

해상 상태 및 선저여유수심을 고려한 연안 내 선박의 최적 항로 결정

이원희¹·유원철¹·최광혁¹·함승호^{2,†}·김태완¹
서울대학교 조선해양공학과¹
창원대학교 조선해양공학과²

Determination of Optimal Ship Route in Coastal Sea Considering Sea State and Under Keel Clearance

Wonhee Lee¹·Wonchul Yoo¹·Gwang-Hyeok Choi¹·Seung-Ho Ham^{2,†}·Tae-wan Kim¹
Department of Naval Architecture and Ocean Engineering, Seoul National University¹
Department of Naval Architecture and Marine Engineering, Changwon National University²

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ship route planning is to find a route to minimize voyage time and/or fuel consumption in a given sea state. Unlike previous studies, this study proposes an optimization method for the route planning to avoid the grounding risk near the coast. The route waypoints were searched using A* algorithm, and the route simplification was performed to remove redundant waypoints using Douglas-Peucker algorithm. The optimization was performed to minimize fuel consumption by setting the optimization design parameters to the engine rpm. The sea state factors such as wind, wave, and current are also considered for route planning. We propose the constraint to avoid ground risk by using under keel clearance obtained from electronic navigational chart. The proposed method was applied to find the optimal route between Mokpo and Jeju. The result showed that the proposed method suggests the optimal route that minimizes fuel consumption.

Keywords : Coastal route(연안 항로), Under keel clearance(선저여유수심), Optimal route(최적 항로), A* algorithm(A* 알고리즘)

1. 서론

일반적으로 선박이 운항할 때 연료 소모량 또는 항해 시간을 최소화 하는 최적의 항로를 도출하는 것을 목표로 한다. 연료 소모량 혹은 항해 시간을 최소화하기 위한 연구로 A* 알고리즘 (Park & Kim, 2014), Dijkstra 알고리즘 (Sen & Padhy, 2015), 등시선법 (Rho, 2013), 3차원 동적 프로그래밍 (Lin et al., 2013), 유전 알고리즘 (Szlapczynska, 2015)을 사용한 연구가 있다. 하지만 언급한 연구들은 모두 대양을 항해하는 선박을 대상으로 연구를 진행하였다. 대양에서는 장기간의 항해 시간과 넓은 항해 공간으로 인해 다양한 항로를 선택할 수 있으며, 섬이나 육지에 대한 충돌 회피는 고려하지만, 수심은 일반적으로 고려하고 있지 않다. 또한 연안에서의 속도 제한과 같은 규정도 대양을 운항하는 선박은 고려할 필요가 없다.

하지만 연안에서 운항하는 선박의 경우 주변의 수많은 섬을 회피해야 하는 동시에 좌초를 피하기 위해 수심 정보도 고려해야 한다. 이러한 특성 때문에 Dijkstra 알고리즘이나 동적 프로그래밍을 사용할 경우 격자를 세밀하게 구성해야 하며 격자로 구성된 그래프에서 연료 소모량을 계산할 경우 연산량이 많이 증가한다. 또한, 유전 알고리즘과 같은 전역 최적화 알고리즘의 경우, 좌초 위험이 없는 지역까지 검토하기 위해 불필요한 연산이 반복 수행되어 많은 시간이 소요된다. 실제 연안 여객선이나 어선을 대상으로 하는 서비스를 제공하기 위해서는 자동차의 내비게이션과 유사하게 빠른 시간에 최적 항로를 도출해야하기 때문에 빠른 연산이 가능한 알고리즘이 필요하다. 마지막으로 연안 선박에 관한 규정도 고려해야 한다.

본 논문에서는 최적 항로 알고리즘을 제안하는데 있어서 실제 연안 선박에 적용할 수 있도록 빠른 시간 내에 최적 항로를 계산할 수 있는가에 더욱 초점을 맞추도록 한다. 기존에 사용하는

알고리즘 중 격자를 생성하는 Dijkstra 알고리즘이나 연산이 반복 수행되는 유전 알고리즘은 빠른 계산 시간이 필요한 연안의 최적 항로에 적합하지 않다. 이를 보완하기 위해 격자를 생성하지 않는 A* 알고리즘을 목적에 맞게 변형한 Sparse A* 알고리즘을 사용하였다. 최적화 문제 중 목적 함수에 해당하는 연료 소모량을 추정하기 위해 해상 상태에 따른 선박의 성능을 수식으로 나타내었다. 제약 조건으로 전자해도 상의 선저여유수심(UKC: Under Keel Clearance) 정보를 활용하여 좌초 위험을 계산하였고, 연안 선박의 속도 규정을 반영하였다. 이렇게 정의된 문제들을 Sparse A* 알고리즘을 이용하여 빠르게 항로를 계산하고 Douglas-Peucker 알고리즘을 사용하여 불필요한 번침점을 제거하였다. 이어지는 장에서는 최적화 문제의 정식화하고, 이를 풀기 위한 알고리즘을 상세히 설명하였다.

2. 최적 항로 문제의 정식화

연안 내 항로를 계획할 때 좌초, 해상 상태, 연안 선박에 관한 규정을 고려해야 한다. 우선 좌초는 선박이 해저에 얹히거나 부딪힌 것을 의미한다. 연안 근처는 수심이 얕은 지역과 섬이 있어 좌초가 일어나지 않도록 항로를 계획해야 한다. 선저여유수심을 고려하여 좌초 위험을 계산하며 좌초 위험이 있는 지역은 지나가지 못하도록 설정하였다.

다음으로 고려한 요소는 해상 상태이다. 항해 거리가 증가할수록 연료 소모량이 증가하기 때문에 선박은 연료 소모량을 줄이기 위해 일반적으로 최단 거리 항로로 향한다. 하지만 최단 거리 항로의 해상 상태가 좋지 않으면 연료 소모량이 증가하여 우회하는 항로의 연료 소모량이 더 적게 나올 수 있다. 파고가 높거나 바람의 세가 강하면 선박의 저항이 증가하여 선박의 속력이 감소하는데 선박의 속력 감소를 보정하기 위해 더 많은 동력이 필요하게 되므로 연료 소모량이 증가하게 된다. 따라서 최단 거리 항로가 아닌 해상 상태가 좋은 항로로 우회하면 항로 길이가 길어도 연료 소모량이 줄어들게 된다.

마지막으로 고려한 요소는 연안 선박에 관한 규정이다. 항구 근처에서 기상, 조류 및 수심 때문에 발생하는 사고를 막기 위해 선박의 속력이 제한되는 구역이 존재한다. 예를 들어, 목포항 근처에는 선속 12노트 이하로 운행해야 한다는 세칙이 존재하는데 이를 최적화 제약 조건에 반영하여 연안의 실제 항해와 유사한 모델을 설계하였다.

최적 항로 문제는 앞서 언급한 좌초, 해상 상태, 연안 내 규정을 고려하여 여러 항로 대안 중 하나를 선택해야 하는 최적화 문제이다. 최적 항로를 구하기 위한 최적화 목적함수와 제약 조건을 다음과 같이 정의하였다.

$$\begin{aligned} & \text{Minimize } FOC(\mathbf{X}, \mathbf{n}) \\ & \text{Subject to } ETA(\mathbf{X}, \mathbf{n}) - ETA_{req} \leq 0 \\ & \quad v(\mathbf{X}_{rule}, \mathbf{n}) \leq v_{rule} \end{aligned} \quad (1)$$

최적화 모델에서 설계 변수는 항로의 위치 (\mathbf{X})와 엔진 회전

수 (\mathbf{n})이다. 목적함수로는 연료 소모량 (FOC: Fuel Oil Consumption)이 최소가 되도록 설정하였다 (식 (1)). 첫 번째 제약 조건은 선박이 주어진 시간 안에 도착해야 한다는 제약 조건으로 ETA (Estimated Time of Arrival)는 예상 도착시각, ETA_{req} 는 정해진 도착시각이다. ETA_{req} 는 선박마다 다르며 최적화 모델을 계산할 수 있도록 선박의 추진성능을 고려하여 설정되어야 한다. 두 번째 제약 조건은 항구 근처를 지나가는 선박의 속력을 제약해야 한다는 조건으로, v 는 선박의 속력, X_{rule} 은 제한 속력이 있는 지역, v_{rule} 은 제한 속력을 의미한다.

3. 연료 소모량 추정 방법

선박이 운항하면서 파랑, 바람, 조류를 만나면 선박의 속도 및 저항이 변하여 이를 고려한 동력을 계산해야 한다. 운항 시 필요한 동력을 계산하기 위해서는 정확한 부가 저항을 계산해야 하지만 계산 시간이 오래 걸리고 방법이 복잡하다. 그러므로 본 연구에서는 정수 중 저항과 주요 부가저항인 파랑 부가저항 및 바람 부가저항을 고려하여 제동 동력을 계산하고 선박의 속력에 조류를 반영하여 연료 소모량을 계산한다.

3.1 저항성능 추정

저항성능은 식 (2)를 기반으로 하여 계산한다 (ISO, 2002).

$$R_{TT} = R_{TS} + \Delta R_{Wave} + \Delta R_{Air} \quad (2)$$

식 (2)에서 R_{TT} 는 운항 중 선박의 전 저항, R_{TS} 는 정수 중 선박의 전 저항, ΔR_{Wave} 는 파랑 부가저항, ΔR_{Air} 는 바람 부가저항을 의미한다. 전 저항은 정수 중 선박의 전 저항, 파랑 부가저항, 바람 부가저항을 합한 값이다.

$$R_{TS} = \frac{1}{2} \rho S V^2 ((C_{FS} + \Delta C_F) \frac{S + S_{BK}}{S} + C_R) \quad (3)$$

$$C_{FS} = \frac{0.075}{(\log Rn - 2)^2} \quad (4)$$

정수 중 전 저항은 식 (3)을 기반으로 하여 계산한다 (ISO, 2002). 식 (3)에서 ρ 는 해수 밀도, S 는 침수 표면적, S_{BK} 는 빌지 킬 침수 표면적, C_{FS} 는 평판마찰계수, ΔC_F 는 마찰수정계수, C_R 은 잉여저항계수이다. 평판마찰계수는 ITTC 1957 추정식을 사용하며 식 (4)와 같이 계산한다. 식 (4)에서 Rn 은 레이놀즈수이다.

$$\Delta R_{Wave} = 2 \int_{-\pi}^{\pi} G(\alpha - \chi) \left[\int_0^{INF} S(f) \frac{\Delta r(f, \alpha)}{\zeta_A^2} df \right] d\alpha \quad (5)$$

$$\Delta R_{Air} = \frac{1}{2} \rho_A A_T V_R^2 C_{AA0} k(\theta) \quad (6)$$

파랑 부가저항은 식 (5)를 기반으로 하여 계산한다 (ISO,

2002). 식 (5)에서 f 는 입사파의 주파수, G 는 입사파의 방향 분포, $S(f)$ 는 입사파의 주파수 분포, a 는 입사파의 방향, $\Delta r/\xi$ λ_A 는 규칙파 중 응답함수이다. 바람 부가저항은 식 (6)을 기반으로 하여 계산한다 (ISO, 2002). 식 (6)에서 ρ_A 는 공기 밀도, A_T 는 수면 아래 정면 방향 투영면적, V_R 는 바람에 대한 선박의 상대 속력, C_{AAO} 는 정면 공기저항계수, $k(\theta)$ 는 풍향영향계수이다. 정면 공기저항계수와 풍향영향계수는 ISO 15016:2002를 기반으로 하여 계산한다.

3.2 추진성능 추정

연료 소모량을 계산하기 위해서 제동 동력을 계산해야 하며 제동 동력은 저항과 선박 계수를 이용하여 계산한다. 유효동력, 전달동력, 제동 동력은 식 (7), (8), (9)와 같이 표현된다.

$$P_E = R_{TT} V \tag{7}$$

$$P_D = P_E / \eta_D \tag{8}$$

$$P_B = P_D / \eta_T \tag{9}$$

P_E 는 유효동력, P_D 는 전달동력, P_B 는 제동 동력, V 는 선박의 속력, η_D 는 추진효율, η_T 는 축 전달효율이다. 식 (7)에서 선박의 속력은 시운전 자료를 기반으로 선박의 속력과 엔진 회전수 상관관계를 이용하여 계산한다.

$$\eta_D = \eta_H \eta_R \eta_O = \frac{1-t}{1-\omega} \eta_R \eta_O \tag{10}$$

추진효율과 축 전달효율은 식 (10)을 기반으로 하여 계산한다. η_H 는 선각효율, η_R 는 상대회전효율, η_O 는 프로펠러 단독효율, t 는 추진감소계수, ω 는 반류계수이다.

3.3 조류에 의한 선속 변화

선박의 속도에서 조류는 상대속도로 반영되어 연료 소모량을 추정하기 위해 사용된다. 조류가 있으면 표류 때문에 원하는 위치로 갈 수 없으므로 적절한 방향각의 조절이 필요하다. 방향각 조절 값 θ 는 식 (11)을 통해 구할 수 있다 (Park and Kim, 2014).

$$\frac{V \sin(p+\theta) + u_{cy}}{V \cos(p+\theta) + u_{cx}} = \tan p \tag{11}$$

$$\delta V = \sqrt{(V \cos(p+\theta) + u_{cx})^2 + (V \sin(p+\theta) + u_{cy})^2} - V \tag{12}$$

식 (11)에서 V 는 선박의 속력, u_{cx} 는 조류의 x축 성분, u_{cy} 는 조류의 y축 성분을, p 는 실제 선박의 진행 방향을 의미한다. 위 식을 방향각 조절 값 θ 에 대해 정리하면 선박 속력의 변화 δV 를 식 (12)와 같이 쓸 수 있다.

3.4 연료 소모량 추정

앞서 계산한 제동 동력과 조류에 의한 선속 변화를 고려하여 연료 소모량을 추정할 수 있다. 식 (13)은 전체 항로를 N 개로 나누어 구간별로 제동 동력과 선박의 속력을 계산하여 총 연료 소모량을 계산하는 식이다.

$$FOC = \sum_{i=1}^N f_{rate} P_B(n_i) \frac{l_i}{V(n_i) + \delta V} \tag{13}$$

식 (13)에서 N 은 구간, f_{rate} 는 시간, 마력당 연료 소모량, $P_B(n)$ 는 엔진 회전수가 n 일 때 제동 동력, l_i 는 항해 거리, $V(n)$ 는 엔진 회전수가 n 일 때 선박의 속력, δV 는 조류에 의한 선박의 속력 변화량이다. 엔진 회전수와 해상 상태에 따라 연료 소모량이 달라지며 제동 동력과 선박의 속력을 계산하여 연료 소모량을 계산한다.

4. UKC를 이용한 좌초 위험 계산

연안에서 항해하는 선박은 좌초 위험을 필수적으로 고려해야 한다. 좌초 여부를 판단하기 위해 선저여유수심을 계산하여 사용한다. 선저여유수심은 선체 아래의 여유 거리를 의미하며 선박이 항해할 때 충분한 선저여유수심이 확보되지 않으면 좌초가 일어날 수 있다 (Fig. 1). 이를 수식으로 표현하면 식 (14)와 같다. 식 (14)에서 H 는 수심, T_{max} 는 선박의 최대 흘수, UKC는 선저여유수심이다 (Galor, 2011).

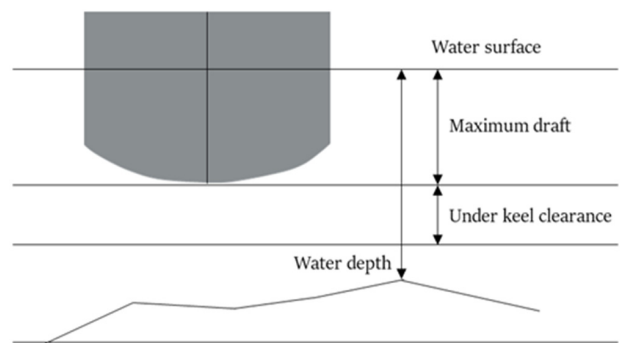


Fig. 1 Under keel clearance

$$H - T_{max} > UKC \tag{14}$$

흘수, 수심 측량의 불확실성, 밀도 변화, 선박의 6자유도 운동 등을 고려하여 선저여유수심을 계산해야 한다 (PIANC, 2014). 하지만 선박과 환경 조건이 변하지 않는다고 가정하여 선저여유수심은 흘수로부터 일정 비율(β)을 곱하여 식 (15)와 같이 계산한다. 파랑의 영향이 적은 수역일 경우에 선저여유수심 계수는 0.1 혹은 0.15를 사용하며, 파랑의 영향이 큰 수역일 경우에 선저여유수심 계수는 0.3 이상의 값을 사용한다 (Park and Chun, 2009). 연안 수역은 파랑의 영향을 크게 받으므로

선저여유수심 계수를 0.3으로 설정하였다.

$$UKC = \beta T_{max} = 0.3 T_{max} \quad (15)$$

5. 최적 항로 알고리즘

2장에서 설명한 것처럼 최적 항로를 계산하기 위해서는 설계 변수인 항로의 위치와 엔진 회전수를 계산해야 한다. 본 연구에서는 초기 항로를 탐색하기 위해 실공간에서 A* 알고리즘을 사용한다. 하지만 A* 알고리즘은 전역 최적 해에 가까운 지역 최적 해를 계산하는 알고리즘이기 때문에 초기 항로를 그대로 사용하기 어렵다. 따라서 초기 항로를 후처리하는 과정이 필요하며 후처리한 항로에 대해 최적의 엔진 회전수를 찾아야 한다. 정리하면, 최적 항로를 찾는 과정은 아래 4가지 단계로 나누어진다.

- Step 1. 출항지와 입항지 설정
- Step 2. Sparse A* 알고리즘을 사용하여 초기 항로 탐색
- Step 3. Ramer-Douglas-Peucker 알고리즘을 사용하여 초기 항로를 후처리
- Step 4. 연료 소모량을 최소화하는 최적 엔진 회전수 탐색

5.1 Sparse A* 알고리즘

A* 알고리즘은 경로 탐색에 많이 사용되는 알고리즘으로 전역 최적 해에 가까운 지역 최적 해를 찾으며 계산 시간이 짧아 많이 사용되는 알고리즘이다. 본 연구는 실공간에서 항로를 탐색해야 하므로 Dijkstra 알고리즘과 같은 그래프 기반 알고리즘을 사용하기 어려워 A* 알고리즘을 사용하였다. 격자 기반이 아닌 실공간에서 A* 알고리즘을 사용하여 경로를 탐색하기 때문에 잘못된 지역 최적 해를 찾아낼 가능성이 있다. 이러한 단점을 보완하기 위해 A* 알고리즘을 변형한 Sparse A* 알고리즘을 사용하였다 (Szczerba, 2000).

동일한 환경에서 A* 알고리즘은 전역 최적화 방법인 Dijkstra 알고리즘에 비해 계산 시간은 21.9%, 연산량은 30.7% 낮다 (Yukel & Sezgin, 2005). 또한, 최적 경로 문제에서 유전 알고리즘을 사용할 때 연산량이 크게 증가하는 부분은 장애물과 간섭을 계산하는 부분이다 (Szlupczynska & Smierzchalski, 2007). 연안에는 육지와 섬이 다수 존재하기 때문에 Dijkstra 알고리즘을 사용할 때 격자를 조밀하게 생성해야 하며 이는 연산량이 크게 증가하는 이유가 된다. 격자를 생성하지 않는 유전 알고리즘을 사용하면 장애물과 간섭을 반복적으로 계산하므로 불필요한 연산시간이 증가한다. 이러한 이유로 본 논문에서는 격자를 생성하지 않고 장애물과 간섭 계산을 최소화하여 빠른 시간 내에 최적 경로를 계산하는 Sparse A* 알고리즘을 사용하였다.

Sparse A* 알고리즘은 최소 힙(min heap)을 이용하여 잘못된 지역 최적 해를 찾는 경우를 방지하도록 설계된 알고리즘이다. 알고리즘을 사용하는 데 필요한 변수는 시작점, 도착점, 최소 탐색 길이, 최대 회전 각도, 탐색 방향 수, 최대 항로 길이가

다. 탐색 방향 수는 최대 회전 각도 내 이동 가능한 위치의 수를 의미한다. 탐색 방향 수가 많을수록 더 좋은 해를 찾을 수 있지만, 알고리즘의 연산 시간이 증가한다. 최대 항로 길이는 장애물을 만났을 때 너무 멀리 돌아가는 것을 방지하기 위해 설정한다. 알고리즘의 수행 과정은 다음과 같다.

- (1) Step 1. 비용이 0인 시작점을 갖는 최소 힙을 생성한다.
- (2) Step 2. 최소 힙 최상단 노드(root)의 경로를 불러와 마지막 위치를 현재 위치로 설정한다. 현재 위치에서 진행 방향으로 최대 회전 각도 내 최소 탐색 길이만큼 이동 가능한 위치를 탐색한다. 진행 방향이 없는 시작점에서는 360° 방향으로 이동 가능한 위치를 탐색한다.
- (3) Step 3. 현재 위치에서 이동 가능한 위치까지의 비용 함수를 계산한다. 시작점부터 현재 위치까지 이동한 거리와 현재 위치에서 도착점까지 직선거리의 합이 최대 항로 길이보다 작거나 같으면 계산한 비용과 경로를 최소 힙에 추가한다.
- (4) Step 4. 현재 위치에서 이동 가능한 위치가 없으면 최소 힙의 최상단 노드(root)를 제거한다.
- (5) Step 5. 현재 위치와 도착점의 거리가 최소 탐색 길이보다 작을 때까지 최소 힙의 최상단 노드에 대해 Step 2-4를 반복한다.
- (6) Step 6-1. 현재 위치와 도착점의 거리가 최소 탐색 길이보다 작으면 도착점을 경로에 추가하여 비용을 업데이트 하고 알고리즘을 종료한다.
- (7) Step 6-2. 현재 위치와 도착점의 거리가 최소 탐색 길이보다 크지만, 최소 힙이 비어 있어 탐색을 진행할 수 없을 때 알고리즘을 종료한다. 이러한 경우는 해를 찾지 못한 경우이며 최대 회전 각도나 탐색 회수와 같은 알고리즘 변수를 변경하여 알고리즘을 수행하면 경로를 찾을 수 있다.

Sparse A* 알고리즘은 Fig. 2와 같이 현재 위치에서 갈 수 있는 모든 위치에서 평가함수를 계산하고 평가함수가 가장 작은 위치로 이동하는 단계를 반복하여 수행한다. A* 알고리즘을 실공간에서 그대로 사용하게 되면 Fig. 2의 파란색 경로와 같이 지역 최적 해에 빠질 가능성이 있다. Sparse A* 알고리즘 중 최소 힙을 사용하여 지역 최적 해에 빠질 가능성을 최소화하였다.

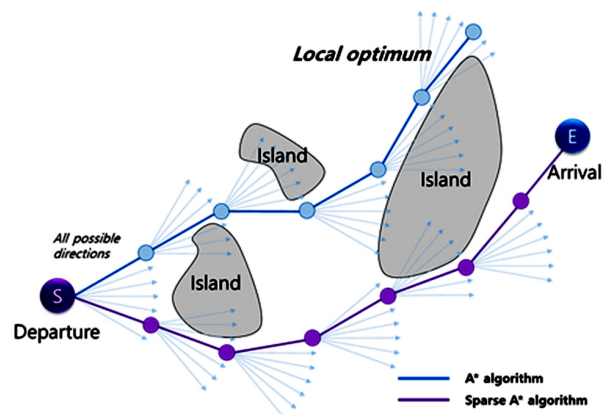


Fig. 2 Comparison between A* and Sparse A* algorithm

평가함수를 계산하기 위해 엔진 회전수를 설정해야 하며 알고리즘을 시작할 때 엔진 회전수는 가능한 엔진 회전수 중 가장 작은 값으로 설정한다. 출항지와 입항지까지 소요된 시간이 제약 조건을 위반하면 엔진 회전수를 증가시킨다. 소요된 시간이 주어진 항해 시간과 차이가 없을 때까지 A* 알고리즘을 반복 수행하여 최적 엔진 회전수를 계산한다.

식 (16)과 같이 평가함수 $f(x)$ 를 비용 함수 $g(x)$ 와 휴리스틱 함수 $h(x)$ 의 합으로 나타내었다. $g(x)$ 는 현재 위치에서 위치 x 까지 도달하는데 드는 비용이며 $h(x)$ 는 위치 x 에서 도착지까지 드는 비용이다.

$$f(x) = g(x) + h(x) \tag{16}$$

비용 함수는 식 (17)과 같이 정의하였다. 식 (17)에서 f_{rate} 는 시간, 마력당 연료 소모량, $P_B(n)$ 은 엔진 회전수 n 일 때 해상 상태를 고려하여 다음 위치로 가는데 필요한 제동 동력, d_x 는 현재 위치에서 다음 위치 x 까지의 거리, $v(n)$ 은 엔진 회전수 n 일 때 속력이다. 현재 위치에서 다음 위치 x 까지의 거리와 엔진 회전수를 이용하여 비용 함수를 계산한다.

$$g(x) = f_{rate} P_B(n) \frac{d_x}{v(n)} \tag{17}$$

휴리스틱 함수는 식 (18)과 같이 정의하였다. 휴리스틱 함수에서 $d_{x-final}$ 은 다음 위치 x 에서 도착지까지의 거리, $gr(x)$ 는 선저여유수심을 고려한 좌초 여부이다. 식 (19)와 같이 $gr(x)$ 를 계산하며 다음 위치 x 에서 좌초가 일어나면 $gr(x)$ 의 값을 ∞ 로 설정하여 탐색할 수 없는 위치로 설정하였다. 현재 위치에서 다음 위치 x 까지의 엔진 회전수와 다음 위치 x 에서 도착지까지의 속력이 같고 해상 상태가 같다는 가정을 적용하여 식 (18)의 제동 동력 $P_B(n)$ 은 비용 함수에서 계산한 결과를 사용하였다.

$$h(x) = f_{rate} P_B(n) \frac{d_{x-final}}{v(n)} + gr(x) \tag{18}$$

$$gr(x) = \begin{cases} 0 & \text{if } H(x) - T_{max} > UKC \\ \infty & \text{if } H(x) - T_{max} \leq UKC \end{cases} \tag{19}$$

5.2 Douglas-Peucker 알고리즘

Sparse A* 알고리즘으로 초기 항로를 탐색하면 많은 변침점이 생성되며 이 중 상당수는 불필요하다. 따라서 항로를 단순화하는 후처리 과정을 통해 불필요한 변침점을 제거해야 한다. 본 연구에서는 항로를 단순화하기 위해 Douglas-Peucker 알고리즘을 사용하였다 (Douglas & Peucker, 1973).

Douglas-Peucker 알고리즘 과정은 Fig. 3에 나타내었다. 초기 항로에서 시작점과 끝점을 잇는 직선을 구하고 그 직선에서 가장 거리가 먼 점을 찾는다 (Fig. 3(a), Fig. 3(b)). 그 거리가 설정한 임계 거리보다 크다면 거리가 먼 점을 그대로 유지한다. 초기 항로의 모든 점에 대해 이 과정을 반복 수행한다 (Fig. 3(c),

Fig. 3(d)). 임계 거리보다 작은 모든 점을 제거하면 Fig. 3(e)와 같이 단순화된 항로를 얻게 된다. 초기 항로를 단순화할 때 육지나 섬과 같은 장애물과 교차하는지 확인해야 하며 장애물과 교차할 경우 임계 거리가 작더라도 변침점을 제거하지 않는다.

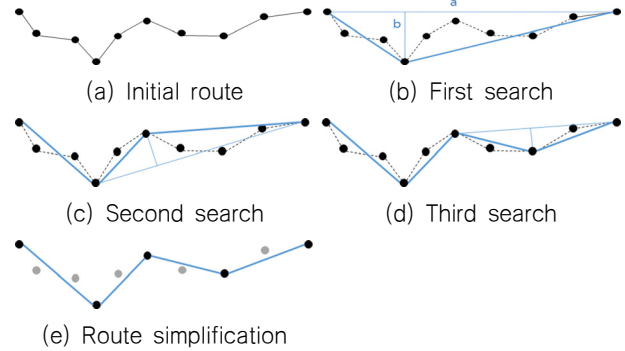


Fig. 3 Example of route simplification Douglas-Peucker algorithm

Fig. 4는 임계 거리에 따른 항로 단순화 결과이다. Fig. 4(a)는 적당한 임계 거리를 설정하여 최적 해의 손실 없이 단순화된 결과이며 Fig. 4(b)는 너무 큰 임계 거리를 설정하여 최적 해의 손실이 일어난 결과이다. 임계 거리를 크게 하면 최적 해의 손실이 일어날 수 있어 적당한 값을 선택해야 한다. 식 (20)에서 임계 거리 ϵ 는 Sparse A* 알고리즘에 사용한 최소 탐색 길이 (L)와 단순화하는 부분의 변침점과 가장 가까운 장애물과 거리 ($D(x)$) 중 작은 값을 이용하여 계산한다. 주변에 장애물 존재 여부에 따라 임계 거리가 달라지도록 하여 장애물 근처의 항로는 단순화되지 않도록 설정하였다.

$$\epsilon = \min(L, D(x)) \tag{20}$$

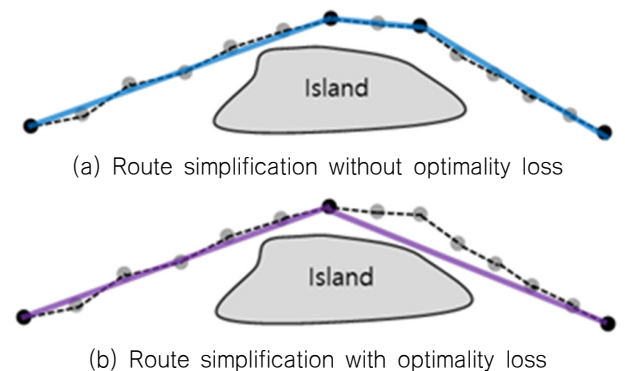


Fig. 4 Example of route simplification according to threshold distance

5.3 연료 소모량 최소화 알고리즘

앞서 언급한 방법을 이용하여 항로를 결정하였으므로 연료 소모량이 최소가 되는 최적의 엔진 회전수를 찾아야 한다. 2장에서 설명한 최적화 모델 중 항로 탐색 알고리즘을 수행하여 항로

위치가 결정되었으므로 연료 소모량이 최소가 되는 엔진 회전수를 찾아야 한다. 설계 변수인 항로 위치를 제외하고 엔진 회전수에 대해 최적화 모델을 구성하여 최적의 엔진 회전수를 계산한다. 최적화 모델은 목적함수와 제약 조건식 (1)에서 항로의 위치 (X)를 제외하여 설정한다.

위 연료 소모량 최소화 모델은 비선형이며 제약 조건이 있으므로 제약 조건이 있는 비선형 최적화 모델이다. 이러한 최적화 문제를 푸는 방법으로 SQP (Sequential Quadratic Programming) 방법을 사용하였다.

6. 사례 연구

최적 항로 계획 알고리즘을 사용하여 목포에서 제주까지 가는 항로를 항해하는 시뮬레이션을 수행하였다. 이 시뮬레이션에서는 해역 내 기상조건이 일반적이었을 때의 항로와 기상조건이 나쁠 때의 항로를 비교하고 분석한다. 기상조건이 일반적이었을 때 항로를 최단 거리 항로라 할 수 있으며 기상조건이 나쁠 때 항로가 어떻게 달라지는가에 대해 분석한다. 해역 내 기상조건은 BNI이 3에서 6 사이의 값을 가지도록 설정하였다.

6.1 시뮬레이션 선박

Table 1은 시뮬레이션에서 사용된 선박의 제원 및 주요 계수를 나타낸 표이다.

Table 1 Principal dimensions of the simulation ship

Factor	Value
Length between perpendiculars	189.0 m
Depth	9.7 m
Breadth	27 m
Draft	6.5 m
Displacement	15,180 m ³
Surface area	4,218 m ²
Block coefficient	0.5164
Quasi-propulsive coefficient	0.640

정수 중 선박의 속력을 초당 엔진 회전수 n 에 관한 함수로 표현하면 식 (21)과 같다. V 의 단위는 m/s이고 n 의 단위는 RPS (revolution per second)이다.

$$V(n) = -0.0354 + 4.0634n - 0.0343n^2 + 0.0059n^3 \quad (21)$$

6.2 시뮬레이션 조건

본 연구에서 시뮬레이션 조건은 다음과 같다. Sparse A* 알고리즘을 수행할 때 현재 위치에서 다음 위치까지의 거리인 최소 탐색 길이를 1 km로 설정하였다. 총 항해 시간의 제약 조건

은 4시간 30분이며 목포에서 10시에 출항하는 시나리오이다. 목포항 항만시설 운영세칙에 따라 목포항 주변에서 제한 속력 12노트를 넘지 않도록 제약 조건을 설정하였다. 주기관의 시간, 마력 당 연료 소모량은 125 g/ps·h이다. 육지, 섬, 수심 자료는 전자해도 정보를 기반으로 하였으며 기상정보는 Beaufort Number (BN)에 기반하여 4km 격자를 사용하였다. BN은 Table 2에 나타난 3~6 사이의 값을 사용하였으며 조류의 속도 범위는 2m/s 이내의 값을 설정하였다 (Huler, 2005).

Table 2 Beaufort number

Beaufort number	Wind speed	Wave height
3	3.4 - 5.5 m/s	0.6 - 1.2 m
4	5.5 - 7.9 m/s	1.2 - 2.0 m
5	8.0 - 10.7 m/s	2.0 - 3.0 m
6	10.8 - 13.8 m/s	3.0 - 4.0 m

6.3 시뮬레이션 결과

Fig. 5는 최적 항로 알고리즘을 이용하여 목포(A)에서 출항하여 제주(B)에 입항하는 항로를 계산한 결과이다. 본 시뮬레이션에서 선박은 A와 B 사이를 항해하며 기상조건을 고려하지 않은 항로 1과 기상 조건을 고려한 항로 2를 계산하였다. 항로 1과 항로 2의 계산 시간은 각각 6.8초, 8.7초였으며 기상 조건을 고려 여부로 인해 계산 시간의 차이가 발생했다. 기상조건은 Fig. 5(b)와 같이 특정 구역에서 BNI이 높았다.

기상조건에 따라 Sparse A* 알고리즘 결과가 다르게 나오며 비교를 위해 Fig. 5(a)와 같이 두 항로를 계산하였다. 항로 1은 기상조건을 고려하지 않고 최적 항로 알고리즘을 수행한 결과이다. 기상조건을 고려하지 않았기 때문에 정수 중 저항만 고려가 되며 이는 거리가 Sparse A* 알고리즘의 평가함수가 된 것과 같다. 따라서 항로 1은 거리를 Sparse A* 알고리즘의 평가함수로 설정하여 도출한 항로이다. 항로 2는 기상조건을 고려하여 최적 항로 알고리즘을 수행한 결과이다. 기상조건을 고려하여 6.1절과 같이 연료 소모량을 Sparse A* 알고리즘의 평가함수로 설정하여 알고리즘을 수행한다. Table 3은 각 항로의 기상 조건에 따른 연료 소모량을 계산한 결과이다.

Table 3에서 보이듯, 항로 1은 항로 2에 비해 항해 거리가 2.56% 감소했지만, 기상조건에 의해 생기는 부가저항 때문에 2.09% 더 많은 연료 소모량을 소비한다. 또한, 항로 1은 항로 2보다 기상조건이 나쁘므로 항로 1은 위험한 항로이다. 항로 2는 항해 거리가 항로 1보다 길지만, 기상조건에 의해 생기는 부가저항이 상대적으로 작아 더 적은 연료 소모량이 필요한 것을 확인하였다.

Table 3에서 보면, BNI이 6일 때 항로 1의 동력 증가량이 24~26%로 상당히 높은 것을 볼 수 있다. 하지만 항로 2의 연료 소모량은 항로 1의 연료 소모량보다 2.09%로 상대적으로 적게 나타난다. 이는 항로 2의 항해 거리가 항로 1의 항해 거리보다 길어 정해진 도착시각을 맞추기 위해 항로 2의 항해 속력이

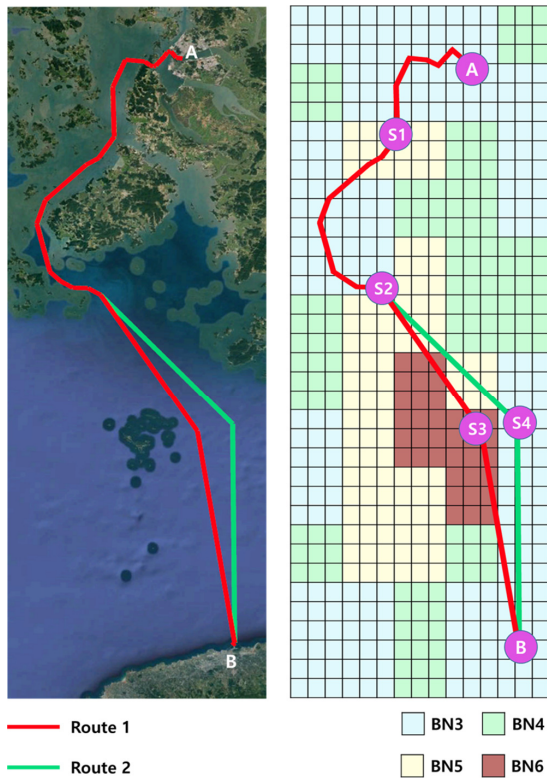


Fig. 5 (a) The route between Mokpo and Jeju (left)
 (b) Determination of route based on Beaufort Number (right)

Table 3 Result of optimal route algorithm from Mokpo to Jeju

Route 1	Increased ratio of power (%)	Sail distance (NM)	Fuel consumption (ton)
A-S1	4.30	20.44	2.29
S1-S2	9.39	31.40	5.58
S2-S3	24.45	23.76	4.82
S3-B	26.49	31.42	6.54
Sum	-	107.02 [100 %]	19.23 [100 %]
Route 2	Increased ratio of power (%)	Sail distance (NM)	Fuel consumption (ton)
A-S1	4.35	20.44	2.43
S1-S2	9.36	31.40	5.97
S2-S4	4.23	25.46	4.56
S4-B	3.95	31.92	5.87
Sum	-	109.22 [102.56 %]	18.83 [97.91 %]

높아졌기 때문이다. 운항 중 동력은 항해 속력의 세제곱에 비례하기 때문에 필요한 동력이 항해 속력이 높아지면 동력 역시 크게 높아진다. 항로 1의 S1-S2 부분과 항로 2의 S1-S2를 보면 이 차이를 알 수 있다. S1-S2를 항해하는 데 필요한 연료 소모량은 항로 1에서 5.58 ton이며 항로 2에서 5.97 ton이다. 따라서 기상조건에 의한 동력 증가량과 항해 거리 증가에 의한 동력

증가량이 상쇄되어 총 연료 소모량 차이가 상대적으로 적게 나오는 결과를 확인하였다.

7. 결론

본 연구는 전자해도 정보와 기상정보를 이용하여 최적 항로를 계산하는 방법을 제시하였다. 빠른 시간 내에 항로를 계산해야 하는 연안 선박의 특성을 고려하여 실공간에서 Sparse A* 알고리즘을 사용하였으며, Sparse A* 알고리즘의 평가함수는 연료 소모량으로 설정하였다. 수심을 고려한 UKC를 사용하여 좌초 위험이 있는 지역은 피하도록 항로를 탐색했다. 앞서 언급한 방법을 기반으로 하여 Sparse A* 알고리즘을 통해 도출한 항로를 단순화하고 기상정보와 선박 제원을 고려하여 연료 소모량을 최소화하는 최적의 엔진 회전수를 계산했다. 본 연구에서 제안한 방법을 사용하여 기상 상태 변화에 대한 최적 항로를 계산하였으며 연안에서 연료 소모량이 최소가 되면서 상대적으로 안전한 항로를 제시하였다.

본 연구에서 제안한 방법의 신뢰성과 정확성을 높이기 위해 정확한 연료 소모량 추정 방법에 관한 연구가 필요하며 실제 운항 사례에도 적용하는 연구가 필요하다. 또한, 연안에서 발생하는 충돌이나 전복 등의 위험을 고려하기 위해 선박 주변의 교통 상황이나 선박의 안정성을 최적 항로 알고리즘에 반영하는 연구가 필요하다.

후기

본 연구는 (a) 해양수산부 재원으로 해양과학기술진흥원과 한국형 e-Navigation 사업단(IMO 차세대 해양안전 종합관리체계 기술개발), (b) BK21+ 해양플랜트 창의인재양성 사업단, (c) 서울대학교 해양시스템공학연구소의 지원을 받아 수행된 연구 결과의 일부임을 밝히며, 이에 감사드립니다.

References

Douglas, D. & Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2), pp.112-122.

Galor, W., 2011. Determination of dynamic under keel clearance of maneuvering ship. *The Journal of Korbin*, 8(1), pp.53-60.

Huler, S., 2005. *Defining the wind: the beaufort scale, and how a 19th-century admiral turned science into poetry*. Broadway Books.

ISO, 2002. Guidelines for the assessment of speed and power performance by analysis of speed trial data. *ISO/DIS 15016*, pp.1-45.

Lin, Y.H., Fang, M.C. & Thong, S.K., 2013. The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic Elements. *Applied Ocean Research*, 43, pp.184-194.

Park, J. & Kim, N., 2014. A basic research on optimum under keel clearance for entrance channel. *The Korean Association of Ocean Science and Technology Societies 2009 Conference*, Changwon, Republic of Korea, 28-29 May 2009.

PIANC, 2014. *Harbour approach channels design guidelines*. PINAC report No. 978-2-87223-210-9.

Rho, M., 2013. Determination of an economical shipping route considering the effects of sea state for lower fuel consumption. *International Journal of Naval Architecture and Ocean Engineering*, 5(2), pp.246-262.

Sen, D. & Padhy, C.P., 2015. An approach for development of a ship routing algorithm for application in the north Indian ocean region. *Applied Ocean Research*, 50, pp.173-191.

Szczerba, R. & Glickstein, I., 2000. Robust algorithm for real-time route planning, *IEEE Transactions on Aerospace and Electronic System*, 36(3), pp.869-878.

Szlapczynska, J. & Smierzchalski, R., 2007. Adopted isochrone method improving ship safety in weather routing with evolutionary approach. *International Journal of Reliability, Quality and Safety Engineering*, 14(6), pp.635-645.

Szlapczynska, J., 2015. Multi-objective weather routing with customised criteria and constraints. *Journal of Navigation*, 68(2), pp.338-354.

Yukel, T. & Sezgin, A., 2005. *An Implementation of Path Planning Algorithms for Mobile Robots on a Grid based Map*. Ondokuz Mayıs University, Electrical & Electronics Engineering Department, 55139 Kurupelit-SAMSUN- TURKEY.



이원희



유원철



최광혁



함승호



김태완