

동적 블록수송환경을 위한 트랜스포터 일정계획

이운식^{1*} · 임원일¹ · 구평희¹ · 주철민²

¹부경대학교 시스템경영공학과 / ²동서대학교 시스템경영공학과

Transporter Scheduling for Dynamic Block Transportation Environment

Woon-Seek Lee¹ · Won-Il Lim¹ · Pyung-Hoi Koo¹ · Cheol-Min Joo²

¹Systems Management and Engineering, Pukyong National University

²System and Management Engineering, Dongseo University

This paper considers a transporter scheduling problem under dynamic block transportation environment in shipbuilding. In dynamic situations, there exist the addition or cancellation 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 heuristic algorithms which minimize the weighted sum of empty transporter travel times, delay times, and tardy times. Four heuristic algorithms for transporter scheduling are proposed and their performance is evaluated.

Keyword: transporter scheduling, block allocation, dynamic block transportation environment, shipbuilding

1. 서론

한국조선공업협회 발표 자료에 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 넘어섰으며, 국내 총무역수지 흑자 108억불의 93.3%를 차지할 정도로 국가 기간 중추 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 조선

산업의 낮은 생산성은 일본대비 10% 가격경쟁력 우위를 불안하게 하고 있는 실정이다. 2010년 150억불의 신조선 수출로 세계시장 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 절재절단, 부분품을 제공단계의 좀 더 큰 구조물로 만드는 블록조립, Dock내에서 블록들을

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

E-mail : iewlee@pknu.ac.kr

투고일(2008년 04월 14일), 심사일(1차 : 2008년 04월 15일), 게재확정일(2008년 06월 09일).

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 조선소 내의 여러 곳에 분산되어 관리된다. 조선소에서는 이러한 다양한 형상과 무게를 갖는 블록들을 각 플랜트가 원하는 시간에 pick-up하고 delivery하기 위해 트랜스포터(Transporter)라는 고가의 블록운반용 특수차량을 이용한다. 트랜스포터는 100, 200, 300, 350, 500톤 급 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 선박 건조 기간의 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 수송요구량, 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 블록 할당과 운반순서를 효율적으로 결정하는 4가지 휴리스틱 알고리즘들을 제안하고 알고리즘의 성능평가도 비교한다. 마지막으로, 제 5장에서는 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 시간*이라 한다)과 요구되는 delivery 시각 이후에 블록운반이 완료될 때 발생하는 지연시간(이하 *tardy 시간*이라 한다)의 2가지 벌과시간(penalty time)을 유발한다. 더구나, 블록들은 조선소 내의 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 현위치까지 이동하는 동안 공차운행시간이 발생된다.
- (2) 가능하면 블록의 pick-up시각에 맞추어 블록의 출발지에 도착한다. 이때, pick-up시각 이후에 트랜스포터가 도착하면 *delay 시간*이 발생된다.
- (3) 출발지에 도착 후, 블록을 적재하기 위해서는 블록의 형상과 무게에 따라 일정한 *적재시간*이 소요된다.
- (4) 트랜스포터는 블록을 적제한 후 도착지까지 이동한다. 이때 *적재운행시간*이 발생되며 적재시간 및 하역시간은 모두 여기에 포함된다.
- (5) 트랜스포터가 delivery 시각 이후에 도착하게 되면 *tardy 시간*이 발생된다.
- (6) 운반 후, 트랜스포터는 다음 블록을 운반하기 위해 현 위치에서 대기한다.

조선소 내의 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 목표는 동적인 블록수송환경에서 트랜스포터의 이동 거리에 따른 이동시간을 최소화하고 트랜스포터의 공차운행 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 delivery 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 일정계획문제는 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. Savelsbergh(1985)는 TSPTC에 대한 interchange 휴리스틱 알고리즘을 제안하였다. Desrosiers 등(1988)은 최소 차량대수를 구하기 위해 확장된 라그랑주 기법을 기초로 한 알고리즘을 제시했다. Dumas 등(1995)은 최적해를 찾기 위한 동적계획법 알고리즘을 제안한 반면 Langevin 등(1993)은 branch-and-cut 알고리즘에 의해 풀 수 있는 two-commodity flow problem를 개발하였다. Bianco 등(1998)은 선후제약식을 갖는 TSPTC를 풀기 위한 2단계 동적계획법에 기초한 알고리즘을 제안하였다. Gendreau 등(1998)은 insertion과 post-optimization 기법을 결합한 휴리스틱 알고리즘을 개발하였다. Pesant 등(1998)은 logic constraint programming에 기초한 기존의 논문과 차별된 알고리즘을 제시하였다. Calvo (2000)는 greedy insertion 기법에 기초한 휴리스틱 알고리즘을 제안하였다. Bard 등(2002)은 시간제약을 갖는 차량 경로문제에서의 최소 차량대수 결정문제를 다루었으며 branch-and-cut에 기초한 알고리즘을 개발하였다.

수송 시스템에서 차량대수결정문제를 다루는 또 다른 분야로는 자동화된 제조시스템에서의 AGV 시스템들과 관련된다. Maxwell and Muckstadt(1982)는 계획기간동안 수요요구량이 주어질 때, AGV의 최소 대수를 결정하는 수리모형을 제안하였다. 고창성 등(2000)은 컨테이너 수송 시스템에서 차량대수 및 차량경로 결정을 위한 insertion 알고리즘을 제시하였다. Koo 등(2004)은 컨테이너 수송 시스템에서 수송요구량을 수행하기 위한 차량대수 및 차량경로를 동시에 결정하는 2단계 휴리스틱 알고리즘을 개발하였다.

조선생산공정에서 블록의 생산 및 배치와 관련된 최근 연구들을 정리하면 다음과 같다. 박명환 등(1995)는 국내 조선산업의 개요와 산업공학적인 관점에서의 응용분야들을 조사하였다. 고시근 등(1999)은 조선소 내에서 곡블록 조립공정에서의 조립일정계획문제를 다루었고 유전자 알고리즘을 이용한 일정계획의 수립 방안을 제안하였다. 하태룡 등(2000)은 조선소 내에서 선각 평블록 조립공정에서의 효율적 일정계획 수립을 위한 혼합 유전자 알고리즘을 제시하였다. 정귀훈 등(2001)은 조선 도장공정에서의 효율적 공간일정계획 수립을 위한 시간할당, 공간할당, 부하균형 등을 고려한 일정계획 알고리즘을 개발하였다. 주철민 등(2006)은 계획기간동안 블록의 모든 요구량이 미리 알려져 있는 정적인 수송환경에서의 트랜스포터의 일정계획을 효율적으로 수립할 수 있는 휴리스틱 알고리즘을 제안하였다. 그러나, 조선소 내에서의 트랜스포터 일정계획에 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

주철민 등(2006)에서 제시한 최적화 모형의 경우, 의사결정변수를 정의하는데 있어, 많은 인덱스를 사용함으로써 변수들의 수를 증가시켜 문제의 복잡성을 증대시키고 있다. 본 논문에서는 의사결정변수의 정의를 새롭게 하여 문제의 복잡성을 줄이고 목적식을 개선한다. 사용되는 기호는 다음과 같다.

- x_i : 블록 i 의 운송시작 시간
- r_i : 블록 i 의 pick-up 가능 시간(ready time)
- c_i : 블록 i 의 요구되는 도착 시간(delivery time)
- t_{ik} : 트랜스포터 k 의 블록 i 를 운반하는데 소요되는 시간 (적재, 하역시간 포함)
- e_{ijk} : 트랜스포터 k 의 블록 i 를 운반한 후 블록 j 를 pick-up하기 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
- $y_{ik} = \begin{cases} 1 & \text{if (트랜스포터 } k \text{에 의해 블록 } i \text{가 운반되어지면)} \\ 0 & \text{otherwise} \end{cases}$
- $y_{ijk} = \begin{cases} 1 & \text{if (트랜스포터 } k \text{에 의해 블록 } i \text{후 블록 } j \text{가 운반되어지면)} \\ 0 & \text{otherwise} \end{cases}$
- 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(무한대)
- α, β, γ : 가중치

개선된 최적화 모형은 다음과 같다.

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

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

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

$$x_i + \sum_{k=1}^{N_T} t_{ik} y_{ik} + \sum_{k=1}^{N_T} e_{ijk} y_{ijk} \leq x_j + M \cdot \left(1 - \sum_{k=1}^{N_T} y_{ijk}\right), \quad i = 1, \dots, N_B \quad j = 1, \dots, N_B \quad j \neq i \quad (3)$$

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

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

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

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

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

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

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

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

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

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

상기 최적화 모형의 목적함수는 총 공차운행시간, 총 delay 시간, 그리고 총 tardy 시간의 가중합으로 표현된다.

제약식 (1)은 모든 블록은 pick-up 가능 시간 이후에 운반될 수 있음을 나타내며, 제약식 (2)는 tardy 시간을 산출하기 위한 식이다. 제약식 (3)~제약식 (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)~제약식 (6)은 트랜스포터 k 에 할당된 모든 블록들은 반드시 그리고 한번만 그 트랜스포터의 운반순서상에 나타나야 됨을 제약한다. 제약식 (7)~제약식 (8)은 각 트랜스포터에 운반 시작과 종료시 한 블록만을 할당함으로써 전체적으로 한번에 하나의 블록만 운반해야 함을 제약한다.

블록의 수를 N , 트랜스포터의 수를 T 라 할 때, 본 논문에서 제시한 최적화 모형에 대한 문제의 복잡성은 주철민 등(2006)의 최적화 모형에 비해, 의사결정변수는 $3N(T-1)$ 개, 제약식은 $N^2T + 3N(T-1)$ 개가 감소되어 문제의 복잡성이 현저히 줄어들게 됨을 알 수 있다.

4. 휴리스틱 알고리즘

본 논문에서는 출발지와 도착지, pick-up과 delivery 시각, 톤수, 결선, 고장 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 블

록을 할당하고 어떤 순서로 운반할 것인지 결정하는 것이다. 이를 위해 본 논문에서 제시하는 트랜스포터 일정계획의 전체 절차는 <그림 1>과 같이 요약할 수 있다. 운반 대상 블록들의 출발지와 도착지, pick-up과 delivery 시각, 톤수 등의 요구조건이 주어지면 트랜스포터의 종류별로 운반가능 블록의 크기를 그룹화하여 문제를 작은 크기로 분할한다. 즉, 유사 무게의 블록들을 그룹화하고 한 종류의 트랜스포터를 이용하여 운반하는 문제로 축소된다.

4.1 휴리스틱 알고리즘의 개발

앞서 제시한 최적화 모형을 근간으로, 본 논문에서는 주철민 등(2006)이 제안한 휴리스틱 알고리즘(이하 H1이라 한다). 이 규모가 더 큰 문제에 대해서도 효율성을 유지할 수 있는지를 비교하기 위해, 4가지 휴리스틱 알고리즘을 추가적으로 제안하고 이의 성능을 비교하고자 한다.

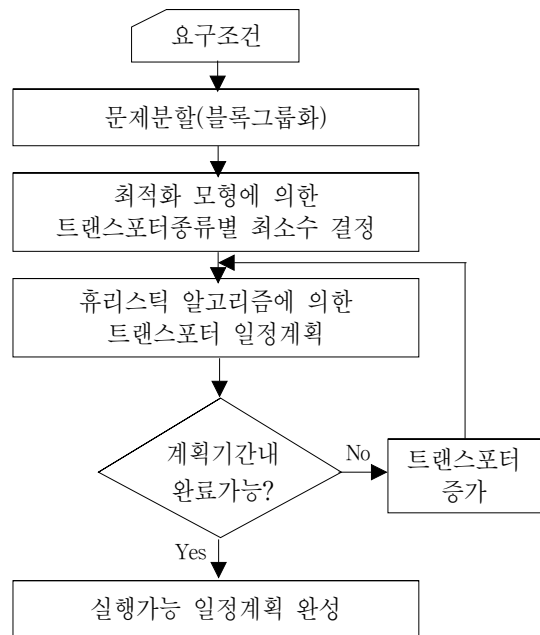


그림 1. 트랜스포터 일정계획 절차

(1) 트랜스포터 주도형 블록매치 알고리즘(H2)

이 방법은 가장 빨리 준비되는 트랜스포터를 기준으로, 아직 운반되지 않은 블록 중에서 운송완료 시간이 최소화되도록 블록을 할당하는 방법이다. 이는 트랜스포터의 위치와 각 블록의 준비시간에 대한 정보만 알면 의사결정이 가능한 방법으로 현장에서 비교적 어렵지 않게 적용할 수 있는 방법이다. 그 절차는 아래와 같다.

단계 1 : 현재시각(T_{now})을 가장 빨리 준비되는 트랜스포터의 가용시점(VT)로 초기화한다.

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

단계 2 : 현재시각에 아직 운반되지 않은 블록리스트(BL)를 작성한다.

단계 3 : 준비시간이 가장 빠른 트랜스포터를 선정하고 다음의 할당 규칙에 의하여 운반 블록을 결정한다.

단계 3.1 : 트랜스포터의 준비시간 기준의 가능한 블록들을 대상으로 할당한다.

단계 3.2 : 운행예상시간(공차운행시간+적재운행시간)을 기준으로 가능한 delivery 시각 내에 운반이 완료되도록 블록 할당한다.

- 가장 가까이 있는 블록을 대상으로 한다(Closest Unscheduled Block).
- 가능하지 않다면, 지연시간이 최소화되도록 할당한다.

단계 4 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록을 BL 에서 삭제한다.
- 해당 트랜스포터가 선택된 블록을 운반완료 하는 시간을 계산한다.

단계 5 : BL 의 상태에 따라

- IF $|BL| = 0$, THEN 알고리즘 종료한다.
- IF $|BL| > 0$, THEN 단계 1로 간다.

(2) 블록 주도형 트랜스포터 할당 알고리즘(H3)

이 할당 정책은 블록이 준비되는 시점에 트랜스포터의 준비시간에 대하여 현 블록에의 공차운행시간+적재운행시간이 가장 빠른 트랜스포터를 선택하는 방법이다. 이는 각 트랜스포터의 예상완료시점과 위치에 대한 정보만 알면 의사결정이 가능한 방법으로 현장에서 비교적 어렵지 않게 적용할 수 있는 방법이다. 그 절차는 아래와 같다.

단계 1 : 아직 운반되지 않은 블록리스트(BL)를 Pick-up 시각을 기준으로 작성한다.

단계 2 : 현재시각(T_{now})를 가장 빨리 준비되는 블록의 준비시간으로 초기화한다.

단계 3 : 블록이 선정되면, 다음의 할당 규칙에 의하여 운반 트랜스포터를 결정한다.

단계 3.1 : 블록의 준비시간 기준의 트랜스포터들을 대상으로 할당한다.

단계 3.2 : 운행예상시간(공차운행시간+적재운행시간)을 기준으로 가능한 delivery 시각 내에 운반이 완료되도록 트랜스포터 할당한다.

- 가장 빨리 블록을 운송할 수 있는 트랜스포터를 선택한다.
- 가능하지 않다면, 지연시간이 최소화되도록 트랜스포터 선택한다.

단계 4 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록을 BL 에서 삭제한다.

- 해당 트랜스포터가 선택된 블록을 운반완료 하는 시간 계산한다.

단계 5 : BL 의 상태에 따라

- IF $|BL| = 0$, THEN 알고리즘 종료한다.
- IF $|BL| > 0$, THEN 단계 1로 간다.

(3) 혼합형 할당 알고리즘(H4)

이 할당 정책은 모든 트랜스포터의 가용시점과 모든 블록의 준비시점의 조합에 있어서 가장 빨리 운반할 수 있는 트랜스포터와 블록의 조합을 선택하여 할당하는 방법이다.

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

단계 2 : 모든 트랜스포터와 BL 의 조합인 트랜스포터-블록리스트(TBL)를 작성한다.

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

단계 3.1 : 모든 트랜스포터의 준비시간과 블록의 준비시간을 기준으로 할당한다.

단계 3.2 : TBL 에 대하여 \min (공차운행시간+적재운행시간)이 되는 트랜스포터와 블록의 조합으로 가능한 delivery 시각 내에 블록을 운반한다.

- 가능하지 않다면, 지연시간이 최소화되도록 트랜스포터 선택한다.

단계 4 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록을 BL 에서 삭제한다.
- 해당 트랜스포터가 선택된 블록을 운반완료 하는 시간을 계산한다($VT = T_{now} + \text{공차운행시간} + \text{적재운행시간}$).

단계 5 : BL 의 상태에 따라

- IF $|BL| = 0$, THEN 알고리즘 종료한다.
- IF $|BL| > 0$, THEN 단계 1로 간다.

(4) 혼합형 Look-Ahead 할당 알고리즘(H5)

이 할당 정책은 모든 트랜스포터의 가용시점과 모든 블록의 준비시점의 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 1 : 운반가능하지만 아직 운반되지 않은 블록리스트(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 준비시간과 블록의 준비시

간을 기준으로 할당한다.

단계 3.2 : $TBBL$ 에 대하여 \min (트랜스포터의 현 위치에서 B_1 (첫 번째 블록)으로의 공차운행시간 + B_1 적재운행시간 + B_1 에서 B_2 (두 번째 블록)로의 공차운행시간 + B_2 적재운행시간)이 되는 트랜스포터-블록-블록의 조합에 대하여 가능한 delivery 시각 내에 B_1 블록을 운송한다.

- 가능하지 않다면, 지연시간이 최소화되도록 트랜스포터 선택한다.

단계 4 : 선정된 블록 및 트랜스포터에 대한 데이터를 갱신한다.

- 해당 블록을 BL 에서 삭제한다.
- 해당 트랜스포터가 선택된 블록을 운반완료 하는 시간 계산한다($VT = T_{now} + \text{공차운행시간} + \text{적재운행시간}$).

단계 5 : BL 의 상태에 따라

- IF $|BL| = 0$, THEN 알고리즘 종료한다.
- IF $|BL| > 0$, THEN 단계 1로 간다.

4.2 성능평가

기존의 휴리스틱 알고리즘(H1)과 본 연구에서 제시한 4가지 휴리스틱 알고리즘들을 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 실제 거리 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

표 1. CPLEX 최적해 대비 목적값 차이 비교

| 휴리스틱 | 블록 (TP) | GAP | | | | | | | | | | GAP 평균 | |
|------|---------|------|------|------|------|------|------|------|------|------|------|--------|------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| H1 | 5(1) | 0.90 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 | 0.38 | 0.15 |
| H1 | 8(1) | 0.35 | 0.00 | 0.00 | 0.17 | 0.04 | 0.07 | 0.12 | 0.28 | 0.13 | 0.15 | 0.15 | 0.13 |
| H1 | 10(1) | 0.21 | 0.19 | 0.19 | 0.24 | 0.22 | 0.21 | 0.05 | 0.26 | 0.17 | 0.43 | 0.43 | 0.22 |
| H1 | 13(2) | 0.02 | 0.23 | 0.10 | 0.46 | 0.17 | 0.03 | 0.36 | 0.00 | 0.23 | 0.43 | 0.43 | 0.20 |
| | | | | | | | | | | | | | 0.18 |
| H2 | 5(1) | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.09 | 0.29 | 0.08 | 0.00 | 0.00 | 0.05 |
| H2 | 8(1) | 0.00 | 0.00 | 0.00 | 0.07 | 0.17 | 0.00 | 0.00 | 0.07 | 0.03 | 0.00 | 0.00 | 0.03 |
| H2 | 10(1) | 0.05 | 0.28 | 0.12 | 0.30 | 0.00 | 0.15 | 0.01 | 0.39 | 0.40 | 0.22 | 0.22 | 0.19 |
| H2 | 13(2) | 0.00 | 0.07 | 0.00 | 0.74 | 0.28 | 0.35 | 0.24 | 0.00 | 0.36 | 0.25 | 0.25 | 0.23 |
| | | | | | | | | | | | | | 0.13 |
| H3 | 5(1) | 0.90 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 | 0.38 | 0.38 | 0.15 |
| H3 | 8(1) | 0.35 | 0.00 | 0.00 | 0.17 | 0.04 | 0.07 | 0.12 | 0.28 | 0.13 | 0.15 | 0.15 | 0.13 |
| H3 | 10(1) | 0.21 | 0.19 | 0.19 | 0.24 | 0.22 | 0.21 | 0.05 | 0.26 | 0.17 | 0.43 | 0.43 | 0.22 |
| H3 | 13(2) | 0.02 | 0.23 | 0.10 | 0.46 | 0.17 | 0.03 | 0.36 | 0.00 | 0.23 | 0.43 | 0.43 | 0.20 |
| | | | | | | | | | | | | | 0.18 |
| H4 | 5(1) | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.09 | 0.29 | 0.08 | 0.00 | 0.00 | 0.05 |
| H4 | 8(1) | 0.00 | 0.00 | 0.00 | 0.07 | 0.17 | 0.00 | 0.00 | 0.07 | 0.03 | 0.00 | 0.00 | 0.03 |
| H4 | 10(1) | 0.05 | 0.28 | 0.12 | 0.30 | 0.00 | 0.15 | 0.01 | 0.39 | 0.40 | 0.22 | 0.22 | 0.19 |
| H4 | 13(2) | 0.02 | 0.07 | 0.10 | 0.65 | 0.23 | 0.02 | 0.24 | 0.00 | 0.32 | 0.27 | 0.27 | 0.19 |
| | | | | | | | | | | | | | 0.12 |
| H5 | 5(1) | 0.00 | 0.13 | 0.00 | 0.00 | 0.00 | 0.00 | 0.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
| H5 | 8(1) | 0.16 | 0.07 | 0.13 | 0.07 | 0.27 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 0.07 |
| H5 | 10(1) | 0.08 | 0.18 | 0.14 | 0.07 | 0.00 | 0.09 | 0.01 | 0.17 | 0.01 | 0.03 | 0.03 | 0.08 |
| H5 | 13(2) | 0.02 | 0.11 | 0.45 | 0.37 | 0.43 | 0.09 | 0.02 | 0.21 | 0.47 | 0.38 | 0.38 | 0.25 |
| | | | | | | | | | | | | | 0.11 |

1. ()안은 사용된 트랜스포터의 수를 나타냄.

2. GAP = (휴리스틱목적값 - CPLEX 최적목적값) / CPLEX 최적목적값.

- 하역 시간 : (L+10) ~ (L+20)분
- 공차운행속도 : 250 m/분
- 블록운송속도 : 135 m/분
- $\alpha, \beta, \gamma = 1$

실험은 생성된 각각의 데이터에 대해 기존의 알고리즘(H1)과 Visual Basic으로 구현된 4가지 휴리스틱 알고리즘들의 결과를 CPLEX를 이용하여 구한 최적해와 비교·검증하였다.

80개의 실험문제들을 대상으로 소규모 문제(블록수 13개 이하)와 대규모 문제(블록수 15개 이상)로 나누어 5가지 휴리스틱 알고리즘들을 비교하였다. CPLEX로 최적해를 얻을 수 있는 5, 8, 10, 13개의 블록을 가지는 소규모 문제들에 대하여 비용 측면에서 최적해와의 비교결과를 보면, 휴리스틱 알고리즘 H1, H2, H3, H4, H5의 효율성은 각각 최적해와 비교하여 평균 18.0%, 13.0%, 18.0%, 12.0%, 11.0% 정도의 차이를 보여주고 있다(<표 1> 참조). 따라서, 휴리스틱 알고리즘 H5(혼합형 Look-Ahead 할당 알고리즘)가 평균적으로 가장 우수함을 알 수 있다. 또한, 휴리스틱 알고리즘 H4의 결과가 H1, H2, H3와 비교해서 더 우수한 결과를 보여주고 있다. 이는 트랜스포터와 블록 간의 더 많은 조합을 기준으로 평가하는 것이 더 효율적인 결과

를 제공하게 됨을 의미한다. 또한, 휴리스틱 알고리즘 H5와 같이 트랜스포터와 블록의 조합을 선택한 후 다음에 할당될 블록의 운송 완료시간을 함께 고려하는 Look-Ahead 방식은 더욱 우수한 결과를 보여준다. 블록수 15, 20, 30, 40을 갖는 대규모 문제들에 대해서는 CPLEX에 의한 최적해를 구할 수 없어 휴리스틱 알고리즘 중 가장 우수한 H5를 기준으로 나머지 4가지 휴리스틱 알고리즘과 비교해 본 결과, 평균 7%부터 13%까지 상대적으로 훨씬 높은 효율성을 보여주었다(<표 2> 참조). <표 2> 내에서 음수값을 갖는 수치는 해당 실험문제에 대해 해당 휴리스틱 알고리즘이 H5에 비해 상대적으로 우수한 해를 제공했음을 의미한다. 본 연구에서 사용한 실험문제들의 블록수송 요구정보들은 랜덤하게 생성됨에 따라 실험문제들의 구조적 특성에 의해, 일부분의 수송정보만을 이용하여 의사결정을 하는 휴리스틱 알고리즘의 결과가 최적해에 비해 상당히 나쁘게 나타나는 경우가 있었다. 그러나, 조선소 현장에서는 생산일정계획에 따른 실제 수송정보들이 계획기간에 걸쳐 비교적 순차적으로 잘 나열되어 있어, 휴리스틱 알고리즘의 적용결과는 매우 우수할 것으로 예상된다.

최적화 모형은 전형적인 NP-hard 문제이므로 실시간에 우수한 해를 얻는 것이 불가능하나, 휴리스틱 모형은 해를 찾는 계

표 2. H5 대비 휴리스틱 알고리즘 목적값 차이 비교

| 휴리스틱 | 블록 (TP) | GAP | | | | | | | | | | 평균 GAP |
|------|---------|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| H1 | 15(2) | -0.07 | 0.12 | -0.01 | 0.13 | 0.08 | 0.15 | 0.06 | 0.02 | 0.32 | 0.61 | 0.14 |
| H1 | 20(2) | 0.13 | 0.06 | 0.05 | -0.22 | 0.06 | 0.41 | 0.41 | 0.20 | 0.03 | -0.08 | 0.11 |
| H1 | 30(3) | 0.54 | 0.22 | 0.14 | 0.29 | 0.06 | -0.05 | -0.08 | 0.15 | 0.33 | 0.21 | 0.18 |
| H1 | 40(3) | 0.18 | 0.08 | 0.13 | 0.12 | -0.02 | 0.06 | 0.15 | 0.21 | 0.25 | -0.05 | 0.11 |
| | | | | | | | | | | | | 0.13 |
| H2 | 15(2) | 0.06 | 0.00 | 0.00 | 0.20 | 0.04 | 0.03 | 0.13 | 0.00 | -0.04 | 0.00 | 0.04 |
| H2 | 20(2) | 0.12 | 0.00 | 0.19 | 0.01 | -0.02 | 0.05 | 0.16 | 0.24 | 0.02 | 0.07 | 0.08 |
| H2 | 30(3) | 0.32 | 0.12 | 0.04 | 0.28 | 0.31 | 0.19 | 0.00 | 0.02 | 0.07 | 0.10 | 0.14 |
| H2 | 40(3) | 0.06 | 0.11 | 0.05 | 0.07 | 0.10 | 0.16 | 0.04 | 0.15 | 0.05 | 0.09 | 0.09 |
| | | | | | | | | | | | | 0.09 |
| H3 | 15(2) | -0.07 | 0.12 | -0.01 | 0.13 | 0.08 | 0.15 | 0.06 | 0.02 | 0.32 | 0.61 | 0.14 |
| H3 | 20(2) | 0.13 | 0.06 | 0.05 | -0.22 | 0.06 | 0.41 | 0.41 | 0.20 | 0.03 | -0.08 | 0.11 |
| H3 | 30(3) | 0.54 | 0.22 | 0.14 | 0.29 | 0.06 | -0.05 | -0.08 | 0.15 | 0.33 | 0.21 | 0.18 |
| H3 | 40(3) | 0.18 | 0.08 | 0.13 | 0.12 | -0.02 | 0.06 | 0.15 | 0.21 | 0.25 | -0.05 | 0.11 |
| | | | | | | | | | | | | 0.13 |
| H4 | 15(2) | 0.06 | 0.00 | 0.02 | 0.20 | 0.04 | 0.04 | 0.13 | -0.01 | -0.03 | 0.00 | 0.04 |
| H4 | 20(2) | 0.12 | 0.00 | -0.03 | 0.01 | -0.03 | 0.05 | 0.13 | 0.23 | 0.00 | 0.02 | 0.05 |
| H4 | 30(3) | 0.24 | 0.12 | 0.02 | 0.24 | 0.31 | 0.10 | -0.03 | 0.05 | 0.06 | 0.10 | 0.12 |
| H4 | 40(3) | 0.02 | 0.11 | 0.04 | 0.06 | 0.10 | 0.10 | 0.04 | 0.06 | 0.03 | 0.09 | 0.06 |
| | | | | | | | | | | | | 0.07 |

1. ()안은 사용된 트랜스포터의 수를 나타냄.

2. GAP = (휴리스틱목적값 - [H5]목적 값)/[H5]목적 값.

표 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 | 0 | 5,210 | 1 | 22 | 0 | 5,237 | 1 | 4,377 | 1 |
| 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 | 0 | 834 | 0 |

산시간이 무시할 수 있을 정도로 적어 실제문제에 적용 시 효율적인 일정계획의 수립이 가능하리라 여겨진다(<표 3> 참조). <표 3>은 가장 우수한 해를 제시하는 휴리스틱 알고리즘 H5를 기준으로 작성되었다. <표 3>에서 나타나듯이, 블록수가 10인 경우에 주철민 등(2006)의 결과에서는 최대 40시간(144,000초) 이상이 소요되었으나, 개선된 최적 모형의 적용에서는 최적해의 계산시간이 최대 5,725초로 나타났다. 비록 실험문제들이 다르더라도 동일한 성능의 컴퓨터에서 동일한 규모의 문제에 대한 최적해의 계산속도는 현저한 차이를 보임을 알 수 있었으며 이는 제약식 수와 의사결정변수 수의 감소에서 기인한다. <표 3>은 소규모 문제들에 대한 CPLEX와 휴리스틱 알고리즘 H5의 계산시간을 비교한 것이다. 모든 실험문제들에 대해 휴리스틱 알고리즘 H5는 1초 이하의 계산속도를 보여주고 있어, 문제의 규모(트랜스포터의 수 및 블록의 수)가 증가할수록 휴리스틱 알고리즘 H5의 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 트랜스포터를 운영하면서도 더욱 극대화할 수 있을 것으로 기대된다. 나아가 트랜스포터 운전자의 단말기를 통한 실시간 통신 시스템이 구축된다면 더욱 실질적이고 효율적인 트랜스포터의 운영을 기대할 수 있을 것이다.

5. 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 4가지 휴리스틱 알고리즘들을 제안하였다. 80개의 실험문제들을 대상으로 소규모 문제(블록수 13개 이하)와 대규모 문제(블록수 15개 이상)로 나누어 5가지 휴리스틱 알고리즘을 비교하였다. 소규모 문제들에 대해서는 휴리스틱 알고리즘 H5가 비용 측면에서 최적해에 비해 평균 11.0%의 차이를 보였고 계산시간 측면에서는 모든 대상문제들을 1초 내에 계산하는 탁월한 효율성을 입증하였다. 대규모 문제들에 대해서는 휴리스틱 알고리즘 H5가 나머지 4가지 휴리스틱 알고리즘에 비해 평균 7%부터 13%까지 상대적으로 훨씬 효율성이 높음을 알 수 있었다. 따라서, 문제의 규모가 증가할수록 휴리스틱 알고리즘 H5의 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 보다 적은 트랜스포터를 운영하면서도 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, *Ops. 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, *Opns. 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 O.R.*, 29, 33-51.
- Chung, K. H., Baek, T. H., Min, S. G., Kim, H. S., Park, J. C., Cho, K. K., and Park, C. K. (2001), Development of the Spatial Scheduling System and Its Applications in Shipbuilding Industry, *IE Interfaces*, 14(4), 394-402.
- 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, *Opns. 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, *Opns. Res.*, 46, 330-335.
- Ha, T. R., Moon, C. U., Joo, C. M., and Park, J. C. (2000), A Hybrid Genetic Algorithm for Scheduling of the Panel Block Assembly Shop in Shipbuilding, *Management Science*, 17(1), 135-143.
- 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.
- Koh, S. G., Park, J. C., Choi, Y. S., and Joo, C. M. (1999), Development of a Block Assembly Scheduling System for Shipbuilding Company, *IE Interfaces*, 12(4), 586-594.
- 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.
- Maxwell, W. L. and Muckstadt, J. A. (1982), Design of Automated Guided Vehicle Systems, *IIE Trans.*, 14, 114-124.
- Park, M. H., Lee, W. S., Ock, Y. S., and Lee, T. E. (1995), A Review of Korean Shipbuilding Industry and Industrial Engineering Research, *IE Interfaces*, 8(2), 5-20.
- 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. Opns. Res.*, 4, 285-305.



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: DK디엔씨(주) CS3팀
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



주철민

고려대학교 산업공학 학사
한국과학기술원 산업공학 석사
한국과학기술원 산업공학 박사
현재: 동서대학교 시스템경영공학과 교수
관심분야: 네트워크 이론, 일정계획, 시뮬레이션, 정보시스템