	2	0.1	-47.28
5	362*	2	1936	832	2	0.1	129.83
6	0*	3	250.4	119	3	0.1	11800
7	136	2	1시간	29	2	0.1	-78.38
8	0	4	1시간	470	2	0.1	46900

표3의 결과를 보면 제안된 알고리즘의 해가 좋지 않은 것으로 측정되었다. 실험결과를 통해 이동비용 측면에서는 좋은 결과를 주나 시간제약에 대한 벌금비용이 고려될 경우에는 좋은 결과를 주지 못하였다.

5. 결론

본 논문은 시간제약과 다회방문 차량경로문제를 대상으로 하며 총 운송비용을 최소화하면서 고객서비스시간제약 만족을 최대화하는 것이 목적이다. 고객수가 6~12곳인 경우에 한해 2장에 제시한 수리모형으로 최적해를 비교한 결

과 시간제약을 고려할 경우 좋은 해를 주지 못하는 것으로 나타났다. 이는 경로를 구성할 때 시간제약에 따라 Trip을 결정해주나 이동거리를 기반으로 하여 경로를 결정하기 때문에 시간의 제약조건을 만족하지 못하고 있다. 이는 향후에 초기해를 가능해로 유지하면서 개선이 될 수 있도록 메타 휴리스틱기법을 이용한 알고리즘을 고려해 볼 수 있다.

참고문헌

- [1] Barrie M. Baker and M.A. Ayechew, 2003, A genetic algorithm for the vehicle routing problem, *Computers & Operations Research*, 30, 787-800.
- [2] Clarke, G., Wright, J.W., 1964, Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581.
- [3] Jose Brandao, Alan Mercer, 1997, *European Journal of Operational Research*, 100, 180-191.
- [4] Lenstra, J., Rinnooy Kan, A., 1981, Complexity of vehicle routing and scheduling problems, *Networks*, 11, 221-227.
- [5] Or, I., 1976, Traveling salesmand-type combinatorial problems and their relation to the logistics of blood banking. Ph.D. thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.
- [6] Osman, I.H., 1993, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, 41, 421-451.
- [7] Taillard, E., Laporte, G., Gendreau, M., 1996, Vehicle routing with multiple use of vehicles, *Journal of the Operational Research Society*, 47, 1065-1070.
- [8] Potvin, J.Y., Rousseau, J.M., 1995, An exchange heuristic for routing problems with time windows, *Journal of the Operational Research Society*, 46, 1433-1446.