KIGAS Vol. 21, No. 6, pp 61~69, 2017
(Journal of the Korean Institute of Gas)
https://doi.org/10.7842/kigas.2017.21.6.61

플랜트 사고 대응 훈련을 위한 탈출 및 조치 경로 설계 기법 개발

†김형진 · 박찬국 · 이재용 · 이춘식
고등기술연구원 플랜트엔지니어링본부
(2017년 10월 17일 접수, 2017년 2017년 12월 26일 수정, 2017년 12월 27일 채택)

Development of Escape and Rescue Path-taking Method for Plant Accident Response Training

†**Hyoung Jean Kim · Chan-Cook Park · Jae Yong Lee · Chun Sik Lee**
Institute for Advanced Engineering, 175-28 Goan-ro 51beon-gil Baegam-myeon Cheoin-gu Yongin-si Gyeounggi-do, 449-863, Korea
(Received October 17, 2017; Revised December 26, 2017; Accepted December 27, 2017)

요 약

플랜트 사고 발생 시 현장 운전원, 제어실 운전원 및 소방관 등이 취해야 할 가장 중요한 사항은 사고 현장으로부터의 탈출과 사고 현장으로의 사고 진압 처리를 위한 진입일 것이다. 이 두 가지 중요한 행동은 서로 상반된 방향으로 진행해야 하는 조치이며, 사고 대응 이동 경로에 대한 훈련을 평상시에 훈련함으로써, 사고의 확산을 방지하고 효율적인 사고 대응을 할 수 있다. 이와 같은 필요성에 의해 본 연구에서는 플랜트 사고 대응 훈련을 위한 탈출 및 조치 경로 설계 기법을 개발하였다. 활용 방안으로는, 운전원 및 소방관들의 사고 발생 시점의 플랜트 내 실시간 위치로부터 사고 탈출 및 조치 경로를 계산하여 플랜트 안전훈련시스템에 이동 경로 정보를 제공함으로써, 안전 훈련 시나리오에 적용하여 현실적이고 효과적인 훈련 효과를 제공할 수 있을 것으로 기대된다.

Abstract - In case of plant accident, the most important measures that field operators, control-room operators and fire fighters must take are the escape from and going into the accident sites. These two different actions are reverse directional moving actions. By training operators and fire fighters with counter-accident path taking measurements, we can prevent the small accidents from becoming large-scale accidents, and can take efficient measurements in case of actual plant accidents. Out of necessities of path-taking training, in this research, we developed the escape and rescue path-taking method for plant accident response training. We can calculate the escape and rescue routes from a operator or fire fighter's current location as of accident happening and provide route data which in turn can be used as the safety training scenario. We expect this path-taking method can enhance the effectiveness and reality of escape and rescue training scenarios.

Key words : plant, training, path, escape, rescue

I. 서 론

가스 누출, 화재, 폭발과 같은 공정 플랜트 사고 발생 시 현장 운전원, 제어실 운전원 및 소방관 등은 신속 정확한 조치를 취함으로써 인명과 재산 손실을 방지할 수 있다. 이를 위해 가장 기본적인 요건은 사고 현장으로부터의 탈출과 사고 현장으로의 사고 진압 처리를 위한 진입이라고 볼 수 있다. 이 두 가지 중요한 행동은 서로 반대되는 방향으로 진행해야 하는 조치이지만 유사한 이동 경로를 가진다는 점에서는 유사하다. 사고 대응을 위한 이동 경로를 기반으로 안전 훈련을 평상시에 실행함으로써, 사고의 확산을 방지하고 효율적인 사고 대응을 할 수 있기 때문에 본 연구에서는 플랜트 사고 대응 훈련을 위한 탈출 및 조치 경로 설계 기법을

†Corresponding author:hyoung@iae.re.kr

개발하였다.

복잡한 플랜트 내부 위험 장비 배치 및 이동 가능 경로, 급박한 사고 상황, 평상시 플랜트 내부 구조에 익숙하지 않은 소방관 등의 환경 하에서는 운전원과 소방관들이 우왕좌왕함으로써 탈출 및 조치를 위한 이동이 신속 정확하게 최적의 경로로 진행될 수 없기 때문에, 긴급 상황에서의 소중한 시간을 낭비할 수 있으며 그 과정에서 인적, 물적 피해가 커질 수 있다.

운전원 및 소방관들의 사고 발생 시점의 플랜트 내 실시간 위치로부터 사고 탈출 및 조치 경로를 계산하여 플랜트 안전훈련시스템에 이동 경로 정보를 제공함으로써, 안전 훈련 시나리오에 적용하여 현실적이고 효과적인 훈련 효과를 제공할 수 있을 것으로 기대된다.

Ⅱ. 최단 경로 선행 연구

본 연구의 핵심 로직인 최단 경로 알고리즘은 선행 연구 사례가 있다. 대표적으로 다익스트라 (Dijkstra) 알고리즘, 벨먼-포드 (Bellman-Ford) 알고리즘, A* (A Star) 알고리즘, 플로이드-와셜 (Floyd-Warshall) 알고리즘, 존슨 (Johnson) 알고리즘, 비터비 (Viterbi) 알고리즘[1]-[7] 등이 있으며 이들을 비교하면 Table 1과 같다.

다익스트라 알고리즘[2]은 Fig. 1과 같이 출발점으로부터 목표점까지의 모든 방향에 걸쳐 거리 (weight) 계산을 하여 최단 경로를 찾는다. 여기서 'weight'란 일반적인 사전적 의미에서 '가중치'를 뜻하며, 경로 문제에서는 weight가 높으면 그만큼 거리 (distance)가 멀거나 비용 (cost)이 많이 드는 등 경로를 통과하기 어렵다는 의미이다. 특정 지점에서 이동 가능한 모든 방향의 경우 수를 계산하기 때문에 계산량이 많고 계산 속도가 느리다.

Table 1. Comparisons of shortest path algorithms

알고리즘명	문제 유형	특징
Dijkstra	single source	양수 weight
Bellman-Ford	single source	양수/음수 weight
A*	single pair	계산 속도 향상을 위한 heuristic 사용
Floyd-Warshall	all pairs	
Johnson	all pairs	Floyd-Warshal보다 빠른 계산 속도
Viterbi	stochastic path	additional probabilistic weight

벨먼-포드 알고리즘[3]은 음수 (-, negative)의 거리를 고려할 수 있다는 점에서 다익스트라 알고리즘과 다르고 나머지는 동일하다. 즉, 다익스트라 알고리즘은 일방 통행만 가능하지만, 벨먼-포드 알고리즘은 양방향 통행이 가능하므로 더 현실적인 경로 설정이 가능하다.

A 스타 알고리즘[4]은 Fig. 2와 같이 출발점으로부터 목표점까지의 직선 방향선을 중심으로 최단 거리에 해당될 확률이 낮은 부분인 직선 방향성에서 멀리 떨어진 지점은 계산에서 제외하고, 최단 거리에 해당될 확률이 높은 부분인 직선 방향선 주위에 위치한 지점을 중심으로 거리 (weight) 계산을 하다가 장애물에 부딪히면, 그 지점에서부터 나

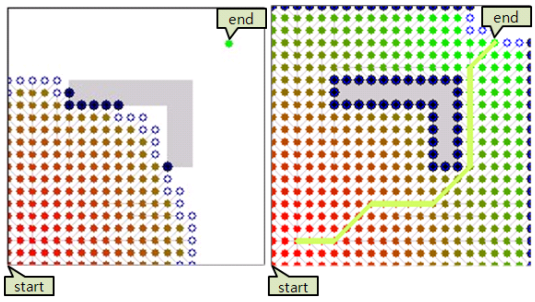

Fig. 1. Shortest path finding with Dijkstra algorithm.

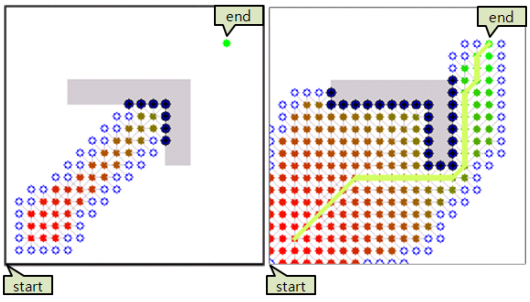

Fig. 2. Shortest path finding with A* algorithm.

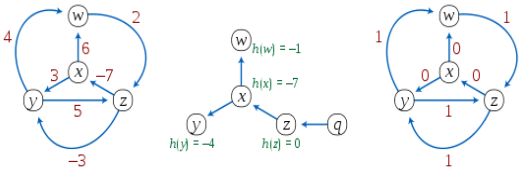

Fig. 3. Johnson algorithm.
(a) original graph with negative edges
(b) shortest path tree found by Bellman-Ford algorithm
(c) reweighted graph with no negative edges

머지 탐색하지 않은 지점에 대해 계산한다. 즉, 다익스트라 알고리즘은 모든 지점에 대해 계산을 하는 반면, A 스타 알고리즘은 최단 거리에 해당될 확률이 높은 부분만 선별해서 계산함으로써, 계산량과 계산 시간을 절약해주기 때문에 다익스트라 알고리즘에 비해 효율적인 알고리즘이다.

플로이드-와셜 알고리즘[5]은 계산 결과로서 특정 최단 경로 선정은 불가능하고 경로 길이의 합계만 계산 가능하다. 또한, 모든 경로가 연결되었을 때만 경로 길이 계산이 가능하다. 즉, 다익스트라 알고리즘은 현실 상황과 동일하게 특정 두 지점을 연결하는 경로가 단절되거나 존재하지 않는 경우(sparse graph)도 계산이 가능하지만, 플로이드-와셜 알고리즘은 모든 지점을 연결하는 모든 경로가 존재 (dense graph)한다고 가정하여 경로 길이 합계를 계산하므로, 현실적인 상황을 반영하기 어렵고, 다익스트라 알고리즘에 비해서 계산량과 계산 시간이 많이 드는 비효율적인 알고리즘이라고 볼 수 있다.

존슨 알고리즘[6]은 벨먼-포드 알고리즘과 다익스트라 알고리즘을 조합한 알고리즘으로, 음수 (-, negative) 거리 (weight)를 가진 경로 문제를 벨먼-포드 알고리즘을 적용해서 음수 거리를 제거한 경로 문제로 변환한 후, 다익스트라 알고리즘을 적용할 수 있도록 한다.

비터비 알고리즘[7]은 가장 발생 확률이 높은 은닉 상태 시퀀스를 찾기 위한 다이나믹 프로그램밍 알고리즘으로, 결과적으로 은닉 마르코프 모델에서 이벤트 발생 시퀀스를 찾아낸다. 주로 통신 분야에서 적용되어 왔으며, 최근에는 음성 인식 분야에서도 활용된다.

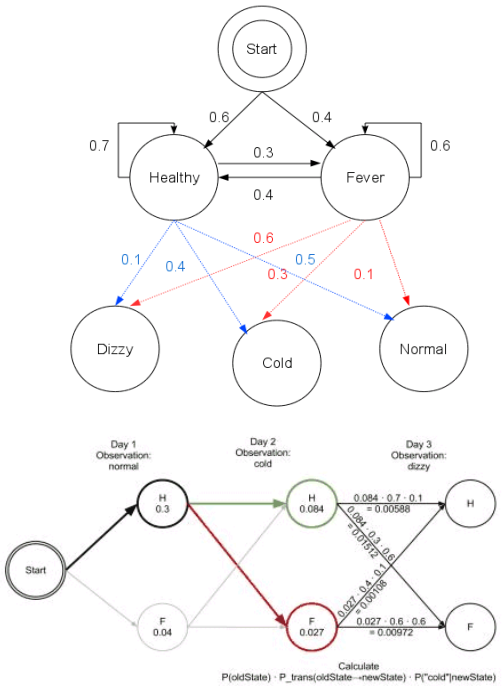

Fig. 4. Viterbi algorithm.
(a) Given hidden Markov model
(b) Viterbi algorithm result of the most likely path

대표적인 최단 경로 알고리즘 중에서 다양한 분야에 응용할 수 있는 범용성이 높고, 자동차 네비게이션과 같은 일상 생활에서 활용 사례를 쉽게 찾아볼 수 있으며, 구현이 가장 용이하고 일반적이며 기본적인 다익스트라 알고리즘을 본 연구에 적용하였다. 다익스트라를 비롯한 최단 경로 알고리즘은 네비게이션, 통신 분야 등에서 널리 응용되고 있으나, 플랜트 안전 분야에서 연구개발이나 활용 사례로 보고된 경우는 모바일 어플리케이션을 이용한 최적 대피경로 설정에 관한 Cho[10]의 논문이 거의 유일하다고 할 수 있다.

Ⅲ. 탈황 공정 플랜트 구성 및 적용

본 알고리즘 적용 대상으로 석유화학 공정에서의 탈황 공정 유닛 (RDS Unit, Residue Desulfurization Unit)을 선택하였다. 탈황 공정 유닛은 크게 Fig. 5와 같은 반응기와 Fig. 6과 같은 열교환기로 이루어져 있으며 Fig. 7과 같이 반응기 topside에

Fig. 5. Reactor of Residue Desulfurization (RDS) Unit.

Fig. 6. Heat exchanger of Residue Desulfurization (RDS) Unit.

Fig. 7. Topside of Reactor of Residue Desulfurization (RDS) Unit.

있는 플랜지의 상부 및 하부 또는 Fig. 6의 열교환기 shell side 및 tube side에서 누출, 화재와 같은 안전 사고 발생 사례가 보고되고 있다.

본 연구 적용 대상인 탈황 공정은 동일한 2개의 트레인 (train)으로 구성되어 있으며, 1개의 트레인은 Fig. 8과 같은 레이아웃 (layout)으로 배치되어 있으며, 주요 장비로는 Table 2와 같이 5개의 반응

Table 2. Leak and fire locations of equipments in Residue Desulfurization (RDS) Unit

train	equipment	number of equipment	location of leak, fire	number of location of leak, fire
1	reactor	5	upper flange	1
			lower flange	1
	heat exchanger	2	shell side	2
			tube side	2
2	reactor	5	upper flange	1
			lower flange	1
	heat exchanger	2	shell side	2
			tube side	2

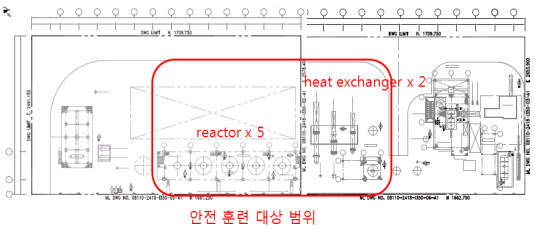

Fig. 8. Layout of Reactor of Residue Desulfurization (RDS) Unit train 1.

기와 2개의 열교환기로 이루어져 있다. 각 반응기는 위와 아래 부분에 각각 1개씩 부착되어 있는 플랜지 합계 2군데에서 누출, 화재가 주로 발생하며, 열교환기는 shell side 및 tube side 2군데씩, 합계 4군데에서 누출 및 화재가 주로 발생한다. 따라서, 본 연구에서 가정한 2개 트레인의 누출, 화재 발생 지점 개수 총 합계는 (10 +8) x 2 = 32 군데이다.

다익스트라 알고리즘 적용을 위해서는 Fig. 9와 같은 그래프를 설정해야 한다. 즉, vertex, edge, weight 3개 요소를 설정해야 한다. 플랜트 환경으로 변환해보면 vertex는 운전원이나 장비의 위치, edge는 이동 경로, weight는 이동 거리이다. Vertex는

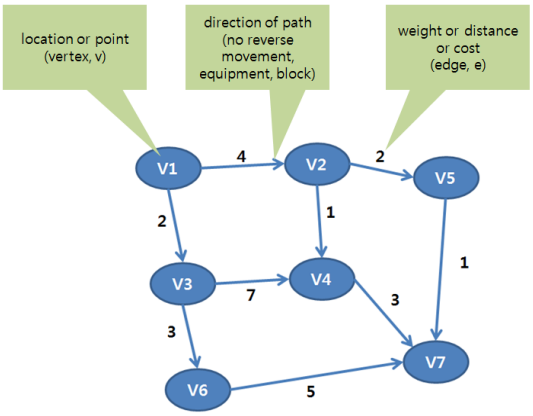

Fig. 9. Graph configuration of Dijkstra algorithm.

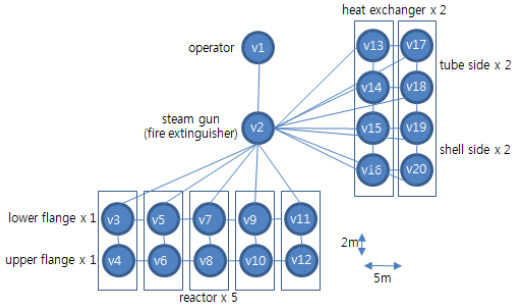

Fig. 10. Simplified layout of Residue Desulfurization (RDS) Unit.

장비는 고정 위치, 운전원이나 소방관은 이동 가능한 위치이나, 본 연구에서는 운전원과 소방관도 고정 위치로 가정하여 연구를 수행하였다. 향후에는 운전원과 소방관의 실시간 이동 상황을 반영한 실시간 이동 위치를 기반으로 본 알고리즘을 개발할 수 있다. 모든 vertex와 그외의 모든 vertex간에 edge가 존재하는 것이 아니고, 일부의 vertex간에만 edge가 존재한다. 플랜트 환경에서도 이동 경로가 모든 경우에 존재하지 않으며, 이동이 가능한 경로가 일부 존재하는 것과 유사하다. 이동 거리는 실제 플랜트 환경에서의 거리와 최대한 유사하게 설정하면 된다. 본 연구에서는 2차원 평면으로 가정하였으나, 향후에는 3차원 거리를 반영하여 연구를 수행할 수 있다. 반응기는 높이가 20m 정도이며 사고가 발생하는 지점은 반응기 topside에 있는 플랜지이므로, 평지에서 topside까지의 20m 정도의 거리 이동을 알고리즘에 반영할 수 있다.

Table 3. Weight/distance/cost of Residue Desulfurization (RDS) Unit path

vertex	location of leak, fire
v1	operator
v2	steam gun (fire extinguisher)
v3	lower flange of reactor 1
v4	upper flange of reactor 1
v5	lower flange of reactor 2
v6	upper flange of reactor 2
v7	lower flange of reactor 3
v8	upper flange of reactor 3
v9	lower flange of reactor 4
v10	upper flange of reactor 4
v11	lower flange of reactor 5
v12	upper flange of reactor 5
v13	tube side 1 of heat exchanger 1
v14	tube side 2 of heat exchanger 1
v15	shell side 1 of heat exchanger 1
v16	shell side 2 of heat exchanger 1
v17	tube side 1 of heat exchanger 2
v18	tube side 2 of heat exchanger 2
v19	shell side 1 of heat exchanger 2
v20	shell side 2 of heat exchanger 2

탈황 공정 유닛 환경에서 운전원이나 소방관이 사고 발생 시 탈출하거나 대응하기 위한 최적 이동 경로를 사전에 안전 훈련에 적용하기 위한 다익스트라 알고리즘 적용을 위해서는 플랜트 레이아웃의 단순화가 필수적이므로, Fig. 10과 같이 단순화된 탈황 공정 플랜트 레이아웃 모델을 구성하였다.

Fig. 10의 레이아웃을 기준으로 Table 3과 같이 다익스트라 알고리즘의 vertex를 결정하였다. Vertex 1은 운전원이나 소방관의 현재 위치, vertex 2는 steam gun이나 소화기와 같은 화재 진압 장비,

Table 4. Weight/distance/cost of Residue Desulfurization (RDS) Unit vertex

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17	v18	v19	v20
v1	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v2	∞	∞	25	∞	20	∞	10	∞	8	∞	10	∞	25	10	8	10	25	20	20	25
v3	∞	25	∞	2	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v4	∞	∞	2	∞	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v5	∞	20	5	∞	∞	2	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v6	∞	∞	∞	5	2	∞	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v7	∞	10	∞	∞	5	∞	∞	2	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v8	∞	∞	∞	∞	∞	5	2	∞	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
v9	∞	8	∞	∞	∞	∞	5	∞	∞	2	5	∞	∞	∞	∞	∞	∞	∞	∞	∞
v10	∞	∞	∞	∞	∞	∞	∞	5	2	∞	∞	5	∞	∞	∞	∞	∞	∞	∞	∞
v11	∞	10	∞	∞	∞	∞	∞	∞	5	∞	∞	2	∞	∞	∞	∞	∞	∞	∞	∞
v12	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	2	∞	∞	∞	∞	∞	∞	∞	∞	∞
v13	∞	25	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2	∞	∞	5	∞	∞	∞	∞
v14	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	2	∞	2	∞	∞	5	∞	∞	∞
v15	∞	8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2	∞	2	∞	∞	5	∞	∞
v16	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2	∞	∞	∞	∞	∞	5
v17	∞	25	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	∞	∞	2	∞	∞	∞
v18	∞	20	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	2	∞	2	∞	∞
v19	∞	20	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	2	∞	2	∞
v20	∞	25	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	2	∞	

vertex 3~12는 5개 반응기 topside에 부착된 상·하부 플랜지이며, vertex 12~20은 2개 열교환기의 shell and tube side이다.

Fig. 8 및 Fig. 10의 레이아웃을 기준으로 Table 4와 같이 다익스트라 알고리즘의 weight/distance/cost를 결정하였다. Table 4은 양방향 edge의 weight를 모두 게재하였지만, 계산 시에는 대각선을 기준으로 우측 혹은 좌측 중 한 부분만 입력하면 된다. 최단 경로 계산을 위한 플랜트 형상 및 거리 데이터 입력 시에 Table 4의 모든 값을 입력할 필요 없이 우측 또는 좌측의 데이터 등 전체 입력 데이터 분량의 1/2만 입력하면 된다. 우측은 정방향, 좌측은 역방향이므로 본 연구와 같이 정방향만을 고려하는 경우는 우측 부분만 입력하면 된다. 본 연구에서는 distance가 여기에 해당된다. Fig. 5의 현장 반응기 사진을 보면 반응기와 반응기 사이의 간격을 대략 5m 정도로 가늠해볼 수 있으며, Fig. 7의 현장 플랜트 사진을 참고하여, 상부 및 하부 플랜지간 간격을 2m로 가정하였다. 마찬가지로 열교환기간의 간격을 5m, shell & tube side간 간격을 2m로 가정하였다. 물론, 현장에서의 측정 거리와는 약간의 차이가 발생할 수 있으나, 단순화된 레이아웃 모델기반의 알고리즘 검증이라는 연구 목적을 위해서는 대략적인 거리로 가정하여 적용할 수 있을 것이다.

운전원이나 소방관은 반드시 소방 장비를 먼저 확보한 후에 사고 장소로 이동하도록 설계하였다. 따라서, v1은 반드시 v2로밖에 이동할 수 없도록 단일 경로로 구성하였다.

플랜트 현장에서 반응기 플랜지에 접근하려면 20m 상당의 높은 반응기 topside까지 먼저 이동해야 하지만, 단순화 모델에서는 이와 같은 3차원 이동 거리는 고려하여 않고 2차원적 평면만으로 가정하였기 때문에 현장 실물과의 차이점이 발생한다.

상 · 하부 반응기 플랜지는 동일한 반응기 topside에 위치하기 때문에 운전원이나 소방관의 이동이 단일 반응기로 이동하도록 설정하였다. 반면에 열교환기는 지상에 위치하므로 shell & tube side에 독립적으로 이동할 수 있도록 경로를 설정하였다.

위와 같은 기본 경로 설정기반으로 다익스트라 알고리즘을 적용하여 최단 경로를 계산하였다.

Table 5. Input data of shortest path algorithm

input data	equivalent plant configuration	value
vertex 개수	이동 가능한 위치 개수	20
시작 vertex	시작 위치	1
목표 vertex	목표 위치	20
edge 개수	이동 가능한 경로 개수	37
edge에 부과되는 weight	이동 경로 거리	Table 4의 weight 데이터, 대각선 기준 우측 데이터 (전체의 1/2)

본 알고리즘의 입력값과 플랜트에 해당되는 값은 Table 5와 같다.

Ⅳ. 시험 결과 및 분석

Fig. 11과 같이, 시작 위치를 v1, 목표 위치를 v20로 설정하여 결과를 얻었다. 플랜트 안전 훈련 환경에서는 운전원이 2번 열교환기 아래쪽 shell side에서 화재가 발생하여 steam gun이나 소화기를 가지고 조치를 취하려고 화재 현장에 출동하는 훈련 시나리오이다. 이때 최단 이동 거리를 계산하는 경우이다.

Fig. 11에서 중단 단계 지점까지의 최단 거리는 예를 들어 'minimum distance d, minimum vertex v: 37 4'라고 표시되며, 이는 vertex 4까지 도달하는 최단 거리는 35이라는 의미이다. 중간 단계

```
minimum distance d, minimum vertex v: 35 3
decision: visited vertex, visit[3]: 1
distance to to vertex[1] >= distance to from vertex[3] + weight[3][1]: 0 35 100000
distance to to vertex[2] >= distance to from vertex[3] + weight[3][2]: 10 35 100000
distance to to vertex[3] >= distance to from vertex[3] + weight[3][3]: 35 35 0
distance to to vertex[4] >= distance to from vertex[3] + weight[3][4]: 37 35 2
distance to to vertex[5] >= distance to from vertex[3] + weight[3][5]: 30 35 5
distance to to vertex[6] >= distance to from vertex[3] + weight[3][6]: 32 35 100000
distance to to vertex[7] >= distance to from vertex[3] + weight[3][7]: 20 35 100000
distance to to vertex[8] >= distance to from vertex[3] + weight[3][8]: 22 35 100000
distance to to vertex[9] >= distance to from vertex[3] + weight[3][9]: 18 35 100000
distance to to vertex[10] >= distance to from vertex[3] + weight[3][10]: 20 35 100000
distance to to vertex[11] >= distance to from vertex[3] + weight[3][11]: 20 35 100000
distance to to vertex[12] >= distance to from vertex[3] + weight[3][12]: 22 35 100000
distance to to vertex[13] >= distance to from vertex[3] + weight[3][13]: 35 35 100000
distance to to vertex[14] >= distance to from vertex[3] + weight[3][14]: 20 35 100000
distance to to vertex[15] >= distance to from vertex[3] + weight[3][15]: 18 35 100000
distance to to vertex[16] >= distance to from vertex[3] + weight[3][16]: 20 35 100000
distance to to vertex[17] >= distance to from vertex[3] + weight[3][17]: 35 35 100000
distance to to vertex[18] >= distance to from vertex[3] + weight[3][18]: 25 35 100000
distance to to vertex[19] >= distance to from vertex[3] + weight[3][19]: 23 35 100000
distance to to vertex[20] >= distance to from vertex[3] + weight[3][20]: 25 35 100000
minimum distance d, minimum vertex v: 37 4
minimum distance d, minimum vertex v: 35 13
decision: visited vertex, visit[13]: 1
distance to to vertex[1] >= distance to from vertex[13] + weight[13][1]: 0 35 100000
distance to to vertex[2] >= distance to from vertex[13] + weight[13][2]: 10 35 100000
distance to to vertex[3] >= distance to from vertex[13] + weight[13][3]: 35 35 100000
distance to to vertex[4] >= distance to from vertex[13] + weight[13][4]: 37 35 100000
distance to to vertex[5] >= distance to from vertex[13] + weight[13][5]: 30 35 100000
distance to to vertex[6] >= distance to from vertex[13] + weight[13][6]: 32 35 100000
distance to to vertex[7] >= distance to from vertex[13] + weight[13][7]: 20 35 100000
distance to to vertex[8] >= distance to from vertex[13] + weight[13][8]: 22 35 100000
distance to to vertex[9] >= distance to from vertex[13] + weight[13][9]: 18 35 100000
distance to to vertex[10] >= distance to from vertex[13] + weight[13][10]: 20 35 100000
distance to to vertex[11] >= distance to from vertex[13] + weight[13][11]: 20 35 100000
distance to to vertex[12] >= distance to from vertex[13] + weight[13][12]: 22 35 100000
distance to to vertex[13] >= distance to from vertex[13] + weight[13][13]: 35 35 0
distance to to vertex[14] >= distance to from vertex[13] + weight[13][14]: 20 35 2
distance to to vertex[15] >= distance to from vertex[13] + weight[13][15]: 18 35 100000
distance to to vertex[16] >= distance to from vertex[13] + weight[13][16]: 20 35 100000
distance to to vertex[17] >= distance to from vertex[13] + weight[13][17]: 35 35 5
distance to to vertex[18] >= distance to from vertex[13] + weight[13][18]: 25 35 100000
distance to to vertex[19] >= distance to from vertex[13] + weight[13][19]: 23 35 100000
distance to to vertex[20] >= distance to from vertex[13] + weight[13][20]: 25 35 100000
minimum distance d, minimum vertex v: 37 4
minimum distance d, minimum vertex v: 35 17
decision: visited vertex, visit[17]: 1
distance to to vertex[1] >= distance to from vertex[17] + weight[17][1]: 0 35 100000
distance to to vertex[2] >= distance to from vertex[17] + weight[17][2]: 10 35 100000
distance to to vertex[3] >= distance to from vertex[17] + weight[17][3]: 35 35 100000
distance to to vertex[4] >= distance to from vertex[17] + weight[17][4]: 37 35 100000
distance to to vertex[5] >= distance to from vertex[17] + weight[17][5]: 30 35 100000
distance to to vertex[6] >= distance to from vertex[17] + weight[17][6]: 32 35 100000
distance to to vertex[7] >= distance to from vertex[17] + weight[17][7]: 20 35 100000
distance to to vertex[8] >= distance to from vertex[17] + weight[17][8]: 22 35 100000
distance to to vertex[9] >= distance to from vertex[17] + weight[17][9]: 18 35 100000
distance to to vertex[10] >= distance to from vertex[17] + weight[17][10]: 20 35 100000
distance to to vertex[11] >= distance to from vertex[17] + weight[17][11]: 20 35 100000
distance to to vertex[12] >= distance to from vertex[17] + weight[17][12]: 22 35 100000
distance to to vertex[13] >= distance to from vertex[17] + weight[17][13]: 35 35 100000
distance to to vertex[14] >= distance to from vertex[17] + weight[17][14]: 20 35 100000
distance to to vertex[15] >= distance to from vertex[17] + weight[17][15]: 18 35 100000
distance to to vertex[16] >= distance to from vertex[17] + weight[17][16]: 20 35 100000
distance to to vertex[17] >= distance to from vertex[17] + weight[17][17]: 35 35 0
distance to to vertex[18] >= distance to from vertex[17] + weight[17][18]: 25 35 2
distance to to vertex[19] >= distance to from vertex[17] + weight[17][19]: 23 35 100000
distance to to vertex[20] >= distance to from vertex[17] + weight[17][20]: 25 35 100000
minimum distance d, minimum vertex v: 37 4
decision: visited vertex, visit[4]: 1
distance to to vertex[1] >= distance to from vertex[4] + weight[4][1]: 0 37 100000
distance to to vertex[2] >= distance to from vertex[4] + weight[4][2]: 10 37 100000
distance to to vertex[3] >= distance to from vertex[4] + weight[4][3]: 35 37 100000
distance to to vertex[4] >= distance to from vertex[4] + weight[4][4]: 37 37 0
distance to to vertex[5] >= distance to from vertex[4] + weight[4][5]: 30 37 100000
distance to to vertex[6] >= distance to from vertex[4] + weight[4][6]: 32 37 5
distance to to vertex[7] >= distance to from vertex[4] + weight[4][7]: 20 37 100000
distance to to vertex[8] >= distance to from vertex[4] + weight[4][8]: 22 37 100000
distance to to vertex[9] >= distance to from vertex[4] + weight[4][9]: 18 37 100000
distance to to vertex[10] >= distance to from vertex[4] + weight[4][10]: 20 37 100000
distance to to vertex[11] >= distance to from vertex[4] + weight[4][11]: 20 37 100000
distance to to vertex[12] >= distance to from vertex[4] + weight[4][12]: 22 37 100000
distance to to vertex[13] >= distance to from vertex[4] + weight[4][13]: 35 37 100000
distance to to vertex[14] >= distance to from vertex[4] + weight[4][14]: 20 37 100000
distance to to vertex[15] >= distance to from vertex[4] + weight[4][15]: 18 37 100000
distance to to vertex[16] >= distance to from vertex[4] + weight[4][16]: 20 37 100000
distance to to vertex[17] >= distance to from vertex[4] + weight[4][17]: 35 37 100000
distance to to vertex[18] >= distance to from vertex[4] + weight[4][18]: 25 37 100000
distance to to vertex[19] >= distance to from vertex[4] + weight[4][19]: 23 37 100000
distance to to vertex[20] >= distance to from vertex[4] + weight[4][20]: 25 37 100000

Final result: 25
```

Fig. 11. Shortest path calculation results.

지점까지 이르는 최단 거리는 계산 과정에서 다수의

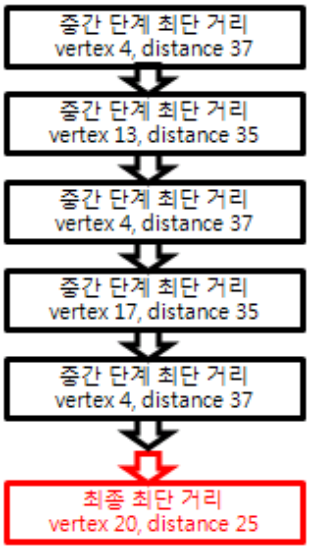

Fig. 12. Middle and final path calculation results.

set가 계산되어 산출된다. 최하단의 'Final result 25'는 최종 목표 지점에 이르는 최단 거리 계산 결과를 나타낸다. 본 실험에서는 최단 거리 계산 결과로서 25m를 얻었다. Fig. 10의 플랜트 레이아웃과 Table 4의 weight 이동 거리를 참고하여 운전원의 경험과 눈짐작으로 최단 거리를 결정하면 v1 → v2 → v20 이동 경로가 최단 거리일 것으로 짐작할 수 있고, 이동 거리는 weight v1 → v2 + weight v2 → v20 = 10 + 25 = 35m로 계산된다. 예상 밖으로 알고리즘을 사용한 최단 거리인 25m에 비해 이동 거리가 10m나 늘어났다. 운전원의 눈짐작과 알고리즘 계산 결과 간에 예상보다 큰 차이를 보여주었다. 플랜트 누출, 화재, 폭발과 같은 급박한 상황에서 25m대 35m의 이동 거리 10m 차이는 골든타임을 놓칠 수 있는 큰 차이라고 할 수 있다. 물론 플랜트 현장의 많은 변수가 모두 고려되지 않았고 모델의 단순화로 인해 현실성은 떨어질 수 있으나 향후 개선을 통해 현실성을 더욱 높일 수 있을 것으로 사료된다.

본 알고리즘의 활용 방안을 고려해보면 대규모 석유화학 플랜트의 복잡한 공정 설비에서 문제 발생시 운전원의 대처 행동을 훈련시키는 목적으로 안전 훈련 시나리오 설정에 활용 가능하다.

사전 연구[11]에서 플랜트 안전훈련 플랫폼 구조를 Fig. 13과 같이 설계하였다. 본 플랫폼 기반의 시스템 구현 시, 시나리오 설정 모듈을 본 알고리

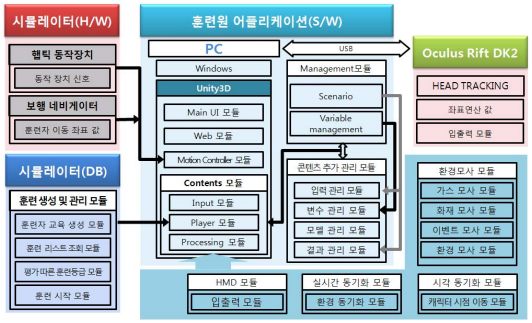

Fig. 13. Structure of interactive training platform.

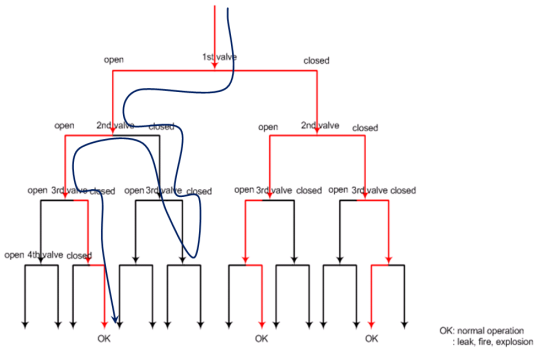

Fig. 14. Concept of interactive training.

즘을 도입하여 시스템을 개발할 수 있다.

또한, Fig. 14와 같이 운전원과 시스템간의 인터랙티브 훈련 시나리오 구성에도 활용이 가능하다.

상호작용적 훈련 개념에서 운전원이 어떤 이동 경로를 취하고 어떤 조치를 취하는 가에 따라 사고 규모가 걷잡을 수 없거나 초기에 대규모 사고로 확장됨을 방지할 수 있다. 운전원은 다양하고 인터랙티브한 이동 경로 훈련 시나리오를 통해 최적의 이동 경로를 습득하고 현실에서 사고가 날 경우에 대비할 수 있게 된다. 본 알고리즘을 적용해서 플랜트 발생 시에 최단 이동 거리를 활용하여 탈출과 조치를 취하도록 안전 훈련을 실행할 수 있다.

V. 결 론

본 연구에서는 3가지 결과를 도출하였다.

첫째, 플랜트 탈출 및 조치 경로 최단 거리 계산을 위한 다양한 알고리즘의 특징과 장단점을 조사 분석하였으며, 본 연구에 적용할 알고리즘으로서 다익스트라 알고리즘을 선정하였다.

둘째, 석유화학 플랜트 탈황 공정을 대상으로 하여 플랜트 레이아웃을 단순화 모델링하였고, 다익스트라 알고리즘 적용을 위해 운전원, 장비 등 이동 가능한 위치 개수, 시작 위치, 목표 위치, 이동 가능한 경로 개수, 이동 경로 거리 등을 설정하였다.

셋째, 플랜트 안전 훈련 시나리오에의 응용을 위한 알고리즘 계산 결과와 운전원의 경험과 눈짐작으로 선택한 이동 거리를 비교하여, 알고리즘이 10m의 이동 거리 감소 효과를 얻었다.

본 연구에서는 플랜트 실제 현장 상황과의 차이점이 있는 부분을 가정하였다.

플랜트 반응기나 열교환기와 같은 장비는 위치가 변경되지 않기 때문에, 단순화된 레이아웃기반으로 계산 수행에 문제가 없지만, 고정 장치가 아닌 운전원이나 소방관 및 소화기 위치는 상황에 따라 항상 변할 수 있는 변수이기 때문에, 본 연구에서와 같이 항상 고정된 위치로 설정할 수 없다. 뿐만 아니라, 운전원과 소화기와 관련된 거리도 이동에 따라서 변할 수 밖에 없다. 결국, 정확한 거리를 설정하기 위해서는 x, y축 좌표기반으로 정확한 거리를 계산해야 한다. 운전원과 소화기의 실시간 위치 정보를 기반으로 최단 경로를 계산하는 알고리즘은 향후 연구에서 고려해볼만한 연구 주제이다.

본 연구에서는 2차원 평면 상 거리만을 고려하였으나, 향후에는 3차원 거리도 고려해볼 수 있다. 특히, 반응기는 높이 20m 정도로 topside에 있는 플랜지에 접근하려면 사다리를 통해서 이동해야 하므로 이동 거리가 평면에 비해서 상당히 많이 늘어나는 점 등을 고려해보면, 더욱 상세한 모델링이 필요하다.

본 연구는 실제 활용되는 단계까지는 아직 이르지 못했으나, 석유화학 플랜트 탈황 공정을 대상으로 개발 중인 안전훈련시스템의 훈련 시나리오 설정 설계에 적용이 가능할 것으로 판단되며, 향후 본 기법과 개념이 적용된 안전훈련시스템을 활용한 훈련을 통해 활용 효과 검토가 가능할 것이다.

감사의 글

본 연구는 국토교통부 플랜트연구사업의 연구비 지원(16IFIP-B087592-04)에 의해 수행되었습니다.

REFERENCES

[1] Shortest path problem, Wikipedia,

http://en.wikipedia.org/wiki/Shortest_path_problem
[2] Dijkstra's algorithm, Wikipedia, http://en.wikipedia.org/wiki/Dijkstra's_algorithm
[3] Bellman–Ford algorithm, Wikipedia, https://en.wikipedia.org/wiki/Bellman–Ford_algorithm
[4] A* search algorithm, Wikipedia, http://en.wikipedia.org/wiki/A*_search_algorithm
[5] Floyd–Warshall algorithm, Wikipedia, http://en.wikipedia.org/wiki/Floyd–Warshall_algorithm
[6] Johnson's algorithm, Wikipedia, http://en.wikipedia.org/wiki/Johnson's_algorithm
[7] Viterbi algorithm, Wikipedia, http://en.wikipedia.org/wiki/Viterbi_algorithm
[8] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, 3rd ed., MIT Press, Cambridge, Massachusetts, (2009)
[9] Xiao, J. X., and Lu, F. L., "An Improvement of the Shortest Path Algorithm Based on Dijkstra Algorithm", *2nd International Conference on Computer and Automation Engineering (ICCAE)*, (2010)
[10] Cho, S. H., Joo, K. D., Kang, H., Park, K. S., and Shin, D. I., "A Mobile Application for Navigating the Optimal Escape Route in Accidents and Emergency Situations", *Korean Journal of Hazardous Materials*, **3**(1), 28-36, (2015)
[11] Kim, H. J., Park, C. C., Lee, J. Y., Lee, C. S., and Yu, J. C., "A Study of Interactive Training Methods for the Safe Operation of City Gas Governor", *Journal of the Korean Institute of Gas*, **21**(1), 34-42, (2017)