

## 시간대 제약이 있는 차량 수·배송 문제를 위한 Grouping Genetic Algorithm

### A Grouping Genetic Algorithm for the Pick-up and Delivery Problem with Time Windows

송정은\*, 김형석\*\* , 이명호\*\*\*, 김내현\*

\* 아주대학교 산업정보시스템공학부

\*\* 대우정보시스템

\*\*\* 세명대학교 인터넷 정보공학부

#### Abstract

OR 기법이 현실에 성공적으로 적용된 분야 중 하나가 차량 경로 문제 풀이며 이 분야에 대한 연구는 꾸준히 진행되어 오고 있다. 시간대 제약이 있는 차량 수·배송 문제는 기존의 차량 배송 문제에서 더욱 진보된 형태의 문제이다. 많은 연구가 있어왔지만, 이러한 형태의 문제들은 NP-hard 형태의 문제이므로 meta-heuristic 기법들이 많이 사용되었다. 유전자 알고리즘에 비해 타부 서치 기법이 많이 사용되었는데, 이는 시간대 제약이 있는 차량 수·배송 문제의 경우 유전자 알고리즘의 근간인 염색체 형태로 표현하기가 쉽지 않기 때문이다. 이에 본 논문에서는 시간대 제약이 있는 차량 수·배송 문제를 다수의 차량을 사용하는 Grouping Genetic Algorithms을 적용하여 이러한 어려움을 해결하였다. 또한, 기존의 유전자 알고리즘들은 초기 해 집단을 임의의 선택 방식으로 구성하였지만, 초기 해 집단 구성을 heuristic 기법을 사용하여 적합도 함수 값이 좋은 해들로 구성하여 기존에 사용된 알고리즘들과 성능을 비교 분석하고자 한다.

#### 1. 서론

차량 경로 문제는 지리적으로 분산된 수요지점의 요구를 충족시키기 위하여 최소의 비용 또는 거리로 차량 경로를 선정하는 문제이다. 제약 조건에 따라 CVRP, MDVRP, VRPTW, SVRP, VRPPD 등으로 구분된다. 이 중 시간대 제약이 있는 차량 경로 문제(Vehicle Routing Problem with Time Windows : VRPTW)는 고객에 대한 서비스를 고려하는 수송 의사결정 문제이다. 이는 중앙 차고지를 출발한 일련의 차량들이 지리적으로 산재해 있는 고객들을 고객이 요구하는 서비스 시간대에 맞추어 방문하여 주어진 서비스를 수행하고 다시 차고지로 돌아오는 최소 비용의 경로를 고려하는 수송 의사결정 문제이다. 그러나, VRPTW 모델은 고객의 서비스 시간을 고려한 해를 도출함으로써 다른 모델과는 달리 현실에 한 발자국 더 근접한 모델이긴 하지만

Pick-up 과 Delivery 중 한 가지만을 다루는 모델로서 실제 문제를 적용하기에는 한계점을 가지고 있다. 이에 고객의 서비스 시간대를 고려하면서도 Pick-up 과 Delivery를 같이 고려하는 Pick-up and Delivery Problem with Time Windows(PDPTW) 모델이 제기되었다. PDPTW 모델은 VRPTW 모델의 제약 조건을 완화한 모델이며, 실제 모델에 근접한 형태를 가지고 있다. 차량 경로 문제는 조합적 특성으로 인하여 수리 모형으로는 표현 가능하지만 최적의 해를 도출하기에는 사실상 불가능한 NP-hard 문제의 특성을 가지고 있다.(Lenstra and Rinnoy Kan, 1981)

이런 이유로 PDPTW 문제를 풀기 위해서 많은 종류의 heuristic 기법이 사용되어 왔다. 이러한 기법을 크게 나누면 구성형, 개선형, 메타휴리스틱으로 구분될 수 있는데(홍성철, 박양병 2004) 구성형은 부분 경로에 특정한 기준에 따라 경로에 포함되지 않은 고객 지점들을 추가하는 방법으로 insertion 기법이 대표적이다.(Solomon, 1987) 개선형은 지역 탐색을 통하여 반복적으로 현재의 차량 경로 해를 수정하여 더 나은 해를 탐색하는 방법이다.(Potvin and Rousseau, 1995). 메타 휴리스틱은 지역 해를 탈출하여 다양한 해 공간을 탐색하도록 고안된 방법이며, 대표적인 방법으로는 타부서치, 유전자 알고리즘, 시뮬레이티드 어닐링 등이 있다.

유전자 알고리즘은 Holland에 의해 처음 제시된 이후로 다양한 분야에서 널리 사용되어 왔다. 주로 단순한 차량 수·배송 문제는 적용이 용이했으나, PDPTW 같은 경우에는 상호 독립적인 두개의 문제가 결합되어 있기 때문에 전통적인 유전자 알고리즘의 적용하기는 사실상 불가능하다.(Giselher Pankrantz, 2005) 이에 수정 유전자 알고리즘 중 Falkenauer(1998)에 의해 제안된 Grouping Genetic Algorithm을 사용하고자 한다.

#### 2. 수리 모형

본 논문에서 다루는 PDPTW의 특성은 다음과 같다. 차량의 대수는 제한이 없고 모든 차량은 동일한 능력을 가지고 있다고 가정하였다. 고객의 요구 특성

에 따라 pick-up 노드와 delivery 노드로 나누어진다. 모든 pick-up 노드의 집합을  $N_i^+$ , delivery 노드들의 집합을  $N_i^-$ 라고 하고,  $N_i^+ \cup N_i^- = V$ 라 한다. 본 모형에서 사용되는 상수와 변수를 정의하면 다음과 같다.

$N_+$  = 픽업 노드들의 집합  
 $N_-$  = 딜리버리 노드들의 집합  
 $q_i$  = 지점  $i$ 에서의 고객 요구량 (픽업 노드에서는 양의 값을, 그 외에는 음의 값을 갖는다.)

$$\forall i \in N, \sum_{k \in M} Z_{ik} = 1 \quad (1)$$

$$\forall i \in V, \sum_{k \in M} \sum_{j \in V} X_{ijk} = 1 \quad (2)$$

$$\forall k \in M, \sum_{i \in V} X_{iok} = 1 \quad (3)$$

$$\forall k \in M, \sum_{j \in V} X_{ojk} = 1 \quad (4)$$

$$(\forall h \in V)(\forall k \in M), \sum_{i \in V} X_{ihk} - \sum_{j \in V} X_{hjk} = 0 \quad (5)$$

$$(\forall j \in V)(\forall k \in M), \sum_{i \in V} X_{ijk} Q \geq y_j \quad (6)$$

$$(\forall i, j \in V \cup O)(\forall k \in M), X_{ijk} = 1 \Rightarrow y_i + q_i = y_j \quad (7)$$

$$y_0 = 0 \quad (8)$$

$$\forall i \in V, y_i \geq 0 \quad (9)$$

$$(\forall i, j \in V)(\forall k \in M), X_{ijk} = 1 \Rightarrow D_i + t_{ij} \leq D_j \quad (10)$$

$$\forall i \in N, p = N_i^+, q = N_i^-, D_p \leq D_q \quad (11)$$

$$D_0 = 0 \quad (12)$$

$Z_{ik}$ 는  $i$ 번째 고객이 차량  $k$ 에 배정 되면 1, 아니면 0의 값을 갖는 이진 변수이다.  
 $X_{ijk}$ 는 고객  $i$ 와  $j$ 를 차량  $k$ 가 방문하게 되면 1, 아니면 0의 값을 갖는다.  $y_j$ 는  $j$  지점을 방문했을 때 차량의 적재량을 나타낸다.

(1) 식은 각 고객의 요구는 차량 1대에만 배정되어야 한다는 것을 의미한다. (2) 각 지점은 한 번만 방문되어야 함을 의미한다. (3)과 (4)는 차량의 출발점과 도착점은 Depot이어야 함을 의미하고, (5) 식은 모든 고객 요구지점에서는 차량이 멈출 수 없음을 뜻한다. (6)~(9)식은 차량 용량에 관한 제약식이고, (10)~(12) 식은 고객 서비스 시간 제약과 precedence 제약을 나타내고 있다.

본 모형의 목적함수는 다음과 같다.

$$\text{Minimize } \sum_{k \in M} \sum_{i \in V} \sum_{j \in V} d_{ij} X_{ijk}$$

위의 제약식은 차량이 이동하는 총 운행거리를 최소화하는 식이다.

## 2 Grouping Genetic Algorithm

기존의 VRP 또는 TSP 문제에 적용하기 위해 사용된 Genetic Algorithm(GA)에서는 순서 기반의 encoding을 주로 사용하였다. 차량의 순회 순서만을 짜기 위해서는 상당히 효과적인 encoding 방법이지만, PDPTW 같은 두 개의 문제가 포함되어 있는 형태인 경우는 순서 기반의 encoding 기법은 의미가 퇴색하고 만다. 이러한 문제점을 극복하고자 Jung and Haghani(2000)가 발표한 논문에서는 random key representation이라고 불리는 4자리의 숫자를 사용하여 PDPTW의 routing 부분과 grouping 부분을 동시에 표현하였다. 하지만, 이 조차도 시간대 제약과 용량 제약 등으로 인해 불가능한 해를 많이 만들어 내게 되었다.

이에 그림 1처럼 각각의 노드로 encoding 하는 것이 아니라 노드들을 하나로 묶어 request로 표현하여 encoding 하는 Grouping Genetic Algorithm(GGA)을 사용하였다.(Giselher Pankrantz, 2005)

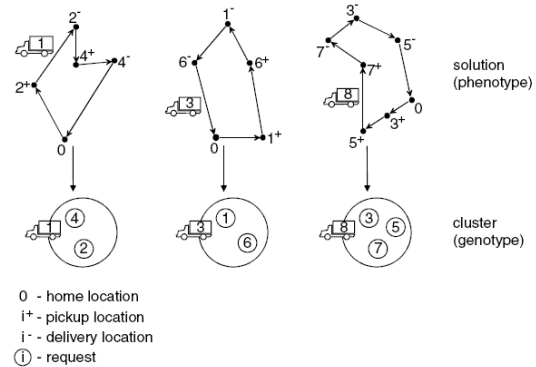


그림1. encoding of GGA

### 2.1.1 초기 해

기존의 GA는 초기 해를 random으로 잡는 경우가 많았다. 이는 자연의 열성과 우성을 그대로 반영하기 위해 좋은 염색체와 나쁜 염색체를 통하여 지역 국한적인 해가 아닌 전역 최적해를 얻고자 함이었다. 하지만, 임의의 선택이 아닌 보다 좋은 해를 가지고서 GA를 실행한다면 그 결과가 더 빨리 얻을 수 있지 않을까란 본 논문에서는 초기 해를 heuristic 기법을 사용하여 선정하였다. 그 순서는 다음과 같다.

#### 2.1.1.1. Ready\_Dist\_Due

$P\_Set$  = Pick Up Node들의 집합  
 $D\_Set$  = Delivery Node들의 집합

- $P\_Set$ 을 Ready time 기준으로 정렬
- $P\_Set$ 을 capacity 별로 끊어서 vehicle들을 형성
- 각 vehicle 내의 P Node들을 거리가 가까운 것끼리 NNH 알고리즘을 통해서 묶는다.
- 각 vehicle 내의 P에 대해 pair를 이루는 D의 Node를 vehicle에 추가
- 각 vehicle 내에서 D는 due time이 빠른 순 부터 정렬

#### 2.1.1.2. Dist\_Ready\_Due

P\_Set = Pick Up Node들의 집합  
D\_Set = Delivery Node들의 집합

- P\_Set을 NNH 알고리즘을 통해서 정렬
- P\_Set을 capacity 별로 끊어서 vehicle들을 형성
- 각 vehicle 내의 P Node들을 Ready time 기준으로 정렬
- 각 vehicle 내의 P에 대해 pair를 이루는 D의 Node를 vehicle에 추가
- 각 vehicle 내에서 D는 due time이 빠른 순서부터 정렬

이러한 2 종류의 heuristic 기법을 사용하여 초기 해를 선정하였다.

### 2.1.2 교배 연산자 및 돌연변이 연산자

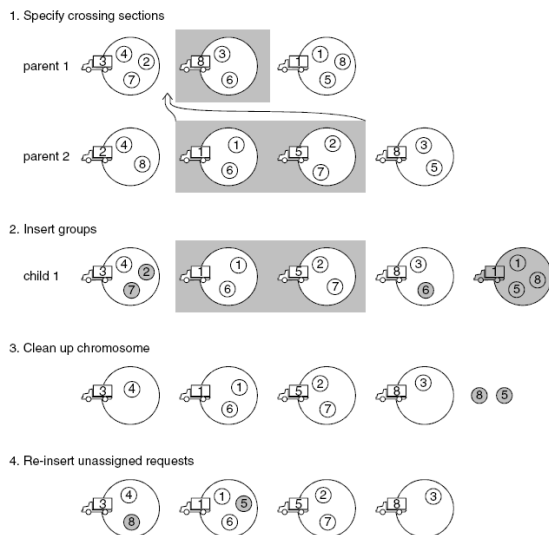


그림 2. Crossover in GGA

교배 연산자는 그림 2에서 표현된 것처럼, Falkenauer가 저술한 Genetic Algorithms and GROUPING PROBLEMS에서 서술된 group-oriented crossover를 사용하였다. 그 순서는 다음과 같다.

Step1. 두 부모 염색체에서 임의로 각각 2곳의 교배점을 선택한다.

Step2. 두 번째 부모의 교배점 사이의 부분을 첫 번째 부모의 첫 번째 교배점에 삽입한다.

Step3. 중복된 차량이 있는 경우에 첫 번째 부모의 염색체에서 중복된 차량을 제거한다. 마찬가지로, 중복된 request가 있는 경우도 첫 번째 차량의 request만 제거하고 삽입된 부분은 보존한다.

Step4. 제거된 차량에 속해있던 request들을 삽입 가능한 후보 차량에 배정한다. 삽입 가능한 차량이 없는 경우 새로 차량을 배정한다.

Step5. 두 번째 부모 염색체에 대하여 반복한다.

돌연변이 연산자 또한 Falkenauer가 저술한 Genetic Algorithms and GROUPING PROBLEMS에서 서술된 group-oriented crossover를 사용하였다. 그 순서는 다음과 같다.

Step1. 염색체에서 임의로 gene을 선택한다.

Step2. 선택된 gene을 삭제하고, gene에 속해있던 request들 또한 삭제한다.

Step3. 삭제된 request들을 기존의 염색체 안에 삽입 가능한 공간에 재 삽입한다. 만약, 가능 공간이 없다면 차량을 새로 배정한다.

### 3. 계산 실험

제안된 알고리즘의 성능평가를 위하여 Li and Lim(2001)에 의해 제안된 56 100-노드의 데이터를 사용하였다. 이는 VRPTW를 풀기 위한 알고리즘 테스트를 위해 사용되어져 온 Solomon의 문제를 수정한 것이다. 픽업과 딜리버리 노드의 demand를 약간 수정하였고, precedence 제약을 위해 각각의 노드가 선행 후행을 나타내는 precedence(i), successor(i) 값을 갖도록 하였다.

GGA를 적용하기 위해 사용된 모수들은 다음과 같다.  $n^{pop} \Rightarrow$  모집단 크기,  $p^{cross} \Rightarrow$  교배 확률,  $p^{mut} \Rightarrow$  돌연변이 확률이며 그 값은  $(n^{pop}, p^{cross}, p^{mut}) = (50, 1.0, 0.5)$  로 설정하였다.

Table1. 결과값

instance	TD	CT	차량 수
nc 101	828.94	67.82	10
nc 102	828.94	84.2	10
nc 103	831.87	148.2	10
nc 104	816.74	186.4	10
nc 105	828.94	72.4	10
nc 106	828.94	78.62	10
nc 107	827.82	79.21	10
nc 108	829.14	89.1	10
nc 109	827.82	134.1	10

Table 2. 기존의 GGA 결과값

instance	TD	CT	차량 수
nc 101	828.94	64.93	10
nc 102	828.94	83.1	10
nc 103	831.87	117.13	10
nc 104	816.74	143.67	10
nc 105	828.94	69.83	10
nc 106	828.94	74.73	10
nc 107	827.82	74.10	10
nc 108	827.82	84.9	10
nc 109	827.82	116.87	10

본 논문에서 제안한 알고리즘은 기존의 GGA와 결과적으로는 큰 차이가 없다. 하지만, 초기 해를 선정하는 과정에서 임의로 선택하는 것이 아니라

heuristic 기법을 사용하였기에 최종 결과 값 도출 시간이 평균적으로 12.31 더 길게 나왔다. 위에 명기된 table은 LC 타입만 적용하여 나온 결과이다.

#### 4. 결론 및 향후과제

본 연구에서는 PDPTW 문제를 풀기 위하여 GGA를 사용하였고, 초기 해 선정 과정에서 제안한 heuristic 기법을 사용하여 해를 도출하였다. LC 타입에 적용하였을 시 기존의 GGA와 비교하여 차량 대수 및 총 운행거리는 거의 동일하게 나왔다. 단지 계산 시간이 평균적으로 12.31 만큼 더 걸렸지만, 이는 heuristic 기법을 사용하였기에 늘어난 결과이다. 기 발표된 GGA 연구를 보면 타부 서치나 기타 알고리즘에 비해 상당히 좋은 결과 값을 보이고 있다. 이미 입증된 알고리즘의 우수성을 개선하기 위해서는 초기 해 선정에 있어서 제시된 heuristic 기법 보다 더욱 다양한 방법을 사용하여 계산 시간을 줄이는 것이 앞으로의 연구 과제이다. 본 연구에서 사용된 자료 구조보다 더욱 효율적이고 최적화된 자료 구조를 사용함으로써 더욱 좋은 결과를 도출한 것으로 예상된다. 이러한 개선된 알고리즘은 실제의 차량 수 · 배송 문제를 푸는 데 있어서 도움이 될 것이다.

#### 참고문헌

- Emanuel Falkenauer(1998)), GENETIC ALGORITHMS and GROUPING PROBLEMS, WILEY. Chichester
- Giselher Pankratz(2005), A Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows, *OR Spectrum*, 27, 21-41.
- Jung S, Haghani A(2000), A Genetic Algorithm for Pick-up and Delivery Problem with Time windows, *Transportation Research Record 1733*, Transportation Research Board, 1-7.
- Lau HC, Liang Z (2001), Pickup und Delivery with Time Windows, Algorithms und Test Case Generation, IEEE Computer Society (ed) Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 333-340
- Li H, Lim A (2001) A metaheuristic for solving the Pickup und Delivery Problem with Time Windows, IEEE Computer Society (ed) Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 160-167