

Hybrid-PSO 해법을 이용한 수요지 제한이 있는 다용량 차량경로문제

이상현[†] · 황선호

국방대학교 운영분석학과

Heterogeneous Fleet Vehicle Routing Problem with Customer Restriction using Hybrid Particle Swarm Optimization

Sang-Heon Lee · Sun-Ho Hwang

Department of Operations Research, Korea National Defense University, Seoul, 122-875

The heterogeneous fleet vehicle routing problem(HVRP) is a variant of the classical vehicle routing problem in which customers are served by a heterogeneous fleet of vehicles with various capacities, fixed costs and variable costs. We propose a new conceptual HVRPCR(HVRP with customer restriction) model including additional customer restrictions in HVRP. In this paper, we develop hybrid particle swarm optimization(HPSO) algorithm with 2-opt and node exchange technique for HVRP. The solution representation is a n-dimensional particle for HVRP with N customers. The decoding method for this representation starts with the transformation of particle into a priority list of customer to enter route and limit of vehicle to serve each customer. The vehicle routes are then constructed based on the customer priority list and limit of vehicle to serve. The proposed algorithm is tested using 8 benchmark problems and it consistently produces high-quality solutions, including new best solutions. The numerical results show that the proposed algorithm is robust and efficient.

Keywords: Heterogeneous Fleet Vehicle Routing Problem, Particle Swarm Optimization, Customer Restriction

1. 서론

2005년 톤-km 기준 화물수송실적을 수송 수단별로 살펴보면, 도로화물 수송실적의 연평균 증가율(2001~2005년)은 약 2.64% 증가한 것으로 분석되어 도로부분에 대한 수송 의존도가 지속적으로 심화되고 있는 것으로 나타났다. 또한 2005년 기능별 국가물류비 추이에서도 도로화물 수송비가 74조 3,131억 원으로 전체 수송비의 96.56%의 비중을 차지하고 있어 국가물류활동의 도로의존도가 일본(44.50%), 미국(84.25%)에 비해서 매우 높은 수준으로 나타났다. 이에 따라 각 기관과 기업은 도로화물 수송비를 절감하고 효율성을 증대시키기 위해 다각도의 노력을 기울이고 있다.

운송비 최적화를 위한 다양한 방법 중 제품을 수송하는 운

송회사의 관점에서 개별 고객들의 수요를 모두 충족시키면서 전체 운송비용을 최소화하는 것을 수리적으로 다룬 문제가 차량경로문제(VRP : Vehicle Routing Problem)이다. VRP는 복수외판원문제의 확장된 개념으로 제한된 용량을 가진 복수차량이 지리적으로 분산되고 수요량이 서로 다른 수요지에 대하여 제품이나 서비스를 효율적으로 수송할 수 있도록 차량 할당 및 운행경로를 결정함으로써 전체 운행비용을 최소화하는 문제이다. VRP는 그 목적과 제약사항에 따라 여러 가지 변형된 형태로 연구되어 왔다.

본 연구에서는 VRP의 여러 가지 변형 중 복수차량유형과 차량별 일일 방문가능 수요지가 정해진 제약사항 하에서 총 차량 운영비용(고정비, 변동비)을 최소화하는 문제를 다룬다. 이를 위해 최소 차량 운영비용을 만족하는 차량 유형의 조합과

[†] 연락처 : 이상현 교수, 122-875 서울시 은평구 수색동 205번지 국방대학교 운영분석학과, Tel : 02-300-2374, Fax : 02-309-6233,

E-mail : leesangh@kndu.ac.kr

2008년 □월 □일 접수; 2008년 □월 □일 수정본 접수; 2008년 □월 □일 게재 확정.

차량별 이동경로, 이동거리를 산출할 수 있도록 Hybrid Particle Swarm Optimization(HPSO) 알고리즘을 제시함으로써 합리적이고 효율성 있는 수송정책 수립에 기여하는데 목적이 있다.

2. 기존해법 고찰

HVRP(Heterogeneous Fleet Vehicle Routing Problem)는 용량이 상이한 차량들의 서로 다른 운행비용(고정비, 변동비)의 발생으로 총 운행비용을 최소로 하는 차량의 조합을 구하는 문제이다. HVRP의 초기연구로 Golden *et al.*(1984)은 우선 수요지를 모두 연결시켜 거대한 경로를 만들고 난 후 전체 비용이 최소화가 되게끔 여러 개의 경로로 나누어서 사용된 차량의 총 고정비(fixed cost)를 최소화하는 것으로서 Giant tour와 3-opt를 이용하는 방법을 제시하였다. Salhi and Rand(1993)는 Clark and Wright(1964)의 Saving기법을 응용하여 초기해를 구한 다음 각 경로별로 포함하고 있는 지점을 제거, 경로 합침, 경로분할, 지점교환의 과정을 차례로 거치는 과정을 여러 번 반복하여 동일한 용량 차량문제의 해를 구하였다. Ochi *et al.*(1998)은 섬모델에 기초한 병렬유전알고리즘을 이용하여 프로세서별로 각기 해 탐색을 진행하면서 주기적으로 이웃섬과 유전자 교환 및 돌연변이를 거치는 진화과정을 통하여 Taillard(1999)의 해와 유사한 결과를 보였다. Taillard(1999)는 Glover(1997)의 AMP

(Adaptive Memory Procedure)를 이용하여 각 차량용량별로 초기해를 구하고 그 해들 중에서 여러 개의 우수한 경로를 선별하여 메모리에 저장시킨 뒤 그들의 조합을 이용하여 소비비용이 소요되는 해집합인 차량경로를 구성하는 방법을 제안하였다. Gendreau *et al.*(1999)은 TSP를 풀기 위해 고안된 GENIUS 삽입기법을 기반으로 타부서치를 적용하였는데 무작위로 고객들을 경로 간에 이동시킨 후 경로 내 지점방문순서를 GENIUS 삽입기법을 이용하여 개선시키는 절차를 반복수행해서 Taillard(1999)보다 우수한 해를 도출하였다. Li *et al.*(2007)은 Dueck(1993)이 개발한 record-to-record 기법에 2-opt, or-opt, 1 point move, 2 point move를 삽입한 HRTR(hybrid record to record travel algorithm)을 사용하여 Taillard(1999)보다 우수한 해를 도출하였으며 수요지가 확장된 200~360개의 문제에도 적용하였다. Choi and Tcha(2007)는 선형계획법에 기반한 column generation 기법에 dynamic programming을 사용하여 기존의 해와 동일하거나 우수한 해를 도출하였으며, Imran *et al.*(2008)은 초기해를 sweep algorithm으로 giant tour를 구한 다음 2-opt로 개선시킨 후 Dijkstra algorithm을 사용하고, VNS(variable neighborhood search)를 사용하여 최종해를 구하였다.

HVRP는 목적함수의 설정에 따라 여러 가지 형태로 구분되는데 기존에 연구되었던 목적함수와 방법을 요약하면 <Table 1>과 같다. Taillard(1999)의 목적함수는 차량용량별 대수가 제한되어 있다는 점에서 차량용량별 대수가 무제한인 Salhi *et*

Table 1. Summary of the HVRP related papers

Authors	Objective Function	Method used
Golden <i>et al.</i> (1984)	$\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Giant-tour • 3-opt
Salhi <i>et al.</i> (1993)	$\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Saving algorithm
Ochi <i>et al.</i> (1998)	$\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Genetic algorithm
Taillard(1999)	$\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Tabu search algorithm • AMP(adaptive memory procedure)
Gendreau <i>et al.</i> (1999)	$\sum(\text{Distance} \times \text{Variable Cost each vehicle})$ $\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Tabu search algorithm • Genius insertion
Li <i>et al.</i> (2007)	$\sum(\text{Distance} \times \text{Variable Cost each vehicle})$ $\sum(\text{Distance}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Record-to-record travel algorithm • 2-opt, or-opt, point move
Choi and Tcha(2007)	$\sum(\text{Distance} \times \text{Variable Cost each vehicle})$ $\sum(\text{Distance}) + \sum(\text{Fixed Cost})$ $\sum(\text{Distance} \times \text{Variable Cost each vehicle}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Column generation + Dynamic programming
Imran <i>et al.</i> (2008)	$\sum(\text{Distance} \times \text{Variable Cost each vehicle})$ $\sum(\text{Distance}) + \sum(\text{Fixed Cost})$ $\sum(\text{Distance} \times \text{Variable Cost each vehicle}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Variable neighborhood-based heuristic
This study	$\sum(\text{Distance}) + \sum(\text{Fixed Cost})$ $\sum(\text{Distance} \times \text{Variable Cost each vehicle}) + \sum(\text{Fixed Cost})$	<ul style="list-style-type: none"> • Particle Swarm Optimization • 2-opt, exchange

al.(1993)과는 다르다. Gendreau *et al.*(1999)과 Li *et al.*(2007)의 HVRP는 기존의 두 가지 목적함수(변동비와 이동거리, 고정비와 이동거리)에 대해서 각각 한 실험을 한 경우이다. 고정비와 변동비를 동시에 고려한 연구는 미비하였으나 최근 Choi and Tcha(2007)와 Imran *et al.*(2008)이 연구하였다.

본 연구에서는 제품이 매일 출하되어 전국에 위치한 공장, 대형 유통점 등의 거래처로 수송되는 상황에서 일일 가용한 시간 내 모든 수요지를 방문하는 최소비용 차량수송계획을 수립하거나, 신규설립 물류센터나 공장 등에서 물품을 수요지로 배달하기 위해 필요한 차량의 종류별 구매량 결정 등의 시장 상황에 적용하기 위하여 기존 HVRP에 수요지 제한이라는 제약사항을 추가하였다. 또한 HVRP 해결을 위하여 알고리즘이 단순하고, 계산시간이 짧으며 대량의 메모리가 필요 없고, 연속형과 비연속형 양쪽 모두 적용가능하다는 장점을 가지고 있는 Particle Swarm Optimization(PSO) 기법을 적용하였다.

즉, 본 연구는 기존의 HVRP 연구에서 방문가능 수요지 수 제한이라는 제약사항을 추가하여 운행차량의 고정비와 변동비를 동시에 고려했다는 것과 메타휴리스틱 기법 중 비교적 최신기법인 PSO를 적용하여 차량의 총 운행비용이 최소가 되는 차량조합과 차량별 이동경로를 산출할 수 있게 알고리즘을 구성하였다는 점에서 타 연구와의 차이점이 있다.

3. 문제 정의 및 수리모형

3.1 문제정의

본 연구의 문제는 기존의 HVRP에 방문가능 수요지 수를 제한하는 것으로 이를 HVRPCR(Heterogeneous Fleet Vehicle Routing Problem with Customer Restriction)로 정의한다. HVRPCR 모형은 전체 네트워크상에 유일한 공급지인 창고(depot)와 N 개의 수요지점이 있다. 창고는 수요지의 수요량을 충족시킬 수 있도록 적재용량이 다양한 차량을 충분히 보유하고 있다. 각 수요지점의 수요량과 위치는 알려져 있으며 차량이 방문하는 수요지 수요량의 합은 차량의 용량을 초과할 수 없다. 이때 모든 수요지는 한 대의 차량에 의해서 한번만 방문되고 운행되는 차량의 총 운행비용은 차량 고정비와 운행거리에 따른 변동비의 합이 최소가 되도록 한다. 문제의 단순화를 위한 가정 사항은 다음과 같다.

- (1) 각 차량은 일정한 수요지 수 하한 이상을 방문하여야 하며 상한을 초과하여 방문할 수 없다.
- (2) 차량의 출발 및 종착은 창고에서만 이루어진다.
- (3) 각 차량경로에 포함된 수요지점 수요량의 합은 최대 적재용량을 초과할 수 없다.
- (4) 차량의 적재용량은 종류별로 다르며 종류별로 충분한 차량을 보유하고 있다.

3.2 수리모형

HVRPCR 모형에 사용되는 변수와 기호들은 다음과 같다.

<Notation>

N	: 수요지 수(고객 수)
Y	: 수요지의 부분집합(subset)
i, j	: 수요지($i, j = 1, 2, \dots, N$; $i, j = 0$: depot)
q_i	: 수요지 i 의 수요량(배달량)($i = 1, 2, \dots, N$)
u	: 차량 종류($u = 1, 2, \dots, U$)
Q_u	: 차량 종류별 적재용량($u = 1, 2, \dots, U$)
k	: 차량 종류별 가용 차량번호($k = 1, 2, \dots, K$)
S_{up}, S_{low}	: 방문가능 수요지 수 상한과 하한($S_{low} = 1$)
f_u	: 차량 종류별 고정비
g_u	: 차량 종류별 변동비
d_{ij}	: 수요지 i 와 j 간 이동거리($i, j = 1, 2, \dots, N$; $i, j = 0$: depot)

$$X_{ijk} = \begin{cases} 1, & i \text{에서 } j \text{까지 차량종류 } u \text{의 } k \text{번째 차량이 운행시} \\ 0, & \text{otherwise} \end{cases} \quad (i, j = 1, 2, \dots, N; i, j = 0: \text{depot})$$

$$(u = 1, 2, \dots, U)$$

$$(k = 1, 2, \dots, K)$$

위에서 정의한 변수로 가장 일반적인 HVRP 모형인 Gendreau *et al.*(1999)의 모형에 수요지 제한사항을 추가하여 구성된 HVRPCR 수리모형은 다음과 같다.

Minimize

$$\sum_{u=1}^U \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N d_{ij} g_u X_{ijk} + \sum_{u=1}^U \sum_{k=1}^K \sum_{j=1}^N f_u X_{0jk} \quad (1)$$

Subject to

$$\sum_{u=1}^U \sum_{k=1}^K \sum_{i=0}^N X_{ijk} = 1 \quad j = 1, 2, \dots, N \quad (2)$$

$$\sum_{u=1}^U \sum_{k=1}^K \sum_{j=0}^N X_{ijk} = 1 \quad i = 1, 2, \dots, N \quad (3)$$

$$\sum_{a=1}^N X_{abuk} - \sum_{c=0}^N X_{bcauk} = 0 \quad \begin{matrix} b = 0, 1, \dots, N \\ k = 1, 2, \dots, K \\ u = 1, 2, \dots, U \end{matrix} \quad (4)$$

$$S_{low} \leq \sum_{i=0}^N \sum_{j=1}^N X_{ijk} \leq S_{up} \quad \begin{matrix} k = 1, 2, \dots, K \\ u = 1, 2, \dots, U \end{matrix} \quad (5)$$

$$\sum_{i \in Y} \sum_{j \in Y} X_{ijk} \leq |Y| - 1 \quad \begin{matrix} k = 1, 2, \dots, K \\ u = 1, 2, \dots, U \\ Y \in N, |Y| \neq 0 \end{matrix} \quad (6)$$

$$X_{ijk} \in \{0, 1\} \quad \forall i, j, u, k \quad (7)$$

식 (1)의 앞부분은 차량의 변동비를 나타내며 뒷부분은 차량

의 고정비를 나타낸다. 식 (2)와 식 (3)은 각 수요지에는 한 대의 차량만이 방문할 수 있다는 일회방문을 의미하여 중복방문이 없도록 제한한다. 식 (4)는 차량이 임의의 수요지점에 도착하면 화물의 하역을 마친 후에 반드시 다른 수요지점이나 창고로 출발해야 한다는 차량흐름의 연속성을 나타내는 제약이다. 식 (5)는 차량이 방문하는 수요지 수의 상한과 하한이 제한되어 있다는 것을 나타낸다. 식 (6)은 부분경로(sub-tour)방지를 위한 조건으로서 Y 는 수요지의 부분집합을 나타낸다. 각 차량에 의해 형성되는 경로들은 반드시 창고에서 최초로 출발하고, 최종적으로 창고로 귀환해야 한다. 따라서 차량 경로는 반드시 공급지와 연결되는 경로가 있어야 한다. 본 모형에서는 제약조건에 차량의 용량과 방문가능 수요지수의 상한과 하한만을 고려하였으며 차량의 총 가용시간이나 총 가용거리 등은 수요지 수의 상한과 하한을 제한하는 것에 부분적으로 포함되어 있어 고려하지 않았다.

4. HVRPCR-HPSO 알고리즘

Particle Swarm Optimization(PSO)은 Kennedy and Eberhart(1995)에 의해 제안된 진화형 계산 기법이다. PSO는 자연선택의 진화 메커니즘이 아닌 새나 물고기 무리와 같은 생체군집의 사회적 행동양식을 바탕으로 하여 확률적 최적화 방법에 대한 연구에서 시작되었다. 이러한 무리가 먹이를 찾아가는 과정에서 무리 전체가 정보를 공유한다는 가설과 무리 내부의 각 particle이 지금까지의 자신의 경험이 무리 전체에 공유되어 있는 정보를 기초로 하여 행동한다는 개념을 최적화 과정에 도입한 기법이라 할 수 있다. PSO는 군집기반 알고리즘으로 병렬처리 특성을 가지고 군집과 개체는 각각 swarm과 particle로 표현된다.

PSO의 각 particle은 최적해를 얻기 위하여 다차원 탐색공간을 날아다니며 자신과 이웃의 경험에 대한 정보를 이용하여 최적의 위치로 이동한다. 이를 위해 particle은 이전에 경험했던 최적의 위치정보에 대한 기억을 유지한다. 또한 PSO는 파라미터의 조절을 통해 개체의 전역탐색과 지역탐색 능력이 균형을 유지할 수 있다. 일반적으로 PSO는 이론의 간결성, 구현의 용이성, 연산의 효율성과 같은 특징을 나타내고 짧은 계산 시간 내에 최적해를 생성할 수 있으며 다른 확률적 방법보다 안정적인 수렴특징을 나타내고 있다.

일반적인 PSO 알고리즘은 모집단의 각 particle의 위치가 랜덤하게 생성되는 것으로 시작된다. 각 particle은 velocity update rule과 position update rule에 의해 전역적인 해 탐색을 시작한다. 탐색과정에서 발견한 해 중에 최적해($pbest$)를 기억하고 있으며, 집단에서 발견한 해 중에 최적해($gbest$)와 그 해의 위치벡터 x^{gbest} 의 정보를 공유한다. 각 particle은 현재 위치벡터와 속도벡터, 그리고 x_{pbest} , x_{gbest} 를 이용해서 식 (8)에 의해 이동하게 되며, 위치벡터의 수정은 현재의 위치와 수정된 속도를 이용해서 식 (9)와 같이 행해진다. 이와 같은 과정을 반복함

으로서 최적해를 찾아가게 된다.

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 \times [x_i^{pbest} - x_{id}(t)] + c_2 r_2 \times [x^{gbest} - x_{id}(t)] \quad (8)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (9)$$

본 장에서는 위에서 구성한 HVRPCR 모형의 목적함수인 최소 운행비용의 차량조합을 산출하기 위해 PSO에 2-opt와 exchange 기법을 삽입한 HVRPCR-HPSO 알고리즘을 구성하였다. Step 1에서 n 개의 particle이 초기화되고, Step 2는 생성된 난수로 차량의 경로를 해석하고 적합도를 평가하게 되며, Step 3~4에서 각 particle의 최고해($pbest$)와 전체 particle의 최고해($gbest$)를 갱신한다. Step 5에서 n 개의 particle은 업데이트되며, Step 6에 의해 알고리즘은 지정된 반복횟수만큼 진행된다. 특히, Step 3에서 빠른 시간 내 해를 도출하기 위하여 2-opt와 exchange 기법을 삽입하여 경로를 개선시켰다. 알고리즘의 전체적인 흐름은 <Figure 1>에 설명하고, 알고리즘에 사용한 기호와 parameter는 다음과 같다.

<Notation>

- i : Particle index; $i = 1, \dots, n$
- d : Dimension index; $d = 1, \dots, D$

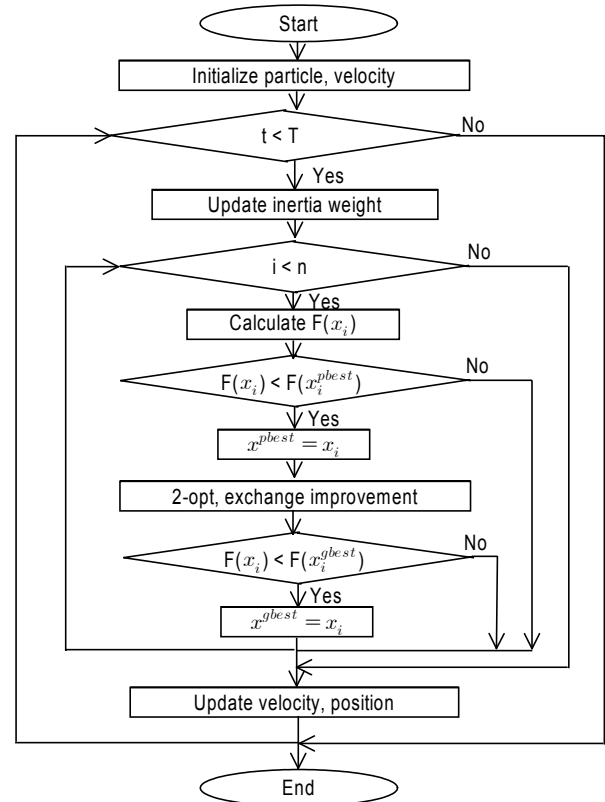


Figure 1. Overview of the HVRPCR-HPSO algorithm

- t : Iteration index; $t = 1, \dots, T$
- $\omega(t)$: t 시점의 관성가중치; $w(1) = w_{up}$, $w(T) = w_{low}$
- $v_{id}(t)$: t 시점에서 particle i 의 d 차원 velocity
- $x_{id}(t)$: t 시점에서 particle i 의 d 차원 position
- $v_i(t)$: t 시점에서 particle i 의 속도벡터;
 - $v_i = [v_{i1}, \dots, v_{iD}]$
- $x_i(t)$: t 시점에서 particle i 의 위치벡터;
 - $x_i = [x_{i1}, \dots, x_{iD}]$
- x_i^{pbest} : particle i 의 가장 좋은 위치벡터;
 - $x_i^{pbest} = [x_{i1}^{pbest}, \dots, x_{iD}^{pbest}]$
- x^{gbest} : 전체 particle 중 최고의 위치벡터;
 - $x^{gbest} = [x_1^{gbest}, \dots, x_D^{gbest}]$
- $F(x_i)$: particle i 의 적합도
- c_1 : $pbest$ 의 가속변수
- c_2 : $gbest$ 의 가속변수
- r_1, r_2 : 0과 1사이의 난수[0, 1]

여기에서 $\omega(t)$ 는 t 시점의 관성가중치로 벡터의 이동속도가 반복횟수가 많아질수록 점점 작아짐을 나타내고 inertia weight 또한 동일한 의미이다. swarm size는 문제에서 생성할 particle의 수를 의미한다.

- Step 1. n 개의 particle을 초기화하기 위해, D 개의 난수로 x_i 를 만들고, $v_i = 0$, $x_i^{pbest} = x_i$, $t = 0$ 으로 설정한다.
- Step 2. <Figure 2>와 같이 particle별로 경로를 해석하고, $F(x_i)$ 를 계산한다.
- Step 3. Update $pbest$: $F(x_i) < F(x_i^{pbest})$ 일 때, $x_i^{pbest} = x_i$ 라 하고, 2-opt와 Exchange 기법을 사용하여 x_i 의 경로를 개선한다.
- Step 4. Update $gbest$: $F(x_i^{new}) < F(x^{gbest})$ 일 때, $x^{gbest} = x_i^{new}$ 라 선정한다.
- Step 5. Update x_i, v_i : 아래의 식 (10), (11), (12)에 따라 x_i, v_i 를 업데이트 한다.

$$W(t) = (w_{up} - w_{low}) \times (T - t) / T + w_{low} \quad (10)$$

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 \times [x_i^{pbest} - x_{id}(t)] + c_2 r_2 \times [x^{gbest} - x_{id}(t)] \quad (11)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (12)$$

Step 6. $t = T$ 일 때, 알고리즘을 종료하고 그렇지 않으면 $t = t + 1$ 로 선정하고 Step 2로 간다.

Step 7. T 시점의 $gbest$ 를 최고해로 선정하고, $gbest$ 가 가지고 있는 D 개의 벡터로 차량의 경로와 운행비용 등을 출력한다.

알고리즘 진행에 따라 벡터공간에서 particle의 위치가 변경되면서 점점 더 좋은 해를 가지게 되고 $pbest$ 와 $gbest$ 의 정보는 update 된다. 그 정보는 D 개의 난수로 구성되고 이는 차량별 경로로 표현이 된다.

4.1 경로표현 및 초기해 생성

PSO의 기본개념은 확률적 최적화 기법에 의해 연속된 공간을 탐색하는 것이다. PSO를 조합최적화문제인 HVRP에 적용하기 위해 난수 표현법을 사용하여 기존의 조합탐색을 연속공간탐색으로 변환하였다. 연속 공간은 총 수요지의 수인 D 차원의 벡터공간이며 구간은 $[0, 1]^D$ 벡터공간이다. 초기해 생성은 수요지마다 무작위로 난수를 발생시키는 것으로부터 시작된다. N 개의 난수를 오름차순으로 정렬하여 수요지별 우선순위를 정하고, 차량별 할당된 수요지 수만큼 차례대로 할당하여 n 개의 particle마다 차량별 경로를 가진 초기해를 구성하였다. 이때, 초기 차량의 수는 노드 수만큼 있다고 가정하였고, 차량의 종류는 해당경로의 수요량의 합에 의해 결정된다. 수요지별로 발생된 난수로 차량별 경로를 할당하는 방법은 <Figure 2>에 나타내었다. <Figure 2>의 priority list는 랜덤에 의하여 생성된 난수를 재배열하여 수요지의 순서를 정하는 목록을 의미한다.

4.2 적합도 평가

n 개 particle의 초기해가 구성되면, particle마다 적합도를 평

Customer :	1	2	3	4	5	6
Random :	0.9721	0.023	0.3489	0.7584	0.1564	0.7319
Customer Priority List	2 → 5 → 3 → 6 → 4 → 1					
Vehicle 1 : 4 Customers	V ₀ → 2 → 5 → 3 → 6 → V ₀					
Vehicle 2 : 2 Customers	V ₀ → 4 → 1 → V ₀					
❖ V ₀ : Depot						

Figure 2. Method to create route using random number

가한다. 적합도는 식 (13)과 같이 단위거리당 차량별 변동비에 거리를 곱한 값이다.

$$\sum_{u=1}^U \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N d_{ij} g_u X_{ijuk} \quad (13)$$

Particle i 에서 최고해를 가지는 $pbest$ 와 전체 particle 중 최고해를 가지는 $gbest$ 는 식 (14), (15)와 같이 적합도 비교를 거쳐 선정한다.

If $F(x_i) < F(x_i^{pbest})$,
then $x_i^{pbest} = x_i, x_i = x_i^{new}$ (14)

If $F(x_i^{new}) < F(x^{gbest})$,
then $x^{gbest} = x_i^{new}$ (15)

특히, 식 (14)에서 $pbest$ 선정 후 2-opt와 Exchange 기법을 적용하여 $pbest$ 의 경로를 개선하여 particle i 의 개선된 $pbest$ 의 벡터공간의 좌표인 x_i^{new} 를 생성하고, 식 (15)에서 x_i^{new} 를 다시 $gbest$ 와 비교하여 x_i^{new} 적합도가 우수하면 $x^{gbest} = x_i^{pbest}$ 로 선정한다.

4.3 경로개선

위에서 언급한 경로개선은 2-opt와 1-1 exchange, 1-1-1 exchange를 사용한다. 2-opt와 exchange는 14개 노드의 TSP 문제인 burma 14(Golden et al., 1984)의 실험결과 PSO만 사용하였을 경우 10회중 4~5회 정도의 최적해가 산출되나, 2-opt와 exchange 기법을 적용하였을 때, 9~10회의 최적해가 산출되었고, 추가 소요되는 시간은 2~3초로 효율적으로 최적해를 산출할 수 있어 본 연구에서 경로개선 2가지 기법 모두 적용하였다.

4.3.1 2-opt

2-opt는 주어진 경로중에서 2개의 노드를 선택하여 바꾸고, 선택된 2개 노드사이의 경로를 역으로 바꾸어 적합도를 반복적으로 평가하여 기존해보다 우수한 해를 찾아내는 방법이다. 본 알고리즘에 적용된 2-opt는 <Figure 3>, <Figure 4>와 같이 단일경로내 2-opt와 경로간 2-opt의 두 가지 경우를 동시에 고려할 수 있도록 알고리즘을 구성하였으며, 그 내용은 다음과 같다.

Step 1. Set N = number of customers in the route.

Step 2. For $i = 1, \dots, (N-1), j = (i+2), \dots, N$.

i, j = sequence number

a. $i+1 \Leftrightarrow j$

b. $i+2 \Leftrightarrow j-1, i+3 \Leftrightarrow j-2, i+4 \Leftrightarrow j-3, \dots$

c. if $F(x_i^{new}) > F(x_i)$, then $x_i = x_i^{new}$, goto Step 2a.

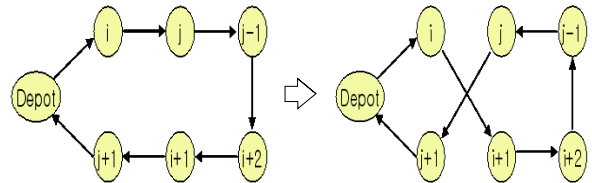


Figure 3. Illustration of 2-opt single route procedure

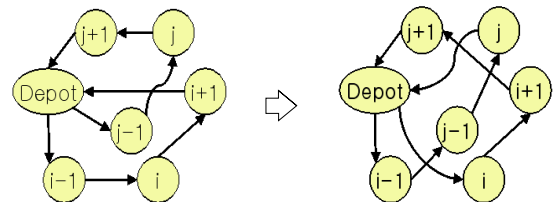


Figure 4. Illustration of 2-opt multiple route procedure

4.3.2 1-1 exchange

1-1 exchange 기법은 선택된 2개의 노드를 서로 바꾸어주는 것으로 2-opt와 유사하다. 2개의 노드를 바꾸어주는 과정은 <Figure 5>에 나타내었고, 구성된 알고리즘은 다음과 같다.

Step 1. Set n =numbers of customer in the route.

Step 2. For $i = 1, \dots, (n-1), j = (i+1), \dots, N, i, j$ = sequence number

a. $i \Leftrightarrow j$

b. if $F(x_i^{new}) > F(x_i^{pbest})$, then $x_i^{pbest} = x_i^{new}$,

goto Step 2a.

4.3.3 1-1-1 exchange

1-1-1 exchange 기법은 3개의 노드를 서로 바꾸어주는 것으로 예를 들면, 1-2-3의 경로는 3-1-2와 2-3-1로 바꾸어 줄 수 있다. 3개의 노드를 선택하여 바꾸어주는 과정은 <Figure 6>과 같으며, 구성된 알고리즘은 다음과 같다.

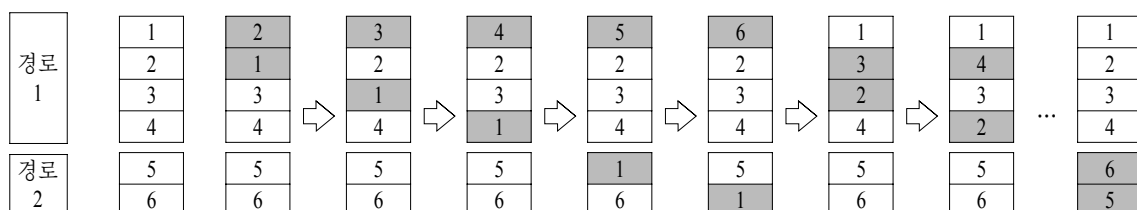


Figure 5. Illustration of 1-1 exchange procedure

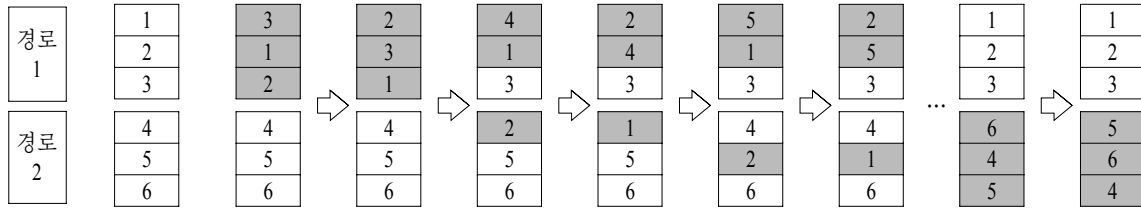


Figure 6. Illustration of 1-1-1 exchange procedure

- Step 1. Set $N =$ numbers of customer in the route.
- Step 2. For $i = 1, \dots, (N-2), j = (i+1), \dots, (N-1), k = (j+1), \dots, N,$
 $i, j, k =$ sequence number
 - a. $i = j, j = k, k = i$
 - b. if $F(x_i^{new}) > F(x_i^{pbest}),$ then $x_i^{pbest} = x_i^{new}$
 - c. $i = k, j = i, k = j$
 - d. if $F(x_i^{new}) > F(x_i^{pbest}),$ then $x_i^{pbest} = x_i^{new},$
 goto step 2a.

4.4 위치 · 속도벡터 업데이트

적합도 평가 과정을 거쳐 particle i 의 최고해 $pbest$ 와 전체 particle의 최고해 $gbest$ 가 결정되면, 앞에서 기술한 식 (7), 식 (8)에 따라 particle은 위치와 속도를 갱신하게 된다. 식 (7)은 현재 위치에서 가장 좋은 해를 가지는 x_i^{pbest} , 집단의 가장 좋은 해 x^{gbest} 와 이전의 속도벡터인 $v_{id}(t)$ 를 기준으로 새로운 속도벡터 $v_{id}(t+1)$ 를 계산하는 것을 나타낸다. 식 (8)은 particle의 위치가 고차원 공간(hyperspace)의 위치에서 변경된 속도벡터 v_i^{next} 와 x_i 에 의해 새로운 위치 x_i^{next} 로 업데이트됨을 의미한다. particle i 의 위치와 속도변경에 대한 개념은 <Figure 7>과 같다.

<Figure 7>에서 $x_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ 와 $v_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ 는 각각 i 번째 particle의 위치벡터와 속도벡터, $x_i^{pbest} = [x_{i1}^{pbest}, x_{i2}^{pbest}, \dots, x_{iD}^{pbest}]$ 는 i 번째 particle의 가장 좋은 해의 위

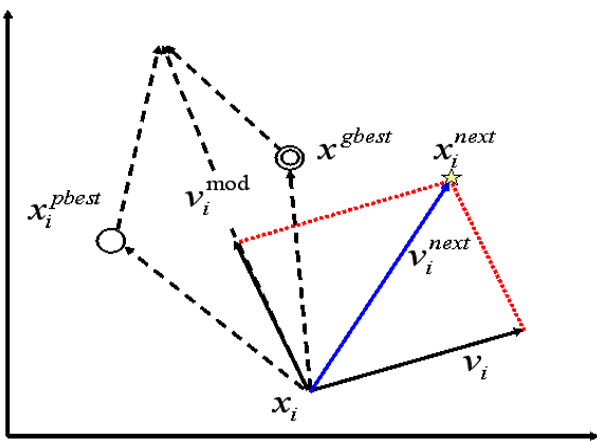


Figure 7. Concept of particle update position and velocity

치를 나타내는 벡터값으로 정의된다. r_1, r_2 는 독립적으로 발생하는 난수로 범위는 $[0, 1]$ 이고, c_1, c_2 는 learning factor로 $pbest$ 와 $gbest$ 의 영향력을 조절하며, 각 차원에서 실행불가능해로 빠지지 않게 하기 위해 v_{id} 의 범위를 $[-V_{max}, V_{max}]$ 로 조절하였다. 즉, 아래의 식 (16), (17)과 같이 $v_{id} > V_{max}$ 이면 $v_{id} = V_{max}, v_{id} < -V_{max}$ 이면 $v_{id} = -V_{max}$ 로 선정하였다.

$$\text{If } v_{id} > V_{max}, \text{ then } v_{id} = V_{max} \tag{16}$$

$$\text{else if } v_{id} < -V_{max}, \text{ then } v_{id} = -V_{max} \tag{17}$$

해를 찾아가는 속도를 조절하기 위해 일정하게 감소하는 inertia weight를 적용하였다. Inertia weight를 크게 하면 전체적으로 속도가 빨라져 탐색을 전역적으로 하게 되며, 작게 하면 국부적인 탐색을 하게 된다. Weight를 너무 크게 설정하면 local minimum에 빠질 확률이 커지게 되고 너무 작게 설정하면 최적해를 찾지 못하고 종료되는 문제가 있으므로, 초기에는 weight를 크게 주었다가 시간이 지남에 따라서 weight를 줄여주는 방식을 사용한다. 식 (9)와 같이 weight를 조절함으로써 local minimum에 빠지는 것을 방지하였다. 즉, $w_{up} = 0.9, w_{low} = 0.4$ 로 설정하면 초기 weight는 $w(1) = 0.9,$ 마지막 weight는 $w(T) = 0.4$ 가 된다.

5. 실험 및 결과분석

본 연구에서 제안하는 HVRPCR-HPSO 알고리즘의 효율성을 증명하기 위해 Golden et al.(1984)의해 제기된 20개의 예제 중 8가지 예제를 선정하였다. 실험은 기존 실험결과와 비교하기 위한 예제실험과 방문가능 수요지가 제한된 상태 하에서의 실험으로 구분하였다. 예제실험에서 목적식을 고정비와 운행거리의 합으로 한 실험을 Group 1로 하고, 고정비와 변동비×거리의 합으로 한 실험을 Group 2로 한다. 본 연구에 적용된 알고리즘은 Microsoft Visual C++ 6.0을 이용하여, Pentium IV(3.0GHz, 512M RAM) 환경에서 수행하였으며, 결과는 알고리즘을 5번 반복 실행한 최고 결과 값이다. 실험예제의 고정비는 Golden et al.(1984)의 값, 변동비는 Taillard(1999)의 값을 사용하였고 차량종류별 대수는 제한하지 않았다. 알고리즘의 파라미터는 swarm size = 100, number of iteration = 1,000, $c_1 = 2, c_2 = 2, w_{up} = 0.9, w_{low} = 0.4$ 를 각각 적용하였다.

5.1 예제실험

Group 1 실험은 Gendreau *et al.*(1999), Group 2 실험은 Choi and Tcha(2007)의 실험결과와 각각 비교하였다.

Table 2. Computational results for the 8 instances of group 1

Instance (No. of nodes)	Best known	Gendreau et al.(1999)	This study
3(20)	961.03	961.03	961.03
4(20)	6437.33	6437.33	6437.33
5(20)	1007.05	1007.05	978.98
6(20)	6516.47	6516.47	6447.56
13(50)	2406.36	2408.41	2407.23
14(50)	9119.03	9119.03	9119.28
15(50)	2586.37	2586.37	2586.37
16(50)	2720.43	2741.50	2728.69

Note) The CPU for solving these problems(Group 1 and Group 2) was between 10 ~ 50 seconds.

Table 3. Computational results for the 8 instances of group 2

Instance (No. of nodes)	Best known	Choi and Tcha (2007)	This study
3(20)	1144.22	1144.22	1144.22
4(20)	6437.33	6437.33	6437.33
5(20)	1322.26	1322.26	1322.26
6(20)	6516.47	6516.47	6447.56
13(50)	2964.65	2964.65	2964.65
14(50)	9126.90	9126.90	9126.90
15(50)	2634.96	2634.96	2634.96
16(50)	3168.92	3168.92	3168.92

Note) The CPU for solving these problems(Group 1 and Group 2) was between 10 ~ 50 seconds.

Group1 실험은 Gendreau *et al.*(1999)의 실험결과와 동일한 수요지를 할당하여 구한 결과로써 <Table 2>와 같이 5, 6번 예제에서 기존에 알려진 최고해보다 우수한 해를 도출하였다. 5번 예제의 결과는 <Table 4>와 같이 총 운행비용 995.79로 차량의 조합은 (A1, B1, D1, E2)에서 (A1, C1, D1, E2)로 변경되어 고정비는 15가 증가하였지만, 총 운행거리는 382에서 356으로 감소하여 기존의 최적해를 1.1% 개선하였다. 6번 예제는 차량

의 조합은 기존해와 동일하지만 총 운행거리가 516에서 448로 확연히 감소하여 총 운행비용 6447.56으로 기존의 최적해를 약 1.1% 개선하였다. 3, 4, 15번 예제는 기존의 최적해와 동일한 해를 구하였으며, 13, 14, 16번 예제는 기존 최적해를 개선시키지 못하였지만 오차범위 0.3% 이내의 유사한 해를 도출하였다.

Group 2 실험은 고정비와 변동비를 동시에 고려한 실험으로 최근 Choi and Tcha(2007)와 Imran *et al.*(2008)의 연구에서 다루어졌다. 실험은 차량별 수요지를 변화시켜가며 본 알고리즘을 적용하여 8가지 예제 중 6번 예제를 제외하고 기존의 최적해와 동일한 해를 도출하였으며, 6번 예제는 Group1과 같은 해로써 기존 최적해를 약 1.1% 개선하였다. 실험결과를 통해 HVRPCR-HPSO 알고리즘이 HVRP에 효율적인 해법임을 확인할 수 있다.

5.2 수요지 제한 실험

본 실험에서는 현실문제에 접근하여 보다 현실성 있는 차량 경로를 구성하기 위해 일일 방문가능 수요지 수를 제한하는 제약조건을 추가하여 실험한다. 실험은 Group 1의 5번 예제를 선택하였으며, 예제실험에서 구한 해는 <Table 4>와 같다. 이는 기존의 최고해와 같은 수요지 수를 할당하여 구한 결과이며, 차량별 수요지와 거리는 각각 1~7, 24~127의 범위를 보였다.

만일 일일 방문가능 수요지가 최대 4곳을 초과하지 못한다고 할 때, 5번 예제의 결과는 일일 운송문제에서 모든 수요지의 요구량을 충족시킬 수 없다. 일일 방문가능 수요지 수의 상한이 4일 때는 차량 10대의 조합(A3, B3, C2, D2), 고정비는 505, 총 운행거리는 509.25, 총 운행비용은 1014.25가 되고 그 결과는 <Table 5>와 같다.

예제실험의 995.79보다 18.46만큼 운행비용이 증가하였지만 일일 방문가능한 수요지의 제약조건을 만족한다는 점에서 보다 현실성 있는 해가 된다. 일일 방문가능 수요지 수의 상한 (S_{up})을 변화시켜가며 실험한 결과 <Table 6>과 같이 상한을 제한하지 않았을 때와 동일한 해를 산출했다.

결론적으로, 본 연구에서 제안하는 HVRPCR-HPSO 알고리즘은 차량의 종류에 따른 차량대수는 제한이 없고 일일 방문가능 수요지 수의 제한이 있을 때, 최적의 차량조합과 차량별 경로, 최소 운행비용을 구할 수 있는 해법으로, 기존의 연구결과와 비교하였을 때 더 우수한 해를 산출할 수 있을 뿐만 아니라 수요지 수 제한에 따른 차량의 조합과 차량별 경로, 최소운

Table 4. Results of instance #5

Vehicle type	Number of customers	Customer sequence	Demand	Distance
E	5	Vo-18-14-6-7-8-Vo	119	87.90
E	6	Vo-1-3-20-2-16-11-Vo	115	83.54
E	9	Vo-4-13-19-17-15-10-9-5-12-Vo	120	132.54
fixed cost : 675		Total Distance : 303.98	Solution cost : 978.98	

Table 5. Best Solution of instance #5(customer windows)

Vehicle type	Number of customers	Customer sequence	Demand	Distance
A	1	Vo-11-Vo	19	24.08
A	1	Vo-6-Vo	15	22.80
B	1	Vo-12-Vo	29	16.12
B	1	Vo-2-Vo	30	42.05
A	2	Vo-19-4-Vo	18	63.88
B	2	Vo-1-8-Vo	30	47.58
C	2	Vo-7-14-Vo	40	66.12
C	3	Vo-15-10-5-Vo	36	68.90
D	3	Vo-18-13-17-Vo	67	68.540
D	4	Vo-3-20-16-9-Vo	70	89.18
fixed cost : 505		Total Distance : 509.25	Solution cost : 1014.25	

Table 6. Best Solution of instance #5(customer windows)

Vehicle type	Number of customers	Customer sequence	Demand	Distance
E	5	Vo-18-14-6-7-8-Vo	119	87.90
E	6	Vo-1-3-20-2-16-11-Vo	115	83.54
E	9	Vo-4-13-19-17-15-10-9-5-12-Vo	120	132.54
fixed cost : 675		Total Distance : 303.98	Solution cost : 978.98	

행비용을 산출 가능함으로 인하여 기존 연구에서 진보하여 보다 현실적인 문제에 접근가능한 알고리즘이라고 할 수 있다.

6. 결론

본 연구에서는 VRP의 여러 가지 변형 중 복수차량 유형을 사용하는 HVRP에 대하여 수요지 제한이라는 제약사항을 추가하여 총 운행비용의 최소값을 산출할 수 있는 HVRPCR 모형을 제시하였으며, HVRP에 HPSO 기법을 적용하여 수송비를 최소화하는 차량조합과 차량별 경로를 구할 수 있는 HVRPCR-HPSO 알고리즘을 제안하였다.

PSO는 지난 10년간 다양한 NP-hard 문제에 적용되어 왔고, 다수의 VRP에도 적용되고 있으나 HVRP에서의 연구는 이루어지지 않음에 따라 본 연구에서는 빠른 수렴성이라는 PSO의 장점을 이용하여 HVRP에 적용하였다. 초기해 생성을 위하여 난수표현법을 사용하였고, 좋은해를 도출하기 위해 2-opt, 1-1 exchange, 1-1-1 exchange 기법을 삽입한 HPSO 알고리즘을 구성하였다. 기존 연구의 실험결과와 비교한 결과 총 16가지의 실험 중 3개의 새로운 최적해를 구할 수 있었고, 10개의 기존 최적해와 동일한 해, 0.3% 오차 범위내의 3개의 해를 구하여 HVRP에서도 PSO가 우수한 성능을 보임을 알 수 있었다.

본 알고리즘은 제품이 매일 출하되어 전국에 위치한 공장, 대형 유통점 등의 거래처로 수송되는 상황에서 일일 가용한 시간내 수요지를 방문하기 위해 일일 방문가능 수요지가 제한

된 상황에서 수송비가 최소화 되는 차량수송계획을 신속하게 수립할 수 있으며, 신규설립 물류센터나 공장 등에서 물품을 수요지로 배달하기 위해 필요한 차량의 종류별 구매대수를 결정하는 상황 등 다양한 현실문제에 적용할 수 있을 것으로 판단된다.

향후 연구방향으로는 다중 차고지에 따른 제약과 배달과 수거가 동시에 이루어지는 문제에 대한 연구가 필요할 것으로 판단되며, 차종별 고정비와 변동비를 단순히 인건비와 차량연비 뿐만 아니라 차량운행에 따른 감가상각비, 보험료, 제세공과금과 엔진오일비, 타이어마모비 등을 현실적으로 산출하여 적용할 필요가 있다. 또한 본 연구에서 제시한 알고리즘을 개선하여 LRP(Location routing problem)와 다차원 공간에서의 경로문제에 적용해 볼 수 있을 것으로 판단된다.

참고문헌

- Imran, A., Salhi, S., and Wassen, N. A. (2009), A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem, *European Journal of Operational Research*, **197**, 509-518.
- Clark, G. and Wright, J. (1964), Scheduling of vehicle from a central depot to a number of delivery points, *Operations Research*, **12**, 568-581.
- Cho, H. S. and Lee, D. M. (2008), 2006 Traffic congestion costs : estimation and trend analysis, *The Korea Transport Institute*, 02.
- Choi, E. and Tcha, D.-W. (2007), A column generation approach to the heterogeneous fleet vehicle routing problem, *Computer and Operations*

- Research*, **34**, 2080-2095.
- Dueck, G. (1993), New optimization heuristics : the great deluge algorithm and the record-to-record travel, *Journal of Computational Physics*, **104**, 86-92.
- Eberhart, R. C. and Shi, Y. (2000), Comparing inertia weights and constriction factors in particle swarm optimization, *Proceedings of the 2000 Congress on Evolutionary Computation*, 84-88.
- Gendreau, M., Laporte, G., Musaraganyi, C., and Taillard E. D. (1999), A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computer and Operations Research*, **26**, 1153-1173.
- Glover, F. (1997), Tabu search and adaptive memory problem programming : advances, applications and challenges, *Interfaces in comp. Sc. and Oper. Res.*, 1-76.
- Golden, B., Assad, A., Levy, L., and Gheysens, F. G. (1984), The fleet size and mix vehicle routing problem, *Computer and Operations Research*, **11**, 49-66.
- <http://ina2.eivd.ch/collaborateurs/etd/problems.dir/vrp.dir/vrp.html>.
- Kennedy. J. and Eberhart. R. C. (1995), Particle swarm optimization, *Proceedings of IEEE international Conference on Neural Networks*, 1942-1948.
- Kennedy. J. and Eberhart. R. C. (2001), *Swarm intelligence*, San Francisco : Morgan Kaufman Publishers.
- Liu, F. H. and Shen, S. Y. (1999), The fleet size and mix vehicle routing problem with time windows, *Journal of the Operational Research Society*, **50**, 721-732.
- Ochi, L. S., Viana, D. S., Drummond, L. M., and Victor, A. O. (1998), A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet, *Future Generation Computation System (FGCS)*, **14**, 285-292.
- Salhi, S., Rand, G. K. (1987), Improvements to vehicle routing heuristic, *Journal of the Operational Research Society*, 293-295.
- Salhi, S., Rand, G. K. (1993), Incorporating vehicle routing into the vehicle fleet composition problem, *European Journal of Operation Research*, **66**, 313-330.
- Taillard, E. D. (1999), A heuristic column generation method for heterogeneous fleet, *Recherche operationnelle(RAIRO)*, **33**, 1-14.