

타부탐색을 이용한 AGVS 일방향 흐름경로 설계

문영훈 · 서윤호[†]

고려대학교 산업시스템 정보공학과

Unidirectional AGVS Flowpath Design using Tabu Search

Young-Hoon Moon and Yoonho Seo

Department of Industrial Systems and Information Engineering, Korea University, Seoul, 136-701

AGV flowpath layout design is one of the most important steps for efficient AGV systems design. Since it was formulated by Gaskins & Tanchoco (1987), a unidirectional AGV flowpath layout design problem has been tackled by many researchers. However, the solution methods were traded off between the solution quality and the computational time. In this paper, a tabu search technique is applied to obtain a good solution for a relatively large problem in reasonable computational time. Specifically, fast construction algorithm for feasible initial solutions, long-term memory structure and neighbor solutions generation are adapted to the problem characteristics and embedded in the tabu search algorithm. Also, sets of computational experiments show that the proposed tabu search algorithm outperforms to the Ko and Egbelu's algorithm (2003).

Keywords: tabu search, unidirectional flowpath, AGV

1. 서론

다양한 시장의 요구에 부응하여 유연한 변화 대처 능력과 동시에 높은 효율이 요구되는 현대의 FMS (Flexible Manufacturing System)에서 AGV (Automated Guided Vehicle)의 사용은 제조시스템의 변경 유연성과 효율을 동시에 만족시킬 수 있는 대안이 될 수 있다. FMS의 성능은 물류시스템의 유연성에 크게 좌우되기 때문에, 1970년대 이후 AGVS는 FMS에서의 효율적인 물류시스템 대안으로 평가받으며 그 사용이 늘어가고 있는 추세에 있다. AGVS를 설계하는 데 AGV 대수, unit load 크기, pickup/drop-off station의 위치, 작업셀 배치, vehicle의 dispatching 전략, 제어시스템 등의 설계 및 결정이 필요하다. 이 중에 AGV 흐름경로(flowpath) 설계 문제는 AGVS의 성능을 결정하는 가장 중요한 요소 중 하나이다. 특히 많은 수의 AGV를 사용하는 제조시스템에서 제어시스템의 부하를 줄이기 위해 일방향 AGV 흐름경로망(unidirectional AGV flowpath network)의 사용이 요구되

며, 지난 20년 동안 이 문제를 해결하기 위해 많은 연구가 이루어졌다. 그러나 문제의 크기가 커질 때에도 좋은 해를 도출할 수 있는 방법론이 아직까지 개발되지 못한 상태이다.

일방향 흐름경로설계(Unidirectional Flow path Design; UFD) 문제는 Gaskins and Tanchoco(1987)가 처음으로 정수계획법을 이용하여 정식화하였고, Goetz(1990)와 Egbelu(1990)가 Gaskins and Tanchoco의 모형을 개선한 알고리즘을 개발하였다. 또한 같은 해 Gaskins and Tanchoco(1990)가 branch-and-bound 기법을 이용한 해법을 제시하였다. 정점-간선 그래프에서 각 간선의 방향을 탐색하며 최적의 가능한 네트워크를 찾아가는 수리모형과 이 모형을 이용한 알고리즘은 최적해를 구할 수 있는 반면, 계산 시간이 과다하게 소요되기 때문에 현실 문제에 적용시키기에는 많은 어려움이 있었다. Seo(1995)와 Egbelu(1995)는 이 문제에 대해 흐름경로 선택 개념을 적용한 혼합정수계획법 모형을 제시하였고 branch-and-bound 기법을 기반으로 한 휴리스틱 알고리즘을 개발하였다. 이 연구에서는 물류량이 존재하는 작업

본 연구는 한국과학재단 목적기초연구(과제번호: R01-2002-000-00232-0) 지원으로 수행되었음.

[†]연락처 : 서윤호 교수, 136-701 서울 성북구 안암동 1-5 고려대학교 산업시스템정보공학과, Fax : 02-929-5888,

E-mail : yoonhoseo@korea.ac.kr

셀 쌍을 흐름링크(flow link)라고 정의하고, 전체 물류비를 최소화하는 방향에서 각각의 흐름링크에 하나의 흐름경로를 선정하고, 선택된 흐름경로를 반영하여 일방향 네트워크를 설계하는 방법을 제시하였다. 그 후 최근에 Ko and Egbelu(2003)는 각 흐름링크에 최단거리경로를 설정하는 순서를 탐색하여 일방향 흐름경로망을 설계는 휴리스틱 알고리즘을 발표하였다. 그들의 논문에 따르면 알고리즘의 실행시간과 결과값이 그때까지의 다른 알고리즘에 비해 우수한 것으로 판명되었다. 그러나 이 알고리즘의 실행시간은 흐름링크의 개수에 의존적이기 때문에 문제 사이즈가 커짐에 따라 기하급수적으로 증가하며, 이 알고리즘의 pairwise interchange에서 이용하는 preprocess 알고리즘은 거의 같은 간선방향을 결정하기 때문에 다양한 해를 탐색하지 못하는 단점이 있다.

본 논문의 목적은 크기가 상대적으로 큰 UFD 문제에 대해 효율적인 해법을 제시하는 것이다. 이를 위하여 흐름경로 선택 문제의 새로운 수리모형을 개발하였고, 흐름경로 선택과 간선 방향의 탐색을 통한 타부탐색(Tabu Search) 알고리즘을 개발하였다. 제안된 알고리즘은 컴퓨터 실험을 통하여 Ko & Egbelu 알고리즘에 비하여 문제의 크기에 상관없이 목적함수 값에서 또한 20개 이상의 셀을 가진 문제에 대해서는 실행시간까지도 우수한 것으로 판명되었다.

2. 문제 정의

아래 <그림 1>에 6개 작업셀을 가진 AGV Guidepath Layout을 도식적으로 표현하였으며, 이는 무방향 AGV 흐름경로망으로 간주될 수 있다. 각 작업셀의 pickup과 drop-off station은 정점으로 표현된다. AGV 일방향 흐름경로 설계 문제는 무방향 AGV 네트워크와 각 작업셀 간의 물류량이 주어지면, 총 물류비를 최소화하는 흐름경로의 방향을 결정하는 것이다.

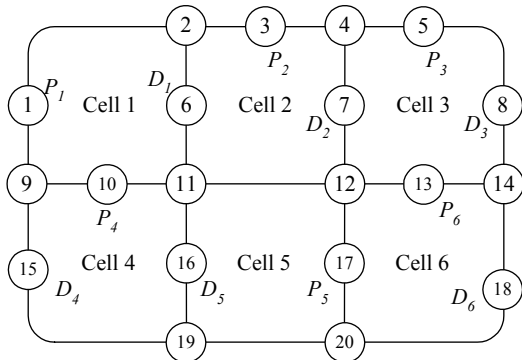


Figure 1. Example of an AGV Guidepath Layout.

UFD 모형은 무방향 네트워크의 간선 방향을 탐색하여 최소 물류비의 일방향의 가능한 AGV 흐름경로망을 찾는 간선탐색 방식(Gaskins and Tanchoco, 1987; Goetz and Egbelu, 1990)과 물류

가 존재하는 작업셀 간에 상호 방향이 일치하는 흐름경로를 선정하는 흐름경로 선택 방식으로 나눌 수 있다(Seo and Egbelu, 1995; Ko and Egbelu, 2003). 본 연구에서는 Seo and Egbelu(1995)이 제시한 흐름경로 선택 모형보다 개선된 수리모형을 제시하고, 흐름경로 선택과 간선 방향 탐색 방식을 조합한 휴리스틱 알고리즘을 개발한다. 수리모형 개발에 필요한 기호와 변수를 다음에 정의한다.

기호 정의:

- i, j : 작업셀 번호 인덱스, *i.e.* $i, j = 1, \dots, N$.
- (a, b) : 정점 a 와 b 를 연결하는 간선.
- p : 경로를 의미하는 인덱스.
- q_{ij} : 작업셀 i 와 j 간의 물류량.
- d_{ijp} : 작업셀 i 와 j 간의 경로 p 의 거리.
- W_{ab} : 간선 (a, b) 에 걸리는 최대 물류부하.
- ω_{ijp}^{ab} : 간선 (a, b) 가 셀 i 와 j 간 경로 p 에 포함된 경우 1, 그 이외 경우 0을 갖는 2진 계수.
- $y_{ijp} = \begin{cases} 1, & \text{작업셀 } i \text{와 } j \text{ 간 경로 } p \text{가 선정된 경우,} \\ 0, & \text{그 이외의 경우} \end{cases}$
- $z_{ab} = \begin{cases} 1, & \text{간선 } (a, b) \text{가 선정된 경우,} \\ 0, & \text{그 이외의 경우} \end{cases}$

Flowpath Selection Model for UFD:

$$\text{Minimize } \sum_{\substack{\forall i, j \\ i \neq j}} \sum_{\forall p} q_{ij} d_{ijp} y_{ijp}$$

subject to

$$\sum_{i, j} \sum_p q_{ij} \omega_{ijp}^{ab} y_{ijp} \leq W_{ab}, \tag{1}$$

$$\sum_p y_{ijp} = 1, \forall i, j, \tag{2}$$

$$z_{ab} + z_{ba} = 1, \forall (a, b), \tag{3}$$

$$\omega_{ijp}^{ab} y_{ijp} \leq z_{ab}, \forall i, j, p \text{ and } (a, b), \tag{4}$$

$$z_{ab} - z_{bc} = 0, z_{cb} - z_{ba} = 0, \forall b \text{ with degree of } 2, \tag{5}$$

$$z_{ab}, y_{ijp} \in \{0, 1\}, \forall i, j, p \text{ and } (a, b). \tag{6}$$

제약식 (1)은 간선에 추가되는 총 물류량을 제한하고, 제약식 (2)는 임의의 작업셀 i 와 j 간에 하나의 경로가 선택되는 것을 보장한다. 제약식 (3)은 간선의 일방향성을 보장하고, 제약식 (4)는 경로가 선택되기 위해서는 그 경로에 포함된 일방향 간선이 포함되어야 함을 의미한다. 제약식 (5)는 차수가 2인 정점의 한편 간선의 방향과 다른 한편 간선의 방향이 일치해야 함을, 제약식 (6)은 정수변수를 보장한다. 정점의 차수는 그 정점에 연결된 간선의 수를 의미한다.

본 모형은 모든 작업셀 쌍 각각에 대해 상호 방향이 일치하

는 하나씩의 흐름경로를 선택하고, 차수가 2인 노드에서의 일 방향 제약만 보장하면 가능한 일방향 물류망을 설계할 수 있다. 그러나 무방향 AGV 네트워크에서 작업셀 간에 정의 가능한 흐름경로의 수가 매우 클 수 있으므로 실제문제에 적용하는데 어려움이 있으므로 발견적 해법 개발이 필요하다.

3. 타부탐색 기반 해법 개발

본 논문에서는 제안된 UFD 모형의 특성을 적용하여 타부탐색 알고리즘을 개발한다. 타부탐색은 Glover(1989, 1990)에 의해 정형화되었으며, 시뮬레이티드 어닐링, 유전 알고리즘과 함께 복잡한 조합최적화 문제에 우수한 성능을 내는 메타 휴리스틱 기법이다. 타부탐색은 타부 리스트를 이용해 중복이나 순환을 피하며 이웃해를 검색하므로 지역해에 머무르지 않고 빠른 시간에 근사 최적해를 찾는 장점이 있다. 타부탐색과 같은 메타 휴리스틱에서 각종 결정모수와 구성요소를 문제의 특성에 맞게 설계하는 것이 알고리즘 성능을 높일 수 있는 요소이다. 타부탐색 알고리즘의 설계를 위해서는 초기해 생성 방법, 이웃해와 이동 방법, 타부 리스트의 크기, 장·단기 메모리 구조, 탐색 깊이와 최대 탐색 반복수 등의 결정이 중요하다.

본 연구에서는 초기해 생성, 이웃해의 이동과 장·단기 메모리 구조를 문제의 특성에 맞게 설계하였으며, 실험을 통해 타부 리스트 크기, 탐색 깊이, 최대 탐색 반복수의 적정 값을 결정한다.

기호 정의:

L_{bi}, L_{mi} : 각각 무방향, 양방향 Network.

$N_{cell}, N_{link}, N_{dep}, N_{rep}$: 각각 셀과 flow link 개수, 최대 탐색 깊이, 최대 탐색 반복수.

Q = 모든 물류흐름량의 집합, *i.e.*, $Q = \{q_{ij} \mid i, j = 1, \dots, N_{cell}, i \neq j, q_{ij} > 0\}$.

$LTM = N_{cell} \times N_{cell}$ 행렬, *i.e.*, LTM_{ij} = 흐름링크 i 에 우선순위 j 가 할당된 횟수.

$\Pi_{mi}, \Pi_{cur}, \Pi_{best}, \Pi(a,b)$ = 각각 초기해, 현재해, 최상해, 그리고 현재해에서 간선 (a,b) 의 방향을 바꿔 생성된 이웃해.

$Z(\Pi)$ = Π 의 목적함수 값.

$T = \{(a,b, Z((a,b))) \mid \text{방향이 바뀐 간선}(a,b) \text{와 그때 목적함수 값}\}$
 $Dijkstra(L_{cur}, p)$ = Network L_{cur} 에서 우선순위 p 인 흐름링크의 최단경로.

P_{ij} = 흐름링크 i 에 우선순위 j 가 할당될 확률.

Φ = 흐름링크 우선순위조합.

$\Phi(i)$ = 우선순위조합 Φ 에서 흐름링크 i 의 우선순위.

k, r = 각각 현재 탐색 깊이, 현재 탐색 반복수.

3.1 전체 타부탐색 알고리즘

타부탐색은 최선의 이웃해를 탐색하는 기본에 문제의 특성

을 반영한 알고리즘을 설계하는 것으로 해석할 수 있다. 주어진 해로부터 이웃해를 탐색하기 위한 inner-loop와 다양한 초기해를 탐색하기 위한 outer-loop로 구성된다. inner-loop에서 이웃해 탐색 시 지역해에 빠지는 것을 방지하기 위하여 최근 탐색된 해가 다시 탐색되는 것을 방지하기 위한 단기 메모리인 타부 리스트와 outer-loop에서 다양한 초기해를 구하기 위한 장기 메모리(long-term memory; LTM)를 이용한다.

본 논문에서는 흐름링크 각각에 최단거리경로 선택 우선순위를 부여하여 초기해를 구하고, 초기해에 의한 네트워크에서 간선의 방향을 바꾸는 방법을 통하여 이웃해를 찾는다. 이 방식은 간선탐색과 흐름경로 선택 개념을 조합한 것으로 좋은 해를 찾을 수 있는 기반이 된다.

UFD Tabu Search 알고리즘:

Step 1. 초기화: Q, L_{bi} 입력; N_{dep}, N_{rep} 결정; $r=1; LTM_{ij}=0$ 로 설정한다.

Step 2. 모든 흐름링크에 대해, 물동량 크기에 따라 우선순위 조합 Φ 를 정한다.

Step 3. Φ 로부터 초기해 Π_{mi} 와 $Z(\Pi_{mi})$ 를 구한다(초기해 생성 알고리즘).

$\Pi_{cur} = \Pi_{best} = \Pi_{mi}; Z(\Pi_{cur}) = Z(\Pi_{best}) = Z(\Pi_{mi}); k = 1$ 로 설정하고, LTM을 수정한다.

Step 4. 현재해 Π_{cur} 의 모든 간선 (a, b) 에 대하여, 이웃해 $\Pi(a, b)$ 를 탐색한다.

4.1. $\Pi^*(a,b) = \min_{\forall (a,b)} \{Z(\Pi(a,b))\}$ 인 해로 이동한다. 단 $\Pi(a,b)$ 는 $((a,b) \notin T, \text{또는 } (T_Z(a,b) > Z(\Pi(a,b)))$ 이고 $(a,b) \in T$ 이어야 한다.

4.2. $\Pi_{cur} = \Pi^*(a,b); T \leftarrow \Pi_{cur}$ 로 설정한다. 만약 $Z(\Pi_{cur}) < Z(\Pi_{best})$ 이면, $\Pi_{best} = \Pi_{cur}; Z(\Pi_{best}) = Z(\Pi_{cur})$ 로 설정한다.

4.3. 만약 $k < N_{dep}$ 이면, $k = k + 1$ 로 설정하고 step 4로 간다. 그렇지 않으면 step 5로 간다.

Step 5. 만약 $r > N_{rep}$ 이면 알고리즘을 멈추고 $L_{mi} \leftarrow \Pi_{best}$ 로 설정한다. 그렇지 않으면 LTM을 고려하여 우선순위조합 Φ 결정(흐름링크 우선순위 결정 알고리즘)하고, $r = r + 1$ 로 설정한 다음, step 3로 간다.

3.2 이웃해와 이동

현재 네트워크에서 간선의 방향을 바꿔 이동 가능한 이웃해를 구하고, 그 이웃해의 목적함수 값을 평가한다. 그 중 가능한 네트워크를 구성하며 목적함수 값의 개선이 최대인 이웃해를 찾는다. 빠른 시간에 좋은 이웃해를 찾기 위해 한 개의 간선 방향을 바꾸는 경우와 두 개의 간선을 동시에 바꾸는 경우를 모두 고려하였다. 네트워크의 가능성(feasibility)은 모든 노드 사이의 최단거리경로를 구할 수 있는 Floyd 알고리즘을 이용하여 검사될 수 있다(Aho *et al.*, 1974). 즉, 최단거리경로의 거리 합이 무한대가 되는 네트워크는 infeasible인 것으로 한다.

3.3 타부 리스트와 열망수준

타부 리스트는 이동된 해의 변경된 간선과 목적함수 값을 기록하는 자료구조이다. 따라서 현재해에서 간선의 방향을 바꿔찾은 이웃해가 타부 리스트에 있다면, 그 이웃해로의 이동은 사이클을 피하기 위해 제한된다. 그러나 과거에 타부해가 생성될 당시의 목적함수 값보다 현재 고려되고 있는 이웃해가 더 좋을 가능성이 있고, 그런 가능성을 배제하지 않기 위하여 열망 수준을 이용한다. 열망 수준은 타부 리스트에 기록될 때 이동되는 해의 목적함수 값으로 한다. 현재 이웃해가 타부 리스트에 있으나, 그 목적함수 값이 열망 수준보다 좋은 경우, 타부 제약이 해제되고 그 이웃해로의 이동이 가능하게 된다. 이때 타부리스트에 있는 목적함수 값도 따라서 수정된다. 타부 리스트의 크기는 실험을 통해서 inner-loop 횟수의 1/2로 정했다.

3.4 장기 메모리 구조

다양한 초기해 생성은 최적해를 찾을 가능성을 높여주고 탐색시간을 단축시킬 가능성을 높여준다. 흐름링크를 물류량이 있는 작업셀 쌍으로 정의하면, 흐름경로 선택 문제는 모든 흐름링크에 대하여 방향이 일치하는 경로를 구해 총 물류비를 최소화하는 일방향 흐름망을 설계는 것으로 해석될 수 있다. 다양한 초기해 생성을 위해 흐름링크의 우선순위를 LTM에 저장한다. LTM은 $N_{link} \times N_{link}$ 행렬로, LTM_{ij} 는 i 번째 흐름링크가 j 우선순위를 가진 횟수를 의미한다. 흐름링크의 우선순위를 구할 때, 각 흐름링크에 대한 우선순위가 골고루 분배될 수 있도록 LTM_{ij} 값의 역수 확률을 적용하여 각 흐름경로의 우선순위를 결정한다. 흐름링크 i 에 우선순위 j 가 할당될 확률 P_{ij} 는 식 (7)과 같이 정의되며, 흐름링크 우선순위 결정 알고리즘은 다음에 기술되어 있다.

$$P_{ij} = \left(\frac{\sum_i LTM_{ij}}{LTM_{ij}} \right) / \sum_i \left(\frac{\sum_i LTM_{ij}}{LTM_{ij}} \right) \quad (7)$$

흐름링크 우선순위 결정 알고리즘:

Step 1. $\Psi = \phi$ 로 설정한다.

Step 2. $i=1$ 부터 N_{link} 까지, step 3~5를 반복한다.

Step 3. $r=U(0, 1)$ 분포를 따르는 임의의 수.

Step 4. $j=1$ 부터 $N_{link}-1$ 까지 중에서 $\sum_{k=1}^j P_{ik} \leq r < \sum_{k=1}^{j+1} P_{ik}$ 인 j 를 찾는다.

Step 5. 만약 $j \in \Psi$ 이면, step 3로 간다. 그렇지 않으면 $\Phi(i) = j$, $\Psi = \Psi \cup j$ 로 설정하고 step 2로 간다.

3.5 초기해 생성

흐름링크의 우선순위 Φ 가 주어지면 순서에 따라 흐름링크에

대한 최단거리경로를 구하고 그 경로에 포함된 간선 방향을 네트워크에 반영한다면 초기 가능해를 구할 수 있다. 초기해 생성 알고리즘은 다음과 같다.

초기해 생성 알고리즘:

Step 1. $\Pi_{ni} = L_{ni}$; $i=1$ 로 설정한다.

Step 2. $\Pi_{ni} \leftarrow Dijkstra(\Pi_{ni}, \Phi(i))$; $i=i+1$ 로 설정한다.

Step 3. 모든 간선의 방향이 설정되거나, $i > N_{link}$ 이면 알고리즘을 끝낸다. 그렇지 않으면 step 2로 간다.

3.6 파라미터 결정

빠른 시간에 우수한 해를 얻기 위해 컴퓨터 실험을 통하여 최대 탐색 깊이 N_{dep} 와 최대 탐색 반복 횟수 N_{rep} 를 결정한다. 실험 결과에 대한 선형회귀 분석을 통하여 반복 횟수 N_{dep} 와 N_{rep} 를 결정하기 위한 선형회귀식을 유도한다.

이웃해 탐색은 네트워크의 간선의 방향을 바꾸는 것이므로 inner-loop의 횟수는 간선 개수에 의존적이며, 간선 수는 작업셀 수에 비례하므로 결정변수를 작업셀 개수로 하여 회귀식을 유도한다. 컴퓨터 실험은 작업셀 6~15개 문제에 대하여 반복 횟수를 10~50회로 변화시켜가며 각각에 대해 10회씩 실험하였고, 가장 좋은 해를 찾는 최소 inner-loop 횟수에 대해 회귀식을 구했다. 이 때의 outer-loop의 횟수가 영향을 끼치지 못하게 하기 위해 outer-loop의 횟수는 충분히 큰 70회로 고정시킨 상태에서 실험을 하였으며, 유도된 inner-loop 횟수 결정을 위한 회귀식은 다음과 같다.

$$N_{dep} = 2.33 \times N_{cell} + 3.73 \quad (8)$$

초기해는 흐름링크에 흐름경로를 선택하는 순서에 따라 생성되기 때문에 outer-loop의 횟수는 흐름링크 수에 의존적이다. 흐름링크 수는 작업셀 수의 제곱에 비례하므로 결정변수를 흐름링크 수로 하였다. 이 실험에서도 inner-loop 횟수는 충분히 고정시켰으며, 셀 6~15개 문제에 대하여 outer-loop 횟수를 10~70회로 변화시켜가며 각각에 대해 10회씩 실험하여 가장 좋은 해를 찾는 최소 outer-loop 횟수에 대해 회귀식을 구했으며, 유도된 outer-loop 횟수 결정을 위한 회귀식은 다음과 같다.

$$N_{rep} = 0.45 \times N_{link} + 24 \quad (9)$$

일반적으로 타부탐색 알고리즘의 실행시간은 N_{dep} 와 N_{rep} 의 곱에 비례한다. 따라서 파라미터 N_{dep} 와 N_{rep} 의 적정값 설정은 우수한 해를 구하기 위해 필요하다.

4. 수치실험 및 분석

본 논문에서 제안된 타부탐색 알고리즘의 성능을 알아보기 위하여 최근 개발된 Ko and Egbelu (2003) 알고리즘과 비교해 보았

다. Ko & Egbelu 알고리즘은 preprocess, complete 그리고 pairwise-interchange 알고리즘으로 구성 되어 있다. Ko & Egbelu 알고리즘은 흐름링크 우선순위 조합에 대한 pairwise-interchange를 통하여 해의 개선을 기대하기 때문에 실행시간이 N_{link}^2 에 비례한다. 반면 본 논문에서 제시한 타부탐색 알고리즘의 실행시간은 식 (8, 9)에서 보듯이 $N_{link} \times N_{cell}$ 에 비례한다. $N_{link} \cong (N_{cell}-1) \times N_{cell}$ 이므로 문제 크기가 커질수록 본 논문에서 제안된 알고리즘이 더 빠른 시간에 해를 구할 수 있을 것으로 예상된다. 본 알고리즘은 Visual C++ 6.0으로 구현되었으며, Pentium 4 프로세서 2.6GHz를 가진 PC에서 실험하였다.

4.1 실험결과 및 분석

50에서 100 사이의 유니폼 분포 내 임의 물류량과 셀 개수 4, 6, 9, 12, 20, 30 문제에 대해 각각 10번씩 실험을 수행하여, 본 연구에서 제안된 TS 알고리즘과 Ko & Egbelu 알고리즘의 실행 결과를 아래 <표 1>에 비교, 요약하였다.

<표 1>에서 보듯이 실행시간은 Ko & Egbelu 알고리즘과 비교하여 셀 4, 6, 9, 12개 문제에 대하여 각각 550, 69, 11, 4배 증가하였으나, 총 물류비에 있어서는 약 10% 정도 우수한 일방향 흐름망을 설계할 수 있었다.

그러나 문제 크기가 커질수록 총 물류비에서의 개선율은 9~11% 정도로 일정하게 유지되는 반면 실행 시간에서의 차이는 급격히 줄어들어 셀 20개 이상인 문제에서는 본 논문에서 제안한 TS 알고리즘이 더 빠른 시간에 해를 구할 수 있었다. 아래 <그림 2>에 문제 크기에 따른 알고리즘 실행시간과 <그림 3>에 목적함수 값을 비교하였다.

Ko & Egbelu 알고리즘에서는 모든 가능한 흐름링크 우선순위 조합을 탐색하는 반면, 본 논문의 알고리즘은 장기 메모리를 이용하여 다양한 초기해를 구할 수 있기 때문에 실행시간에서의 개선이 있는 것으로 해석된다. 또한 간선 방향을 변화시켜 이웃해를 탐색하기 때문에 최적해를 발견할 가능성이 커지며, 따라서 Ko & Egbelu 알고리즘에 비해 목적함수 값에서의 10% 이상의 개선효과가 있었다.

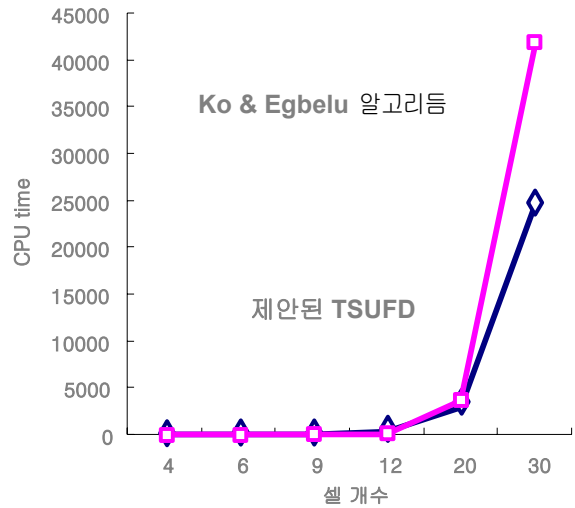


Figure 2. 실행시간 비교.

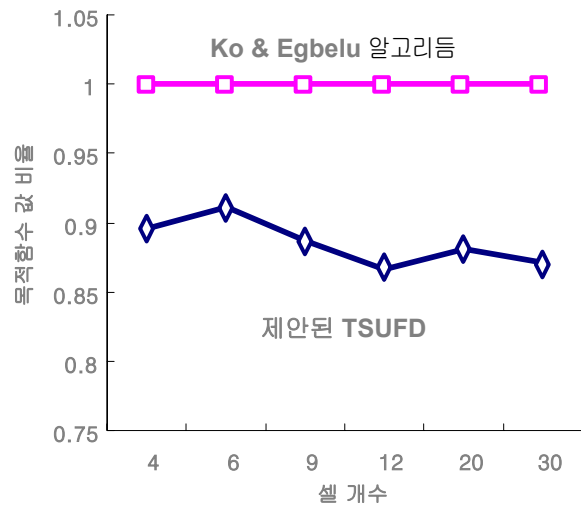


Figure 3. 목적함수 값 비교.

Table 1. Experimental results.

셀 개수	Proposed TS 알고리즘		Ko & Egbelu		평균 총 물류비 개선율 (%)	실행시간 증가율(times)
	평균 총 물류비	실행시간 (sec.)	평균 총 물류비	실행시간 (sec.)		
4	8,497	1.1	9,490	0.01	10.5	110
6	24,337	6.9	26,720	0.1	8.9	69
9	64,387	56.7	72,619	4.5	11.3	12.6
12	145,039	232	167,399	55	13.4	4.2
20	512,963	2,863	582,501	3,633	11.9	0.79
30	985,987	24,472	1,132,385	41,468	12.9	0.59

5. 결 론

본 연구는 일방향 AGV 흐름경로 설계를 위한 수리모형을 개발하였고, 기존 연구의 장점과 문제의 특성을 반영한 타부탐색 알고리즘을 개발하였다. 알고리즘의 성능 비교를 위해 Ko and Egbelu의 논문을 참조하였다. 비교 결과 목적함수 값 총 물류비에서 약 10% 이상 개선된 해를 구할 수 있었고, 셀 개수가 20개 이상인 문제에서는 실행시간도 더 빠른 것으로 분석되었다.

본 연구에서 제시된 타부탐색 알고리즘은 문제의 크기가 큰 도로 설계, 일방향 AGV 흐름망 설계 및 컨베이어 설계 등에 적용될 수 있을 것이다.

참고문헌

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974), *The Design and Algorithm of Computer Algorithms*, Addison Wesley.

- Gaskins, R. J. and Tanchoco, J. M. A. (1987), Flow path design for automated guided vehicle systems, *International Journal of Production Research*, 25, 667-676.
- Glover, F. (1989), Tabu search-Part I, *ORSA Journal on Computing*, 1, 190-206.
- Glover, F. (1990), Tabu search-Part II, *ORSA Journal on Computing*, 2, 4-32.
- Goetz, W. G., Jr. and Egbelu, P. J. (1990), Guide path design and location of load pick-up/drop-off points for an automated guided vehicle system, *International Journal of Production Research*, 28, 927-941.
- Kaspi, M. and Tanchoco, J. M. A. (1990), Optimal flow design of unidirectional AGV systems, *International Journal of Production Research*, 28, 1023-1030.
- Ko, K.-C. and Egbelu, P.J. (2003), Unidirectional AGV Guidepath Network Design: A Heuristic Algorithm, *International Journal of Production Research*, 41, 2325-2343.
- Seo, Y. and Egbelu, P. J. (1995), Flexible guidepath design for automated guided vehicle systems. *International Journal of Production Research*, 33, 1135-1156.



문영훈

고려대학교 산업시스템 정보공학과 학사
현재: 고려대학교 산업시스템 정보공학과 석사과정
관심분야: 생산 정보 시스템, Heuristics



서윤호

고려대학교 산업공학과 학사
Pennsylvania State Uni. 산업공학 석사, 박사
현재: 고려대학교 산업시스템 정보공학과 부교수
관심분야: 제조, 조립 및 물류 시스템의 VR 적용 지능 설계