

An Economic Ship Routing System Based on a Minimal Dynamic-cost Path Search Algorithm

Sang-Yeon Joo[†] · Tae-Jeong Cho[†] · Jae-Mun Cha[†] · Jin-Ho Yang^{**} · Yung-Keun Kwon^{***}

ABSTRACT

An economic ship routing means to sail a ship with a goal of minimizing the fuel consumption by utilizing weather forecast information, and various such systems have been recently studied. For a successful economic ship routing system, an efficient algorithm is needed to search an optimal geographical path, and most of the previous systems were approaching to that problem through a minimal static-cost path search algorithm based on the Dijkstra algorithm. To apply that kind of search algorithm, the cost of every edge assigned with the estimated fuel consumption should be constant. However, that assumption is not practical at all considering that the actual fuel consumption is determined by the weather condition when the ship will pass the edge. To overcome such a limitation, we propose a new optimal ship routing system based on a minimal dynamic-cost path search algorithm by properly modifying the Dijkstra algorithm. In addition, we propose a method which efficiently reduces the search space by using the A* algorithm to decrease the running time. We compared our system with the shortest path-based sailing method over ten testing routes and observed that the former reduced the estimated fuel consumption than the latter by 2.36% on average and the maximum 4.82% with little difference of estimated time of arrival.

Keywords : Economic Ship Routing Problem, Minimal Dynamic-cost Path Search, Dijkstra Algorithm Variant

최소동적비용 경로탐색 알고리즘 기반 선박경제운항시스템

주 상 연[†] · 조 태 정[†] · 차 재 문[†] · 양 진 호^{**} · 권 영 근^{***}

요 약

선박경제운항이란 기상예측정보를 활용하여 연료소모량을 최소화하도록 선박을 운항하는 것으로서 최근 다양한 선박경제운항 시스템이 연구되고 있다. 성공적인 선박경제운항을 위해서는 효율적인 최적의 지리적 경로탐색 방법이 필요한데 기존의 시스템에서는 주로 Dijkstra 알고리즘 기반의 최소정적비용 경로탐색 알고리즘으로 접근하고 있다. 그러한 접근법을 적용하기 위해서는 특히 연료소모량으로 정의되는 간선의 비용을 고정해야 하는데 선박이 그 간선을 실제 지날 때의 기상 상황에 따라 연료소모량이 변할 수 있다는 점에서 적절하지 않은 가정이다. 이에 본 논문에서는 그러한 단점을 극복하기 위해 Dijkstra 알고리즘을 변형한 최소동적비용 경로탐색 알고리즘을 제안한다. 또한, 실행시간을 단축하기 위해 A* 알고리즘을 활용하여 탐색공간을 효과적으로 줄이기 위한 방법도 제시한다. 총 10개의 테스트 노선에 대해서 본 논문에서 제안된 시스템을 기존의 단순한 최단거리 운항방법과 비교한 결과, 운항소요시간은 거의 차이가 없으면서도 연료소모량을 평균 2.36%, 최대 4.82% 개선시킬 수 있었다.

키워드 : 선박경제운항 문제, 최소동적비용 경로탐색, 다익스트라 알고리즘 변형

1. 서 론

선박경제운항이란 연료소모량을 최소화하도록 선박을 경제적으로 운항하는 것을 말하는데, 최근의 고유가 상황과

환경보호 필요성에 의해 그 중요성이 더욱 부각되고 있다. 과거에는 연료소모량을 줄이기 위해서 적절한 감속 운항을 하는 단순한 방법을 사용하였다. 이것은 선박 엔진의 출력을 낮추면 연료소모량 대비 선박 출력의 효율이 좋아지는 점을 이용한 것이지만 운항 시간이 늘어난다는 점에서 한계가 있다. 감속 운항 방법 대신 최근에는 기상예측 정보를 활용하여 운항 소요 시간이 비슷하면서도 연료소모량을 줄일 수 있는 방법에 대해 연구가 진행되고 있다. 예를 들면, 진화 전략 기법을 통해 선박의 엔진출력을 세부구간별로 최적화하는 방법이 제안되어 36개 운항 스케줄에 대해 시뮬레이션을 수행한 결과 운항소요시간은 큰 차이가 없으면서 총

* 이 논문은 현대중공업 지원에 의한 울산대학교 전기공학부 일류화 연구비에 의하여 연구되었음.

[†] 준 회원: 울산대학교 전기공학부 학사과정

^{**} 준 회원: 현대중공업 선박해양연구소 책임연구원

^{***} 정 회원: 울산대학교 전기공학부 조교수

논문접수: 2012년 9월 11일

수정일: 1차 2012년 10월 19일

심사완료: 2012년 10월 23일

* Corresponding Author: Yung-Keun Kwon(kwonyk@ulsan.ac.kr)

연료소모량을 평균 1.3% 개선시켰다[1]. 하지만 이 시스템은 지리적으로 최적의 항로를 찾는 것이 아니라 입력으로 어떤 항로가 주어질 경우 그 항로에 대해서 최적의 세부구간 출력을 탐색한다는 한계점이 있다. 따라서 지리적인 최적 항로를 찾는 방법이 필요한데, 이를 위해서 그래프 모델에서의 최소정적비용 경로탐색 알고리즘이 활용될 수 있다. 실제로 SORAS 등 여러 기존 시스템들[2-4]에서는 해양지리상의 일정한 격자점들을 정점으로 하고, 정점들 사이의 간선(edge)마다 연료소모량 추정값을 비용(cost)으로 할당하는 그래프 모델을 정의하고, 동적 프로그래밍으로 구현된 Dijkstra 알고리즘을 통해 출발정점에서 목표정점에 이르는 최소정적비용 경로를 구하였다. 즉, 간선의 비용을 예상연료소모량으로 정의하는 가중 그래프(weighted graph)로 문제를 표현함으로써 어렵지 않게 최소정적비용 경로를 구할 수 있다. 하지만 실제 상황에서는 간선의 비용이 시간에 따라 변하는 (기상 상황이 변하므로) 동적비용 (dynamic-cost) 그래프 모델이라는 점에서 최소정적비용 경로탐색 알고리즘의 적절한 변형이 필요하다.

이에 본 논문에서는 현실적인 상황을 최대한 반영하기 위하여 Dijkstra 알고리즘을 변형한 동적비용 그래프 모델에서의 최소동적비용 경로탐색 알고리즘을 제안하고 이를 바탕으로 개선된 선박경제운항시스템을 구현한다. 또한 계산의 복잡성을 줄이기 위해 경로탐색 알고리즘의 탐색 범위를 효율적으로 제한하는 방법을 함께 구현하였다. 본 논문에서는 10개의 테스트 노선에서 각각 12회 운항 스케줄(즉, 총 120개 스케줄)에 대해 최단거리 기반의 운항방법과 본 논문에서 제안하는 시스템의 성능을 비교하였다. 그 결과 기존 시스템에 비해 본 시스템의 경제운항방법은 운항소요시간에 큰 차이가 없으면서 평균 2.36%, 최대 4.82%의 연료소모량이 감소함을 관찰할 수 있었다.

본 논문의 전체 구성은 다음과 같다. 2장에서는 선박 경제운항 시스템 구현에 필요한 배경지식을 설명하고, 3장에서는 이 논문에서 제안하는 시스템에 대해 설명한다. 그리고 4장에서는 시뮬레이션 결과에 대해 분석을 하고, 5장에서 결론을 맺는다.

2. 배경 지식

이 장에서는 경제운항시스템을 구현하는 데 필요한 배경지식을 설명한다. 2.1절에서는 경제운항 시스템 구축에 활용된 기상예보데이터와 지리정보데이터에 대해서 설명한다. 2.2절에서는 이 연구에서 사용한 연료소모량 추정식에 대해서 살펴본다.

2.1 지리정보 및 기상예보 데이터

경제운항 시스템을 구축하기 위해서는 유효한 해상 항로를 찾는 과정에서 전 세계 지리정보 데이터가, 연료소모량을 추정하는 과정에서는 기상정보 데이터가 각각 필요하다. 먼저, 지리정보 데이터는 미국국립지리원(<http://www.ngdc.gov/>)로부터 획득한 1,938개의 벡터데이터를 바탕으로 7200×3600

픽셀의 비트맵¹⁾ 정보를 구축하여 육지와 해상을 구분하였다. 한편, 기상정보 데이터는 유럽중기예보센터(<http://www.ecmwf.int/>)에서 2011년 1월 1일부터 2011년 12월 31일까지 1년 동안의 세계 해양을 경위도 각각 1.5도 단위로 나는 격자점에 대해서 6시간(0시, 6시, 12시, 18시)마다 이뤄진 예보 데이터이며 본 시스템에서는 많은 기상정보들 중에서 연료소모량 추정식에 필요한 파향, 파고, 풍향, 풍속 등 네 가지 정보를 획득하였다.

2.2 연료 소모량 추정식

파도와 바람은 대표적인 해양 기상정보이다. 이들 기상정보의 크기와 방향성은 선박운항에 많은 변수를 만들게 된다. 일반적으로 선박의 엔진은 출력에 따른 속도의 기대치가 있는데, 이러한 기상환경들은 기대속도와 실제속도의 차이를 만드는 원인이 된다. 즉, 기상환경에 따라 선체에 부가저항이 생기게 되면, 부가저항은 선속저항으로 이어지게 되고, 여기서 저하된 선속을 기대치까지 보정하기 위해서는 추가적인 엔진의 출력 증가가 필요하고, 이는 결과적으로 연료소모량의 증가를 불러오게 된다.

본 논문에서는 기존의 해상 상태에 따른 선속 저하량의 계산 방법 중 기존의 연구 결과에서 사용된 Townsin-Kwon 연료추정식[5]계산식을 활용하여 연료소모량을 추정하였다. 이 식을 적용하기 위해서는 선박의 길이, 너비, 깊이 등의 9가지 선박제원 정보(4.1절에서 설명)와 파고, 파향, 풍속, 풍향과 같은 4가지 기상정보, 그리고 선박 운항방향, 운항속도, 운항거리와 같은 3가지 운항정보가 입력으로서 필요하다.

3. 제안된 선박경제운항시스템

3.1 시스템 개념

본 논문에서 제안하는 선박경제운항 시스템의 전체적인 구성은 Fig. 1과 같이 초기항로 탐색, 최적 항로 탐색, 최적 출력 탐색의 3단계로 구성된다. 먼저, 첫 번째 단계에서 출항지와 입항지의 경위도가 주어지면 2.1절에서 설명한 세계 지리정보 데이터를 바탕으로 A* 알고리즘을 통해 지리적 최단 경로(P)를 찾는데 이 경로는 다음 단계에서 최적 항로 탐색을 위한 초기 항로로서 활용된다. 두 번째 단계에서는 탐색범위 설정 알고리즘을 통해 초기 항로 P로부터 일정 거리 이내의 구역을 탐색 공간(G_p)으로 제한한 뒤, 최소동적비용 경로탐색 알고리즘을 사용하여 지리적 최적 경로(P*)를 구한다. 마지막 단계에서는 여러 개의 세부 구간들로 구성되어 있는 지리적 최적 경로 P*에 대해 각 세부구간에서의 최적 엔진출력을 탐색하기 위해 [1]에서 제안된 진화진략기반 출력탐색 알고리즘을 적용한다.

1) 즉, 전 세계 지리적 위치가 0° ~ 360° 범위의 경도와 0° ~ 180° 범위의 위도로 표현되며, (i, j) 픽셀의 비트값은 $\frac{i}{7200} \times 360^\circ$, $\frac{j}{3600} \times 180^\circ$ 지역이 해상인지 아닌지 여부를 나타낸다.

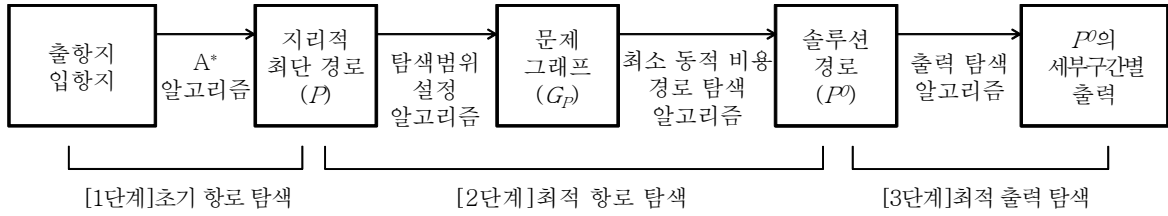


Fig. 1. An economic sailing system proposed in this paper

3.2 A* 알고리즘을 이용한 초기 항로 탐색

본 논문에서는 운항경로라는 해의 표현을 위해 [1]에서 사용된 격자좌표체계와 유사한 형태로 사용한다. 기상예보 데이터가 경위도 1.5도 단위로 주어짐에 따라 전 세계의 해상 지리를 역시 1.5도 단위로 나누었으며 이를 통해 전 세계의 지리를 총 240x120=28,800개의 블록으로 구분하게 된다. 각각의 블록은 선박의 상세위치를 표현하기 위해 하나 이상의 정점을 가질 수 있는데 경로의 자연스런 곡선 표현을 위해 Fig. 2와 같이 블록의 외곽에 8개의 정점을 배치하고, 40개의 방향 중 하나로 이동할 수 있도록 하였다. 이 때, 탐색 속도를 빠르게 하기 위해 이동 블록의 개수를 조절할 수 있도록 하였는데 본 실험에서는 입항지와 출항지 부근의

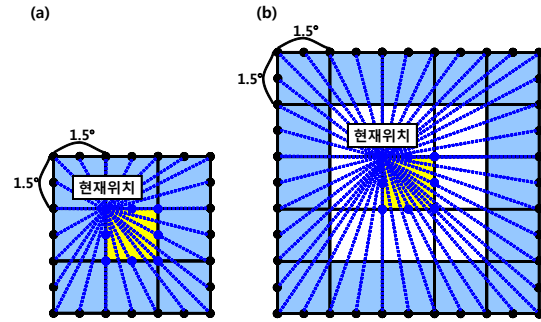


Fig. 2. The grid coordinate system used in this paper. One-block routing nearby departure and arrival ports (a) and two-block routing at other sites

```

A*(G(V,E), v_s, v_g) {
    // G(V,E) : 전세계 지리를 표현한 그래프
    // v_s ∈ V, v_g ∈ V : 출발정점(출항지)과 도착정점(입항지)
    P(v_s) ← null; // P(v): 정점 v_s부터 정점 v까지의 최단 경로를 표현한 문자열
    g(v_s) ← 0, f(v_s) ← g(v_s) + h(v_s, v_g); // h(v,w) : 정점 v와 w사이의 지리적 최단거리
    g(v) ← ∞, f(v) ← ∞ ∀ v ∈ V \ {v_s};
    C ← ∅;
    while (TRUE) {
        w ← argmin_{v ∈ V \ C} f(v);
        if ( w = v_g ) break;
        for ( ∀ v ∈ {v | v ∈ V \ C and (w,v) ∈ E} ) {
            if (collision_check(w, v)) continue; // (w, v)가 육지와 충돌하면 collision_check(w, v)는 TRUE
            l ← g(w) + h(w,v);
            if ( l < g(v) ) {
                g(v) ← l;
                f(v) ← g(v) + h(v, v_g);
                P(v) ← P(w) ⊕ v; // ⊕ : 문자열 연결(string concatenation)
            }
        }
        C ← C ∪ {w};
    }
    return P(v_g); // A* 알고리즘의 결과
}
    
```

Fig. 3. A* Algorithm' for the geographical shortest path

블록에서는 Fig. 2a와 같이 한 칸 블록 이동, 그 이외의 지역에서는 Fig. 2b와 같이 탐색 속도가 빠른 두 칸 블록 이동을 하였다. 입·출항지 블록에서 한 칸 블록 이동을 사용한 이유는 입·출항지 같은 항구 부근에는 만이나 섬들이 존재하여 좀 더 미세한 항로 탐색이 필요하기 때문이다. 이와 같은 식으로 전 세계 지리를 그래프 $G(V, E)$ 로 표현한 후, Fig. 3과 같은 복잡도가 $O(|V|^2)$ 인 A^* 알고리즘을 적용하여 지리적 최단 경로 P 를 구한다. 즉, 지리적 최단 경로 P 는 모든 해상의 기상 이 동일할 경우 최적 항로가 될 수 있지만, 기상이 다를 경우 최적 항로의 탐색이 필요하다. 이를 위해 본 논문에서는 Dijkstra 알고리즘을 변형한 최적항로 탐색 알고리즘(최소동적비용 경로탐색 알고리즘)을 제안하는데, 경로 P 는 탐색 알고리즘의 복잡성을 줄이기 위한 탐색 범위 설정을 위한 초기 항로로 사용된다.

3.3 최소동적비용 경로탐색알고리즘을 사용한 지리적 최적경로 탐색

본 경제운항시스템에서 제안하는 최소동적비용 경로탐색 알고리즘은 Dijkstra 알고리즘을 적절하게 변형한 알고리즘이다. 그런데, Dijkstra 알고리즘은 너비우선탐색을 기반으로 하기 때문에 입항지와 출항지 사이의 거리가 비교적 먼 경우 매우 많은 양의 계산이 필요하다. 이러한 문제점을 완화시키기 위해 본 연구에서는 경로탐색 알고리즘의 탐색 범위를 지리적 최단경로 P 의 주변으로 제한하였다. 구체적으로 설명하면, 전 세계 지리를 표현한 그래프 $G(V, E)$ 와 지리적 최단 경로 $P = v_1v_2 \dots v_n$ 가 주어질 때, 최소동적비용 경로탐색알고리즘을 위한 유효탐색공간 $G_P(V', E')$ 는 $G(V, E)$ 의 부분그래프로서 다음과 같이 정의된다.

$$G_P(V') = \bigcup_{i=1}^{n-1} \{v \in V \mid dist(v, v_i v_{i+1}) \leq \delta\} \quad \text{and}$$

$$G_P(E') = \{e(v, w) \in E \mid v \in G_P(V') \text{ and } w \in G_P(V')\}$$

이 때, $dist(v, v_i v_{i+1})$ 는 정점 v 와 세부구간 $v_i v_{i+1}$ 사이의 최단거리를 뜻하며 δ 는 유효탐색공간의 크기를 조절하기 위한 파라미터이다. 이와 같이 유효탐색공간을 정의한 이유는 일반적으로 선박의 항로는 대권항로(지리적 최단경로)에서 크게 벗어나지 않으며 대권항로로부터 크게 벗어나면 운항거리가 커져서 연료소모량이 크게 증가할 가능성이 높기 때문이다.

Fig. 4는 최소동적비용 경로탐색을 위한 유효탐색공간 설정의 실제 적용 사례를 보여주며, 탐색범위가 효율적으로 좁혀져 있음을 알 수 있다. 한편, 최적항로 탐색의 속도를 높이기 위해서 또 하나의 제약 조건을 설정한다. 이는 해상에서 선박들이 현재 진행 방향 대비 후진을 하거나 항해 도중 왔던 경로를 되돌아가는 경우가 없다는 것을 고려하여 최적 항로 탐색 시 40가지 진행 방향(Fig. 2 참조) 중에서



Fig. 4. An example of constrained space for minimal dynamic-cost path search

후방으로의 진행 방향은 탐색하지 않도록 한 것이다.

이제 유효탐색공간 G_P 가 결정되면 그 범위 내에서 최적항로를 탐색하게 된다. 앞서 살펴본 바와 같이 기존의 최적항로 탐색 연구들에서는 일반적으로 Dijkstra 알고리즘 적용을 위해 간선의 비용을 고정값으로 설정한다. 하지만 선박이 해당 간선을 운항하는 시간에 따라 기상 환경이 변할 수 있으므로 간선의 비용을 시간에 따른 함수로 표현하는 것이 더 정확하다 (즉, 동적비용 모델). 이에 본 논문에서는 선박이 해당 지역을 지나갈 때의 시간을 계산하여 그 시간에서의 기상예보를 고려한 예상연료소모량을 구하고 이를 해당 간선의 비용으로 정하는 최소동적비용 경로탐색 알고리즘을 제안한다. Fig. 5는 본 연구에서 제안한 Dijkstra 알고리즘을 변형한 최소동적비용 경로탐색 알고리즘의 구조이다 (복잡도는 Dijkstra 알고리즘과 같이 $O(|V|^2)$ 이다). 이 알고리즘의 $h(w, v, P^0(w))$ 함수에서 보듯이, 간선의 비용이 선박의 통과 과정을 고려하여 동적으로 정의됨을 알 수 있다.

3.4 진화전략 기반 최적 출력 탐색

3.3절에서 설명한 최소동적비용 경로탐색알고리즘(Dijkstra-variant)에 의해 구해진 지리적 최적 경로는 여러 개의 세부 구간들로 구성된다. 이 지리적 최적 경로는 기본적으로 선박의 출력을 일정한 값으로 고정시킨 상태에서 찾아 낸 경로이므로 각 세부구간별로 출력값을 변화시키면(선박의 속도를 조절) 연료소모량을 더욱 낮출 수 있는 항로를 구할 수 있다. 이를 위해 본 논문에서는 기존 연구에서 활용되었던 진화 전략(Evolutionary Strategy; ES)기반의 출력 최적화 알고리즘[1]을 이용하여 세부구간별 최적출력을 구한다. 또한, 적절한 범위 내의 출력 탐색이 되도록 하기 위해 ES기반 출력탐색 운항법에 의한 도착시간이 고정출력 운항법에 의한 도착시간에 비해 0.3% 이내의 오차범위에서 출력탐색을 하도록 하였다.

```

Dijkstra-variant( $G_P(V, E), v_s, v_g$ ) {
    //  $G_P(V, E)$  : 유효 탐색 공간 그래프
    //  $v_s \in V, v_g \in V$  : 출발정점(출항지)과 도착정점(입항지)
     $P^0(v_s), P^0(v_g) \leftarrow null$ ; //  $P^0(v)$ : 정점  $v$ 까지의 솔루션 경로
     $f(v_s) \leftarrow 0, f(v) \leftarrow \infty \forall v \in V \setminus \{v_s\}$ ; //  $f(v)$ : 경로  $P^0(v)$ 를 따라 항해시 예상연료소모량
     $C \leftarrow \emptyset$ ;
    while ( $V \setminus C \neq \emptyset$ ) {
         $w \leftarrow \operatorname{argmin}_{v \in V \setminus C} f(v)$ ;
        if ( $w = v_g$ ) break;
        for ( $\forall v \in \{v \mid v \in V \setminus C \text{ and } (w, v) \in E\}$ ) {
            if (collision_check( $w, v$ )) continue;
             $l \leftarrow f(w) + h(w, v, P^0(w))$ ; //  $h(w, v, P^0(w))$  : 경로  $P^0(w)$ 를 따라 항해시
                // 동적으로 결정되는 구간( $w, v$ )에서의 예상연료소모량
            if ( $l < f(v)$ ) {
                 $f(v) \leftarrow l$ ;
                 $P^0(v) \leftarrow P^0(w) \oplus v$ ; //  $\oplus$  : 문자열 연결(string concatenation)
            }
        }
         $C \leftarrow C \cup \{w\}$ ;
    }
    return  $P^0(v_g)$ ; //최소동적비용 경로탐색 알고리즘의 결과
}
    
```

Fig. 5. A minimal dynamic-cost path search algorithm by modifying Dijkstra algorithm

4. 시뮬레이션 결과

4.1 파라미터 설정

Table 1은 본 논문의 시뮬레이션에서 사용된 선박 제원에 관한 파라미터를 나타낸다. 여기에는 선박의 길이, 너비, 깊이 등 9가지의 파라미터가 포함되는데 이러한 정보들은 2.2절에서 설명한 연료소모량 추정식에서 사용된다. 본 논문에서는 기존 논문[1]에서 사용된 재화 중량 300,000톤급 VLCC(Very Large Crude Carrier)의 제원 정보를 활용하였다.

4.2 경로별 운항방법별 평균

본 논문에서 제안한 시스템의 성능을 검증하기 위하여 10개의 테스트 노선에 대하여 2011년 1월부터 12월까지 1개월 간격으로 출발날짜를 변경하며 시뮬레이션을 실행하였으며 (즉, 총 120가지의 운항 스케줄), Table 2는 그 결과를 보여준다. 그 표에서 “A*-FIX”와 “A-ES”는 A*알고리즘을 이용하여 최단경로를 구한 후 세부 구간에 대하여 고정 출력으로 운항하는 방법과 3.4절에서 설명한 진화전략(ES)기반 출력최적화 알고리즘을 적용한 방법을 각각 의미한다. 또한, “DV-FIX”와 “DV-ES”는 3.3절에서 설명한 최소동적비용

Table 1. Ship information parameters for simulation

| | Item | Value (단위) |
|--------------|---|--------------|
| 선박제원 파라미터 | 길이 (Length) | 315.0 (m) |
| | 너비 (Breadth) | 57.2 (m) |
| | 깊이 (Depth) | 30.4 (m) |
| | 홀수 (Draft) | 20.8 (m) |
| | 블록 계수 (Block Coefficient) | 0.8 |
| | 부력 중심 (Longitudinal Center of Buoyancy, Forward Direction in m) | 10.17 (m) |
| | 프로펠러 직경 | 9.1 (m) |
| | 프로펠러 효율 | 0.6 |
| | 단위시간 마력당 사용 연료량 (SFOC) | 125.0 (g/hr) |

경로탐색 알고리즘을 이용하여 최적 경로를 탐색한 후 마찬가지로 고정 출력 혹은 진화전략기반 출력최적화 알고리즘을 적용한 방법을 의미한다. 한편, 공정한 비교를 위하여 모든 운항방법에 대해 동일한 도착시간에 관한 테드라인이 설정된다. 그 표에서 “평균(표준편차)”는 해당 테스트 노선에 대해 12회(매월 1회) 운항에 따른 연료소모량의 평균 및 표

Table 2. Performance comparison between A* path search and the minimal dynamic-cost path search algorithms (Unit : ton)

| 테스트 노선 | | A*-FIX | A*-ES | | DV-FIX | | DV-ES | |
|--------|------------------------|--------------------|----------------------------------|------------------|--------------------|------------------|----------------------------------|------------------|
| | | 평균 (표준편차) | 평균 (표준편차) | A*-FIX 대비 개선율 | 평균 (표준편차) | A*-FIX 대비 개선율 | 평균 (표준편차) | A*-FIX 대비 개선율 |
| 그룹 A | Busan → Honolulu | 1026.27 (15.93) | 1002.93 (23.13) | 2.27 % | 1020.79 (8.43) | 0.53 % | 1011.81 (8.06) | 1.40 % |
| | Busan → Oakland | 1243.18 (46.66) | 1195.78 (32.62) | 3.81 % | 1222.79 (18.30) | 1.64 % | 1211.30 (16.63) | 2.56 % |
| 그룹 B | Busan → San Antonio | 2495.82 (16.29) | 2477.16 (17.89) | 0.74 % | 2467.43 (6.14) | 1.13 % | 2451.22 (5.19) | 1.78 % |
| | Busan → Sydney | 1103.22 (26.29) | 1090.79 (16.18) | 1.12 % | 1095.36 (8.89) | 0.71 % | 1085.72 (8.46) | 2.88 % |
| | Busan → Puerto Montt | 2417.24 (23.82) | 2387.15 (8.72) | 1.24 % | 2402.23 (8.09) | 0.62 % | 2384.45 (7.19) | 1.35 % |
| 그룹 C | Busan → Santos | 2875.18 (99.43) | 2774.41 (67.07) | 3.50 % | 2755.56 (40.10) | 4.16 % | 2736.31 (39.97) | 4.82 % |
| | Busan → Rio De Janeiro | 2810.55 (94.03) | 2729.56 (65.64) | 2.88 % | 2714.55 (27.97) | 3.41 % | 2696.81 (27.70) | 3.72 % |
| | Durban → Santos | 887.84 (10.73) | 874.83 (5.66) | 1.46% | 883.91 (4.51) | 0.44% | 873.49 (3.41) | 1.61% |
| | Sydney → Oakland | 1384.04 (11.21) | 1370.52 (8.21) | 0.97% | 1376.98 (6.71) | 0.51% | 1367.82 (5.77) | 1.17% |
| | Busan → Topolobampo | 1369.06 (34.58) | 1349.95 (20.32) | 1.39% | 1353.11 (15.54) | 1.16% | 1343.04 (14.25) | 1.90% |

평균 1.93 %

평균 1.44 %

평균 2.36 %

준편차이며, “A*-FIX대비 개선율”은 “A*-FIX”운항 방법의 평균 연료소모량에 비해 해당 운항 방법들의 평균 연료소모량이 몇 % 줄었는지를 나타낸다 (굵은 글씨는 네 가지 운항방법 중 가장 좋은 결과를 표시한다). 실험결과, “A*-ES”, “DV-FIX”, “DV-ES” 등 모든 운항방법들이 “A*-FIX”보다 최소 0.44%, 최대 4.82% 연료소모량을 개선시켰으며, 세 운항 방법의 전체 평균 개선율은 1.93%, 1.44%, 2.36%이었다. 또한, “A*-ES”가 2개 노선(그룹 A)에서 가장 성능이 좋았던 반면 “DV-ES”는 8개 노선(그룹 B와 C)에서 가장 성능이 좋았으며, 결국 전체적으로는 최소동적비용 경로탐색 알고리즘과 진화연산기반 출력탐색 알고리즘을 차례로 사용한 경우(“DV-ES”모델)가 가장 성능이 좋았다. 그러나 두 알고리즘의 결합이 항상 시너지 효과를 내지는 못함을 알 수 있다. 즉, 그룹 A의 노선에서는 최소동적비용 경로탐색 알고리즘으로 발견한 최적항로가 오히려 진화연산기반 출력탐색 알고리즘의 성능향상을 방해함으로써 “A*-ES”보다 못한 성능을 보인 것이다. 또한 두 알고리즘의 시너지 효과의 양상이 다를 수도 있다. 그룹 B 노선에서는 최소동적비용 경로탐색 알고리즘과 진화연산기반 출력탐색 알고리즘이 모두 성능개선에 기여하였지만 그룹 C 노선에서는 최소동적비용 경로탐색 알고리즘이 성능개선에 상대적으로 크게 영향을 미쳤음을 알 수 있다. 이러한 결과는 비록 “DV-ES”가 가장 좋은 방법이기도 하지만 선박경제운항시스템의 성능 극대화를 위해서는 지리적 최적경로탐색과 세부구간 최적출력탐색이 적절하게 혼합되어 사용되어야함을 뜻한다.

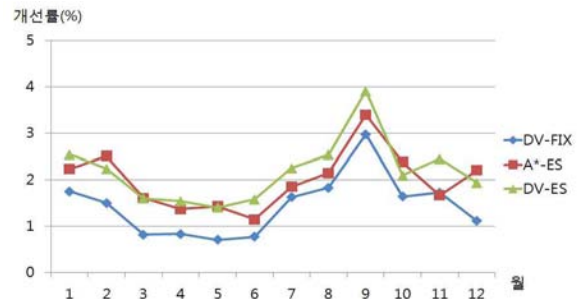


Fig. 6. Monthly average result of the routing methods

다음으로 시기에 따른 성능 개선의 특성이 있는지를 살펴 보기 위하여 “A*-ES”, “DV-FIX”, “DV-ES” 등의 운항방법이 “A*-FIX” 대비 월별 평균 연료소모량을 얼마나 개선시키는지를 비교하였으며, Fig. 6은 그 결과를 나타낸다. 그 그림에서 보이듯이, 성능개선의 폭이 월별로 변화가 있지만, Table 2와 마찬가지로 “DV-ES”가 전체적으로 가장 좋은 성능을 보인다. 특히, 9월에 가장 높은 성능개선을 보였으며 5월에는 가장 작은 성능개선을 보인다. 또한, 2월이나 10월의 경우에는 “A*-ES”가 “DV-ES”보다 더 좋은 성능을 보여 주는데, 이는 Table 2의 결과에서처럼 최소동적비용 경로탐색 알고리즘과 세부구간 최적출력탐색 알고리즘의 효과적인 결합이 필요함을 설명한다.

Fig. 7은 부산에서 출항하여 시드니로 입항하는 노선에 대해서 Fig. 3의 A*알고리즘에 의한 구해진 경로와 Fig. 5의

Dijkstra-variant 알고리즘에 의해 구해진 경로가 지리적으로 얼마나 다른지를 보이기 위해 별도로 제작한 시각화 프로그램을 통해 그 일부를 보인 것이다. 이 때, 엔진의 출력은 두 경로 모두 같은 값으로 고정되어 있으며 해상 블록의 색상이 밝을수록 파고가 높음을 의미한다. 그 그림에서 Dijkstra-variant 알고리즘에 의해 구해진 경로가 A*알고리즘에 의한 구해진 경로에 비해 높은 파고 지역을 효과적으로 피해가고 있음을 알 수 있다. 이러한 악천후 회피 경로가 연료 소모량을 감소 가능성을 높일 수 있다. 실제로 이 예제에서 Dijkstra-variant 알고리즘에 의해 구해진 경로 거리는 8,391.002km로서 A*알고리즘에 의해 구해진 경로 길이 8,277.462km보다 113.540km 더 멀었지만, 도착시간은 전자의 경로가 후자의 경로보다 약 19시간 빨리 도착하였을 뿐만 아니라 연료소모량도 1187.16톤에서 1114.12톤으로 6.5% 감소시켰다. 이처럼 본 연구에서 제안한 Dijkstra-Variant 알고리즘은 효과적인 지리적 최적경로를 탐색하는 데 큰 잠재력을 가지고 있다고 할 수 있다.

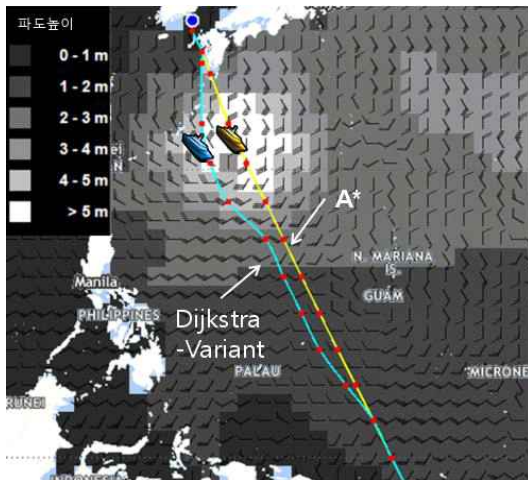


Fig. 7. Weather avoidance example by DV-FIX method

5. 결 론

기존의 선박경제운항시스템에서는 비용(cost)을 고정된 값(실제거리나 예상연료소모량)으로 정하는 정적비용 그래프 모델에서 Dijkstra알고리즘을 적용함으로써 최소정적비용 경로를 탐색을 시도해오고 있는데, 이는 선박의 항해 시간에 따라 비용이 동적으로 변할 수 있다는 실제 상황을 충분히 반영하고 있지 못하다. 이에 본 논문에서는 그러한 단점을 극복하기 위해 선박이 해당 간선을 지날 때의 시간을 고려한 예상연료소모량을 비용으로 정하는 동적비용 그래프 모델을 정의하고 Dijkstra 알고리즘을 적절하게 변형함으로써 최소동적비용 경로탐색 알고리즘을 제안하였다. 또한, 기존의 세부구간별 출력최적화 알고리즘과 자연스럽게 결합할 수 있도록 하였다. 실제로 10개 노선에서 12개월 운항 스케줄(총 120가지)에 대해서 본 논문에서 제안된 시스템의 성

능을 최단거리기반 운항방법과 비교한 결과, 운항소요시간이 차이가 없으면서도 평균적으로 연료소모량을 평균 2.36%, 최대 4.82% 개선시킬 수 있음을 확인하였다.

본 연구 결과물을 향후 더욱 개선하기 위해서는 다음과 같은 연구들이 필요할 것이다. 첫째, 본 논문에서는 지리적 경로최적화와 세부구간 출력최적화가 단계적으로 실행됨으로써 테스트 노선에 따라 두 최적화가 항상 시너지 효과를 내지는 못하였다. 이러한 문제를 해결하기 위해서는 경로탐색과 출력탐색이 동시에 이루어져야하며 이를 위한 적절한 탐색 알고리즘 개발이 필요할 것이다. 둘째, 본 논문에서 제안하는 Dijkstra-variant는 일종의 그리디(greedy) 알고리즘인데 이런 형태의 알고리즘은 초반의 잘못된 경로 탐색으로 인해 좋지 못한 결과를 초래하기 쉽다. 따라서 진화 연산과 같은 전역탐색 알고리즘과 적절하게 결합할 경우 더 나은 성능 향상을 기대할 수 있다. 셋째, 항해사들에게는 디지털 시스템보다 예로부터 내려오는 항해 방법과 노하우에 대한 믿음이 커서, 이와 같은 경제운항시스템의 수용에 어려움이 있는 실정이다. 따라서 일방적인 시스템의 권고보다는 하이브리드 형태의 상호대화형 시스템의 개발을 하여 항해 보조 시스템으로서의 가치를 높이는 것이 필요하다. 넷째, 경제운항시스템에서는 연료소모량 추정 계산과 기상 예보의 정확도에 따른 성능 의존도가 높다. 특히 기상 예보의 경우, 언제나 오차가 존재할 수 있으므로, 이를 동적 보정하는 기법을 도입하여 성능의 강건함을 유지할 수 있도록 해야 할 것이다.

참 고 문 헌

- [1] Ho-Seop Jang and Yung-Keun Kwon, "An Economic Ship Routing System by Optimizing Outputs of Engine-Power based on an Evolutionary Strategy", *Journal of Korea Information and Communications Society*, Vol.36, No.4, pp.412-421, 2011.
- [2] Kyong Soon Choi, Myung Kyu Park, Jin Ho Lee, and Gun Il Park, "A Study on the Optimum Navigation Route Safety Assessment System using Real Time Weather Forecasting", *Journal of the Korean Society of Marine Environment and Safety*, Vol.13, No.2, pp.133-140, 2007.
- [3] Chinmaya Prasad Padhy, Debabrata Sen, and Prasad Kumar Bhaskaran "Application of wave model for weather routing of ships in the North Indian Ocean", *Natural Hazards*, Vol.44, No.3, pp.373-385, 2008.
- [4] J. K. Panigrahi, J. K. Tripathy, P. A. Umesh "Optimum tracking of ship routes in 3g-WAM simulated rough weather using IRS-P4 (MSMR) analysed wind fields", *Journal of the Indian Society of Remote Sensing*, Vol.36, No.2, pp.149-158, 2008.
- [5] R. L. Townsin, Y. J. Kwon, "Estimating the influence of weather on ship performance", *Transactions RINA*, Vol.135, pp.191-209, 1993.



주 상 연

e-mail : psycossise@gmail.com
2006년 3월~현 재 울산대학교 컴퓨터정보
통신학과 학사과정
관심분야: 인공지능, 최적화알고리즘

양 진 호

e-mail : aceyang@hhi.co.kr
1996년 부산대학교 조선해양공학과(학사)
1998년 부산대학교 조선해양공학과
(공학석사)
2000년~현 재 현대중공업 선박해양연구소 책임연구원
관심분야: 조선IT융합, 선박운동조정



조 태 정

e-mail : ctj22kr@gmail.com
2006년 3월~현 재 울산대학교 컴퓨터정보
통신학과 학사과정
관심분야: 인공지능, 모바일 프로그래밍



권 영 근

e-mail : kwonyk@ulsan.ac.kr
1999년 서울대학교 전산학과(학사)
2001년 서울대학교 컴퓨터공학부(공학석사)
2006년 서울대학교 컴퓨터공학부(공학박사)
2008년~현 재 울산대학교 전기공학부
조교수
관심분야: 최적화 이론 및 실제, 복잡계산시스템, 시스템생물학,
소셜네트워크 등



차 재 문

e-mail : zettabyte@naver.com
2006년 3월~현 재 울산대학교 컴퓨터정보
통신학과 학사과정
관심분야: 웹, SE