

# 납품시간창과 다종차량을 고려한 다종제품 동적로트크기결정 및 디스패칭 문제를 위한 유전 알고리즘

김병수<sup>1</sup> · 채승규<sup>2</sup> · 이운식<sup>3\*</sup>

<sup>1</sup> 인천대학교 산업경영공학과 / <sup>2</sup> 부경대학교 일반대학원 시스템경영공학과

<sup>3</sup> 부경대학교 시스템경영공학부

## Genetic Algorithms for a Multi-product Dynamic Lot-sizing and Dispatching Problem with Delivery Time Windows and Multi-vehicle Types

Byung Soo Kim<sup>1</sup> · Syungkyu Chae<sup>2</sup> · Woon-Seek Lee<sup>3</sup>

<sup>1</sup>Department of Industrial and Management Engineering, Incheon University

<sup>2</sup>Department of Systems Management and Engineering, Graduate School, Pukyong National University

<sup>3</sup>Division of Systems and Management and Engineering, Pukyong National University

This paper analyzes a multi-product inbound lot-sizing and outbound dispatching problem with multi-vehicle types in a third-party logistics distribution center. The product must be delivered to the customers within the delivery time window and backlogging is not allowed. Replenishing orders are shipped by several types of vehicles with two types of the freight costs, i.e., uniform and decreasing, are considered. The objective of this study is to determine the lot-size and dispatching schedules to minimize the total cost with the sum of inbound and outbound transportation and inventory costs over the entire time horizon. In this study, we mathematically derive a mixed-integer programming model and propose a genetic algorithm (GA1) based on a local search heuristic algorithm to solve large-scale problems. In addition, we suggest a new genetic algorithm (GA2) with an adjusting algorithm to improve the performance of GA1. The basic mechanism of the GA2 is to provide an unidirectional partial move of products to available containers in the previous period. Finally, we analyze the results of GA1 and GA2 by evaluate the relative performance using the gap between the objective values of CPLEX and the each algorithm.

**Keywords:** Delivery time windows, Dynamic lot-sizing, Multi-product, Multi-vehicle types, Dispatching, Genetic algorithms, Local search heuristic

### 1. 서론

제 3자 물류의 흐름에서 생산업체로부터 납품된 완제품은 저장과 분배를 위해 제 3자 물류 창고로 운송되며, 보관되고 있

는 일부의 제품은 고객의 수요에 따라 제 3자 물류창고로부터 고객에게 분배되고 납품된다. 제 3자 물류회사는 제조업체와 제 3자 물류회사간의 전략적 계약 하에서 물품의 재고관리와 운송을 대행한다. 대부분의 경우에 제조업체로부터 제 3자 물

이 논문은 부경대학교 자율창의학술연구비(2014년)에 의해 연구되었음.

\* 교신저자 : 이운식 교수, 608-737 부산광역시 남구 용소로 45 부경대학교 시스템경영공학부, Tel : 051-629-6482, Fax : 051-629-6478,

E-mail : iewslee@pknu.ac.kr

2014년 6월 11일 접수, 2014년 9월 25일 수정본 접수, 2014년 10월 10일 게재 확정.

류창고로의 제품의 운송은 여러 가지의 운송 수단(차량, 트럭, 선박, 비행기 등)을 운용하게 되며 다양한 적재용량을 갖는 운송 수단의 조합은 수송비용과 제 3자 물류창고에서의 공급비용과 재고비용에 영향을 미치게 된다. 또한, 제 3자 물류창고로부터 고객에게 납품되는 제품들은 고객과의 공급계약에 의해 특정한 벌과비용(Penalty cost) 없이 납품될 수 있는 납품 허용기간을 배려 받을 수 있으며 이를 납품시간창(Delivery Time Windows)이라 하고 각각의 수요에 대해 독립적으로 정해진 납품시간창은 가장 이른 납품시간과 가장 늦은 납품시간으로 구성되는 시간 구간으로 형성되며 제품은 이 구간 내에 고객에게 인도되어야 한다. 본 논문에서는 제 3자 물류창고에서 다양한 운송 수단을 고려한 다종 제품의 인바운드 로트크기결정 및 운송계획과 납품시간창을 고려한 아웃바운드 디스패칭 계획을 동시에 결정하는 동적 로트크기결정 및 디스패칭 문제를 연구한다. 수요는 계획기간 동안 매 기간 동적(Dynamic)으로 발생하는 동적 수요를 고려하고 추후조달(backlogging)은 허용되지 않는다.

관련 연구들을 살펴보면 다음과 같다. 본 논문에서 다루고 있는 납품시간창을 고려한 연구는 Lee *et al.*(2001)에 의해 시작되었다. 그들은 추후조달이 허용되지 않는 경우와 허용되는 경우에 대해 각각  $O(T^2)$ 과  $O(T^3)$  알고리즘을 제안하였다. Jaruphongsa *et al.*(2004)은 제한된 창고용량과 납품시간창을 고려한 동적 로트크기결정 문제를 다루었다. Hwang and Jaruphongsa(2006)는 새로운 분해원리를 기초로 납품시간창을 고려한 동적 로트크기결정문제에 대한  $O(NT^3)$  알고리즘을 개발하였다. Hwang(2007)은 Lee *et al.*(2001)의 모형에서 추후조달이 허용되지 않는 경우에 대한 개선된  $O(\text{Max}\{T^2, NT\})$  알고리즘을 제안하였다. 최근, Hwang과 Jaruphongsa(2008)는 주요 수요에 대해서는 납품시간창을 허용하고 부수적인 수요에 대해서는 시간창을 허용하지 않는 동적 로트크기결정 문제를 다루었고 해법으로  $O(N^2T^2)$  알고리즘을 개발하였다. Lee(2010)는 생산과 수송이 통합된 제 3자 물류환경에서 납품시간창을 고려한 동적 로트크기결정 문제에 대한 알고리즘을 제안하였다. Kim and Lee(2010)는 납품시간창을 허용하고 단일 운송차량을 고려하여 동적 로트크기결정 문제의 효율적인 해를 구하기 위한 유전알고리즘을 제시하였다. Kim and Lee(2011)는 다종의 차량을 고려한 동적 공급계획과 아웃바운드 디스패칭에 대해 지역해 탐색방법을 포함하는 유전알고리즘을 제시하였다. 또한 Kim *et al.*(2012)은 Kim and Lee(2010)의 문제 상황 하에 두 가지 고객 수요 패턴을 고려한 효율적인 유전알고리즘을 제안하였다. 최근에 Seo and Lee(2014)는 Kim and Lee(2011)의 문제 상황 하에서 작은 규모의 문제에 대한 최적해를 효율적으로 찾을 수 있는 동적계획법 알고리즘을 제시하였다.

지금까지의 연구들은 납품시간창과 다종의 제품을 고려한 아웃바운드 디스패칭에 관한 연구 혹은 납품시간창이 존재하지 않는 상황 하에 인바운드 로트크기결정에 관한 연구를 다

루었다. 그러나 납품시간창, 다종의 제품, 그리고 다종의 차량을 모두 고려한 인바운드 로트크기결정과 아웃바운드 디스패칭에 관한 연구는 존재하지 않는다. 따라서 본 연구의 목적은 납품시간창과 다종의 차량을 고려한 다종제품에 대한 인바운드 동적 로트크기결정과 아웃바운드 디스패칭 계획을 동시에 수립할 수 있는 효율적 알고리즘을 제시하는데 있다. <Table 1>은 납품시간창을 고려한 제 3자 물류 로트크기 결정에 관한 기존 연구들에 관한 정리하고 있으며 본 논문과 관련 연구분야의 연구들과 차별성을 보임으로써 본 연구의 필요성을 제시한다.

**Table 1.** Classification of research for 3PL lot-sizing with time windows

Product Transportation		Single	Multiple
Without		Lee <i>et al.</i> (2001) Jauphongsa <i>et al.</i> (2004) Hwang and Jaruphongsa(2006) Hwang and Jaruphongsa(2008)	-
With	Single Type	Kim and Lee(2010) Kim <i>et al.</i> (2012)	-
	Multiple Types	Kim and Lee(2011) Seo and Lee(2014)	This study

## 2. 최적화 모형

본 연구에서 다루는 다종제품 각각은 동일한 중량과 부피를 가지는 제품이라고 가정한다. 따라서 문제의 목적은 관련 비용을 최소화하는 아래의 두 가지 의사결정변수들을 결정하는 것이다.

$x_{int}$  = 기간  $t$ 에서 차량종류  $n$ 에 의해 3PL 창고로 공급되는 제품  $i$ 의 공급량

$d_{ikt}$  = 기간  $t$ 에서 제품  $i$ 에 대한 수요  $k$ 의 디스패칭 납품량

위의 두 가지 의사결정변수들의 결정에 의한 종속변수는 다음과 같다 :

$I_{it}$  = 제 3자 물류창고에서 기간  $t$ 에서의 제품  $i$ 에 대한 재고량.

또한, 본 논문에서 다루어질 최적화 모형을 효과적으로 표현하기 위해 다음과 같이 모수들을 정의한다 :

$TD_{ik}$  = 제품  $i$ 에 대한 수요  $k$ 의 총괄수요량

$E_{ik}$  = 제품  $i$ 에 대한 수요  $k$ 의 가장 이른 납품시간

$L_{ik}$  = 제품  $i$ 에 대한 수요  $k$ 의 가장 늦은 납품시간

- $K_{it}$  = 기간  $t$ 에서 제품  $i$ 를 구입하는데 드는 준비비용
- $p_{it}$  = 기간  $t$ 에서 제품  $i$ 의 단위당 구입비용
- $S_{it}$  = 기간  $t$ 에서 제품  $i$ 에 대한 디스패칭 준비비용
- $c_{it}$  = 기간  $t$ 에서 제품  $i$ 에 대한 단위당 디스패칭비용
- $h_{it}$  = 기간  $t$ 에서 제품  $i$ 에 대한 단위당 재고유지비용
- $M_i$  = 계획기간  $T$ 동안에 발생하는 제품  $i$ 에 대한 수요의 개수
- $L$  = 다중 제품의 종류
- $A_n$  = 차량형태  $n$ 에 대한 운송비용
- $W_n$  = 차량형태  $n$ 에 대한 적재용량

본 논문에서 고려되는 제 3자 물류창고의 운영에 영향을 미치는 주요 요소들로는 공급비용 함수(구입비용 + 운송비용), 납품시간차, 아웃바운드 납품비용 함수, 그리고 재고유지비용 함수가 있다. 다종의 제품  $i$ 에 대해 기간  $t$ 에서 차량 형태(차량의 종류)  $n$ 에 의한 3PL 창고로의 공급량과 관련된 비용함수는 다음과 같다.

$$P_{it}(x_{int}) = K_{it}\delta\left(\sum_{n=1}^N x_{int}\right) + p_{it}\sum_{n=1}^N x_{int} + A_n \left\lceil \sum_{i=1}^L x_{int}/W_n \right\rceil \quad (1)$$

여기서  $\lceil a \rceil$ 는  $a$ 를 초과하는 정수 중 가장 작은 정수이고,  $\delta(\cdot)$ 는 0-1 정수변수를 의미한다. 상기의 공급비용함수는 다수의 고정비용을 갖는 오목함수가 된다.

본 논문에서는 특정한 벌과 비용(Penalty Cost) 없이 제품의 보관이 허용되는 시간구간인 납품시간차를 고려한다.  $d_{ikt}$ 를 제품  $i$ 에 대한 수요  $k$ 의 양이라 할 때,  $d_{ikt}$ 는 기간  $t$ 에서의 제품  $i$ 에 대한 수요  $d_k$ 의 디스패칭 납품량이라 정의한다. 제품  $i$ 에 대한 수요  $k$ 의 납품시간차는 가장 이른 납품시간  $E_{ik}$ 와 가장 늦은 납품시간  $L_{ik}$ 로 구성되는 시간구간( $E_{ik}, L_{ik}$ )로 형성되며 이 구간 내에 제품은 고객에게 반드시 인도되어야 한다. 이를 수식으로 표현하면 다음과 같다.

$$\sum_{t=E_{ik}}^{L_{ik}} d_{ikt} = d_{ik}, \quad k = 1, 2, \dots, M_i \quad (2)$$

여기서,  $M_i$ 는 계획기간  $T$ 동안에 발생하는 제품  $i$ 에 대한 수요  $k$ 의 개수를 나타내며  $d_{ikt}$ 는 의사결정변수이다. 따라서, 납품시간차( $E_{ik}, L_{ik}$ ) 내에서 “어떤 제품이, 언제, 얼마만큼 디스패칭 하는지?”가 중요한 의사결정의 요소가 되며 이에 따라 납품비용이 발생하게 된다. 본 연구에서 납품비용은 고정비용함수를 고려하며 그 형태는 다음과 같다.

$$S_{it}\delta\left(\sum_{k=1}^{M_i} d_{ikt}\right) + c_{it}\sum_{k=1}^{M_i} d_{ikt} \quad (3)$$

여기서  $S_t$ 와  $c_t$ 는 각각 기간  $t$ 에서 3PL 창고로부터 고객으로

의 디스패칭 고정비용과 단위당 배달비용을 나타낸다. 상기에 정의된 파라미터, 의사결정변수 및 수식들을 사용하여 본 문제에 대한 최적화 모형은 다음과 같은 혼합정수모형(P)으로 정형화된다.

$$(P) \text{ Min } \sum_{t=1}^T \sum_{i=1}^L \left( \sum_{n=1}^N P(x_{int}) + h_{it}I_{it} + S_{it}\delta\left(\sum_{k \in M_i} d_{ikt}\right) + c_{it}\sum_{k \in M_i} d_{ikt} \right) \quad (4)$$

$$\text{s.t. } \sum_{n=1}^N x_{int} + I_{i,t-1} - \sum_{k=1}^{M_i} d_{ikt} = I_{i,t}, \quad i = 1, \dots, L; t = 1, \dots, T; \quad (5)$$

$$\sum_{n=E_{ik}}^{L_{ik}} d_{ikt} = TD_{ik}, \quad k = 1, \dots, M_i; i = 1, \dots, L, \quad (6)$$

$$d_{ikt} \geq 0, \quad k = 1, \dots, M_i; i = 1, \dots, L; t = E_{ik}, \dots, L_{ik}, \quad (7)$$

$$d_{ikt} = 0, \quad k = 1, \dots, M_i; i = 1, \dots, L; t = 1, \dots, E_{ik} - 1 \quad (8)$$

$$d_{ikt} = 0, \quad k = 1, \dots, M_i; i = 1, \dots, L; t = L_{ik} + 1, \dots, T; \quad (9)$$

$$I_{i0} = I_{iT} = 0, \quad i = 1, \dots, L, \quad (10)$$

$$x_{int} \geq 0, \quad I_{it} \geq 0, \quad i = 1, \dots, L; n = 1, \dots, N; t = 1, \dots, T. \quad (11)$$

### 3. 유전 알고리즘

본 논문에서는 문제(P)에 대한 우수한 해를 효율적으로 탐색하기 위한 방안으로 유전 알고리즘을 제안한다.

#### 3.1 해의 표현 및 초기해의 생성

본 문제의 두 가지 의사결정변수들은 차량형태에 따른 공급량( $x_{int}$ )과 디스패칭량( $d_{ikt}$ )이다. 디스패칭량에 대한 염색체(Chromosome) 표현은 디스패칭량의 구조적 특성상 염색체 설계를 위해 2차원 행렬 표현법을 각 제품에 대해 사용한다. 초기 디스패칭량( $d_{ikt}$ )이 생성되면, 기간  $t$ 에서 모든 수요의 합이 결정되고, 이를 이용하여 각 기간의 총공급량( $x_{i \cdot t}$ )이 결정된다.

여기서  $x_{i \cdot t} = \sum_{n=1}^N x_{int}$ 를 의미한다. 대부분의 로트크기결정 문제의 유전알고리즘 표현은 각 시점의 로트의 개수가 염색체로 표현된다. 본 논문에서는 각 기간의 총공급량( $x_{i \cdot t}$ )의 표현을 위하여 ‘0-1’ 인코딩 체계를 가지고 염색체는 계획기간  $T$ 의 수와 같은 개수의 0 혹은 1 중의 한 개의 숫자로 표현되고 이러한 표현이 제품의 종류  $L$ 의 수만큼 존재하는 2차원 염색체 표현을 제안하였다. 각 기간의 총공급량( $x_{i \cdot t}$ )의 생성절차는 다음과 같다.

Step 1 : 각 제품  $i$ 와 각 시점  $t$ 에 대한 염색체에 0 또는 1의 값 중 랜덤하게 하나의 값을 입력한다.

Step 2 : 각 제품  $i$ 에 대해 디스패칭량이 최초로 값을 가지는 특정 시점  $p$  이전에 입력된 주문량의 염색체가 모두 0이라면  $p$ 시점의 주문량 염색체는 1의 값으로 수정한다.

이렇게 입력받은 주문량의 염색체값은 다음에 소개할 수송 차량 할당을 위한 지역해 탐색 알고리즘과 Fitness Value를 계산하기 위해 실수 배열 값으로 Decoding 과정이 요구된다. Decoding을 위한 내용은 다음과 같다.

0 : 기간  $t$ 때의 수송량이 0

1 : 시점  $t$ 이후 염색체가 0이 아닌 시점이 나올 때까지의 수요에서 시점  $t$ 의 직전 재고를 뺀 수요량을 만족하는 가장 작은 정수

### 3.2 수송차량 할당을 위한 지역해 탐색 알고리즘

차량 형태를 고려하지 않은 주문량  $x_{i,t}$ 의 값이 정해졌으므로 차량 형태를 고려한 주문량  $x_{int}$ 의 값을 계산하기 위해 다음과 같은 알고리즘을 사용한다. 주문량  $x_{i,t}$ 의 값보다 작은 차량 수송량을 가진 차량 형태  $n$ 에 대해서만 차량의 운송단위당 운송비용인  $UC_{nt}$ 의 값으로 계산하고 주문량  $x_{i,t}$ 의 값보다 큰 차량 수송량을 가지는 차량 형태  $n$ 에 대해서는  $UC_{nt}$ 의 값을 가장 큰 양의 정수( $M$ )를 할당한다.

Step 1 : if ( $x_{i,t} \geq W_n$ ) {  
     차량의 운송단위당 운송비용  $UC_{nt} = A_{nt}/W_{nt}$   
 } else {  $UC_{nt} = M$  }

계산된  $UC_{nt}$ 의 값 중 가장 작은 값의 차량 형태를 선택한다. 동일한  $UC_{nt}$ 의 값을 가지는 차량 형태가 2개 이상이라면 한번에 수송 가능한 차량의 양이 가장 작은 차량 형태를 선택한다. 그 후 선택된 차량의 이용 횟수를 구한 후 선택된 차량으로 수송할 양만큼 주문량에서 차감하고 이를 수정한다. 수정된  $x_{i,t}$ 의 값이 모든 차량 형태 중 가장 작은 수송량을 가지는 차량 형태보다 작다면 Step 3으로 아니라면 Step 1로 가서 다시 반복 수행한다.

Step 2 : 최소  $UC_{nt}$ 인 차량 형태  $p$ 를 선택

if ( $x_{i,t} \geq W_p$ ) {  
     시점  $t$ 일 때, 차량 유형  $p$ 의 이용 수  $y_{tp}$ 는 다음과 같다:  
      $y_{tp} = \lfloor x/W_p \rfloor$   
     update  $x' \leftarrow x - (y_{tp} \times W_p)$   
     if ( $x \geq \text{Lowest}(W_n)$ ) {  
         go to step 1  
     } else { go to step 3 }  
 } else { go to step 3 }

Step 3의 경우 초기 주어진 주문량이 가장 작은 차량보다 작은 양이거나 다른 차량에 이미 할당이 되고 난 후 남은 자투리를 계산하는 방식이다. 먼저  $UC_{nt}$  값이 아닌 차량 하나의 이용 가격이 가장 작은 차량 형태를 선택하고 선택된 차량 형태가 이전에 선택된 적이 있다면 계산된 차량의 이용 수에 1을 더하고 아니라면 현재 선택된 차량의 수치를 1로 둔다.

Step 3 : 최소  $A_n$ 인 차량 형태  $z$ 를 선택

if ( $p = k$ ) {  
      $y_{tp} + \lfloor x/W_z \rfloor$   
 } else {  $y_{tz} = \lfloor x/W_z \rfloor$  }

### 3.3 교차변이 및 돌연변이

유전 알고리즘에서 자식하는 기본적으로 선정된 부모해들을 임의로 짝을 지어 교차변이나 돌연변이 연산으로 생성하나, 가장 우수한 해가 일정비율 이상 부모가 되도록 조정하며 본 논문에서 교차변이 및 돌연변이 비율은 각각 부모해의 40%와 10%이다. 디스패칭량( $d_{ikt}$ )의 교차변이는 Kim and Lee (2010)에서 제안한 다차원 형태의 염색체 교차변이를 이용하였다. 구체적인 알고리즘의 설명을 위해 해집단 내의 임의의 두 개의 디스패칭( $d_{ikt}$ ) 염색체  $d_1 = (d_{ikt}^1)$ ,  $d_2 = (d_{ikt}^2)$ 를 선택한 후 다음과 같이 교차변이를 수행한다.

Step 1 : 두 개의 임시행렬  $P = (p_{ikt})$ ,  $Q = (q_{ikt})$ 을 생성한다 :

$p_{ikt} = \lfloor (d_{ikt}^1 + d_{ikt}^2)/2 \rfloor$  그리고  
 $q_{ikt} = (d_{ikt}^1 + d_{ikt}^2) \bmod 2$ .

Step 2 : 행렬  $Q$ 를 두 개의 행렬  $Q_1$ ,  $Q_2$ 로 분할한다 :

$Q = Q_1 + Q_2$ ,  
 where  $\sum_{t=1}^T q_{ikt}^1 = \sum_{t=1}^T q_{ikt}^2$ , for  $\forall k, i$ .

Step 3 : 다음과 같이 두 개의 자손  $d_1^*$ ,  $d_2^*$ 을 생성한다 :

$d_1^* = p_{ikt} + Q_1$  그리고  $d_2^* = p_{ikt} + Q_2$ .

주문량( $x_{i,t}$ )에 대한 교차변이는 one cut-point 교차변이 방법을 이용하여 수행한다. 교차변이 후 각 제품  $i$ 에 대해 디스패칭량( $d_{ikt}$ )이 최초로 값을 가지는 특정 시점  $p$  이전에 입력된 주문량의 염색체가 모두 0이라면  $p$ 시점의 주문량 염색체는 1의 값으로 수정한다. 이를 수정하지 않을 경우 계약을 벗어나고 재고가 음수값을 가지는 경우가 발생하게 된다.

디스패칭량( $d_{ikt}$ )에 대한 돌연변이 수행은 동일한 납품시간창 내에 임의의 시점  $t$ 에서 발생하는 디스패칭량 전체를 납품시간창 내 임의의 다른 시점으로 보내는 방법이고 주문량( $x_{i,t}$ )에 대한 돌연변이는 돌연변이 확률에 따라 임의의 유전자(Gene)를 선택하여 유전자의 값이 0이라면 1로, 1이었다면 0의 값을 할당한다.

다. 교차변이와 마찬가지로 각 제품  $i$ 에 대해 디스패칭량이 최초로 값을 가지는 특정 시점  $p$  이전에 입력된 주문량의 염색체가 모두 0이라면 시점  $p$ 의 주문량 염색체는 1의 값으로 수정한다.

이렇게 기존 세대와 교차변이, 돌연변이 확률에 따라 생성된 자식 세대는 적합도(Fitness) 값을 낮은 순서대로 순위로 매겨서 해집단(Population)의 수만큼 다음 세대의 부모해가 된다.

### 3.4 차량 할당에 대한 조정 알고리즘

앞서 소개한 유전알고리즘 내의 수송차량 할당을 위한 지역해 탐색 알고리즘은 조정기능 없이 단순히 단방향으로만 작동하여 해의 산출은 용이하나, 최적해와 비교하여 볼 때 상대적으로 열악한 해를 제공할 수 있다. 그 이유는 유전알고리즘 내에 제시된 수송차량 할당을 위한 지역해 탐색 알고리즘은 단방향에 의한 차량할당으로 인해 부분 할당 수송차량들의 취합 조정가능성을 배제하고 있다. 따라서 해의 성능의 향상을 위한 조정 메커니즘을 추가적으로 제안한다.

Step 1 : 주문량  $x_{int}$ 에 대해 각 차량의 운송 가능량을 나눈 나머지 값( $R_{int}$ )을 구한다.

```

for ( i = 1, i ≤ L, i + 1 ) {
    for ( t = 1, t ≤ T, t + 1 ) {
        for ( n = 1, n ≤ N, n + 1 ) {
             $R_{int} = x_{int} \bmod W_n$ 
             $E_{int} = W_n - R_{int}$ 
        }
    }
}
    
```

Step 2 : 특정 시점  $k$ 에 대해 다음과 같은 과정을 수행한다.

```

for ( t = k - 1, t > 0, t - 1 ) {
    for ( n = 1, n ≤ N, n + 1 ) {
        if (  $R_{ink} < E_{int}$  ) {
            if (  $Max(t)$  ) {
                 $x_{int} + R_{ink}$ 
                 $x_{int} - R_{ink}$ 
                for ( t =  $Max(t)$ , t ≥ k, t + 1 ) {
                    I +  $R_{ink}$  }
                for ( t = k, t ≥ T, t + 1 ) {
                     $I_{i,t} - R_{i,n,k}$  }
            }
             $x'_{int} = x_{int}$ 
             $I'_{it} = I_{it}$ 
        }
    }
}
    
```

Step 3 :  $\text{if} \left( A_n \left[ \sum_{i=1}^L x'_{int} / W_n \right] + h \cdot I' \right) > A_n \left[ \sum_{i=1}^L x_{int} / W_n \right] + h_{it} \cdot I_{it} \left\{ \right.$

```

for ( t = k - 1, t ≥ 0, t - 1 ) {
    for ( n = 1, n ≤ N, n + 1 ) {
         $x_{int} = x'_{int}$ 
         $I = I'_{it}$ 
    }
}
else {
     $x_{int} = x_{int}$ 
     $I_{it} = I_{it}$ 
}
    
```

이러한 차량 할당 알고리즘이 추가된 유전 알고리즘(GA2)의 전체 흐름도는 <Figure 1>과 같다.

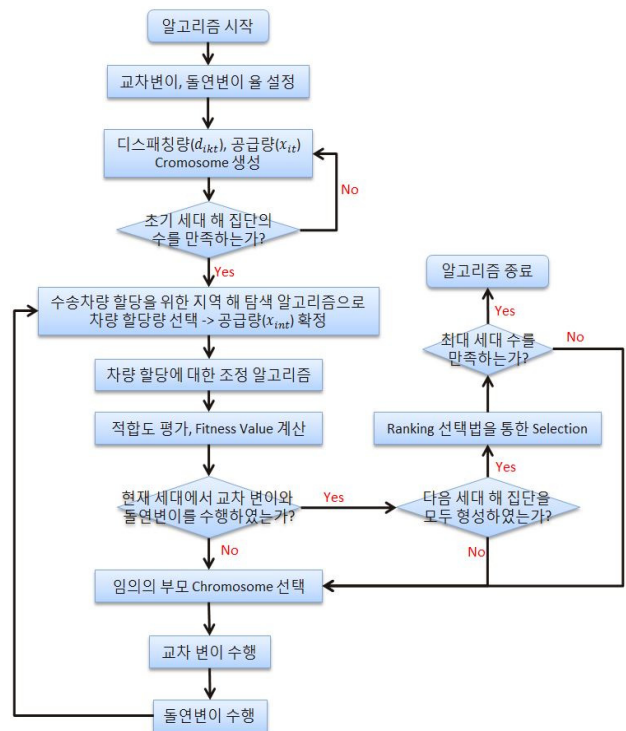


Figure 1. Flow chart of GA2

본 연구에서 조정기능없이 단방향으로만 작동하는 차량할당 방법을 이용하는 유전 알고리즘을 GA1, 차량 할당에 대한 조정 알고리즘이 추가된 유전 알고리즘을 GA2라고 정의하고 CPLEX에 의해 얻어진 해와 비교·분석한다.

## 4. 알고리즘 성능 분석

### 4.1 성능 분석을 위한 실험 환경

본 논문에서 제시한 GA1과 GA2의 성능을 검증하기 위하여, 여러 실험 데이터를 생성하여 최적해와 비교·평가하였으며 실험조건들은 다음과 같다.

- (1) 계획기간( $T$ )은 12, 15, 18의 3가지로 한다.
- (2) 제품수( $L$ )는 3, 5, 7의 3가지로 한다.
- (3) 차량 형태( $N$ )는 3, 5, 7의 3가지로 한다.
- (4) 납품시간창(Window)은 3, 4, 5의 3가지로 한다.
- (5) 각 제품에 대해 납품되는 총 수요 갯수( $TD$ )는 3, 5, 7, 9, 11의 5가지로 하고, 각각의 수요량은  $N(\mu_{ik}, \sigma_{ik}^2)$ 를 따른다.
  - 1) 평균  $\mu_{ik}$ 는 균등분포  $U(100, 250)$ 을 따른다.
  - 2) 표준편차  $\sigma_{ik}^2$ 는 50%의 확률로  $\mu_{ik}$ 와  $\mu_{ik}/5$ 를 택한다.
  - 3) 계획기간이 12일 때는 수요를 3, 5, 7을 고려하고 계획기간이 15일 때는 3, 5, 7, 9를 고려하며 계획기간이 18일 때는 3, 5, 7, 9, 11을 고려하여 계획기간이 길어질수록 수요의 개수 또한 증가하도록 한다.
- (6) 각 제품, 각 시점에 따른 재고유지비용( $h_{it}$ )은 1.0의 값으로 동일하게 둔다.
- (7) 각 제품, 각 시점에 따른 주문준비비용( $K_{it}$ )와 디스패칭 준비비용( $S_{it}$ )의 값은 다음과 같다:
  - 1)  $K_{it} = TW_{ik}^2 \times U_{ik}/2$ ,
  - 2)  $S_{it} = TW_{ik}^2 \times U_{ik}/3$ , 여기서  $TW_{ik}$ 는 납품시간창을 의미한다.
- (8) 각 제품, 각 시점에 따른 구매비용( $p_{it}$ )과 디스패칭 비용( $c_{it}$ )의 값은 다음과 같다:
  - 1)  $p_{it} = \frac{K_{it}}{100}$ ,
  - 2)  $c_{it} = \frac{S_{it}}{100}$ .
- (9) 차량의 적재용량은 차량의 수  $N$ 에 따라 다음과 같이 정한다.
  - 1)  $N=3$ : 150, 250, 350
  - 2)  $N=5$ : 150, 250, 350, 450, 550
  - 3)  $N=7$ : 150, 250, 350, 450, 550, 650, 750
- (10) 운송비( $A_n$ )는 적재용량( $W_n$ )에 따라 다음과 같이 2가지 경우로 한다:
  - 1) 적재용량에 비례할 때( $2 \times W_n$ ):
    - ①  $N=3$ : 300, 500, 700
    - ②  $N=5$ : 300, 500, 700, 900, 1100
    - ③  $N=7$ : 300, 500, 700, 900, 1100, 1300, 1500
  - 2) 적재용량에 비례하여 감소할 때( $n=1$ 일 때 ( $2 \times W_n$ ),  $n>1$ 일 때 ( $1.8 \times W_n$ )):
    - ①  $N=3$ : 300, 450, 630
    - ②  $N=5$ : 300, 450, 630, 810, 990
    - ③  $N=7$ : 300, 450, 630, 810, 990, 1170, 1350
- (11) 상기에 주어진 실험조건에 대해 3개의 결과표본을 발생시키며 성능 분석을 위한 자료에는 3개의 결과표본에 대한 평균으로 분석하도록 한다.

## 4.2 성능 분석

제시된 알고리즘의 성능을 평가하기 위한 척도로 해의 상대비율편차를 이용하였다. 상대비율편차는 최적해에 대한 휴리스틱 해의 편차의 비를 의미하며 식 (12), 식 (13)과 같은 식으로 표현된다.

$$\text{Gap 1} = \frac{Z_{GA1} - Z_{CPLX}}{Z_{CPLX}}, \quad (12)$$

$$\text{Gap 2} = \frac{Z_{GA2} - Z_{CPLX}}{Z_{CPLX}}, \quad (13)$$

여기서,  $Z_{GA1}$  = 유전 알고리즘 GA1으로 산출된 목적함수값,  $Z_{GA2}$  = 유전 알고리즘 GA2로 산출된 목적함수값, 그리고  $Z_{CPLX}$  = Branch and Bound 노드수를 500,000으로 제한하여 산출된 최선해(Best Solution)의 목적함수값으로 정의한다. 그리고 상대비율편차가  $\alpha\%$ 라면 휴리스틱에 의한 해가 최적해에 비해  $\alpha\%$ 만큼 초과한다는 것이므로 상대비율편차가 적을수록 제시된 알고리즘의 성능이 좋다는 것을 의미한다.

<Table 2>는 규모가 작은 문제 중 대표 경우인 제품 종류가 3, 계획 기간이 15, 운송비가 적재용량에 비례하는 경우에 대한 실험결과이다. GA1의 상대비율편차인 Gap 1이 2.63%, GA2의 상대비율편차인 Gap 2가 1.24%로 규모가 작은 문제들에 대해 비교적 제시된 GA1 및 GA2의 알고리즘이 최적해에 근사한 해를 제공해주는 것을 알 수 있다.

<Table 3>는 규모가 큰 문제 중 대표 경우인 제품 종류가 7, 계획 기간이 18, 운송비가 적재용량에 비례하여 감소하는 경우에 대한 실험결과이다. GA1의 상대비율편차의 값이 0% 이하인 경우는 드물게 발생하지만 GA2 즉, 조정 알고리즘을 적용한 후의 최선해 도출은 Solution Gap이 0% 이하인 경우를 자주 볼 수 있다. 본 연구에서는 노드 500,000의 제약에 의해 최적해(Optimum Solution)가 아닌 최선해(Best Solution)를 결과값으로 도출하는 경우가 종종 발생하기 때문에 만약 GA1값과 GA2값이 최선해값보다 효율적인 값을 가진다면 Gap 1과 Gap 2 값이 음의 값을 가지게 된다. 또한 메모리의 용량 제약에 의해 해를 찾지 못하고 계산이 중지되는 경우도 9개의 데이터에서 발생하였으며 이 경우 상대비율편차를 도출하지 못하였기 때문에 상대비율편차의 평균값 계산에서는 제외하였다. 도출된 실험의 결과로부터 GA1의 상대비율편차인 Gap 1의 평균은 (-0.62)%, 그리고 GA2의 상대비율편차인 Gap 2의 평균은 (-1.68)%를 나타내었다. 이는 GA2가 GA1보다 상대적으로 더 효율적인 해를 제공해 줌을 알 수 있다. 그 이유는 제시된 GA2의 차량 할당 후 조정 알고리즘이 차량의 이용 대수를 감소시킴으로써 주문비용을 추가적으로 감소시키는 해를 탐색할 수 있기 때문이다. 제시된 GA1과 GA2의 계산시간은 평균적으로 각각 146.9초, 151.8초로 비교적 짧은 시간에 계산됨을 알 수 있다.

Table 2. Test results for small sized problems

Products	Periods	Ship Mode	V types	Demand	Window	CT (Sec)	GT_1 (Sec)	Gap 1	GT_2 (Sec)	Gap 2	GA_1-GA_2			
3	15	Uniform	3	3	3	10.7	22.5	0.86%	24.7	0.56%	-0.30%			
					4	53.3	30.9	2.73%	33.1	1.53%	-1.18%			
					5	42.6	33.8	3.81%	36.5	2.81%	-0.97%			
				5	3	40.0	32.1	0.79%	34.6	0.69%	-0.10%			
					4	43.0	30.9	0.02%	33.1	0.00%	-0.02%			
					5	19.8	22.8	4.34%	24.9	0.74%	-3.57%			
				7	3	26.8	24.1	0.54%	26.6	0.00%	-0.54%			
					4	24.3	29.1	6.19%	31.3	1.19%	-4.94%			
					5	12.5	30.5	0.83%	32.8	0.00%	-0.83%			
				9	3	32.2	26.1	2.60%	28.6	1.30%	-1.28%			
					4	49.2	31.5	1.60%	34.0	0.00%	-1.60%			
					5	55.0	33.1	2.60%	35.5	1.30%	-1.28%			
				5	3	3	25.9	30.7	4.94%	34.7	4.14%	-0.77%		
						4	23.2	31.1	0.39%	35.2	0.00%	-0.39%		
						5	53.4	48.3	6.21%	52.8	1.61%	-4.53%		
			5		3	13.9	35.5	2.69%	39.4	1.59%	-1.08%			
					4	29.3	33.0	3.09%	37.1	1.09%	-1.98%			
					5	8.9	28.4	5.76%	32.3	4.46%	-1.24%			
			7		3	158.7	52.2	3.55%	56.5	1.85%	-1.67%			
					4	24.3	27.3	3.04%	31.2	1.44%	-1.58%			
					5	27.3	27.6	2.09%	32.0	0.00%	-2.09%			
			9		3	90.4	56.7	0.60%	61.0	0.60%	0.00%			
					4	59.2	29.6	1.90%	33.3	1.60%	-0.30%			
					5	62.7	30.0	1.80%	34.0	0.60%	-1.19%			
			7		3	3	23.9	25.1	1.95%	30.5	1.65%	-0.30%		
						4	18.3	36.9	6.29%	42.5	2.19%	-4.01%		
						5	41.2	33.9	4.93%	39.4	3.73%	-1.16%		
				5	3	63.6	46.4	0.74%	52.5	0.54%	-0.20%			
					4	53.2	46.6	0.55%	51.8	0.55%	0.00%			
					5	38.6	39.4	3.97%	45.4	0.00%	-3.97%			
				7	3	26.1	20.6	2.73%	25.8	2.33%	-0.39%			
					4	26.6	30.0	0.21%	35.5	0.00%	-0.21%			
					5	19.9	38.5	0.83%	44.8	0.73%	-0.10%			
				9	3	31.3	22.4	1.60%	27.7	0.00%	-1.60%			
					4	32.0	32.5	5.00%	38.6	3.70%	-1.25%			
					5	63.9	41.8	2.90%	47.2	0.00%	-2.90%			
				Average					39.6	33.1	2.63%	37.1	1.24%	-1.38%

**Table 3.** Test results for large sized problems

Products	Periods	Ship Mode	V types	Demand	Window	CT (Sec)	GT_1 (Sec)	Gap 1	GT_2 (Sec)	Gap 2	GA_1-GA_2				
7	18	Decreasing	3	3	3	140.0	86.2	5.09%	93.0	0.00%	-5.09%				
					4	198.4	99.6	6.30%	105.9	3.30%	-2.90%				
					5	144.7	85.0	1.80%	91.8	1.10%	-0.69%				
				5	3	394.5	92.5	0.14%	100.0	0.00%	-0.14%				
					4	375.4	85.3	2.10%	91.8	1.30%	-0.79%				
					5	-	85.1	-	92.3	-	-0.04%				
				7	3	752.1	161.2	-2.28%	168.7	-2.57%	-0.30%				
					4	599.0	124.8	-3.86%	132.2	-3.96%	-0.10%				
					5	438.2	88.8	1.01%	95.0	0.00%	-1.01%				
				9	3	268.2	166.0	-0.89%	173.6	-2.38%	-1.52%				
					4	607.6	130.9	2.35%	138.2	2.15%	-0.20%				
					5	-	131.3	-	138.7	-	0.00%				
				11	3	1121.8	106.3	-0.62%	112.5	-1.42%	-0.81%				
					4	729.2	140.1	-2.59%	146.9	-3.47%	-0.91%				
					5	945.4	141.4	3.49%	148.5	1.49%	-1.97%				
				5	3	3	-	96.7	-	107.6	-	0.00%			
						4	1293.3	113.5	-4.57%	124.5	-5.91%	-1.42%			
						5	-	105.7	-	116.1	-	-0.98%			
					5	3	1763.4	119.6	-2.48%	130.1	-3.46%	-1.01%			
						4	2501.5	125.1	-1.14%	137.7	-1.14%	0.00%			
						5	4834.0	158.5	-0.89%	169.9	-2.38%	-1.52%			
					7	3	2955.8	95.3	-6.15%	106.2	-7.56%	-1.52%			
						4	780.4	74.3	-5.01%	85.7	-5.48%	-0.50%			
						5	1072.3	100.5	-2.89%	112.4	-3.28%	-0.40%			
					9	3	624.3	100.8	0.17%	113.4	0.00%	-0.17%			
						4	899.0	78.9	0.85%	91.3	0.00%	-0.85%			
						5	1235.3	103.7	1.44%	114.1	0.00%	-1.44%			
					11	3	745.9	106.3	1.26%	117.2	0.00%	-1.26%			
						4	1078.8	111.8	-1.25%	124.1	-3.13%	-1.94%			
						5	1482.4	108.1	1.98%	119.2	0.68%	-1.29%			
			7		3	3	-	234.3	-	250.9	-	-0.12%			
						4	5367.3	151.6	-4.18%	167.8	-4.85%	-0.70%			
						5	-	203.1	-	220.8	-	-0.01%			
					5	3	906.6	112.4	-1.18%	127.9	-2.66%	-1.52%			
						4	2148.7	173.6	-3.22%	189.1	-4.48%	-1.32%			
						5	3660.0	213.6	-1.01%	229.4	-1.11%	-0.10%			
					7	3	1733.1	183.5	-5.48%	199.4	-6.14%	-0.70%			
						4	-	223.4	-	240.0	-	-1.63%			
						5	-	276.5	-	294.5	-	-1.13%			
					9	3	1996.5	209.7	1.03%	224.5	0.73%	-0.30%			
						4	2475.3	236.2	1.40%	251.3	1.10%	-0.30%			
						5	-	293.1	-	308.3	-	-0.13%			
					11	3	2571.9	226.1	-1.83%	243.4	-3.60%	-1.83%			
						4	-	245.6	-	261.1	-	0.00%			
						5	-	302.2	-	316.7	-	-0.61%			
				Average						1436.5	146.9	-0.62%	158.3	-1.68%	-0.92%



### 4.3 민감도 분석

성능 분석을 통해 비교적 규모가 큰 문제에서 상대적으로 유전 알고리즘의 성능이 CPLEX 계산보다 성능이 우수함을 알 수 있었다. 그러나 본 연구에서 가장 규모가 큰 문제는 제품 종류가 7, 계획 기간이 18, 차량의 종류가 7, 수요의 수가 11인 실험이다. 따라서 보다 큰 규모의 문제에 대한 알고리즘의 성능 예측을 위해 본 절에서는 제 4.2절에서 설계된 알고리즘에 영향을 미치는 주요 인자들(제품종류( $L$ ), 계획기간( $T$ ), 차량 종류( $N$ ))에 대한 민감도 분석을 시행하였다. <Figure 2>, <Figure 3>, 그리고 <Figure 4>는 GA1과 GA2의 주요 인자들이 변화함에 따른 상대비율편차를 나타낸다.

<Figure 2>는 제품 종류에 따른 민감도 분석으로 GA1보다는 GA2의 성능이 상대적으로 우수함을 알 수 있고, 제품의 종류가 증가할수록 성능이 우수함을 알 수 있다.

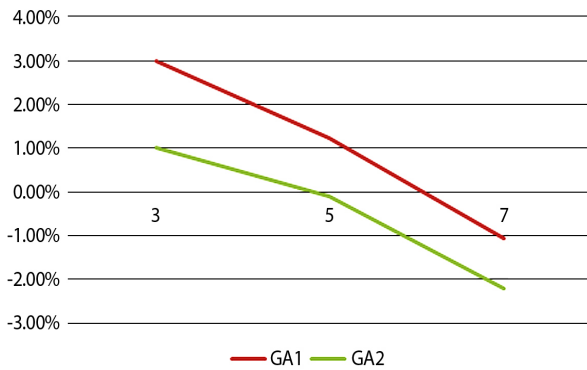


Figure 2. Sensitivity analysis for product type

<Figure 3>은 계획 기간에 따른 민감도 분석으로 계획 기간이 증가할수록 좋은 해를 찾아주는 경향은 보이나 그 정도가 미미함을 알 수 있다.

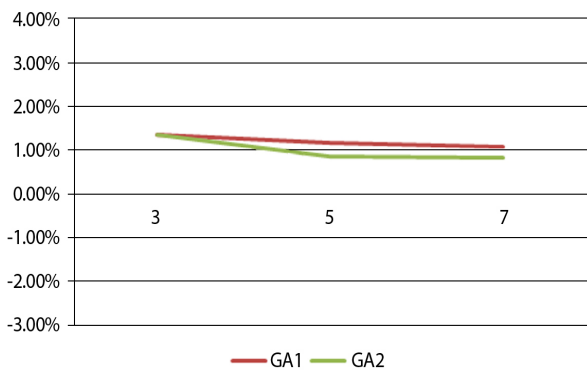


Figure 3. Sensitivity analysis for planning horizon

<Figure 4>는 차량 종류에 따른 민감도 분석으로 제품 종류에 따른 민감도 분석과 비슷한 경향을 보이나 성능 변화폭은 조금 덜함을 알 수 있다. 전반적으로 인자의 수가 증가함에 따라 유전 알고리즘의 성능이 우수해짐을 알 수 있다.

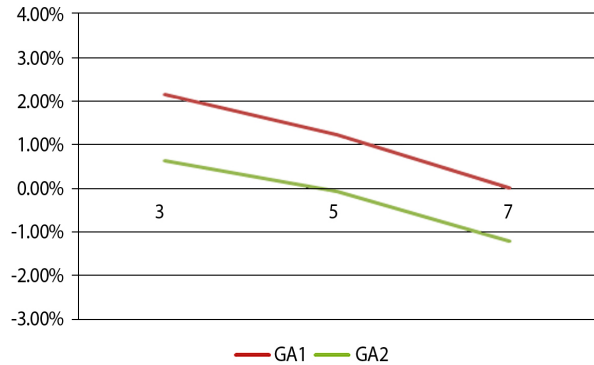


Figure 4. Sensitivity analysis for vehicle type

## 5. 결론 및 추후연구

본 연구에서는 제 3자 물류 창고에서 납품기간차과 다종의 차량을 고려한 동적수요를 갖는 다종제품의 인바운드 로트크기 결정 및 아웃바운드 디스패칭 문제를 연구하였다. 본 문제를 풀기위한 혼합정수계획모형의 수리모형을 제시하였고 제안된 수리모형을 검증하기 위하여 문제 P의 수리모형을 OPL (Optimization Programming Language)을 이용하여 모델링하였으며, Solver Engine은 ILOG사의 CPLEX 12.5를 이용하여 계산에 소요된 시간과 최선해를 찾아보았다. 또한, 현실적으로 큰 규모를 가지는 문제를 풀기 위해 지역해 탐색 휴리스틱 알고리즘을 기반으로 하는 유전 알고리즘 GA1과 GA2를 제안하였다. GA1과 GA2의 해의 값과 CPLEX를 통해 계산된 최선해와 비교하여 제안된 유전 알고리즘들의 성능을 평가하였다. 조정 알고리즘을 적용한 GA2가 GA1에 비해 상대적으로 우수한 해를 제공함으로써 그 성능의 우수성을 검증하였다. 또한 제시된 유전 알고리즘들의 성능은 제품종류( $L$ ), 계획기간( $T$ ), 그리고 차량 종류( $N$ )가 증가함에 따라 더욱 우수해 짐을 보였다. 따라서 제안된 유전알고리즘은 실제 큰 규모의 문제들에 대해 더욱 우수한 성능을 보일 것으로 기대된다.

향후 연구과제로는 현재 유전 알고리즘의 해집단, 교차변이, 돌연변이 확률을 변화시킴으로써 최선해의 계산시간과 목적함수값의 변화를 파악하여 더 우수한 최선해를 짧은 계산시간에 제공하는 인자를 찾아 보다 효율적인 알고리즘을 개발하고자 한다. 또한, 문제 P에 대해 제품의 종류에 따른 서로 다른 제품크기(무게 및 부피 등)와 팔레트 단위로의 차량 적재 등을 고려하여 보다 현실적인 문제의 연구로 확장할 계획이다.

## 참고문헌

Hwang, H. C. and Jaruphongsa, W. (2006), Dynamic Lot-sizing Model with Demand Time Windows and Speculative Cost Structure, *Operations Research Letters*, **34**, 251-256.  
 Hwang, H. C. (2007), An Efficient Procedure for Dynamic Lot-Sizing

- Model with Demand Time Windows, *Journal of Global Optimization*, **37**, 11-26.
- Hwang, H. C. and Jaruphongsa, W. (2008), Dynamic Lot-sizing Model for Major and Minor Demands, *European Journal of Operational Research*, **184**, 711-724.
- Jaruphongsa, W., Cetinkaya, S., and Lee, C. H. (2004), Warehouse Space Capacity and Delivery Time Window Considerations in Dynamic Lot-sizing for a Simple Supply Chain, *International Journal of Production Economics*, **92**, 169-180.
- Kim, B. S. and Lee, W. S. (2010), A Genetic Algorithm for a Dynamic Lot-Sizing Problem with Delivery Time Windows, *Journal of the Korean Production and Operations Management Society*, **21**(4), 383-398.
- Kim, B. S. and Lee, W. S. (2011), An Adaptive Genetic Algorithm for a Dynamic Lot-sizing and Dispatching Problem with Multiple Vehicle Types and Delivery Time Windows, *Journal of the Korean Institute of Industrial Engineers*, **37**(4), 331-341.
- Kim, B. S., Lee, W. S., and Koh, S. G. (2012), A genetic algorithm for dynamic inbound ordering and outbound dispatching problem with delivery time windows, *Engineering Optimization*, **44**(7), 859-874.
- Lee, C. Y., Cetinkaya, S., and Wagelmans, A. P. M. (2001), A Dynamic Lot-sizing Model with Demand Time Windows, *Management Science*, **47**, 1384-1395.
- Lee, W. S. (2010), A Dynamic Ordering Shipping, and Outbound Dispatching Problem with Delivery Time Windows, *Journal of the Korean Management Engineering Society*, **15**(3), 15-23.
- Seo, W. C. and Lee, W. S. (2014), A Dynamic Lot-sizing and Outbound Dispatching Problem with Delivery Time Windows and Heterogeneous Container Types, *Journal of the Korean Institute of Industrial Engineers*, **40**(4), 435-441.