

# 동적 블록수송환경을 위한 네트워크 흐름모형 기반의 트랜스포터 일정계획

이운식<sup>1\*</sup> · 임원일<sup>2</sup> · 구평희<sup>1</sup>

<sup>1</sup>부경대학교 시스템경영공학과 / <sup>2</sup>DK유엔씨(주) CS2팀

## Transporter Scheduling Based on a Network Flow Model for Dynamic Block Transportation Environment

Woon-Seek Lee · Won-Il Lim · Pyung-Hoi Koo

<sup>1</sup>Systems Management and Engineering, Pukyong National University

<sup>2</sup>CS2 Team, DK UNC Co., Ltd.

This paper considers a transporter scheduling problem under dynamic block transportation environment in shipbuilding. In dynamic situations, there exist the addition, cancellation or change of block transportation requirements, sudden breakdowns and maintenance of transporters. The transportation of the blocks in the shipyard has some distinct characteristics. Some blocks are available to be picked up at a specific time during the planning horizon while some other blocks need to be delivered before a specific time. These requirements cause two penalty times: 1) delay times incurred when a block is picked up after a required start time, and 2) tardy times incurred when a block shipment is completed after the required delivery time. The blocks are located at different areas in the shipyard and transported by transporters. The objective of this paper is to propose a heuristic algorithm based on a network flow model which minimize the weighted sum of empty transporter travel times, delay times, and tardy times. Also, a rolling-horizon scheduling method is proposed for dynamic block transportation environment. The performance of the proposed heuristic algorithms are evaluated through a simulation experiment.

**Keyword:** transporter scheduling, block allocation, dynamic block transportation environment, network flow model, shipbuilding

### 1. 서론

한국조선공업협회 발표자료에 따르면 국내 조선산업의 세계 시장 점유율은 2006년 실적을 기준으로, 수주량은 38.3%, 건조량은 34.6%, 수주잔량은 36.3%를 차지하고 있다. 뿐만 아니라 2006년 조선산업 수출액은 220억불을 넘어 섰으며, 국내 총무역수지 흑자 108억불의 93.3%를 차지할 정도로 국가 기간 중

추 산업임에 틀림없다. 또한, 국내 조선산업의 대·중형 9개 조선사 중 8개 조선사가 동남권에 위치하고 있고 국내 조선산업 대형 5개 조선사가 국내 선박 건조량의 95%를 차지하고 있다. 그러나 중국은 조선산업에의 집중적 투자와 가격 경쟁력을 통한 추격으로 한국 조선시장을 위협하고 있으며 일본대비 조선산업의 낮은 생산성은 일본대비 10% 가격경쟁력 우위를 불안하게 하고 있는 실정이다. 2010년 150억불의 신조선 수출로 세

이 논문은 2007년도 정부재원으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2007-D00559).

\*연락처 : 이운식 교수, 608-739 부산광역시 남구 용당동 산 100 부경대학교 시스템경영공학과, Fax : 051-629-6478,

E-mail : iewsee@pknu.ac.kr

투고일(2008년 09월 24일), 심사일(1차 : 2008년 09월 26일, 2차 : 2008년 11월 19일, 3차 : 2008년 12월 16일), 게재확정일(2008년 12월 22일).

게시장 점유율 40% 달성과 함께 세계 조선 선도국 위상 유지를 위해서는 IT 기반형 선박설계·생산 지원시스템의 확보를 통한 지식기반 생산방식으로 전환을 도모함으로써 생산성 향상을 통한 산업고도화가 필수라 하겠다.

조선소의 선박 건조 프로세스를 살펴보면, 설계과정에서 선체를 분할하여 제공품 형태의 다양한 형상과 무게를 갖는 블록(Block)을 정의하고 블록을 구성하는 중·소 부품이 정의되면 선각작업(Hull Construction)에서는 이의 역순으로 철판이나 형강으로부터 부분품을 절단하는 철재절단, 부분품을 제공단계의 좀더 큰 구조물로 만드는 블록조립, Dock내에서 블록들을 준비하여 선박의 형상을 갖춰가는 탑재(Erection)등으로 구성된다. 따라서 많은 수의 블록들은 조립, 선행의장(Pre-Outfitting), 도장(Painting), 의장(Outfitting) 작업 등을 수행하는 여러 플랜트를 거쳐 Dock에서 탑재되기 전까지 조선소내의 여러 곳에 분산되어 관리된다. 조선소에서는 이러한 다양한 형상과 무게를 갖는 블록들을 각 플랜트가 원하는 시간에 pick-up하고 delivery 하기 위해 트랜스포터(Transporter)라는 고가의 블록운반용 특수차량을 이용한다. 트랜스포터는 100, 200, 300, 350, 500 톤급 등으로 종류가 다양하며 종류별로 여러 대의 트랜스포터가 운용된다. 이러한 고가 트랜스포터의 보유와 운영은 많은 로지스틱스(Logistics) 비용을 유발한다. 또한, 트랜스포터의 결합(적재능력을 초과하는 블록(500톤을 초과하는 블록)은 두 대 이상의 트랜스포터를 결합하여 운반함), 트랜스포터의 돌발적인 고장과 정기적인 유지보수 등은 트랜스포터의 일정계획에 제약으로 작용하며 트랜스포터의 비효율적인 운영은 선후 생산공정의 생산지연을 초래함은 물론 선박건조기간의 지연을 유발하게 된다. 더구나, 세계 조선시장의 활황으로 인한 수주물량의 확대는 제한된 조선소 내에서 블록들의 적치와 운반이 심대하게 증가하여 트랜스포터의 효율적인 운영이 한층 요구되고 있다.

본 논문에서는 조선 산업의 생산필드에서 많은 물류비용과 생산지연비용을 유발시키는 트랜스포터의 일정계획 수립방안을 제안하고자 한다. 대상 환경으로는 계획기간동안 각 플랜트의 생산일정에 따른 블록의 수송요구의 추가, 삭제 및 변경, 트랜스포터의 고장 및 유지보수 등이 수시로 발생하는 동적인(dynamic) 수송환경을 고려한다. 이를 위해 제 2장에서는 트랜스포터의 일정계획 환경을 설명하고 관련 연구동향을 소개한다. 제 3장에서는 트랜스포터 일정계획문제에 대한 최적화 모형을 제시한다. 제 4장에서는 트랜스포터의 블록할당과 운반순서를 효율적으로 결정하는 네트워크 흐름모형 기반의 휴리스틱 알고리즘을, 제 5장에서는 연동적 일정계획 알고리즘을 각각 제안하고 알고리즘의 성능평가도 비교한다. 마지막으로, 제 6장에서는 논문의 내용을 요약한다.

## 2. 트랜스포터 일정계획 및 연구동향

조선소 내에서의 블록운반은 몇 가지 특성을 가진다. 운반이

요구되는 블록들 중 일부는 계획기간동안 규정된 출발요구시각 전에 pick-up되기를 요구하는 반면 나머지 블록들은 계획기간동안 규정된 도착요구시각 전에 delivery되기를 요구한다. 각 블록에는 출발지, 도착지, pick-up 시각, delivery 시각, 적재시간 등과 같은 정보가 주어진다. 이러한 블록의 이동과 관련된 요구들은 요구되는 pick-up 시각 이후에 블록이 운반될 때 발생하는 지연시간(이하 *delay* 시간이라 한다)과 요구되는 delivery 시각 이후에 블록운반이 완료될 때 발생하는 지연시간(이하 *tardy* 시간이라 한다)의 2가지 벌과시간(penalty time)을 유발한다. 더구나, 블록들은 조선소내의 여러 지역에 분산 배치되어 있는 관계로 어떤 트랜스포터로 어떤 블록을 운반하느냐에 따라 각 트랜스포터의 공차운행시간에 영향을 미치게 된다. 하나의 블록이 하나의 트랜스포터에 할당될 때, 다음과 같은 순서에 의해 로지스틱스 관련 시간들이 발생하게 된다.

- (1) 사용가능한 트랜스포터는 현재의 위치에서 운반요구 블록의 현 위치까지 이동하는 동안 *공차운행시간*이 발생된다.
- (2) 가능하면 블록의 pick-up시각에 맞추어 블록의 출발지에도착한다. 이때, pick-up시각 이후에 트랜스포터가 도착하면 블록의 *delay* 시간이 발생된다.
- (3) 출발지에 도착 후, 블록을 적재하기 위해서는 블록의 형상과 무게에 따라 일정한 *적재시간*이 소요된다.
- (4) 트랜스포터는 블록을 적재한 후 도착지까지 이동한다. 이때 *적재운행시간*이 발생되며 적재시간 및 하역시간은 모두 여기에 포함된다.
- (5) 트랜스포터가 delivery 시각 이후에 도착하게 되면 블록의 *tardy* 시간이 발생된다.
- (6) 운반 후, 트랜스포터는 다음 블록을 운반하기 위해 현 위치에서 대기한다.

조선소내의 로지스틱스 비용을 절감하기 위해서는 최적의 트랜스포터 운영시스템을 구축하여야 하며 이를 위해 계획기간동안의 블록들의 여러 요구조건을 만족하면서 총 로지스틱스 시간을 최소화하도록 각 트랜스포터별 효율적인 경로 및 일정 계획을 수립해야 한다. 본 논문의 목적은 동적인 블록수송환경에서 트랜스포터의 공차운행시간과 블록의 delay 시간 및 tardy 시간을 최소화하기 위한 블록의 할당과 운반순서를 동시에 결정할 수 있는 일정계획 알고리즘을 제안하는데 있다.

만일 몇몇 블록들이 하나의 특정한 트랜스포터에 의해 운반되기로 사전에 결정되고 결합에 의한 운반형태가 없다면, 이러한 트랜스포터 일정계획문제는 운반하고자 하는 블록의 수를 노드수로 하는 traveling-salesman problem(TSP)와 유사하다. 또한, 운반하고자 하는 블록을 job으로 하는 time-dependent setup time을 갖는 단일기계 일정계획문제 그리고 블록을 하나의 위치로 보는 차량경로문제(vehicle routing problem) 등과 유사하다 할 수 있다. 이는 블록의 운반순서에 의해 트랜스포터의 공차운행시간이 발생하며 공차운행시간을 최소로 하는 운반계획을 결정하는 것이 요지임을 알 수 있다. 그러나 대부분의 TSP와 달리 본 과제는 블록들이 요구하는 pick-up 시각과 deliv-

ery 시각에 대한 제약이 존재하므로 문제를 더욱 어렵게 만들 수 있다(time constraints를 갖는 TSP). 더구나, 본 과제에서 트랜스포터는 종류별로 여러 대를 보유할 수 있으므로 트랜스포터 일정계획 문제는 multiple TSP with time constraints(m-TSPTC)로 해석될 수 있다. 잘 알려진 TSP, multiple TSP, 일반적인 차량 경로문제들이 Laporte and Osman(1995), Crainic and Laporte(1998) 그리고 Chao(2002)에 의해 조사되었다. m-TSPTC는 NP-hard 문제라고 알려져 있다. Savelsbergh(1985)는 차량대수가 미리 주어질 때, 운행비용을 최소화하는 차량 경로문제조차 NP-hard임을 증명하였다. TSP에 대한 최적해를 제공하는 알고리즘은 Baker(1983)에 의해 Branch-and-Bound 기법을 기초로 개발되었다. Salvelsbergh(1985)는 TSPTC에 대한 interchange 휴리스틱 알고리즘을 제안하였다. Desrosiers *et al.*(1988)은 최소 차량대수를 구하기 위해 확장된 라그랑주 기법을 기초로 한 알고리즘을 제시했다. Dumas *et al.*(1995)은 최적해를 찾기 위한 동적계획법 알고리즘을 제안한 반면 Langevin *et al.*(1993)은 branch-and-cut 알고리즘에 의해 풀 수 있는 two-commodity flow problem를 개발하였다. Bianco *et al.*(1998)은 선후제약식을 갖는 TSPTC를 풀기 위한 2단계 동적계획법에 기초한 알고리즘을 제안하였다. Gendreau *et al.*(1998)은 insertion and post-optimization 기법을 결합한 휴리스틱 알고리즘을 개발하였다. Pesant *et al.*(1998)은 logic constraint programming에 기초한 기존의 논문과 차별된 알고리즘을 제시하였다. Calvo(2000)는 greedy insertion 기법에 기초한 휴리스틱 알고리즘을 제안하였다. Bard *et al.*(2002)은 시간제약을 갖는 차량경로문제에서의 최소 차량대수 결정문제를 다루었으며 branch-and-cut에 기초한 알고리즘을 개발하였다.

수송 시스템에서 차량대수 결정문제를 다루는 또 다른 분야로는 자동화된 제조 시스템에서의 AGV 시스템들과 관련된다. Maxwell and Muckstadt(1982)는 계획기간동안 수송요구량이 주어질 때, AGV의 최소 대수를 결정하는 수리모형을 제안하였다. Ko *et al.*(2000)은 컨테이너 수송 시스템에서 차량대수 및 차량경로 결정을 위한 insertion 알고리즘을 제시하였다. Koo *et al.*(2004)은 컨테이너 수송시스템에서 수송요구량을 수행하기 위한 차량대수 및 차량경로를 동시에 결정하는 2단계 휴리스틱 알고리즘을 개발하였다.

조선생산공정에서 블록의 운반과 관련된 최근 연구들을 정리하면 다음과 같다. Joo *et al.*(2006)은 계획기간동안 블록의 모든 요구량이 미리 알려져 있는 정적인 수송환경에서의 트랜스포터의 일정계획을 효율적으로 수립할 수 있는 휴리스틱 알고리즘을 제안하였다. Lim *et al.*(2008)은 계획기간동안 블록의 수송요구가 추가 및 삭제되거나 트랜스포터의 돌발고장 및 유지보수 등이 수시로 발생하는 동적인 수송환경에서의 트랜스포터의 일정계획을 효율적으로 수립할 수 있는 IT 기반의 의사결정시스템의 개발 방안을 제시하였다. 또한, Lee *et al.*(2008)은 동적인 수송환경에서의 트랜스포터의 일정계획을 효율적으로 수립할 수 있는 4가지 휴리스틱 알고리즘을 제안하고 성능을 비교·분석하였다. 그러나 조선소 내에서의 트랜스포터 일정

계획에 대한 연구는 고가의 장비구입으로 인한 투자비용과 고가의 운전비용으로 인해 로지스틱스 비용의 절감을 통한 생산성 향상 측면에서의 중요성에도 불구하고 관련 연구는 미미한 실정이다.

### 3. 최적화 모형

본 논문에서는 Lee *et al.*(2008)이 제시한 수리모형을 사용하여 최적화 모형을 작성하였으며 사용되는 기호는 다음과 같다.

- $x_i$  : 블록  $i$ 의 운송시작 시간
- $r_i$  : 블록  $i$ 의 pick-up 가능 시간
- $c_i$  : 블록  $i$ 의 요구되는 도착 시간 (delivery time)
- $t_{ik}$  : 트랜스포터  $k$ 가 블록  $i$ 를 운반하는데 소요되는 시간 (적재, 하역시간 포함)
- $e_{ijk}$  : 트랜스포터  $k$ 가 블록  $i$ 를 운반한 후 블록  $j$ 를 pick-up 하기 위하여 공차로 이동하는데 소요되는 시간
- $d_i^+$  : 블록  $i$ 가 delivery time 이전에 운반되어지는 경우 그 차이 시간
- $d_i^-$  : 블록  $i$ 가 delivery time 이후에 운반되어지는 경우 그 차이 시간
- $y_{ik}$  = 트랜스포터  $k$ 에 의해 블록  $i$ 가 운반되어지면 1, 아니면 0
- $y_{ijk}$  = 트랜스포터  $k$ 에 의해 블록  $i$  후 블록  $j$ 가 운반되어지면 1, 아니면 0
- $N_T$  : 총 트랜스포터의 수
- $N_B$  : 총 블록의 수
- $O$  : 각 트랜스포터별 최초운반을 위한 가상의 블록
- $E$  : 각 트랜스포터별 최후운반을 위한 가상의 블록
- $M$  : 양의 큰 수( $\infty$ )
- $\alpha, \beta, \gamma$  : 가중치

개선된 최적화 모형은 다음과 같다.

$$\text{Min } z = \alpha \sum_{k=1}^{N_T} \sum_{i=1}^{N_B} \sum_{j=1}^{N_B} e_{ijk} y_{ijk} + \beta \sum_{i=1}^{N_B} (x_i - r_i) + \gamma \sum_{i=1}^{N_B} d_i^-$$

$$\text{s.t. } x_i \geq r_i, \quad i = 1, \dots, N_B \quad (1)$$

$$x_i + \sum_{k=1}^{N_T} t_{ik} y_{ik} + d_i^+ - d_i^- = c_i, \quad i = 1, \dots, N_B \quad (2)$$

$$x_i + \sum_{k=1}^{N_T} t_{ik} y_{ik} + \sum_{k=1}^{N_T} e_{ijk} y_{ijk} \leq x_j + M \cdot (1 - \sum_{k=1}^{N_T} y_{ijk}),$$

$$i = 1, \dots, N_B \quad j = 1, \dots, N_B \quad j \neq i \quad (3)$$

$$\sum_{k=1}^{N_T} y_{ik} = 1, \quad i = 1, \dots, N_B \quad (4)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N_B} y_{jik} + y_{Oik} = y_{ik}, \quad i = 1, \dots, N_B \quad k = 1, \dots, N_T \quad (5)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N_B} y_{ijk} + y_{iEk} = y_{ik}, \quad i = 1, \dots, N_B \quad k = 1, \dots, N_T \quad (6)$$

$$\sum_{j=1}^{N_B} y_{Ojk} + y_{OEK} = 1, \quad k = 1, \dots, N_T \quad (7)$$

$$\sum_{j=1}^{N_B} y_{jEk} + y_{OEK} = 1, \quad k = 1, \dots, N_T \quad (8)$$

$$x_i, d_i^+, d_i^- \geq 0, \quad i = 1, \dots, N_B \quad (9)$$

$$y_{ik} = 0 \text{ 또는 } 1, \quad i = 1, \dots, N_B \quad k = 1, \dots, N_T \quad (10)$$

$$y_{ijk} = 0 \text{ 또는 } 1, \quad i = 1, \dots, N_B \quad j = 1, \dots, N_B \quad k = 1, \dots, N_T \quad (11)$$

$$y_{OEK} = 0 \text{ 또는 } 1, \quad k = 1, \dots, N_T \quad (12)$$

$$y_{Okk}, y_{iEk} = 0 \text{ 또는 } 1, \quad i = 1, \dots, N_B \quad k = 1, \dots, N_T \quad (13)$$

상기 최적화 모형의 목적함수는 총 공차운행시간, 총 delay 시간, 그리고 총 tardy 시간의 가중합으로 표현된다. 제약식 (1)은 모든 블록은 pick-up 가능시간 이후에 운반될 수 있음을 나타내며, 제약식 (2)는 tardy 시간을 산출하기 위한 식이다. 제약식 (3)~식 (8)은 트랜스포터  $k$ 에 의해 운반되어지는 블록들과 이들 블록들의 운반순서를 만들어 낸다. 제약식 (3)은 동일 트랜스포터에 의해 운반되는 선후 블록들의 운송시작 시간을 제약하며, 제약식 (4)는 각 블록은 한 대의 트랜스포터에 의해서만 운반되어야 됨을 나타내며, 제약식 (5)~식 (6)은 트랜스포터  $k$ 에 할당된 모든 블록들은 반드시 그리고 한번만 그 트랜스포터의 운반순서상에 나타나야 됨을 제약한다. 제약식 (7)~식 (8)은 각 트랜스포터에 운반 시작과 종료시 한 블록만을 할당함으로써 전체적으로 한 번에 하나의 블록만 운반해야 함을 제약한다.

#### 4. 네트워크 흐름모형 기반의 일정계획 알고리즘

본 논문에서는 출발지와 도착지, pick-up and delivery 시간, 톤수, 고장, 유지보수 등의 요구조건이 주어진 블록들을 최소의 트랜스포터를 사용하여 계획기간 내에 모든 블록운반 요구사항들을 만족시키는 트랜스포터 일정계획문제를 다룬다. 트랜스포터 일정계획문제에서 중요한 주제는 총 로지스틱스 시간(공차운행시간, delay 시간 및 tardy 시간)을 최소화하기 위해 계획기간 내에 최소의 트랜스포터 운영대수로 각 트랜스포터에 어떤 블록을 할당하고 어떤 순서로 운반할 것인지 결정하는 것이다. 이를 위해 본 논문에서 제시하는 트랜스포터 일정계획의 전체 절차는 <Figure 1>과 같이 요약될 수 있다. 운반대상 블록들의 출발지와 도착지, pick-up and delivery 시간, 톤수 등의 요구조건이 주어지면 트랜스포터의 종류별로 운반가능

블록의 크기를 그룹화 하여 문제를 작은 크기로 분할한다. 즉, 유사 무게의 블록들을 그룹화하고 한 종류의 트랜스포터를 이용하여 운반하는 문제로 축소한다. 축소한 각각의 문제에 대해 필요한 최소의 트랜스포터수를 구하기 위해 Joo *et al.*(2006)이 제시한 최적화 모형을 이용한다.

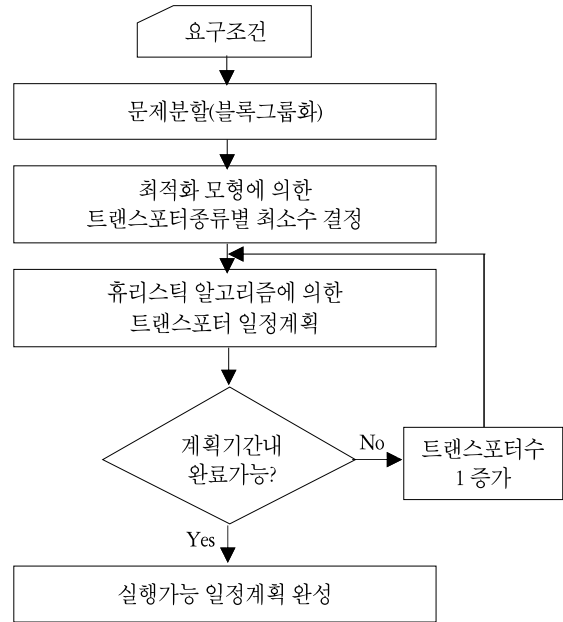


Figure 1. 트랜스포터 일정계획 절차

#### 4.1 휴리스틱 알고리즘의 개발

앞서 제시한 최적화 모형을 근간으로, Lee *et al.*(2008)이 제안한 4가지 휴리스틱 알고리즘들 중 가장 높은 효율성을 보여준 혼합형 Look-ahead 할당 알고리즘(MLA)과 본 논문에서 제안하는 네트워크 흐름모형 기반의 할당 알고리즘(NFA)의 성능을 비교하고자 한다.

##### (1) 혼합형 Look-Ahead 할당 알고리즘(MLA)

이 할당 정책은 모든 트랜스포터의 가용시점과 모든 블록의 pick-up 가능시점의 조합에 있어서 가장 빨리 운반할 수 있는 트랜스포터와 블록의 조합을 선택하여 할당하는 블록 주도형 배치에 다음 블록 운송 완료시간을 함께 고려하여(Look-Ahead) 가장 빠른 완료시간을 가지는 조합으로 할당한다.

단계 0:  $T_{now} = 0$ 으로 놓는다.

단계 1: 운반가능하지만 아직 운반되지 않은 블록 리스트( $BL$ )를 작성한다.

단계 2: 모든 트랜스포터와 두 개의 블록의 조합인 트랜스포터-블록-블록 리스트( $TBBL$ )를 작성한다.

단계 3: 다음의 할당 규칙에 의하여 운반 트랜스포터와 블록을 결정한다.

단계 3.1 : 모든 트랜스포터의 출발가능시간( $VT$ )과 블록의 pick-up 가능시간을 기준으로 할당한다.

단계 3.2 :  $TBBL$ 에 대하여  $min$  트랜스포터의 운반가능 시간 + 트랜스포터의 현 위치에서  $B_i$ (첫 번째 블록)으로의 공차운행시간 +  $B_j$ 적재운행시간 +  $B_j$ 에서  $B_i$ (두 번째 블록)로의 공차운행시간 +  $B_j$ 적재운행시간이 되는 트랜스포터-블록-블록의 조합에 대하여 가능한 delivery 시간 내에  $B_j$ 블록을 운송한다.

- 가능하지 않다면, 지연시간이 최소화되도록 트랜스포터를 선택한다.

단계 4 : 선정된 조합의 첫 번째 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록을  $BL$ 에서 삭제한다.
- 해당 트랜스포터가 선택된 블록을 운송 완료하는 시간을 계산한다.

$$(VT = T_{now} + \text{공차운행시간} + \text{적재운행시간})$$

단계 5 :  $BL$ 의 상태에 따라

- $|BL| = 0$ 이면, 절차를 종료한다.
- $|BL| > 0$ 이면, 트랜스포터들의 출발가능시간과 블록들의 pick-up 가능시간 중 최 소시간으로  $T_{now}$ 를 갱신하고 [단계 1]로 간다.

(2) 네트워크 흐름모형 기반의 할당 알고리즘(NFA)

이 할당 정책은 모든 트랜스포터의 가용시점과 모든 블록의 pick-up 가능시점의 조합에 있어서 네트워크 심플렉스 알고리즘(Network Simplex Algorithm)을 이용하여 얻어진 최적의 조합을 통해 블록을 운송하는 정책이다. 따라서 주어진 트랜스포터의 수만큼의 블록들이 동시에 할당된다.

의사결정 시점은 기 할당된 트랜스포터 중 가장 빨리 운송이 완료되는 트랜스포터의 유희시점으로 하며 이때 pick-up 요구되는 블록이 없다면 의사결정 시점은 블록들 중 가장 빠른 pick-up 요구시각으로 옮겨가게 된다. 이 시점에 할당되지 않은 블록들의 새로운 수송정보를 이용하여 트랜스포터들에 동시에 블록들을 할당한다. 한 번의 블록 할당이 이루어지면 이를 고정시키고 트랜스포터들과 남아 있는 블록들을 대상으로 이를 반복한다. 그 일정계획 절차는 다음과 같다.

단계 0 :  $T_{now} = 0$ 으로 놓는다.

단계 1 : 운반가능하지만 아직 운반되지 않은 블록리스트( $BL$ )를 작성한다.

단계 2 : 다음의 할당 규칙에 의하여 운반 트랜스포터와 블록을 결정한다.

단계 2.1 : 모든 트랜스포터의 출발가능시간( $VT$ )과 블록의 pick-up 가능시간을 기준으로 할당한다.

단계 2.2 : 주어진  $n$ 대의 트랜스포터 집합( $T$ )과  $m$ 개의 블록 집합( $BL$ )을 이용하여 최소비용 흐름 네트워크를 구성한다(<Figure 2> 참조). <Figure 2>에서 노드  $s_0$

와  $t_0$ 에서의 순흐름량은  $min(m, n)$ 으로 동일하고 다른 모든 노드의 순흐름량은 0이며 모든 호의 흐름용량은 1이다.  $T$ 와  $BL$ 를 연결하는 호에 대한 비용은  $(\alpha \times \text{공차운행시간} + \beta \times \text{delay시간} + \gamma \times \text{tardy시간})$ 으로 한다. 또한,  $s_0$ 와  $T$  및  $BL$ 와  $t_0$  간의 호에 대한 비용은 0이다.

단계 2.3 : 네트워크 심플렉스 알고리즘을 이용하여 최적해를 구한다.

단계 3 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록들을  $BL$ 에서 삭제한다.
- 해당 트랜스포터가 선택된 블록을 운송 완료하는 시간을 계산한다.

$$(VT = T_{now} + \text{공차운행시간} + \text{적재운행시간})$$

단계 4 :  $BL$ 의 상태에 따라

- $|BL| = 0$ 이면, 절차를 종료한다.
- $|BL| > 0$ 이면, 트랜스포터들의 출발가능시간과 블록들의 pick-up 가능시간 중 최 소시간으로  $T_{now}$ 를 갱신하고 [단계 1]로 간다.

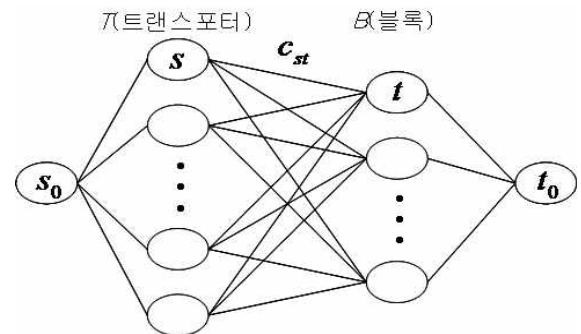


Figure 2. 최소비용 흐름 네트워크

4.2 성능평가

본 연구에서 제시한 네트워크 흐름모형 기반의 휴리스틱 알고리즘 NFA를 검증하기 위하여 본 논문에서는 여러 가지 실험 데이터를 생성하여 그 효율을 최적해와 비교 · 검증하였다. 데이터는 아래의 조건하에서, pick-up 시간, delivery 시간, 적재 및 하역 시간이 서로 다른 총 80개의 실험문제들을 랜덤하게 생성하였다.

- 계획기간 : 480분
- 이동거리(플랜트배치) : ‘D’조선소의 실제거리 사용(42개 플랜트)
- 대상 블록수 : 5, 8, 10, 13, 15, 20, 30, 40
- 각 대상 블록 수별 10개씩 생성
- pick-up 가능 시간(R) : 0 ~ 360분
- 요구되는 도착 시간 : (R+120) ~ (R+240)분
- 적재 시간(L) : 20 ~ 40분
- 하역 시간 : (L+10) ~ (L+20)분

- 공차운행속도 : 250 m/분
- 블록운송속도 : 135 m/분
- $\alpha, \beta, \gamma = 1$

실험은 생성된 각각의 데이터에 대해 Visual Basic으로 구현된 휴리스틱 알고리즘 MLA와 NFA의 결과를 CPLEX를 이용하여 구한 최적해와 비교·검증하였다. 제 3장에서 소개한 최적화 모형에서, 현재 블록의 수를  $M$ , 트랜스포터의 수를  $T$ 라 하고 블록의 수가  $N$ 개 추가된다고 할 때, 개략적으로 제약식은  $(2M+2T+2N+3)N$ 개, 변수는  $2(M+N+1)NT+3N$ 가 증가하게 된다. 이러한 제약식과 변수의 증가로 인해 CPLEX를 이용하여 최적해를 찾기 위해서는 많은 컴퓨터 메모리 용량을 요구하게 되며, 더 큰 규모의 문제들을 대상으로 한 효율성에 대한 검증은 최적해의 계산시간 문제로 인해 제한적인 실험을 할 수 밖에 없었음을 밝힌다.

80개의 실험문제들을 대상으로 소규모 문제(블록수 13개 이하)와 대규모 문제(블록수 15개 이상)로 나누어 휴리스틱 알고리즘 MLA와 NFA를 비교하였다. CPLEX로 최적해를 얻을 수 있는 5, 8, 10, 13개의 블록을 가지는 소규모 문제들에 대하여 비용 측면에서 최적해와의 비교결과를 보면, 휴리스틱 알고리

즘 MLA와 NFA의 효율성은 각각 최적해와 비교하여 평균 11.0%과 10.0% 정도의 차이를 보여주고 있다(<Table 1> 참조). 따라서 본 논문에서 제안한 휴리스틱 알고리즘 NFA가 MLA(혼합형 Look-Ahead 할당 알고리즘) 보다 평균적으로 더 우수함을 알 수 있다. 이는 트랜스포터들과 블록들 간의 최적 조합(주어진 트랜스포터의 수만큼의 최적 블록 할당)을 동시에 고려하는 것이 하나의 트랜스포터와 하나의 블록 간의 한 가지 조합만을 찾는 Look-Ahead 방식 보다는 더 우수한 결과를 보여줌을 나타낸다. 그러나 그 차이는 1% 정도로 매우 근소하므로 휴리스틱 알고리즘 MLA도 유용한 대안이 될 수 있음을 보여준다.

블록수 15, 20, 30, 40을 갖는 대규모 문제들에 대해서는 CPLEX에 의한 최적해를 구할 수 없어 휴리스틱 알고리즘들 중 가장 우수한 NFA를 기준으로 MLA와 비교해 본 결과, 평균 1% 정도 상대적으로 높은 효율성을 보여주었다(<Table 2> 참조). <Table 2> 내에서 음수값을 갖는 수치는 해당 실험문제에 대해 MLA가 NFA에 비해 상대적으로 우수한 해를 제공했음을 의미한다. 본 연구에서 사용한 실험문제들의 블록수송 요구정보들은 랜덤하게 생성됨에 따라 실험문제들의 구조적 특성에 의해, 휴리스틱 알고리즘 MLA가 우수한 결과를 보여주는 표본들도 발생하였다. 그러나 조선소 현장에서는 생산일정계획에 따

Table 1. CPLEX 최적해 대비 목적함수값 차이 비교

휴리스틱	블록 (TP)	GAP										GAP 평균
		1	2	3	4	5	6	7	8	9	10	
MLA	5(1)	0.00	0.13	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.02
MLA	8(1)	0.16	0.07	0.13	0.07	0.27	0.00	0.00	0.00	0.00	0.02	0.07
MLA	10(1)	0.08	0.18	0.14	0.07	0.00	0.09	0.01	0.17	0.01	0.03	0.08
MLA	13(2)	0.02	0.11	0.45	0.37	0.43	0.09	0.02	0.21	0.47	0.38	0.25
												0.11
NFA	5(1)	0.00	0.01	0.00	0.00	0.00	0.00	0.09	0.29	0.08	0.00	0.05
NFA	8(1)	0.00	0.00	0.00	0.07	0.17	0.00	0.00	0.07	0.03	0.00	0.03
NFA	10(1)	0.05	0.28	0.12	0.30	0.00	0.15	0.01	0.39	0.40	0.22	0.19
NFA	13(2)	0.00	0.00	0.00	0.01	0.23	0.02	0.34	0.05	0.19	0.28	0.11
												0.10

주) ()안은 사용된 트랜스포터의 수를 나타냄.

$$GAP = (\text{휴리스틱 목적함수 값} - \text{CPLEX 최적 목적함수 값}) / \text{CPLEX 최적 목적함수 값}$$

Table 2. NFA 대비 MLA 목적함수값 차이 비교

휴리스틱	블록 (TP)	GAP										평균 GAP
		1	2	3	4	5	6	7	8	9	10	
MLA	15(2)	0.06	0.04	-0.15	0.45	0.08	0.04	0.08	-0.13	0.10	0.62	0.12
MLA	20(2)	-0.01	-0.06	-0.09	-0.28	-0.08	0.00	0.03	0.09	-0.05	-0.13	-0.06
MLA	30(3)	0.29	0.03	-0.13	0.11	-0.13	-0.24	-0.16	0.00	-0.04	0.10	-0.02
MLA	40(3)	0.07	0.03	0.08	0.03	-0.12	0.01	0.09	0.03	0.00	-0.08	0.01
												0.01

주) ()안은 사용된 트랜스포터의 수를 나타냄.

$$GAP = (\text{MLA 목적함수 값} - \text{NFA 목적함수 값}) / \text{NFA 목적함수 값}$$

른 실제 수송정보들이 계획기간에 걸쳐 비교적 순차적으로 잘 나열되어 있어, 휴리스틱 알고리즘 NFA의 적용결과가 MLA에 비해 더 우수할 것으로 예상된다.

최적화 모형은 전형적인 NP-hard 문제이므로 실시간에 우수한 해를 얻는 것이 불가능하나, 휴리스틱 모형은 해를 찾는 계산시간이 무시할 수 있을 정도로 적어 실제문제에 적용 시 효율적인 일정계획의 수립이 가능하리라 여겨진다(<Table 3> 참조). <Table 3>은 소규모 문제들에 대한 CPLEX와 휴리스틱 알고리즘 NFA의 계산시간을 비교한 것이다. <Table 3>에서 보듯이, 휴리스틱 알고리즘 NFA는 모든 문제들에 대해 1초 내에 해를 찾는 탁월한 계산속도를 보여주고 있어, 문제의 규모(트랜스포터의 수 및 블록의 수)가 증가할수록 휴리스틱 알고리즘 NFA의 효율성은 더욱 증대될 것으로 기대된다.

D조선소의 실제 현장 데이터로 하루 총 36개의 블록이동을 요구하고 6대의 트랜스포터를 이용하여 주·야간 운영하였던 현장데이터를 사용하여 제안된 휴리스틱 알고리즘들을 수행한 결과, 총 3대의 트랜스포터로 주간만 운영하여도 모든 블록의 이동을 완료할 수 있었다. 이러한 최적 일정계획을 통해, 현재 35%인 트랜스포터 가동율을 적은 수의 트랜스포터를 운영하면서도 더욱 극대화할 수 있을 것으로 기대된다.

### 5. 연동적 일정계획 알고리즘(RSA)

본 절에서는, 동적 블록수송환경에 효율적으로 적용하기 위해, 제 4장에서 제안된 휴리스틱 알고리즘 NFA에 시간창(Time Window)의 개념을 적용한 연동적 일정계획(Rolling Scheduling) 알고리즘 RSA를 제안한다. 여기서, 전체 계획기간 중 일부 제한된 시간대에서의 확정된 수송정보의 관리를 위한 시간영역의 크기를 시간창이라 부르기로 한다. 예를 들어, 7개 블록들의 pick-up 가능시간(분)이 각각 4, 12, 35, 67, 100, 150, 250이고, 의

사결정 시점이 12분이라고 가정하자. 이 때, 시간창이 100분이 라면 pick-up 가능시간이 112분 보다 작은 5개 블록들만을 수송 정보로 확정하여 일정계획을 수립하게 된다는 것을 의미한다.

앞서 제안한 NFA는 운반가능한 모든 블록들과 주어진 트랜스포터들을 대상으로 네트워크 심플렉스 알고리즘을 적용하여 블록을 할당하였으나 이는 최초로 모든 운반대상 블록들을 사전에 알고 있는 정적 수송환경 하에서 가능하다. 실제 조선현장에서의 동적 수송환경에서는 실시간으로 블록 수송의 추가 및 변경, 트랜스포터의 돌발고장 및 유지보수 등의 요청이 나타날 것이며 의사결정시점에 확정된 제한된 수송정보만을 이용하여 일정계획을 수립하는 것이 바람직할 것이다. 그럼에도 불구하고 실제 조선현장에서는 시간창의 개념을 활용한 실시간 일정계획을 수립하지 않고 있어 수송정보의 변경에 따른 뒤늦은 일정계획의 변경으로 물류지연이 많이 발생되고 있는 것이 현실이다. 따라서 주어진 시간창 내에서의 확정된 수송정보만을 이용하여 트랜스포터들의 준비시간을 기준으로 의사결정을 하는 것이 바람직하며 이를 위한 연동적 일정계획 알고리즘 RSA의 절차를 다음과 같이 제안한다.

단계 0 :  $T_{now} = 0$ 으로 놓는다.

단계 1 : 다음의 규칙에 의하여 운반가능한 블록리스트(BL)를 작성한다.

단계 1.1 : 출발가능시간(VT)이 가장 빠른 트랜스포터를 기준으로 주어진 시간창 내의 블록들을 BL에 추가한다. BL의 수가 트랜스포터의 수보다 적을 경우, 그 다음 트랜스포터의 출발가능시간을 기준으로 시간창 내의 블록들을 BL에 추가한다.

단계 1.2 : 가장 늦은 출발가능시간을 갖는 트랜스포터에서의 시간창 내의 BL의 수가 트랜스포터의 수보다 적다면, 운반 가능하지만 아직 운반되지 않은 블록 중 가장 빠른 pick-up 가능시간을 갖는 블록의

Table 3. 해시간 비교

(단위 : 초)

블록 (TP)	1		2		3		4		5	
	Opt	Heu	Opt	Heu	Opt	Heu	Opt	Heu	Opt	Heu
5(1)	0	0	0	0	0	0	0	0	0	0
8(1)	1	0	1	0	0	0	0	0	2	0
10(1)	366	0	71	0	10	0	331	0	39	0
13(2)	92	1	5,210	1	22	1	5,237	1	4,377	0
블록 (TP)	6		7		8		9		10	
	Opt	Heu	Opt	Heu	Opt	Heu	Opt	Heu	Opt	Heu
5(1)	1	0	0	0	0	0	0	0	0	0
8(1)	0	0	0	0	0	0	1	0	0	0
10(1)	219	0	45	0	7	0	83	0	874	0
13(2)	708	0	3,174	1	5,725	1	336	1	834	1

주) ()안은 사용된 트랜스포터의 수를 나타냄.

pick-up 가능시간을 기준으로 시간창 내의 블록을 BL로 설정한다.

단계 2 : 다음의 할당 규칙에 의하여 운반 트랜스포터와 블록을 결정한다.

단계 2.1 : 모든 트랜스포터의 출발가능시간과 BL의 pick-up 가능시간을 기준으로 ( $\alpha \times$  공차운행시간 +  $\beta \times$  delay시간 +  $\gamma \times$  tardy시간)을 비용으로 하고 모든 호의 흐름용량을 1로 하는 최소비용 흐름 네트워크를 작성한다.

단계 2.2 : 네트워크 심플렉스 알고리즘을 이용하여 최적해를 구한다.

단계 3 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록들을 BL에서 삭제한다.
- 트랜스포터들이 선택된 블록을 운송 완료하는 시간을 계산한다.

$$(VT = T_{now} + \text{공차운행시간} + \text{적재운행시간})$$

단계 5 : 운반 가능한 블록의 상태에 따라

- 운반가능한 블록의 수 = 0이면, 절차를 종료한다.

- 운반가능한 블록의 수 > 0이면, 트랜스포터들의 출발가능시간과 블록들의 pick-up 가능시간 중 최소시간으로  $T_{now}$ 를 갱신하고 [단계 1]로 간다.

제 4.2절에서의 실험문제들을 대상으로, 시간창을 60분, 120분, 180분, 240분으로 설정하여 NFA와 대비하여 RSA의 목적함수값의 차이를 비교해 보았다(<Table 4> 참조). <Table 4>에서 보듯이, 시간창이 60분일 경우에 좁은 시간영역으로 인하여 미세하게 목적함수값의 차이가 블록의 수가 15, 20, 30, 40인 경우에 발생하였으며, 이는 좁은 시간영역과 수송정보의 랜덤성에서 기인한다 하겠다.

시간창이 60분이고 GAP이 발생하였던 블록의 수 15, 20, 30, 40인 경우에 대해, Paired t검정을 이용하여 NFA와 RSA 목적함수값의 차이에 대한 유의성을 검증한 결과를 <Table 5>에 나타내었다. <Table 5>에서 보듯이 유의수준 5%하에서 4가지 경우 모두가 채택구간에 포함되므로 NFA와 RSA 목적함수값들 간에 차이가 없음을 검증할 수 있었다. 120분, 180분, 240분의 경우에는 시간창이 커지면서 보다 많은 정보를 이용하여 의사결

Table 4. NFA 대비 RSA의 목적함수값 차이 비교

시간창	블록 (TP)	GAP										평균 GAP	
		1	2	3	4	5	6	7	8	9	10		
W = 60	5(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	8(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	10(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	13(2)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	15(2)	0.000	0.000	-0.181	0.000	0.000	0.000	0.000	0.000	0.070	-0.037	0.000	-0.015
	20(2)	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
	30(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.005	-0.001
	40(3)	0.000	0.000	0.044	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004
												-0.001	
W = 120 W = 180 W = 240	5(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	8(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	10(1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	13(2)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	15(2)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	20(2)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	30(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	40(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
												0.000	

주) ()안은 사용된 트랜스포터의 수를 나타냄.

$$GAP = (RSA \text{ 목적함수 값} - NFA \text{ 목적함수 값}) / NFA \text{ 목적함수 값}$$

Table 5. Paired t 검정 결과

블록 수	표본평균	표본표준편차	$\hat{t}$ 검정통계량	비교	결과
15	-0.0148	0.0640	-0.7315	$ \hat{t}  < t_{0.025}(9) = 2.262$	차이 없음
20	0.0008	0.0025	1.0000	$ \hat{t}  < t_{0.025}(9) = 2.262$	차이 없음
30	-0.0005	0.0016	-1.0000	$ \hat{t}  < t_{0.025}(9) = 2.262$	차이 없음
40	0.0044	0.0139	1.0000	$ \hat{t}  < t_{0.025}(9) = 2.262$	차이 없음



정을 하게 되어 휴리스틱 알고리즘 NFA와 해의 차이가 없음을 보여준다. 또한 RSA는 주어진 실험문제들에 대해 최대 2초 내에 해를 찾는 계산속도를 보여주었다. 따라서 NFA를 기반으로 하는 연동적 일정계획 알고리즘 RSA는 동적 수송환경을 갖는 조선소 현장에서 시간창 내의 확정된 일부 수송정보만을 가지고 효율적으로 일정계획을 수립할 수 있을 것으로 기대된다.

## 6. 결론

본 논문에서는 조선소에서 계획기간동안 각 플랜트의 생산일정에 따른 블록의 수송정보의 변경, 트랜스포터의 돌발고장 및 유지보수 등이 수시로 발생하는 동적인(dynamic) 수송환경에서의 트랜스포터 일정계획문제를 다루었다. 먼저, 총 로지스틱스 시간을 최소화하도록 각 트랜스포터별 효율적인 경로 및 일정 계획을 수립할 수 있는 네트워크 흐름모형 기반의 휴리스틱 알고리즘 NFA를 제안하였다. 80개의 실험문제들을 대상으로 소규모 문제(블록수 13개 이하)와 대규모 문제(블록수 15개 이상)로 나누어 기존의 휴리스틱 알고리즘 MLA와 비교하였다. 소규모 문제들에 대해서는 휴리스틱 알고리즘 NFA가 비용 측면에서 최적해에 비해 평균 10.0%의 차이를 보였고 휴리스틱 알고리즘 MLA에 비해 평균 1% 정도의 우수성을 나타내었다. 계산시간 측면에서는 모든 대상문제들을 1초 내에 계산하는 탁월한 효율성을 입증하였다. 대규모 문제들에 대해서는 NFA가 MLA에 비해 평균 1% 정도로 근소하나마 효율성이 높음을 알 수 있었다. 따라서 문제의 규모가 증가할수록 휴리스틱 알고리즘 NFA의 효율성은 더욱 증대될 것으로 기대된다. 또한, 동적 수송환경에서의 실시간 의사결정을 위해 연동적 일정계획 알고리즘 RSA를 제안하였고 NFA와 비교하여 차이가 없음을 paired  $t$ 검정을 이용하여 검증하였다. 따라서 NFA를 기반으로 하는 RSA는 동적 수송환경을 갖는 조선소 현장에서 유용하게 사용될 수 있을 것으로 기대된다.

D조선소의 실제 현장 데이터로서 하루 총 36개의 블록이동을 요구하고 6대의 트랜스포터를 이용하여 주·야간 운영하였던 현장데이터를 사용하여 제안된 휴리스틱 알고리즘을 수행한 결과, 총 3대의 트랜스포터로 주간만 운영하여도 모든 블록의 이동을 완료하는 효율성을 확인할 수 있었다. 따라서 제안된 휴리스틱 알고리즘 NFA와 RSA를 조선소 현장에 적용한다면 동적 수송환경에서의 트랜스포터의 운영 효율성을 더욱 극대화할 수 있을 것으로 기대된다. 나아가 트랜스포터 운전자의 단말기를 통한 실시간 통신 시스템이 구축된다면 더욱 실질적이고 효율적인 트랜스포터의 운영을 기대할 수 있을 것이다.

## 참고문헌

- Baker, E. K. (1983), An Exact Algorithm for the Time-Constrained Traveling Salesman Problem, *Qns. Res.*, **31**, 939-945.
- Bard, J. F., Kontoravdis, G., and Yu, G. (2002), A Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows, *Transportation Sci.*, **36**, 250-269.
- Bianco, L., Mingozzi, A., and Sicciardelli, S. (1998), Dynamic Programming Strategies for the Traveling Salesman Problem with Time Windows and Precedence Constraints, *Qns. Res.*, **45**, 365-378.
- Calvo, R. W. (2000), A New Heuristic for the Traveling Salesman Problem with Time Windows, *Transportation Sci.*, **34**, 113-124.
- Chao, I. M. (2002), A Tabu Search Method for Truck and Trailer Routing Problem, *Computers and OR.*, **29**, 33-51.
- Crainic, T. G. and Laporte, G. (1998), *Fleet Management and Logistics*, Kluwer Academic Publishers.
- Desrosiers, J., Sauve, M., and Soumis, F. (1988), Lagrangian Relaxation Methods for Solving the Minimum Fleet Size Multiple Traveling Salesman Problem with Time Windows, *Management Sci.*, **34**, 1005-1022.
- Dumas, Y., Desrosiers, J., Gelinas, E., and Solomon, M. M. (1995), An Optimal Algorithm for the Traveling Salesman Problem with Time Windows, *Qns. Res.*, **45**, 367-371.
- Gendreau, M., Hertz, A., Laporte, G., and Stan, M. (1998), A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows, *Qns. Res.*, **46**, 330-335.
- Joo, C. M., Lee, W. S., Koo, P. H., and Lee, K. B. (2006), Transporter Scheduling for Block Transportation in Shipbuilding, *J. of the Korea Management Engineers Society*, **11**(3), 169-179.
- Ko, C. S., Jung, K. H., and Shin, J. Y. (2000), Determination of Vehicle Fleet Size for Container Shuttle Service, *Management Science*, **17**(2), 87-95.
- Koo, P. H., Lee, W. S., and Jang, D. W. (2004), Fleet Sizing and Vehicle Routing for Container Transportation in a Static Environment, *OR Spectrum*, **26**, 193-209.
- Langevin, A., Desrochers, M., Kesrosiers, J., and Fournis, F. (1993), A Two Commodity Flow Formulation for the Traveling Salesman and Makespan Problems with Time Windows, *Network*, **23**, 631-640.
- Laporte, G. and Osman, H. (1995), Routing Problems : a Bibliography, *Annals of OR.*, **61**, 227-262.
- Lee, W. S., Lim, W. I., Koo, P. H., and Joo, C. M. (2008), Transporter Scheduling for Dynamic Block Transportation Environment, *IE Interfaces*, **21**(3), 274-282.
- Lim, W. I., Lee, W. S., Koo, P. H., and Kim, D. J. (2008), A Transporter Scheduling System for Dynamic Block Transportation Environment, *J. of the Korea Management Engineers Society*, **13**(2), 55-71.
- Maxwell, W. L. and Muckstadt, J. A. (1982), Design of Automated Guided Vehicle Systems, *IIE Trans.*, **14**, 114-124.
- Pesant, G., Gendreau, M., Potvin, J.-Y., and Rousseau, J.-M. (1998), An Exact Constraint Logic Programming Algorithm for the TSP with Time Windows, *Transportation Sci.*, **32**, 12-29.
- Savelsbergh, M. W. P. (1985), Local Search in Routing Problems with Time Windows, *Ann. Qns. Res.*, **4**, 285-305.
- The Korea Shipbuilder's Association, www.koshipa.or.kr.



**이운식**

고려대학교 산업공학 학사  
한국과학기술원 산업공학 석사  
한국과학기술원 산업공학 박사  
현재: 부경대학교 시스템경영공학과 교수  
관심분야: OR응용, 시스템최적화, 컴퓨터  
시뮬레이션 응용



**임원일**

부경대학교 시스템경영공학 학사  
부경대학교 시스템경영공학 석사  
현재: DK유엔씨(주) CS2팀  
관심분야: OR응용, 시스템최적화



**구평희**

한양대학교 산업공학 학사  
미국 Purdue대학교 산업공학 석사  
미국 Purdue대학교 산업공학 박사  
현재: 부경대학교 시스템경영공학과 부교수  
관심분야: 생산 및 물류 시스템의 설계 및  
운영