

선박을 이용한 화물 운송 중개 최적화 방안 연구[☆]

A Study on the Optimization for Brokering Between Cargos and Ships

서 상 구*
Sang-Koo Seo

요 약

본 논문은 선박을 이용한 화물의 운송에 있어서 선박과 화물의 중개 최적화에 대하여 기술한다. 중개를 위한 시간 제약, 중량 제한, 선호도 등 여러 기준들을 정의하여 정형화하고 이를 이용하여 최적화 문제 모형을 수립하였다는 것이 연구의 주요 핵심이고, 실험을 통하여 문제의 복잡도와 최적해 탐색 비용의 적정한 한계치를 파악하였을 뿐 아니라 그 과정으로서 성능평가 방법을 제시한 점도 중요한 연구 결과라고 하겠다. 제시된 최적 중개 방안은 화물과 선박의 시간적 제약 조건과 선박의 중량 제약 조건을 이용하여 화물의 선박에 대한 선호도와 선박의 화물에 대한 선호도를 유도하고, 최적화 문제의 목적 함수에서 이들 선호도를 이진 결정변수의 계수로 활용하여 제약 조건을 만족함과 동시에 전체적인 선호도 값의 합이 큰 선박과 화물의 쌍이 중개하도록 하였다. Davis-Putnam 기반의 최적화 프로그램을 이용한 실험에서 결정변수의 개수가 90 여개 이하의 문제 크기에 대하여 적절한 시간 내에 최적해를 구하는 것을 확인하였다.

Abstract

This paper presents a study on the optimization for brokering between cargos and ships for future e-logistics. The primary contribution of this research is that we establish an optimization model by formalizing the criteria for the brokering such as time constraints, weight constraints, and preference values between cargos and ships. Another important contribution is that we not only investigate the complexity and the tractability of the optimal brokering problem but present how to evaluate the performance of the optimization program through an experiment. We first derive the preference values between cargos and ships using the time and the weight constraints. These preference values between each pair of cargos and ships are assigned to corresponding binary decision variables as coefficients in the objective function. The optimization model selects pairs of cargos and ships in a way that the sum of the preference values is maximized while satisfying given constraints. Experiment shows that the Davis-Putnam based optimization program finds optimal solutions in reasonable time for the problems with less than 90 decision variables.

Keyword : e-logistics, brokering optimization, electronic commerce

1. 서 론

1990년대 이후 급속히 보급된 인터넷과 제반 기술의 발전으로 기업간 혹은 기업과 개인간의 거래에서 인터넷을 이용한 e-비즈니스가 다양한 분야에서 도입되고 있으며, 효과적인 전자상거래 프레임워크와 그 응용에 대한 연구가 활발하게 진행되고 있다.

UN/CEFACT와 OASIS가 공동으로 개발한 ebXML, Microsoft를 중심으로 하는 BizTalk.org의 BizTalk, RosettaNet의 RosettaNet 등이 XML 기반의 주요 전자상거래 프레임워크를 형성하고 있다. 이러한 전자상거래 프레임워크는 기업간 업무 프로세스를 통합, 연계하여 거래업무가 보다 신속하고 효과적으로 이루어지게 하고, 거래 기업의 다양한 요구사항에 유연하게 대응하여 상품을 보다 효율적으로 운송할 수 있는 새로운 물류 시스템을 필요로 한다[1,2,3,4].

기업간 상거래로 발생하는 물류 중 국제 운송은 그 규모나 복잡도 면에서 중요도가 크다고 하겠다.

* 정 회 원 : 광운대학교 경영정보학과 교수
skseo@kw.ac.kr(제 1저자)

☆ 본 논문은 2002년도 광운대학교 교내학술연구비 지원에 의하여 연구되었음.

화주의 화물 운송에 대한 요구사항은 요금 이외에도 화물의 종류, 중량, 선적 방법, 도착지, 화물 출고일, 희망 도착일 등의 다양한 조건으로 표현된다. 운송 수단으로서 예를 들어 선박의 경우에도 요금, 정기/부정기선, 최대 적재 중량, 컨테이너 최대 적재량, 출발 및 도착 예정 일시, 도착지 및 경유지 등 다양한 제약 조건들이 존재한다. 현재 거래선을 찾고 수출계약과 대금 결제 과정까지의 업무와 화물이 선적되어 해당 지역에 도착한 후 통관업무가 진행되는 과정은 EDI 혹은 XML/EDI를 사용하여 전자적으로 진행되고 있지만, 화주와 운송업체 간의 중개는 대부분 전문 중개사를 통한 전통적인 방식을 취하고 있어, 이에 대한 효과적인 중개 서비스의 정보화는 전자상거래의 매우 필요한 요소라고 하겠다.

본 논문에서는 화주와 운송업체 사이의 최적 중개 방안에 대한 연구를 다룬다. 다수의 화주와 운송업체를 상호 중개하는 문제를 0-1 MKP(Multiple Knapsack Problem[5])과 유사한 형태로 모형화 하였다. 화물과 운송업체 (선박, 항공 등)의 출항 및 도착 날짜와 화물의 컨테이너 수 및 중량을 이용하여 화주와 운송업체의 선호도를 정의하고, 거래 양자간의 선호도를 동시에 반영하기 위하여, 선호도의 곱의 합이 최대가 되도록 최적화 목표함수를 설정하였다. 제약 조건으로서 화물과 선박 각각에 출발 및 도착 시간, 적재 컨테이너 수 및 중량 등의 요구사항을 표현하였다. 최적해를 구하기 위한 방법으로서 0-1 최적화 문제 해결에 잘 알려진 Davis-Putnam[6] 기반의 OPBDP[7] 프로그램을 이용하였다. 문제의 복잡도와 최적화 프로그램의 성능을 평가하기 위하여 실험을 실시하였다.

관련 연구로서, 컨테이너에 화물을 최적 적재하기 위한 연구가 시도된 바 있으나[8], 이는 하나의 컨테이너를 대상으로 하였고, 화주와 운송업체 간의 다양한 요구조건은 고려되지 않았다. 전자상거래에 에이전트 기법을 이용한 많은 연구가 진행되어 왔으며, 주로 가격을 비교하거나 사용자의 구매성향을 분석하여 상품을 추천하고, 경매에서 적정 낙찰가를 제안하며, 마케팅 정책에 의해 사용자의 구매를 유도하는 등의 연구가 있었다[9,10,11,12]. 또, [13]에서는 중개

참여 에이전트의 이해관계 및 연결 조건을 CSP (Constraint Satisfaction Problem)으로 표현하고 구매자 또는 판매자 중심으로 단일 공급량과 다량의 공급량 등의 상거래 형태에 따라 중개 모델을 제시하였다. 유전자 알고리즘(Genetic Algorithm)을 이용하여 다수의 거래 요건을 충족하는 중개 서비스의 최적화를 위한 시도도 있었고[14], 4자 유통방식에서의 지능형 중개 에이전트에 대한 연구로서, 중개 문제를 SPP(Set Packing Problem)으로 모형화 하고 Greedy 휴리스틱에 의한 근접 해를 구하는 연구도 있었다[1]. 이상의 이전 연구들과 본 연구와는 문제 모형의 설정 과정에서 유사한 면이 있지만, 본 논문에서와 같이 화물과 운송업체를 대상으로 양측의 요구사항과 선호도를 고려하여 최적화 모델을 정형화한 이전 연구는 없었을 뿐만 아니라 Davis-Putnam의 최적화 프로그램을 중개 최적화 문제에 적용하여 문제의 복잡도를 분석하였다는 점에서 본 연구결과의 기여도가 있다고 하겠다.

본 논문의 구성은 다음과 같다. 2장에서 최적화 중개 모형 및 고려사항들을 기술하고, 3장에서 실험 환경 및 수행 결과를 설명하고, 끝으로 4장에서 결론을 맺는다.

2. 최적 중개 문제

2.1 가정

본 연구에서는 연구의 범위를 적절히 하기 위하여, 다음과 같은 가정을 하였다. 운송수단은 선박으로 가정하며, 운송업체는 선주로 표현하기로 하고, 선주는 하나의 정기운항의 선박을 갖는 것으로 한다. 화주의 화물의 목적지는 한 곳이며 여러 개의 컨테이너로 구성된 하나의 화물 운송을 가정한다. 또한 하나의 컨테이너에는 한 화주의 화물이 적재되며(FCL: Full Container Load), 화물의 종류는 건식, 습식, 냉장 등 다양한 요구 조건이 있으나 이들은 중개 대상의 선정에서 쉽게 여과될 수 있기 때문에, 본 논문에서

는 화물의 종류가 동일한 것으로 가정한다. 화물의 수량이나 중량이 매우 많은 불가피한 경우를 제외하고는 한 화주의 화물은 하나의 선박을 이용하여 운송하는 것이 경제적이고 또 일반적이므로, 화물의 컨테이너 수와 총 중량은 모든 선박에 대하여 선박의 최대 허용 컨테이너수와 총 허용 적재량보다 작다고 가정한다.

화물과 운송업체의 중개에는 여러 고려 사항들이 있겠지만, 본 연구에서는 다음과 같은 사항들만 요구 사항으로 고려하기로 한다. 먼저, 화물의 출고 가능 시점 및 도착시점 등 화주가 요구하는 운송 일정을 준수하도록, 화물의 출고 가능 시점 이후에 출항하며 화물의 도착 요구 시점 전에 목적지에 도착하는 운송업체가 중개 대상으로 고려되어야 한다. 또, 화물의 컨테이너 조건을 고려하여 한 화주의 화물은 가능한 하나의 운송 스케줄에 할당되는 운송업체가 고려되도록 하고, 끝으로, 운송 업체, 즉, 선박의 최대 적재 중량과 최대 컨테이너 적재 수량을 초과하지 않도록 대상 화물을 고려하도록 하였다.

이상의 가정과 요구사항들은 해운 물류 운송 중개의 현실적인 환경에 대체로 부합하고 있다. 하지만 본 연구의 모형을 육상 또는 항공 물류 운송에도 적용하기 위해서는 한 컨테이너에 여러 화주의 화물을 적재하는 소량 컨테이너 화물 운송(LCL: Less than Container Load)을 고려해야 할 뿐 아니라 화물들이 여러 중간 경유지에 운송되는 운항도 고려해야 할 것이다. 그 밖에, 운송비용 협상, 업체간의 정책적 제휴, 안전성 등도 운송 거래가 체결되는데 영향을 미치는 요소인데, 이러한 점들을 고려한 보다 일반화된 운송 중개 최적화에 대한 연구는 향후 연구로 미루기로 한다.

2.2 문제 모형

화주와 선주의 중개 문제는 MKP(Multiple Knapsack Problem)를 활용하여 적절히 표현할 수 있다. MKP는 이득 값과 무게가 각각 주어진 m 개의 물건을 각각 총 중량제한이 있는 n 개의 가방 (knapsack)

에 나눠 놓되, 가방에 넣은 물건들의 이득 값의 합이 최대가 되도록 하는 것이다. 이 때 각각의 물건은 하나의 단위로 고려되어야 할 경우에는 0-1 MKP이 된다. 즉, 하나의 물건의 일부만 가방에 포함시킬 수 없고 모두 한 가방에 넣든지 ('1') 또는 어느 가방에도 넣지 않든지 ('0') 둘 중의 하나의 경우만 가능하다. MKP (따라서 0-1 MKP도 포함) 최적해를 구하기 위한 프로그램 수행 시간이 입력 데이터 크기에 지수적으로 소요되는 복잡도가 매우 높은 문제로 알려져 있다[5].

화주-선주의 중개문제는 목표함수와 중량제한, 그리고 한 화주의 화물이 여러 선박 가운데 최대한 하나의 선박에 적재 될 수 있는 점 등 0-1 MKP와 유사한 점이 있지만, 화물의 선적 가능 시점과 도착 시점에 대한 제약 사항이 각각의 선박의 출발 가능 시점과 목표지 도착 예정 일시 조건에 고려되어야 하는 점이 추가된다. 즉, 만약 선적 가능 시점과 목표 도착 시점의 제약사항을 출항 또는 도착 후 창고 입고를 위한 대기 비용으로 각각 표현한다면, 특정 화물을 임의의 선박에 대하여 고려할 때의 대기비용은 선박마다 다르게 계산되어야 할 것이다. 화주의 입장에서 화물 출고에서부터 출항까지의 대기시간과 도착지에 도착해서 창고에 입고까지의 대기시간이 짧은 것이 물류 비용을 줄이기 때문에 가능하면 전체 대기시간이 짧은 운항의 선박을 선호할 것이다. 가능하면 한 화주의 화물이 하나의 선박에 적재되는 것을 선호함은 물론이다. 선박의 측면에서는 선박의 최대 적재 컨테이너 수만큼 화물을 적재하되 가능하면 컨테이너당 허용된 최대 중량보다 가능한 가벼운 화물을 선호하게 된다. 따라서, 최적화의 대상인 목표함수는 이 두 가지 측면이 반영되어야 한다.

본 연구에서는 화주 s_i 의 선박 c_j 에 대한 선호도를 $s_pref(s_i, c_j)$ 로 표현하고, 선박 c_j 의 화주 s_i 에 대한 선호도를 $c_pref(s_i, c_j)$ 로 표현하고 아래와 같이 정의한다. 각각의 화주 s_i , $1 \leq i \leq m$, 와 선박 c_j , $1 \leq j \leq n$,에 대한 입력 데이터 값은 표 1과 같이 정의하였다.

〈표 1〉 화주와 선박에 대한 입력 데이터

각각의 화주(Shipper) s_i 에 대해	각각의 선박 (Carrier) c_j 에 대해
$ready_to_ship(s_i)$: 화물의 선적 가능 일시	$depart_time(c_j)$: 선박의 출항 일시
$arrival_due(s_i)$: 화물의 도착 목표 일시	$arrival_time(c_j)$: 선박의 도착지 도착 예정일시
$weight(s_i)$: 화물의 총 중량	$max_weight(c_j)$: 선박의 최대 적재 중량
$count(s_i)$: 화물의 컨테이너 수	$max_count(c_j)$: 선박의 최대 적재 컨테이너 수

$$\begin{aligned}
 & s_pref(s_i, c_j) \\
 &= \left(1 - \frac{depart_time(c_j) - ready_to_ship(s_i)}{MAX_DEPART_WAIT}\right) \times w_1 \\
 &+ \left(1 - \frac{arrival_due(s_i) - arrival_time(c_j)}{MAX_ARRIVAL_WAIT}\right) \times w_2 \\
 & s_pref(s_i, c_j) \\
 &= nil, \text{ if } depart_time(c_j) < ready_to_ship(s_i) \text{ or } \\
 & \quad arrival_due(s_i) < arrival_time(c_j)
 \end{aligned} \tag{1}$$

여기서, MAX_DEPART_WAIT 와 $MAX_ARRIVAL_WAIT$ 는 화물의 출고 후 최대 허용 출항 대기시간과 도착 후 최대 허용 입고대기 시간을 의미하는 상수 값으로서, 3장 실험에서 설명될 출발/도착 시각의 겹침 값을 이용하여 결정된다. w_1 과 w_2 는 가중치로서 출항대기 및 입고대기의 중요도를 의미하며, 만약 선박 c_j 의 출항 및 도착 시점과 화물 s_i 의 출고 가능 및 도착 목표일이 정확히 서로 일치할 경우에는 $s_pref(s_i, c_j)$ 값은 $w_1 + w_2$ 만큼이 된다. 그리고, 어떤 화물의 출고시각 이전에 출항해야하는 선박과 화물의 도착 목표 시각 이후에 도착되는 선박은 그 화물을 운송하기에 적하지 않기 때문에 그와 같은 선박에 대한 화물의 선호도는 0으로 정의하였다. 0으로 계산된 화물 s_i 와 선박 c_j 쌍에 대해서는 3장의 최적화 실험에서 목적 함수를 생성할 때, 이진 결정 변수 x_{ij} 에 대한 항목을 제외시킴으로써 최적화 대상에서 고려되지 않도록 하였다.

선박 c_j 의 화주 s_i 에 대한 선호도 $c_pref(s_i, c_j)$ 는 화물의 컨테이너당 중량이 가벼울수록 화물에 대한 선

호도가 높아지기 때문에 아래와 같이 정의하였다.

$$c_pref(c_j, s_i) = \left(\frac{1}{\frac{weight(s_i)}{count(s_i)}}\right) \times w_3 \tag{2}$$

위 식 (1) 과 식(2)에서 정의한 화주의 선박에 대한 선호도와 선박의 화주에 대한 선호도를 고려하여 최적 중개를 위한 목표함수를 정의하였다. 본 논문에서는 양측의 선호도를 하나의 목표함수로 반영하기 위하여 각 선호도의 곱으로서 표현하기로 하였다. 예를 들어, 상호 선호도가 6과 8인 쌍보다는 7과 7인 쌍이 양측의 선호도를 절충한 형태로서 보다 중개의 효과와 제약 가능성이 높을 것으로 가정하였기 때문이다. 따라서, 이상의 선호도와 제약 조건 고려 사항을 기반으로 화물과 선박 사이의 최적 중개 문제는 다음과 같은 0-1 MKP의 응용 형태로 모형화할 수 있다. 결정 변수 x_{ij} 는 화주 s_i 의 화물을 선박 c_j 에 선적하기로 선정되면 1, 그렇지 않으면 0의 값을 갖는 이진 정수 값의 변수이다.

$$\begin{aligned}
 & \text{Maximize} \\
 & \sum_{i=1}^n \sum_{j=1}^m x_{ij} \times c_pref(s_i, c_j) \times s_pref(s_i, c_j) \tag{3}
 \end{aligned}$$

$$\begin{aligned}
 & \text{Subject to} \\
 & \sum_{j=1}^m x_{ij} \leq 1, 1 \leq i \leq m, x_{ij} \text{ is a binary integer.}
 \end{aligned} \tag{4}$$

$$\sum_{i=1}^m (x_{ij} \times \text{count}(s_i)) \leq \max_count(c_j), \text{ for } 1 \leq j \leq n \quad (5)$$

$$\sum_{i=1}^m (x_{ij} \times \text{weight}(s_i)) \leq \max_weight(c_j), \text{ for } 1 \leq j \leq n \quad (6)$$

위 식에서 m 과 n 은 각각 화물의 수와 선박의 수를 뜻한다. 따라서 목적 함수 식(1)에서 각각의 선박에 적재된 화물들 ($x_{ij} = 1$)에 대하여 상호 선호도의 곱의 합계가 계산되고, 목표함수 값을 최대화하기 위하여 가능하면 계수의 곱이 크면서 제약 조건을 위배하지 않는, 많은 수의 화물-선박 쌍을 선택하게 될 것이다. 첫 번째 제약조건인 식(4)는 하나의 화물이 최대 한 하나의 선박에만 적재될 수 있다는 조건을 나타내고 있다. 경우에 따라 어떠한 선박에도 적재되지 못하는 화물도 있을 수 있기 때문에 등호가 아닌 부등호로 표현하였다. 두 번째와 세 번째 제약 조건은 각각의 선박에 대하여 적재되는 화물의 컨테이너 수와 총 중량의 합이 선박의 최대 허용 컨테이너 수 및 최대 허용 적재량보다 같거나 작아야한다는 조건을 나타낸다. 이 문제 모형은 화물의 컨테이너수가 모두 하나이고 하나의 선박을 대상으로 하면 이진 단일 knapsack 문제로 볼 수 있다. 따라서, 이진 단일 knapsack 문제가 하나의 특수한 형태이기 때문에 다수의 화물과 선박간의 중개 문제는 NP-hard에 속하는 문제 유형으로 규명할 수 있다.

m 개의 화물과 n 개의 선박에 대하여 임의의 화물이 임의의 선박에 적재될 것인지를 결정해야 하므로, 총 $m \times n$ 개의 결정 변수가 필요하다. 각각의 결정변수는 목표함수에서 선호도의 곱을 계수로 하여 하나의 항목으로 존재하게 된다. 각각의 화물은 최대한 하나의 선박에만 적재 가능하다는 제약조건이 m 개 있고, 각각의 선박에 대해서는 적재 가능한 컨테이너 수에 대한 제약 조건과 총 적재 중량에 대한 제약 조건이 각각 있으므로, 총 $m + 2n$ 개의 제약 조건이 주어진다. 총 2^{mn} 만큼의 이진 결정변수들의 조합이 가능하므로

이들 목적 함수와 제약 조건에 대하여 총 2^{mn} 만큼의 계산이 요구되는 매우 높은 처리비용을 요구하는 문제임을 알 수 있다. 예를 들면, 선박 10개와 화물 20개에 대하여, 결정 변수의 수는 $10 \times 20 = 200$ 개 (즉, 목표함수에서 200개의 항목)이고, 화물에 대하여 20개의 제약 조건과, 선박에 대한 컨테이너 수에 대한 제약 조건 (10개)과 최대 중량에 대한 제약 조건 (10개)을 합하면, 총 30개의 제약 조건이 주어진다. 이들에 대하여 2^{200} 만큼, 즉, 대략 10^{60} 번의 결정변수의 조합이 고려되어야 한다.

3. 실험

3.1 최적화 프로그램 및 실험 환경

제안된 문제 모형에 대한 최적화 프로그램의 실행 성능을 분석하기 위하여 실험을 실시하였다. 최적화 알고리즘은 목표 문제의 특성과 문제 해결 정도에 따라 다양하게 분류될 수 있다[15]. 본 연구에서는 최적 중개 문제 해결을 위한 프로그램으로서 OPBDP[7]을 이용하였다. OPBDP는 SAT 문제(Satisfiability Problem) 해결 알고리즘으로 잘 알려진 Davis-Putnam [6]을 일반화한 0-1 최적화 문제 해결기로서 여러 응용 분야에서 그 적용이 연구되고 있다.

실험은 SunOS v5.6을 탑재한 Sun Enterprise 450 서버 (2 x 300MHz CPUs, 512MB Main Memory) 상에서 수행되었다. 실험에 필요한 데이터 생성과 OPBDP용 입력 수식 생성 프로그램은 GNU C/C++ Version 2.8.1로 구현되었다.

3.2 입력 데이터

입력 데이터 생성을 위한 매개변수들과 그 설명은 다음과 같다.

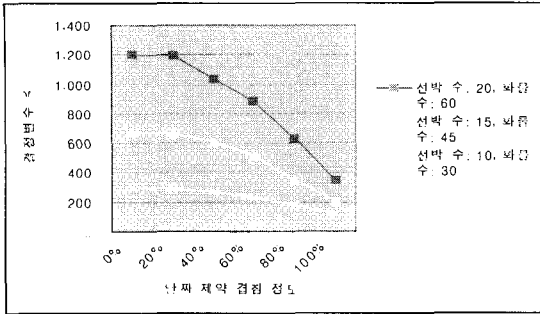
- 중개 대상 화물과 선박의 개수 : 화물 (즉, 화주)의 개수와 선박의 개수는 정수 값으로

지정하였다. 이들 입력 변수는 문제 인스턴스의 크기에 직접적으로 영향을 미치지만, 실제 복잡도는 제약조건의 수와 결정변수의 수에 의하여 결정된다.

- 화물과 선박의 출항 및 도착 일시 표현 : 화물의 출항 가능 시점 (즉, 출고 시점) 및 도착 기한 제약과 선박의 출발 시점 및 도착 예정 시점은 목적 함수에도 선박의 선호도 값으로 나타나고 제약 조건의 사전 제거에도 이용되기 때문에 효과적인 표현 기법이 필요하다. 본 연구에서는 출항 및 도착 시일 정보를 $1 \sim t$ 사이의 임의의 값으로 생성되도록 하되, 화물과 선박의 출발 및 도착 시일 범위를 또한 매개 변수로 지정하여 조정하도록 하였다. 예를 들어, $t = 10$ 일 경우에, 선박의 출발 가능 시간은 $1 \sim 10$ 사이의 임의의 값으로 주어지고, 화물의 출발 예정 시간도 $1 \sim 10$ 사이의 임의의 값을 이용하되, 시간 범위의 겹치는 정도의 값, (예를 들어 5라고 하면),을 더하여 $6 \sim 15$ 사이의 임의의 값이 생성되도록 하였다. 겹침의 정도 값을 고려한 이유는, 화물과 선박의 출발시점 범위가 상호 일치할 경우에는 균등 랜덤 함수에 의하여 대략 절반 정도의 화물과 선박 쌍이 출발 시간 제약 조건을 위배하게 될 것이다. 따라서, 결정 변수와 제약조건이 상당히 줄어들어 상황을 예상할 수 있을 것이다. 만약 겹침의 값을 10으로 하게 되면 모든 선박의 출항 시간이 모든 화물의 출발시간 이후로 지정되어 모든 화물-선박 쌍이 제약 조건을 위배하지 않기 때문에 결정변수와 제약조건의 수는 최대한의 크기가 될 것이다 (물론 이때 화물의 도착 목표 시점보다 선박의 도착예정시각이 모두 이르다고 할 경우이다). 화물의 출발 시각이 아닌 선박의 출항 시간 값에 겹침 값을 더한 이유는, 어떠한 화물의 출발 가능 시간보다도 먼저 출항하는 선박은 중개에서 고려할 이유가 없기 때문이다. 화

물과 선박의 도착 시각에 대해서도 마찬가지로 $1 \sim t$ 사이의 임의의 값을 고려하되, 이번에는 선박의 도착시간이 아닌 화물의 도착 목표 시각에 겹침 값을 더하도록 하였다. 왜냐하면, 앞서와 유사하게, 모든 화물보다도 늦게 도착하는 선박은 중개에서 고려할 필요가 없기 때문이다.

- 선호도 가중치 : 선박의 경우에 가능하면 출발 대기시간과 도착 후 창고 대기시간이 적은 운항을 선호하게 되고, 선박의 경우 컨테이너 당 주어진 최대 중량보다 가능한 가벼운 컨테이너를 선호하게 되는데, 이들 선호도에 대한 관점과 중요도가 다를 수 있기 때문에 출발 대기 시간에 대한 가중치, 도착 후 대기시간에 대한 가중치, 그리고 컨테이너 중량에 대한 가중치를 각각 입력 변수로 지정하였고, 실험에서는 기본 값을 10으로 하였다.
- 화물의 컨테이너 수와 컨테이너 중량 : 각 화주별 컨테이너 수는 $1 \sim 5$ 사이의 임의의 값을 생성하였고, 이때 각 컨테이너 별 중량은 $1 \sim 10$ 사이의 임의의 값을 생성하여 지정하였다.
- 선박의 최대 적재 컨테이너 수와 최대 적재 중량 : 선박은 여러 화주의 화물을 적재하는 것이 일반적이므로, 선박의 최대 적재 컨테이너 수와 최대 적재 중량은 선박의 컨테이너 수와 중량의 범위보다 훨씬 크게 지정하여야 한다. 또, 선박 별로 최대 적재 수와 중량 값이 모두 동일하지 않도록 적절하게 임의의 값을 활용하여야 한다. 본 연구에서는 선박 당 평균 화주의 수에 화물 당 평균 컨테이너 수를 곱한 값으로 하되, 여기에 각 선박 별로 $-5 \sim 5$ 사이의 랜덤 값을 더한 값을 선박의 최대 컨테이너 수로 정하였다. 이는 일정 수치의 편차를 적용한 것으로 합리적인 분포를 생성한다고 생각된다. 선박의 최대 적재 중량도 이와 유사한 개념으로 생성하였다.



(그림 1) 출발/도착 겹침 정도에 따른 결정변수 수의 변화

즉, 선박당 평균 화주 수에 화물당 평균 컨테이너 수와 컨테이너 평균 중량을 곱하여 그 결과 값에 -5 ~ 5 사이의 편차 값을 더하여 화물의 최대 적재 중량으로 하였다.

3.3 문제 인스턴스 생성

성능 평가 대상인 문제 인스턴스는 목표함수와 제약 조건 부분으로 구성된다. 결정변수는 x_{7-9} 의 형태로 생성하였는데 7은 화물의 색인, 9는 선박의 색인을 의미한다. 앞서 설명한 바와 같이, m 개의 화물과 n 개의 선박에 대하여, $m \times n$ 개의 결정변수 및 $m + 2n$ 개의 제약 조건이 가능한데, GNU C/C++의 랜덤 함수 (`srand()` 및 `rand()`) 값에 의하여 생성되는 문제 인스턴스에서 제약조건을 위배하는 결정변수들이 생겨나게 된다. 즉, 화물 s_i 의 출항 가능 일시가 선박 c_j 의 출항 예정 일시보다 늦을 경우에는 화물 s_i 는 선박 c_j 에 적재를 고려할 필요가 없으므로, 화물 s_i 의 제약 조건에 선박 c_j 항을 제거할 수 있고, 선박 c_j 의 컨테이너 수와 총 중량 제약 조건에도 화물 s_i 항을 제거할 수 있다. 다량의 컨테이너로 구성된 화물과 최대 컨테이너 수가 많지 않은 소형 선박 사이에도 이와 유사한 제약 조건 위배가 발생할 수 있다. 임의의 값 생성 결과에 따라 어떤 화물 전체가 모든 선박에 대하여 제약조건을 위배할

수도 있는데, 이런 경우에는 앞서 정의한 식 (4)의 제약식 하나가 줄게 된다. 또한, 화물의 경우에 하나의 선박만 고려하게 되는 경우 (예를 들면, 화물 7번의 경우 선박 3번에만 적재 가능할 때 $x_{7-3} \leq 0$) 에도, 결정변수는 0 또는 1의 이진 값이므로 Tautology로써 이 화물 (즉, 화물 7번)의 제약 조건식을 안전하게 제거할 수 있다. 이와 같이 문제 생성 과정에서 제약조건을 위배하는 쌍들을 제거함으로써, 제약 조건수와 결정변수 수를 많은 경우에 상당히 줄일 수 있었고 실행 성능에서 그 효과는 매우 높았다.

3.4 실험 결과

선박과 화물간의 증개 문제는 결정 변수의 개수나 제약조건의 수가 다량 발생하는 매우 복잡도가 높은 문제임을 앞서 언급하였는데, 실제 실험에서도 입력 크기가 조금만 증가하여도 실행 시간이 지수적으로 증가함을 확인할 수 있었다. 문제 인스턴스의 크기에 따른 최적화 프로그램의 성능을 살펴보기에 앞서, 화물과 선박간의 출발/도착 시간 범위의 겹침 정도가 문제 인스턴스의 복잡도에 미치는 영향을 먼저 분석해보았다.

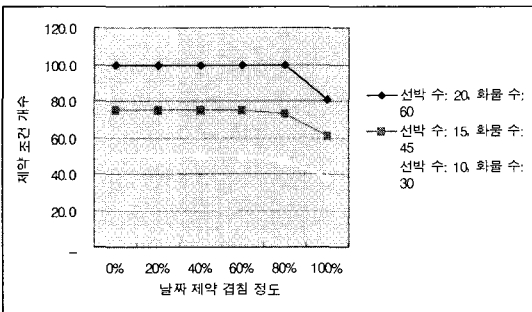
3.4.1 출발/도착 범위 값의 변화에 따른 문제 인스턴스의 크기

실험에서 선박 수와 화물 수를 각각 $\langle 20, 60 \rangle$, $\langle 15, 45 \rangle$, $\langle 10, 30 \rangle$ 세 유형으로 하고, 각 경우마다 겹침 정도 값을 0% ~ 100% 범위에서 20% 간격으로 증가시키며, 각각의 경우마다 문제 인스턴스를 3회씩 생성하여 구해진 결정변수와 제약 조건 수의 평균값을 취하는 실험을 하였다. 실험에 사용된 랜덤 함수 `rand()`를 위한 seed로서 호출시점의 `time()` 함수 값을 제공함으로써 매번 임의의 값이 생성되도록 하였다. 아래 그림은 먼저 결정 변수 수의 감소 정도를 보여주고 있다.

겹침 정도가 0%일 때에는 제거 효과가 전혀 있을

수 없기 때문에, 주어진 선박수와 화물 수에 대한 최대 수의 결정변수를 나타내고 있다 (즉, 화물 수 x 선박 수). 그래프에서 나타나듯이 겹침 정도가 20% 이상의 값에서 결정변수 수가 줄어들기 시작했으며 선박수와 화물 수가 많은 <20, 60>의 경우에 감소 폭이 제일 크게 나타났다. 그 이유는 각각의 선박과 화물의 쌍의 수가 많고 겹침의 정도가 클수록 제약 사항을 위배하는 쌍의 수가 지수에 가까운 비율로 늘어나기 때문인 것으로 판단된다.

아래 그림 2는 겹침 정도의 변화에 대한 제약조건 수의 감소효과를 나타내고 있다. 여기에서도 겹침 정도가 0%일 때에는 문제 인스턴스의 최대 제약 조건수를 나타내고 있다 (즉, 화물수 + 선박수 x 2). 그림에서 보듯이 결정변수 개수의 감소 형태와는 달리 문제 인스턴스의 제약 조건 수의 감소는 겹침의 정도가 80% 정도 이르러야 감소가 발생하였다. 특히, <20, 60>의 경우에는 90% 근처부터 감소가 시작되었다. 결정변수와 달리 제약 조건식 하나가 감소하려면, 하나의 선박에 대하여, 컨테이너 수와 최대 중량 값의 범위를 고려할 때, 날짜 범위를 위배하는 화물의 개수가 55개 정도 되어야 하기 때문에 선박의 제약 조건식이 제거되기란 확률적으로 매우 낮기 때문일 것이다. 또한, 화물의 경우에도, 문제 인스턴스에서 하나의 화물의 제약 조건 식이 제거하려면 19개의 선박과 날짜 범위가 적합하지 않아야 하기 때문에 겹침 정도가 매우 높을 경우에만 가능한 것으로 이해된다.



<그림 2> 출발/도착 겹침 정도에 따른 문제 인스턴스의 제약조건 수의 변화

3.4.2 문제 인스턴스 크기에 대한 OPBDP의 성능

본 실험에서는 OPBDP 최적화 프로그램의 문제 인스턴스 해결 시간을 측정 및 분석하였다. 실질적인 문제의 복잡도는 결정변수의 수와 제약 조건의 개수이므로, 출발/도착 시간의 겹침 값을 0으로 함으로써 결정변수와 제약 조건이 제거되지 않도록 하여 화물 수와 선박 수의 값으로 이용하여 문제 인스턴스를 생성하였다. 아래 표 2는 선박 수를 5로 고정하고, 화물 수를 10 ~ 20으로 변경하면서 OPBDP의 실행 결과를 나타내고 있다. 결정변수 수는 50 ~ 100으로, 제약 조건 수는 20 ~ 30으로 각각 생성되었다. 출발/도착 시간의 겹침 값을 90% 이상으로 하면 더 많은 화물 수와 선박 수에 대한 실행이 가능하였지만, 제거되는 변수와 제약 조건수가 랜덤 함수 값에 따라 불규칙적으로 발생하기 때문에 프로그램 실행 시간을 일관성 있게 관찰하기가 어려웠다. 따라서, 선박 수와 화물 수 값은 적게 하더라도, 변수 수와 제약 조건 수를 일정 크기로 증가시키며 실험을 실시한 것이다. 또, 실행 시간이 오래 걸렸기 때문에 동일한 변수 수와 제약 조건에 대하여 여러 번 실행하지 못하고 1회 실행한 결과만 측정하였다.

표 2에서 보듯이 선박 수가 5일 때, 화물의 수를 1씩 증가 시킬 때마다 결정 변수수가 5씩 증가하고 제약 조건수는 1씩 증가하는데, 최적화 알고리즘의 탐

<표 2> OPBDP 최적화 프로그램 실행 결과

결정변수수	제약조건 수	탐색 노드 수	최대 탐색 깊이	실행 시간 (sec.)
50	20	25430	35	0.3
55	21	36198	34	0.5
60	22	621458	53	8.9
65	23	497241	53	7.9
70	24	1270812	64	21.4
75	25	2331279	62	35.7
80	26	28374930	67	473.4
85	27	136751748	75	1652.4
90	28	296336528	75	5504.7
95	29	408640060	83	7753.0
100	30	481146721	94	9228.9

색노드 수는 기하 급수적으로 증가하고, 실행 시간 또한 매우 가파르게 증가하는 것을 관찰할 수 있었다. 화물 수와 선박 수를 다양하게 설정하여 실험을 수행해 보았으나, 표 2에서 보듯이 결정 변수 수가 대략 90 이상부터는 실행시간이 2시간 가까이 소요되었기 때문에 더 많은 변수 수 또는 제약 조건수에 대한 실행 시간 측정은 현실적으로 쉽지 않기 때문에, 변수 수가 100이고 제약 조건 수가 30일 때의 실행 값으로 제한하였다. 실제로 더 큰 개수 값에 대하여 프로그램을 실행하였으나 10시간을 초과해도 프로그램이 종료하지 않는 사례도 있었다. 비록 동일 출발지와 도착지를 가정하였고, 시간 제약 조건을 위배하는 선박-화물 쌍이 하나도 없는 경우이기는 하지만, 단순한 최적화 알고리즘의 실행만으로는 적절한 시간 내에 최적 중개 결과를 생성하기가 만만치 않다는 것을 본 실험을 통하여 알 수 있다. 선박의 수와 화주의 수가 각각 5개와 18~20개 정도는 현실적으로 소규모의 중개 대상에 해당된다. 중/대규모의 중개, 즉, 수십 개의 선박과 수십 또는 수백의 화주들 간의 중개를 위해서는 휴리스틱 알고리즘을 고안하여 근사해를 구하는 방안이 필수적일 것이다. 따라서 본 실험의 결과는 3.1절의 가정 하에 현실적으로 10여개 이하의 선박과 20여개 이하의 화주들로 구성된 소규모의 운송 중개에 대한 최적해를 구하는데 시도될 수 있을 것이다.

4. 결론

전자상거래의 발전에 기인한 상거래 시스템의 변화는 필수적으로 물적 유통구조의 변화를 요구한다. 본 논문에서는 화물과 운송업체 간의 물류 중개 문제를 연구하였다. 0-1 MKP(Multiple Knapsack Problem)과 유사한 형태로 최적화 문제를 모형화 하였고, Davis-Putnam 기반의 최적화 프로그램인 OPBDP를 이용하여 성능평가 실험을 실시하여, 최적 문제 해결 범위와 가능성을 살펴보았다. 화주와 운송업체의 선호도를 출발 및 도착 대기시간과 화물 중량 값을 이용하여 정의하였고 이를 이용하여 목표함수와 제약 조

건들을 생성하였다. 실험 결과, 화주와 운송업체가 대략 각각 10여개 이상 정도의 문제 크기부터는 최적해 탐색 시간이 매우 과다하게 소요되는, 복잡도가 매우 높은 문제 유형임을 보였고, 이에 대한 현실적인 방안으로서 효과적인 휴리스틱 알고리즘이나 근사해 알고리즘의 필요성을 언급하였다. 본 논문에서는 선박과 컨테이너 화물을 중심으로 표현하였는데 다른 형태의 물류 환경에도 본 연구의 프레임워크 확장될 수 있을 것으로 생각한다. 즉, 본 논문에서는 선박과 화물의 많은 요구 사항들 가운데 출발/도착 시간, 컨테이너 수, 중량 등의 속성들만 고려하였는데, 향후 보다 다양한 운송 수단(선박, 항공, 철도, ...)과 운항 유형(정기/부정기, 직송/경유, ...), 화물특성, 가격 협상 등을 고려한 최적 중개 문제 모형 및 중개 휴리스틱 알고리즘의 연구가 흥미로울 것으로 사료된다.

참고 문헌

- [1] Lau, H.C. and Goh Y.G, "An Intelligent Brokering System to Support Multi-Agent Web-based 4th-Party Logistics," Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02), 2002.
- [2] 황재각, 오세원, 이용준, "EbXML 표준에 기반한 다자간 물류 통합정보시스템," 정보과학회, 2002년 춘계학술대회, Vol.29, No.02, 2002.
- [3] 오동근, 홍정선, 전종미, 김광훈, "ebXML 기반의 전자물류 비즈니스 프로세스 관리 시스템 엔진," 정보과학회, 2003년 춘계학술대회, Vol.30, No.01, 2003.
- [4] 김영일, 이용준, 황재각, "ebXML을 활용한 e-Logistics 통합플랫폼에서의 등록지장소의 설계," 정보과학회, 2003년 춘계학술대회, Vol.30, No.01, 2003.
- [5] Garey, M. R. and Johnson, D. S.. Computers and Intractability: A Guide to the Theory of

- NP Completeness. Freeman, 1979.
- [6] Davis, M. and Putnam, H. "A Computing Procedure for Quantification Theory," Journal of the ACM, Vol. 7, pp.201 -205, 1960.
- [7] Barth, P., "A Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization," MPI-I-95-2-003, MPI Informatic, Germany, 1995 (<http://www.mpi-sb.mpg.de /units/ag2/software/opbdp/>).
- [8] 권아름, 박운재, 김정중, "컨테이너 최적 적재를 위한 알고리즘," 한국정보처리학회, 춘계학술발표대회, 제2권, 제1호, pp188-191, 1995.
- [9] 최중민, "에이전트기술 및 전자상거래," 공학기술학회지, Vol.07, No.02, pp.37~40, 2002.
- [10] Joshi, A. and Singh, M.P., "Multiagent systems on the net," Communications of the ACM, Vol.42, No.3, pp.81-90, March, 1999
- [11] Ma, M., "Agents in E-Commerce," Communications of the ACM, Vol.42, No.3, pp.79-80, March, 1999
- [12] 황병연, 박성철, "전자상거래를 위한 정책지향 매칭 에이전트 시스템의 설계 및 구현," 한국정보처리학회 논문지, Vol.8-D, No.05, pp.623-630
- [13] 정종진, 김지연, 조근식, "에이전트 기반 전자상거래에서 제약만족 기법을 이용한 최적의 중개 모델," 정보처리학회 논문지 제7권 제5호, 2000.
- [14] Easwaran, A.M. and Pitt, J., "Simultaneous Optimization of Multiple Criteria for Efficient Agent Service Brokering," Proceedings of AGENTS'01, May 2001
- [15] OTC, "NEOS Guide Optimization Tree," Optimization Technology Center, <http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/index.html>

● 저 자 소 개 ●



서 상 구

1984년 : 서울대학교 전자계산기공학과 졸업(학사)
 1986년 : 한국과학기술원 전산학과 졸업(석사)
 1995년 : 한국과학기술원 전산학과 졸업(박사)
 1986년~1989년 : 현대전자(주) 소프트웨어 연구소
 1995년~1998년 : 현대전자(주) 정보시스템사업본부
 1999~현재 : 광운대학교 경영정보학과 교수
 관심분야 : 데이터베이스, 전자상거래, etc.
 E-mail : skseo@kwangwoon.ac.kr