

배달 및 수거를 고려한 차량운송계획모델 A VRP Model for Pickup and Delivery Problem

황 홍석*, 조 규성, 홍 창우

Heung-Suk Hwang, Kyu-Sung Cho, Chang-Woo Hong

동의대학교 산업공학과

614-714 부산시 부산진구 가야동 산24

Tel : 051-890-1657, E-Mail : hshwang@hyomin.donggeui.ac.kr

Abstract

본 연구는 Heuristic 알고리즘 및 유전자알고리즘(GA)을 이용하여 수거(Pickup) 및 배달(Delivery)을 동시에 고려한 통합차량운송계획 모델의 개발이다. 본 연구는 기존의 TSP의 문제를 확장 응용하였으며, 이는 한 Route에서 수거지(Origin)와 운반지(Destination)를 포함하는 수요들을 만족하도록 운반되어야 하는 문제이다. 이러한 통합차량경로계획문제(VRP : Vehicle Routing Problem)를 해결하기 위한 접근방법으로 Heuristic 방법을 사용하였으며, 기존의 Saving 알고리즘과 유전자알고리즘(Genetic Algorithm)의 각종 연산자(Operators)들을 계산하여 사용한 TSP문제의 해를 본 연구의 해의 초기해로 사용하였으며 수거 및 배달문제의 특성을 고려하여 해를 구하였다. 본 연구의 결과를 다양한 운송환경에서, 거리산정방법, 가용운송장비 대수, 운송시간의 제한, 물류센터 및 운송지점의 위치 및 수요량 등 다양한 인자들을 고려한 통합시스템으로 프로그램을 개발하고 Sample 문제를 통하여 응용결과를 보였다.

Keyword : PDP, TSP, VRP, GA

1. 서론

최근에는 새로운 정보기술이 발전됨에 따라 대다수 수거 및 배송 관련업체들은 그들의 차량운송계획에 있어서 보다 효과적인 새로운 방법을 중시하게 되었다. 본 연구는 지역내의 수거 및 배달을 하는 소규모의 물류센터의 표준화된 화물(Palletized Cargo)의 운송계획을 위한 모델의 연구이다. 본 연구에서 고려된 시스템은 지역내의 각 고객의 배달(Delivery) 및 수거(Pick-Up)의 수요에 최소의 차량으로 효과적으로 운영하는 시스템을 위한 운송계획을 위한 연구이다. 본 연구에서 개발된 운송계획에 따라 전산프로그램을 개발하고, 고객의 요구(Order)에 신속하게 대응하기 위하여 전산프로그램의 입출력 과정을 신속하게 처리할 수 있도록 고려하였다. 본 연구에서 예제 시스템으로 99 Order와 약 2000 Pallets가 91개 분배창고 지역에서 각 창고에의 배달(Delivery) 및 수거(Pick-Up)를 위한 약 50대의 차량을 운영하는 시스템을 고려하였으며, 이들 차량의 종류를 4가지까지 가능하도록 하고 최적운송계획을 보였다.

으로 운반해야하는 제한사항을 가지는 점이 다르다. 이러한 PDP문제의 주요 목적을 요약하면 다음과 같다 :

- 운송 총 거리의 최소화
- 고객 서비스수준의 최대화
- 고객 요구에 부응하는 서비스 능력의 최대화

또한 PDP문제의 주요 제한사항(Constraints)을 들면 다음과 같다 :

- 운송순서에 출발지와 도착지가 반드시 포함
- 운송 시간의 제한(Time Constraint)
- 요원의 제한(운송요원의 작업시간)
- 운송능력의 제한(Capacity Constraint)
- 운송차량의 제한(Vehicle Duty)

이러한 PDP의 해는 일반적으로 TSP문제의 해를 응용할 수 있으며 다음과 같은 방법들이 사용되고 있다 :

- Dynamic Programming
- Mathematical Programming
- Branch and Bound Method

그림 1은 TSP 문제로부터 PDP문제로 변환하는 예를 들어 보인 것이다.

2. 배달 및 수거시스템

(PDP: Pickup and Delivery Problem)

2.1 PDP 시스템의 개념

PDP시스템은 지역 물류센터에서 각 운송지역에 배달(Delivery) 및 수거(Pick-Up)를 동시에 이루어지는 물류시스템이다. 이러한 PDP문제는 근본적으로 TSP문제에 기반을 두고있으며 실제 운송경로는 TSP문제의 운송경로와 같은 경로를 따르며 다만 수거지점(Pickup Point)에서 배달지점(Delivery)

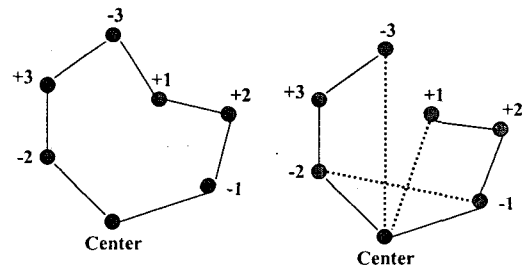


그림 1. TSP문제의 PDP문제로의 변환 예:

위의 그림은 두 개의 수요(Order)의 예이다.
 ●는 운송의 각 Node를 표시하고 +i 는 수거(Pickup)을 표시하며 -i 는 배달(Delivery)을 뜻한다. 그림 1의 우측 부분이 PDP문제이며 Center에서 가장 가까운 Node 1에서 Pickup하고 Node 2에서 Pickup한뒤 Node 1에 Delivery한 후 Center로 돌아오지 못하고 나머지 Node -2, +3를 끝내고 Center로 돌아오는 경로이다. 이는 좌측의 TSP경로를 같은 시계방향의 순서로 운송하였다. 본 연구에서 고려한 PDP시스템은 대부분의 주문을 전형적인 오전 및 오후로 구분되는 1/2일 계획으로 하였으며 동의회물(주)의 운송회사의 예를 들어 보았다.

본 연구의 예제에서 고려한 화물운송 Order는 표준화된 화물의 운송요청(Order)으로 다음과 같이 3가지로 구분하여 처리하였다.

- Group 1 : 4시간 이내에 운송
- Group 2 : 8시간 이내에 운송
- Group 3 : 운송할 담당자에게 1일 2회 전화로 지시(11:00시, 15:00)

운송요구는 항상 전화로 받으며, 운송 담당자는 각 운송지역에 운송할 준비가 항상 되어있는 것으로 가정한다. 본 시스템에 고려된 운송차량의 종류는 소형트레일러(ST), 중형트럭(MT), 대형트레일러(LT) 및 대형운송트럭(LG) 등으로 구분된다. 이들 운반차량은 각각 약어로 표기하여 사용하였다.

2.2 PDP문제의 운송계획

PDP문제의 이론적 모델링을 위하여 먼저 단일 차량 문제를 가정하고 N를 n개의 고객으로부터의 Order 수의 Set으로 정의하고, 운송차량은 물류센터에서 최초로 출발(Origin Center), +0 운송지점 i (수거 +i, 배송 -i)들에 수거 및 배달을 한 후 최종 도착지인 물류센터 -id 도착하는 경로를 따른다. 여기서 총 운송거리를 최소화하는 문제를 정식화한다. 이를 위하여 다음 수식을 정의한다.

$$G_N = (V_N, E_N)$$

여기서,

$$V_N = (+0, -0) \cup (+i, -i | i \in N)$$

$$E_N = ((+0, -0) \cup (+0, +i) | i \in N) \cup (-0, -i) | i \in N) \cup E(K_{2n})$$

$E(K_{2n})$ 는 고객이 요구한 수거 및 배달 총 지점의 Set이다.

2 고객의 경우 예를 들면,

$$|V_N| = 2n + 2$$

$$|E_N| = 2n^2 + n + 2$$

2개의 고객의 경우의 PDP 문제의 가능 해의 예를 들면 그림 2와 같다.

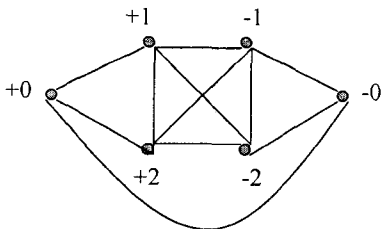


그림 2. 2고객의 PDP Routing 예

여기서 모든 가능한 주어진 PDP문제의 6가지의 가능한 Routing을 열거하면 다음과 같다:

- (+0, +1, -1, +2, -2, -0), (+0, +1, +2, -1, -2, -0)
- (+0, +1, +2, -2, -1, -0), (+0, +2, -2, +1, -1, -0)
- (+0, +2, +1, -2, -1, -0), (+0, +2, +1, -1, +2, -0)

이러한 PDP문제의 가능 해는 다음과 같은 조건을 만족하는 $R \in E$ 의 Node로 구성된다.

- ① $(0, -0) \in R$
- ② $R \cap \delta(v) = 2, \forall v \in V$
- ③ $G(R)$ 가 서로 연결됨
- ④ 모든 고객에 대하여 (+0, -0)사이 존재하는 R에서 +i 가 +0 과 -i의 Path 사이에 존재

이 중 ①에서 ③까지의 조건은 TSP문제에도 해당되며 조건 ④는 PDP문제에 해당되는 조건이다. 본 연구에서 고려한 PDP문제의 운송계획은 운송요청(Order)이 다음과 같이 4 단계를 거쳐서 처리되는 운송계획을 다루었다.

- 단계 1 :
- CRT에 바로 입력처리
 - 수요자가 대화형식으로 Network를 통해서 요청

- 단계 2 :
- Order의 입력 순서를 결정
 - 차량 배치시간을 줄이도록 운송 요구 처리

- 단계 3 :
- 4가지 차량 유형에 따른 운송순서 계획 (Vehicle Route) 결정

- 단계 4 :
- 가능한 운송계획의 출력
 - 본 운송계획은 다음 내용의 입력자료를 필요로 하고 있다.
 - 운송요구(Order) : 운송 량의 크기/출발지 및 도착지(분배창고)를 On-Line System으로 입력 한다.
 - 운송차량(Vehicles) : 운송차량 Type, 용량 및 최대 운송가능시간을 입력
 - 운송개시시간(Begin Time) : 1일 24시간을 기준으로 운송개시 시간을 입력
 - 운송기간(Route Length) : 최대 운송가능 시간을 입력

위의 입력자료로부터 다음과 같은 각 단계의 과정을 거쳐서 최적 운송 계획이 수립된다.

- 단계 1 : 운송요구 및 시스템 운영 Data
- 고객 운송요구(Order)
 - 운송량 크기.
 - 출발지, 도착지
 - 운송차량
 - 운송차량 Type
 - 운송능력
 - 최대운송가능시간
 - 운송 시간 정보
 - 운송개시시간(Begin Time) (1일 24시간 기준시간)
 - 운송기간(Route Length)
- 단계 2 : 운송대안 선정
- 각 운송차량 Type별 및 Route결정
 - 각 Route의 수요자 요구 판정 여부 결정
 - 총 운송시간 산정
 - 각 차량 Type별 최적 Route결정
- 단계 3 : 결과 출력
- 입력 Data
 - 차량 Type별 출발시간 및 Route의 세부계획

2.4 가능 해의 산정

새로운 Order가 접수되면 출발 및 도착 Node 모두를 기존의 시간계획에 포함시켜야 한다. 이를 위하여 운송차량이 출발 Node에 먼저 도착하고 다음으로 도착 Node에 도착되도록 하여야 하며 새로이 추가되는 Node 들로 인하여 타 Node의 계획시간계획을 허용시간 이상으로 지연시키는 일이 없도록 계획하여야 한다. 이를 위하여 가능 해의 산정을 위해 다음 두 가지가 고려된다.

① 기존의 모든 인접 Node들을 분석하고 새로운 Pick-Up Node의 삽입가능성을 판단(Type i Search)하여야 하며, 기존의 Node m, n 사이에 Node i의 삽입가능성을 위하여 다음 식을 고려한다.

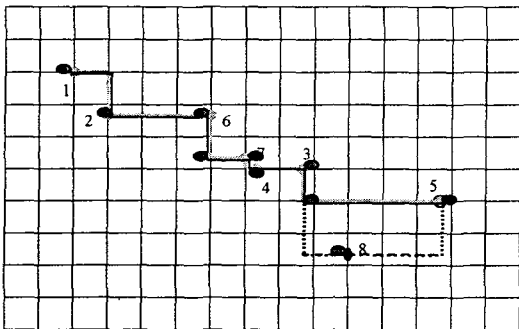


그림 4. 차량경로 계획에 새로운 Node의 삽입 (n_i)

$$e_8 = TT_{3,8} + \max(0, TR_8 - TA_8) + B_8 + TT_{8,5} - TT_{3,5}$$

$$= 15 + \max[0, 145 - (127 + 15)] + 3 + 18 - 20$$

$$= 15 + 3 + 3 + 18 - 20 = 19$$

$$TE_i = TR_i - W_B = 140$$

$$TD_3 + TT_{3,8} = 127 + 15 = 142$$

$$TL_8 = 145 + 5 = 150$$

$$e_8 < R \cdot S_5 ; TE_8 < TD_3 + TT_{3,8} < TL_8$$

$$TE_i \leq TA_i \leq TL_i$$

즉, 증가시간이 Route의 여유시간 보다 같거나 적어야 한다.

$$e_i \leq R \cdot R_n, \text{ 여기서 } R \cdot R_n : \text{Route여유시간}$$

② 첫 번째 Node를 중심으로 관련된 모든 인접 Node에 도착 Node의 삽입가능성을 판단한다.

이는 기존 운송계획을 조정하여 임시적인 시간 Loop를 고려하여 가능하다.

(Type II Search) j Node가 p와 q Node사이에 삽입 가능할 경우, $TE_j \leq TA_p \leq TL_q$

여기서 p, q Node사이에 j Node가 삽입됨으로 인한 증가되는 시간은 $e_j \leq R \cdot S_q$ 이다.

③ 시스템 비용

시스템 비용의 차이는 운행시간의 차이에 관련이 큰 것으로 보고 시간차이를 산출하여 비교하였다. m, n Node사이에 출발 Node i를 삽입하고 p, q Node 사이에 도착 Node j를 삽입할 경우, Node m에서 Node n으로, Node p에서 Node q로 직접 갈 경우보다 증가하는 시간을 산출하면 다음과 같다.

$$X = (TT_{m,i} + TT_{i,n} - TT_{m,n}) + (TT_{p,q} + TT_{i,p} - TT_{p,q})$$

④ 기존 예약고객에 대한 서비스

기존 예약고객들이 새로이 삽입되고 Node들로 인하여 기다리게 되는 시간을 산출하여 비교한다.

$$Y = \frac{\sum_{n=1}^{n+1} \max[(TA_n - TR_n), 0] - \sum_{n=1}^n \max[(TA_n - TR_n), 0]}{n+1}$$

또한 실제 시스템에서 있을 총 시간과 새로운 고객이 들어오지 않을 경우 직접 갈 경우의 총 시스템 시간과의 비율의 변화

$$Z = \frac{\sum_{n=1}^{n+1} \frac{TA_n - TR_n}{TT_{n,n_i}}}{n+1} - \frac{\sum_{n=1}^n \frac{TA_n - TR_n}{TT_{n,n_i}}}{n}$$

여기서 최선의 대안선정은 다음 수식에서 이루어질 수 있다.

$$\text{Min } F_{A,B,C} = AX + BY + CZ$$

여기서 A, B, C는 가중치로서 Simulation 등에 의하여 구할 수 있다. 운송스케줄에서 가능 해를 구하지 못할 경우(Infeasible Solution), 예약고객은 차선의 대안 선택 또는 예약을 수용하지 못하는 경우가 발생한다. 이때에 모든 가능한 대안점토를 위하여 다음 과정의 분석을 한다.

⑤ 모든 인접 Node간의 새로운 Node 삽입 가능성을 검토한다.

Node m, n 사이에 새로운 Node i의 삽입 가능성을 검토한다. m, n Node 사이에 Node i를 고려할 경우 총 증가 시간은 다음과 같다.

$$e = TT_{m,i} + B_i + TT_{i,n} - TT_{m,n}$$

$e \leq RS_n$ 인 경우 새로운 Node는 삽입 가능하다.

2.4 전산 프로그램 개발

위의 운송배치계획 알고리즘에 따라서 다음과 같이 전산프로그램을 개발하였고 동의화물(주)의 문제를 통하여 본 모델의 실용화가능성을 보였다.

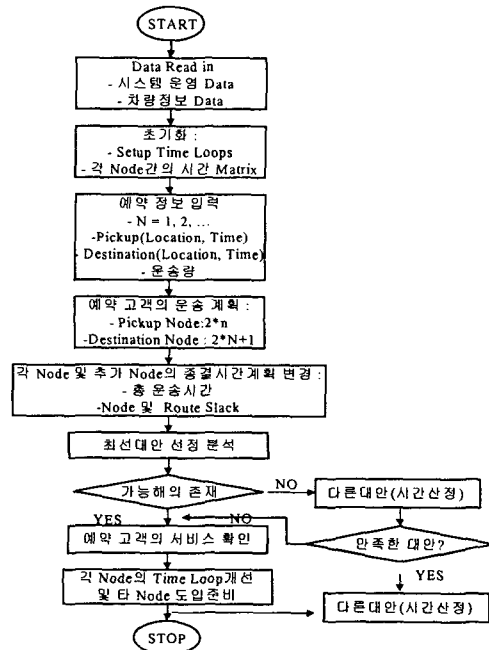


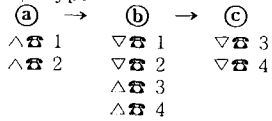
그림 5. 차량운송계획모델의 흐름도

본 프로그램의 흐름 도는 그림 5와 같이 표시하였다. 위의 운송계획산정의 방법에 따라서 산출된 계획을 운반지시를 하여 운송 차량을 할당하는 예를 들면 다음과 같다.

범례 : Pick-up : △ 창고 : ○
 운반 : ▽ 운반방향 : →
 지시 : ≡

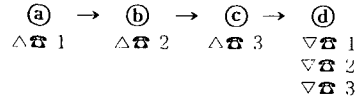
a) 모든 Type의 차량의 운반 할당

※ 차량의 Type



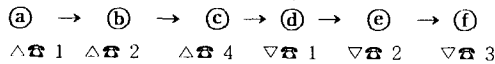
b) 소형트레일러의 운반할당

※ 트레일러형 차량

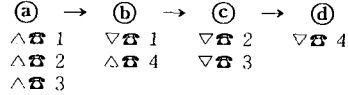


c) 중형트럭의 운반 할당

※ 다른 출발지 → 동일 도착지



d) 대형 트레일러/대형 차량



3. 모델의 응용

본 모델을 동의화물(주)의 부산지역 화물운송계획 문제에 응용하였다. 동의화물(주)은 pallet단위의 표준화된 화물을 운송하며 다음과 같이 4 가지 종류의 7 대 차량으로 운반하며 그림 1과 같이 부산지역의 분배창고들에 운송하는 계획을 세우려고 한다. 모두 58건의 운송 요청(Order)을 받았다. (부록 1 참고)

표 1. 가용차량 Data

차량종류	가용대수	운송능력	운송가능시간
1. 소형트레일러(ST)	3	7 pallets	240:00분
2. 중형 트럭(MT)	1	12 pallets	240:00분
3. 대형 트레일러(LT)	1	14 pallets	240:00분
4. 대형차량(LG)	2	10 pallets	240:00분
계	7		

본 운송계획 모델의 적용결과를 각 차량별 일일운송계획을 요약하면 부록 2와 같으며 이중 소형 트레일러(ST-1~ST-3) 3대 및 중형 트럭의 운송계획을 표 2 및 표 3과 같이 요약하였다.

표 2. 소형트레일러의 운송계획(ST-1~ST-3)

VEHICLE = ST-1
 START TIME = 800.
 DATE = 2000. 3. 15.

```

*****
Stop Site Time Deliver Pickup Order Stay Time
*****
1 D-4 809 7 45 3
2 S-1 814 7 45 3
3 C-1 823 7 36 3
4 J-17 838 7 36 3
5 S-6 853 1 54 2
  1 52 1
6 S-8 857 3 12 3
7 J-15 911 1 54 2
8 J-17 915 1 52 2
  3 12 1
9 D-5 941 5 31 3
10 D-8 1006 5 31 3
    
```

```

11 D-6 1011 2 56 2
12 D-4 1015 2 44 2
13 D-1 1019 1 32 2
14 J-17 1044 2 56 2
  2 44 1
  1 32 1
15 D-1 1110 2 34 2
16 D-4 1114 1 46 2
  1 46 1
17 G-4 1143 2 46 1
  2 34 2
    
```

ROUTE ENDED
 LOCATION = S-11
 TIME =1211
 NO OF PALLETS MOVED = 32

VEHICLE = ST-2
 START TIME = 800.
 DATE = 2000. 3. 15.

```

*****
Stop Site Time Deliver Pickup Order Stay Time
*****
1 D-4 809 7 49 3 * 트레일러
2 C-3 818 7 49 3 * 트레일러
3 D-4 834 7 49 3 * 트레일러
4 C-3 839 7 49 3 * 트레일러
5 D-4 855 7 49 3 * 트레일러
6 C-3 900 7 49 3 * 트레일러
7 D-4 916 7 49 3 * 트레일러
8 C-3 921 7 49 3 * 트레일러
9 D-6 937 7 23 3 * 트레일러
10 J-17 1000 7 23 3 * 트레일러
11 D-6 1026 4 23 3 * 트레일러
12 J-17 1031 4 23 3 * 트레일러
    
```

ROUTE ENDED
 LOCATION = S-11
 TIME =1045
 NO OF PALLETS MOVED = 39

4. 결론

본 연구는 제한된 운송차량과 각각의 운송능력으로 지역 유통물류의 적정 운송계획을 위한 모델의 개발이다. 본 연구에서 먼저 1개의 중앙물류센터와 지역내에 산재해있는 여러개의 분배창고(Warehouse)에 대하여 최소시간(비용)을 전제로 하는 운송계획을 수립하는 방법과 이의 전산프로그램을 개발하였으며, 실무 응용사례를 위하여 동의화물(주)의 문제를 적용한 Sample출력을 제시하였다. 지역 유통물류 시스템의 운송모델로서 유용하게 활용될 수 있도록 계속 보완 연구할 예정이다.

참고 문헌

[1] Cluff, C. K., "Minimization of Tardiness in Many-to Many Pickup and Delivery Systems", Ph. D. Thesis, Case Western Reserve University, 1987.
 [2] Laporte, G., "The Vehicle Routing Problem : An Review of Exact and Approximate Algorithms", European J. Oper. Res. 59, pp.345-358, 1992.
 [3] Laporte, G. and Nobert, Y. and Desrochers, M., "Optimal Routing under Capacity and Distance Restrictions", Oper. Res. pp.1050-1073, 1958.
 [4] Sexton, T. R. and Bodin, L. D., "Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times : I, Scheduling," Transportation Sci. 19(4), pp.378-410, 1985.
 [5] Sexton, T. R. and Bodin, L. D., "Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times : II, Routing," Transportation Sci. 19(4), pp.411-435, 1985.
 [6] Solomon, M. M., " Time Windows Constrained Routing and Scheduling Problems", Transportation Sci., 22, pp.1-13, 1988.