

공급 사슬 상에서 Cross Docking을 고려한 Vehicle Routing Scheduling(VRS)

이경민(kmlee@scm.hanyang.ac.kr), 이영해(yhlee@hanyang.ac.kr)
한양대학교 산업공학과

Abstract

Fierce competition in today's global markets, the heightened expectation of customers have forced business enterprises to invest in, and focus attentions on, their Supply Chains. Also Cross Docking is an essential part of SC, and integrating Cross Docking with vehicle routing scheduling is needed to smoothly link the physical flow of SC. However, there is no the mathematical model which focuses on Cross Docking with vehicle routing scheduling. Therefore, the integrating model considers Cross Docking and vehicle routing scheduling will be developed in this paper. And the solution based on Tabu algorithm to this model will be provided.

1. 서론

오늘날 많은 기업은 다양한 고객 요구 및 서비스 수준의 증가로 치열한 기업간 경쟁에 직면하고 있다. 따라서 기업은 고객 요구에 대한 빠른 반응(Reaction)과 유연성(Flexibility) 그리고, 원하는 제품을 납기 내에 배송함으로써 고객에게 확실성(Reliability)을 줄 수 있어야 한다. 이러한 문제들은 공급사슬(SC: Supply Chain)내의 제품의 흐름(Physical Flow)을 효율적으로 통제함으로써 가능하며, 특히 많은 기업들은 제품 흐름의 효과적인 경영 관리 도구로써 Cross Docking을 채택하고 있다. Cross Docking은 공급자나 생산자로부터 받은 제품을 재고로 보관하지 않고, 즉시 통합(Consolidation), 배송하여 전체 공급 사슬내의 재고량을 감소시키고, 높은 수준의 고객 서비스와 짧은 주문 처리 시간을 통하여 유연성 및 빠른 대응(Responsiveness)을 가능하게 한다. Apte et al.(2000)은 전체 제품 가격의 30%이상이 물류와 관련된 부분에서 발생하기 때문에 Cetinkaya et al.(2003)이 언급한 것과 같이 재고 관련 분야를 직접 개선하는 것보다는 수송(Transportation)과 관련된 부분을 개선하는 것이 더 효과적이라고 하였다.

제품의 흐름과 관련하여 Mosheiov(1998)은 차

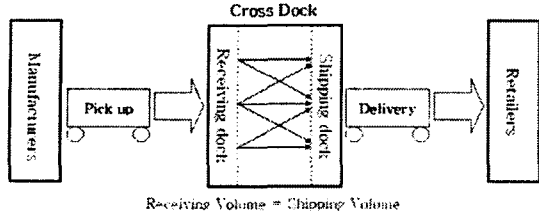
량 경로 문제(VRP: Vehicle Routing Problem)의 일종인 픽업 및 배달(Pick-up and Delivery) 문제를 다루었다. Ceder et al.(2001)는 버스의 환승과정에서 고객의 정체 시간을 최소화함으로써 고객 서비스 수준의 극대화가 가능하다고 보았다. 다수의 버스가 동시에 정류장에 도착할 수 있도록 동시 도착을 위한 버스의 배차 시간 및 스케줄 결정 모델 및 알고리즘을 제시하였다. Lourenco et al.(2001)은 다수의 목적(Multi-Objective)을 갖는 버스 스케줄 문제를 위한 세 가지의 알고리즘을 제안했다. Hwang(2002)은 고객 서비스 수준을 고려한 창고 입지 선정 및 VRP를 위한 유전자 알고리즘(Genetic Algorithm)을 제시하였다.

이와 같이 제품 흐름(Physical Flow)과 관련된 많은 연구가 있었다. 하지만, 비용, 고객 서비스, 유연성(Flexibility) 과 대응(Responsiveness) 등 공급사슬 상에서 제품 흐름을 효율적으로 통제할 수 있는 Cross Docking과 관련된 연구 및 수리적인 접근이 이루어지지 않았다. 따라서 본 논문에서는 공급사슬에서 Cross Docking을 고려한 제품의 수송 문제를 해결할 수 있는 차량의 경로 및 Cross Docking이 이루어지는 시점을 결정하는 구체적인 수리 모델을 제시한다. 또한 Barker et al.(1999)은 Tabu 알고리즘이 VRP 문제에서 가장 좋은 결과를 얻을 수 있다고 언급하였고, 픽업 및 배달 문제가 NP-hard로 정의되어 있기 때문에 본 모델을 위한 새로운 Tabu 알고리즘 및 수치 예제를 제안할 것이다.

2. 공급사슬에서 Cross Docking의 적용

Cross Docking은 공급사슬경영(SCM: Supply Chain Management)에서 적용할 수 있는 효과적인 경영 관리 도구이다. 하지만 창고나 물류 센터에서 공급자로 부터 받은 제품을 바로 배송하는 단순한 창고관리전략(Warehouse Management Strategy)으로 인식되고 있다. 보다 효과적인 Cross Docking의 적용을 위해서는 [그림 1]과 같이 픽업(Pick up), Cross Dock, 배달(Delivery)의 세 단계를 종합적으로 고려해야 한다. 픽업은 생산자로부터 Cross Dock까지의 제품 수송으로 차량의 동시 도착이 이루어진다. Cross Dock은 Receiving Dock에서 Shipping Dock으로 제품이 이동하는 단계이다. 공급자로부터 받은 제품과 소매상으로 배송되는 제품

의 수량이 일치해야 하며, 결국 재고 저장 과정 없이 각 고객별로 제품 통합이 이루어진다. 배달은 Cross Dock에서 통합된 제품들이 다수의 최종 목적까지 배송되는 것을 말한다.



[그림 1] 공급사슬에서의 Cross Docking

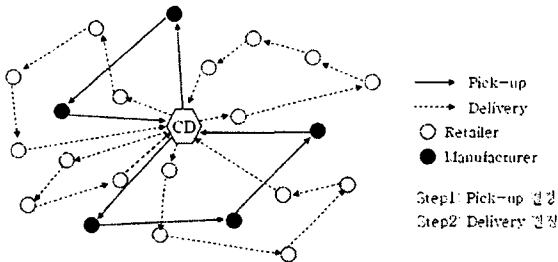
Cross Docking은 제품 및 목적지별로 구분 될 수 있다. 제품별은 제품 통합(Consolidation)을 하지 않고, 하나의 차량에 한 종류의 제품만을 대량으로 적재해서 필요한 여러 고객을 순회, 배달이 이루어지는 것을 말하고, 목적지별은 하나의 차량에 할당된 고객별로 여러 종류의 제품을 통합해서 한번에 배달하는 Cross Docking을 의미 한다.

공급 사슬에는 다수의 Cross Dock이 존재할 수 있다. Cross Dock은 지역적으로 적합한 장소에 독립적으로 위치하고 있으며, 특정 지역 내의 고객들을 담당하고 있다. Barbarosoglu et al.(1999)은 VRP에서 다수의 창고로 이루어진 물류 네트워크 전체를 최적화 하는 것은 불필요하며, 지역적으로 독립된 하나의 창고를 중심으로, 하나의 네트워크를 최적화함으로써 전체 문제를 해결 할 수 있다고 주장하였다.

본 논문의 모델은 Cross Docking을 적용하기 적합한 식품업 산업에서 목적지별 Cross Docking이 이루어지는 환경을 가정한다.

3. Cross Docking 모델

본 연구는 공급사슬에서 Cross Docking을 고려한 차량 경로 스케줄(VRS) 문제로, 전체 Cross Docking 과정에서 픽업(Pick up)과 배달(Delivery) 부분만을 고려하고, Cross Dock에서 이루어지는 작업 시간은 일정하다고 가정한다.



[그림 2] Cross Docking 네트워크

[그림 2]와 같이 n개의 노드로 구성된 네트워크에

서 시간 T 동안 전체 비용을 최소로 하는 차량의 경로와 수를 결정하고, 픽업 과정에 사용된 각 차량의 동시도착시점(Cross Docking Time) 및 배차 스케줄을 결정한다.

기호

- n : 전체 노드의 수 (생산자, 소매상)
- m : 최대 가용한 차량 수
- Q : 각 차량의 최대 적재 용량
- p_i : Pick up 과정인 노드 i 의 공급량
- d_i : Delivery 과정인 노드 i 의 수요량
- c_{ij} : 노드 i 와 j 사이의 차량 수송비용
- c_k : 차량 k 의 운영비용
- y_{ij} : Pick up에서 노드 i 에서 j 로의 총수송량
- z_{ij} : Delivery에서 노드 i 에서 j 로의 총수송량
- t_i : 노드 i 에서의 서비스 시간
- t_{ij} : 노드 i 와 j 사이의 차량 이동시간

결정 변수

- x_{ij}^k : $\begin{cases} 1: \text{차량 } k \text{가 } i \text{에서 } j \text{로 이동하는 경우} \\ 0: \text{그렇지 않은 경우} \end{cases}$
- v_k : $\begin{cases} 1: \text{차량 } k \text{가 사용되는 경우} \\ 0: \text{그렇지 않은 경우} \end{cases}$

ST_i^k : 노드 i 에서 차량 k 의 출발 시간

ET^k : Cross Docking time(Pick up 종료시점)

수리 모형

$$\text{Min} \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m c_{ij} x_{ij}^k + \sum_{k=1}^m c_k v_k \quad (1)$$

subject to

$$\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad \forall j = 1, \dots, m \quad (2)$$

$$\sum_{j=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad \forall i = 1, \dots, n \quad (3)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0 \quad \forall p = 1, \dots, n \quad (4)$$

$$k = 1, \dots, m$$

$$\sum_{j=1}^n x_{0j}^k \leq m \quad \forall k = 1, \dots, m \quad (5)$$

$$\sum_{k=1}^m x_{0j}^k \leq 1 \quad \forall j = 1, \dots, n \quad (6)$$

$$\sum_{k=1}^m x_{i0}^k \leq 1 \quad \forall i = 1, \dots, n \quad (7)$$

$$y_{ij} + z_{ij} \leq Qx_{ij}^k \quad \forall k = 1, \dots, m \quad (8)$$

$$\sum_{i=1}^n p_i = \sum_{i=1}^n d_i \quad (9)$$

$$y_{jl} - y_{ij} \begin{cases} p_j & \text{if } j \in \text{pick up} \\ 0 & \text{if } j \in \text{delivery} \\ -\sum_{i=1}^n p_i & \text{if } j \in 0 \end{cases} \quad (10)$$

$$z_{ij} - z_{jl} \begin{cases} 0 & \text{if } j \in \text{pick up} \\ d_j & \text{if } j \in \text{delivery} \\ \sum_{i=1}^n d_i & \text{if } j \in 0 \end{cases} \quad (11)$$

$$\sum_{i=0}^n \sum_{j=0}^n t_i^k x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T \quad (12)$$

$$\forall k = 1, \dots, m$$

$$ST_j^k = (t_{ij} + ST_i^k + t_j) x_{ij}^k \quad (13)$$

$$\forall k = 1, \dots, m$$

$$ET^k = (ST_i^k + t_{i0}) x_{i0}^k \quad \forall k = 1, \dots, m \quad (14)$$

$$ET^k = ET^{k'} \quad \text{if } x_{i0}^k = x_{i0}^{k'} \quad (15)$$

$$\forall k \neq k', k \text{ and } k' \in 1, \dots, m$$

$$i, j \in \text{pick up}$$

본 연구의 모델은 차량 경로 및 차량 운영비용을 최소로 하기 위해 세 가지의 관련 제약이 존재한다. 식(2)부터 식(7)까지는 차량의 경로를 결정하기 위한 제약 조건으로 출입 회수 및 운영 가능 차량의 제한, 출발 및 도착에 관한 제약 조건으로 각 노드는 하나의 차량에 할당되어야 한다. 식(8)에서 식(11)은 제품의 수량 관련 제약들로 픽업이나 배달 진행에 따라서 차량이 제품을 수송하게 되고, 픽업하는 총수량과 배달하는 총수량이 일치해야 함을 나타낸다. 식(12)은 전체 작업 시간이 일정 시간 T 동안에 이루어져야 하는 것을 의미하고, 식(15)을 통해서 Cross Docking Time 및 차량의 운행 시간 테이블을 얻을 수 있다.

4. Cross Docking을 위한 Tabu 알고리즘

본 알고리즘은 3단계로 구성된다. 1단계에서 탐욕적 알고리즘(Greedy Algorithm)과 α 값에 의하여 초기해가 생성한다. 가장 좋은 값에 α 을 곱한 범위 내의 모든 노드로 후보 리스트를 작성하고, 랜덤한 방법으로 후보 리스트에 속한 하나의 노드를 선택하게 된다. 탐욕적 알고리즘에 α 개념을 추가하여 다수의 다른 초기해를 생성할 수 있다. 2단계는 지역해 탐색 과정으로 설정된 초기해를 개선하기 위하여 k-opt를 사용, 가장 나쁜 영향을 미치는 두 개의 노드를 교환한다. 3단계는 국지해를 얻기 위하여 제거타부리스트(Remove Tabu List)를 사용하게 되며, 위 과정을 1단계부터 반복한다.

Step 1. 초기화 단계

$p_i, d_i, c_{ij}, c_k, n, m, t_i, t_j, T, \alpha$ 을 초기화 한다.

Step 2. 초기해 생성 단계

노드 선정 시 α 을 반영한 선정 가능 후보 리스트를 작성하고, 리스트 내 하나를 랜덤하게 선택한다. 차량 적재 능력 한도까지 위 과정을 반복하고, 한도 초과 시 차량을 추가한다.

Step 3. 지역해 탐색 과정

초기해에서 두개의 나쁜 노드(Worst Node)를 제거, 교환한다. 개선된 경우 지역해로 지정, 일정 회수 이상 반복, 최종 지역해를 선정한다.

Step 4. Tabu 탐색 과정

지역해에서 가장 나쁜 노드 두개를 교환, 이웃해를 작성, 이를 초기해로 step 3을 반복, 이때 제거 타부리스트(Remove Tabu list)를 사용한다.

Step 5. 초기해 반복 생성 과정

다른 초기해로 위 과정을 반복하기 위해서 일정 조건을 만족할 때 까지 step 1로 이동한다.

Step 6. α 의 변화

α 값을 변화 시키고, step 1로 이동한다.

Step 7. 결과 출력

각 차량의 경로 및 배차 시간, Cross Docking Time, 차량별 스케줄을 출력한다.

5. 수치 예제

본 실험의 대상은 [그림 2]에서 언급된 것처럼 하나의 Cross Dock이 다수의 생산자 및 소매상과 관련된 픽업 및 배달을 담당하고 있으며, 편의상 Cross Dock에서의 서비스 시간은 일정하지만 없는 것으로 가정한다.

<표 1> 전체 노드가 50개인 경우 (n = 50)

| α | 운행차량수 | Cross Docking time | 작업시간 | 비용 |
|----------|-------|--------------------|------|---------|
| 0.1 | 17 | 281 | 701 | 29499.2 |
| 0.2 | 17 | 281 | 709 | 29339.6 |
| 0.3 | 17 | 338 | 763 | 29138 |
| 0.4 | 17 | 338 | 790 | 28975.6 |
| 0.5 | 17 | 338 | 764 | 29194 |
| 0.6 | 17 | 338 | 798 | 29558 |
| 0.7 | 17 | 360 | 753 | 29406.8 |
| 0.8 | 17 | 360 | 740 | 29048.4 |
| 0.9 | 17 | 360 | 769 | 29572 |
| 1.0 | 17 | 360 | 862 | 29986.4 |

<표 2> 전체 노드가 100개인 경우 (n = 100)

| α | 운행차량수 | Cross Docking time | 작업시간 | 비용 |
|----------|-------|--------------------|------|---------|
| 0.1 | 27 | 355 | 753 | 45770.8 |
| 0.2 | 27 | 326 | 774 | 47599.6 |
| 0.3 | 27 | 414 | 1077 | 47873.6 |
| 0.4 | 27 | 383 | 800 | 46260.8 |
| 0.5 | 27 | 293 | 757 | 47619.2 |
| 0.6 | 27 | 326 | 841 | 47260.8 |
| 0.7 | 26 | 402 | 937 | 46372.8 |
| 0.8 | 27 | 316 | 828 | 45888.4 |
| 0.9 | 26 | 407 | 914 | 48181.6 |
| 1.0 | 27 | 331 | 880 | 47636 |

6. 결론

공급사슬에서 Cross Docking의 운영은 기업이 직면하고 있는 많은 문제들을 효과적으로 해결하고, 고객서비스의 향상, 기업의 경쟁력 강화 및 효율성 향상을 위한 도구로써 반드시 이루어져야 하는 경영 기법 가운데 하나이다. 본 연구에서는 기

존에 제시되지 않았던 Cross Docking의 새로운 수리적 모델을 제안했고, 픽업(Pick up) 및 배달(Delivery)을 동시에 고려한 VRP 문제의 효과적인 해결 수단인 Tabu 탐색 과정을 통해서 계량적인 접근을 이루었다.

참고 문헌

- [1] Apte, U.M. and S. Viswanathan, "Effective Cross Docking for improving distribution efficiencies," *International Journal of Logistics*, Vol. 3(2000), pp.291-302.
- [2] Baker, B.M. and J. Sheasby, "Extensions to the generalized assignment heuristic for vehicle routing," *European Journal of Operational Research*, Vol. 119(1999), pp.147-157.
- [3] Barbarosoglu, G. and D. Ozgur, "A Tabu search algorithm for the vehicle routing problem," *Computer & Operational Research*, Vol. 26(1999), pp.255-270.
- [4] Ceder, A. and B. Golany and O. Tal, "Creating bus timetables with maximal synchronization," *Transportation Research Part A*, Vol. 35(2001), pp.913-928.
- [5] Cetinkaya, S. and J.H. Bookbinder, "Stochastic models for the dispatch of consolidated shipments," *Transportation Research Part B*, Vol. 37(2003), pp.741-768.
- [6] Hwang, H.S., "Design of supply chain logistics system considering service level," *Computer & Industrial Engineering*, Vol. 43(2002), pp.283-297.
- [7] Kinnear, E. "Is there any magic in Cross - Docking?," *Supply Chain Management*, Vol. 2(1997), pp.249-52.
- [8] Lau, H.C. and M. Sim and K.M. Teo, "Vehicle routing problem with time windows and a limited number of vehicles," *European Journal of Operational Research*, Vol. 148(2003), pp.559-569.
- [9] Lourenco, H.R. and J.P. Paixao, R. Portugal, "Multi objective Metaheuristics for the bus driver scheduling problem," *Transportation Science*, Vol. 35(2001), pp.331-343.
- [10] Mosheiov, G., "Vehicle routing with Pick-up and Delivery: tour - partitioning heuristics," *Computers & Industrial Engineering*, Vol. 34(1998), pp.669-684.