

우선 순위 기반 쌍방향 다개체 동작 계획 방법

A Priority-based Interactive Approach to Multi-agent Motion Planning

지 상 훈[†], 정 연 수¹, 이 범 희²

Ji Sang hoon[†], Jung Youn soo¹, Lee Beom hee²

Abstract It is well known that mathematical solutions for multi-agent planning problems are very difficult to obtain due to the complexity of mutual interactions among multi-agent. Most of the past research results thus are based on the probabilistic completeness. However, the practicality and effectiveness of the solution from the probabilistic completeness is significantly reduced by heavy computational burden. In this paper, we propose a practically applicable solution technique for multi-agent planning problems, which assures a reasonable computation time and a real world application for more than 3 multi-agents for the case of general shaped paths in agent movement. First, to reduce the computation time, a collision map is utilized for detecting potential collisions and obtaining collision-free solutions for multi-agents. Second, to minimize the maximum of multi-agent task execution time, a method is developed for selecting an optimal priority order. Simulations are finally provided for more than 20 agents to emphasize the effectiveness of the proposed interactive approach to multi-agent planning problems.

Keywords : Multi-agents, Motion planning, Collision map, Priority

1. 서 론

빌딩 보안이나 무인화된 항만 해역 분야에서 사용되는 개체들의 운용에는 기능 특화된 소형 개체들로 구성된 그룹 개체가 복합 기능의 단일 개체보다 능률적이다. 그러나 그룹 개체의 동작 계획은 개체들에게 부여된 동력학적 제약성과 개체 수에 따라 높아지는 문제의 복잡도 때문에 수학적인 접근 방법으로 해결되기 어렵다. 이에 많은 연구자들이 수치적인 접근 방법에 기반하여 다개체들의 무 충돌 주행을 계획하였다^[1,2]. 대형 개체의 일부로서 각 개체들의 동작을 계획하는 집중식 주행 계획 방법은 개체의 수가 증가함에 따라 계산 량이 기하 급수적으로 증가하고 주행 계획 단계에서 개체들의 모든 동작 환경 정보를 알고 있어야 한다. 이러한 약점은 집중식 주행 계획 방법이 현실에 적용되기 어렵게 한다^[4]. 그러나 각

개체들이 독립적인 행동의 주체가 되는 분산식 주행 계획 방법에서 발생하는 상호 교착이나 국부 최소화 문제가 집중식 주행 계획 방식에는 발생하지 않는다. 그렇기 때문에 산업 영역이나 무인화된 항만 하역 공정과 같이 모든 개체들의 행동을 예측할 수 있는 제한된 환경에서는 집중식 주행 계획 방식이 다개체의 동작 계획에 유리하다^[3]. 초기 집중식 주행 계획 방법이 가지고 있었던 높은 계산 부하를 해결하기 위하여 긴급 탐색 임의 나무(RRT)나 확률 경로 방식(PRM)과 같은 방법들이 제시되었다^[9,12-14,18]. 그러나 이 방법들을 사용하여 얻은 주행 해의 최적성이 보장되지 않고 개체 수가 늘어났을 경우에는 실시간으로 처리되기에는 계산 량이 많다. 한편으로 다개체 동작 계획 과정에 특정 제약 조건을 설정하여 검색 공간을 줄임으로써 문제의 복잡도를 제한하는 우선 순위 기반 주행 계획과 같은 방법들이 제시되었다^[2]. 우선 순위 기반 다개체 주행 계획 방법인 충돌 맵은 공통 활동 영역을 가지는 두 대의 산업용 로봇의 최적화된 무 충돌 동작 해를 제시할 수 있다^[22,24-25,29]. 최근에 이 기법을 임의 형상의 주행 경로를 가지는 개체들의 주행 계획으

* 본 논문은 서울대학교 자동화시스템공동연구소와 두뇌한국 21

서울대학교 정보기술사업단의 지원에 의하여 연구되었음.

[†] 교신저자: 서울대학교 전기컴퓨터공학부 학생(robot91@snu.ac.kr)

¹ 서울대학교 전기컴퓨터공학부 학생(jung0818@snu.ac.kr)

² 서울대학교 전기컴퓨터공학부 교수(bhlee@snu.ac.kr)

로 확장시키는 방법이 제시되었다.

한편 개체들의 우선 순위 순서는 주행 계획의 결과(각 개체들의 목적지 도착 시간들 중 최대치)에 큰 영향력을 주는 다개체 시스템의 주요 설계 요소이다. 이제까지 우선 순위 분배 기법으로 개체들의 출발점 또는 도착점이 다른 개체의 주행 경로 위에 있는가를 판단하여 우선 순위를 결정하는 우선 순위 그래프 기법과 교차점에 먼저 도달하는 개체에 높은 우선 순위를 부여하는 방법이 개발되었다^[8,23]. 우선 순위 그래프는 충돌 회피를 보장하는 정적 조건에 의해 우선 순위를 분배하기 때문에 개체들의 동적 주행 특성을 반영할 수 없다. 그리고 후자의 방법은 개체의 수가 많은 경우에 적용하기 어렵다.

본 논문에서는 실시간 계산이 가능한 3대 이상의 다개체 주행 계획 방법을 제시하고자 한다. 이를 위하여 개체 간 충돌 여부를 빠르게 판단 할 수 있는 방법으로 다개체 주행을 네트워크 형태로 모델링 한다. 그리고 이 모델을 이용하여 무 충돌 주행 해를 연속적으로 구하는 방법을 제시한다. 마지막으로 개체 수가 증가함에 따라 빠르게 팽창하는 우선 순위 해 공간 안에서 합리적인 시간 내에 최적 주행 해를 보장하는 우선 순위를 도출하는 우선 순위 결정 방법을 제시한다.

본 논문은 다음과 같이 구성된다. 2장에서는 본 논문의 기본 가정들과 충돌 맵 기법에 대하여 기술한다. 그리고 3장에서는 충돌 지역에 대한 특성 변수를 정의하고 이 변수들을 이용하여 다개체의 무 충돌 주행 계획을 위한 네트워크 모델을 제시 한다. 4장에서는 최적 우선 순위를 결정하는 방법을 제시하고 5장에서는 모의 실험 결과를 제시하여 제시된 기법의 효율성을 입증한다.

2. 확장 충돌 맵과 충돌 회피

본 논문에서 사용되고 있는 기본 가정들과 임의의 형상의 주행 경로를 갖는 개체들의 주행 계획에 사용되는 확장 충돌 맵 이론을 소개한다.

2.1. 기본 가정들

[지능환경] 지능 환경은 유비쿼터스 센서 네트워크를 가지는 3D 공간이다^[19,20]. 경로 계획기와 동작 계획기를 가지고 있는 중앙 제어기는 이 센서 네트워크를 통하여 언제라도 각 개체들의 주행 상태 및 목표 위치를 확인할 수 있고, 이를 기반으로 다개체의 주행 계획을 할 수 있다.

[전역오프라인경로계획기] 모든 개체들의 목표 위치와 주행 경로들을 결정하는 전역 오프라인 경로 계획기는

각 개체들의 주행 경로들이 어떠한 개체의 출발점과 도착점을 지나지 않도록 설계할 수 있다.

[개체모델] 각 개체들은 계획된 주행 경로를 따라서 정확하게 이동할 수 있다. 각 개체들은 계산 편의를 위하여 원으로 모델링 한다. 개체가 차지하고 있는 모든 공간상의 영역을 식(1)과 같이 중심점과 반경으로 표현한다.

$$p(t) = [p_x(t), p_y(t)] \quad (1)$$

[개체주행특성] 개체들의 속도 및 가속도 특성은 식(2)와 같다. 각 개체들은 항상 물리적인 특성 안에서 최대 속도로 이동하는 것을 가정으로 한다. 이를 위하여 주행 경로의 곡률 반경에 따른 최대 속도가 제한되는 것이 발생되지 않도록 전역 오프라인 경로계획기가 주행 경로를 설계할 수 있다고 가정한다. 본 논문에서는 개체가 사다리꼴 속도 특성을 가진다고 가정한다.

$$|\dot{p}(t)| \leq v_{\max}, |\ddot{p}(t)| \leq a_{\max} \quad (2)$$

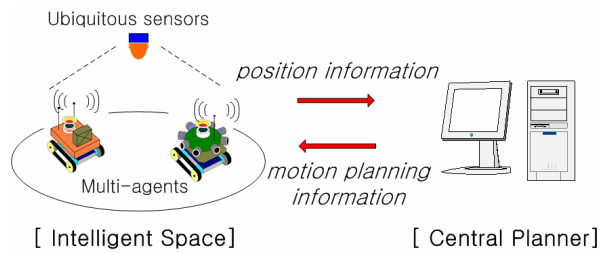


그림 1. 지능 환경과 중앙 동작 계획기

2.2. 충돌 맵을 이용한 충돌 예측

충돌 맵은 작업 영역을 공유하는 두 개체의 무 충돌 동작 계획 방법으로 주요 개념은 다음과 같다. 높은 우선 순위 개체를 agent 1, 낮은 우선 순위를 가진 개체를 agent 2라고 하고, 두 개체의 반경을 각각 r_1 과 r_2 라고 하자. agent 2에 대한 장애물 공간에서 agent 1은 r_1+r_2 의 반경을 가진 장애물로, agent 2는 중심 점에 놓여진 점 개체로 표현된다. 충돌 맵에서는 agent 1의 동작 궤적은 항상 변경되지 않고, 두 개체의 충돌이 예상될 경우에만 agent 2의 동작 궤적을 변경한다. agent 2의 경로가 반경 r_1+r_2 를 가지는 agent 1과 만난다면, 두 개체는 서로 만날 가능성이 있다. 그림 2에서 $p_1(k)$ 는 샘플링 시간 k에서의 agent 1의 중심의 위치이다. 샘플링 시간 k에서의 agent 2의 위치 $p_2(k)$ 는 다음과 같이 표현된다.

$$p_2(k) = p_2(k_0) + \lambda(p_2(k_f) - p_2(k_0)) \quad (3)$$

여기서, $0 \leq \lambda \leq 1$ 이고 $p_2(k_0)$ 는 agent 2의 초기 위치, $p_2(k_f)$ 는 agent 2의 도착 위치이다. 그림 2의 $\lambda_1(k)$ 과 $\lambda_2(k)$ 사이의 영역과 같이 agent 1의 경로와 겹치는 agent 2의 경로 부분을 충돌길이(Collision length)' 라고 명명한다.

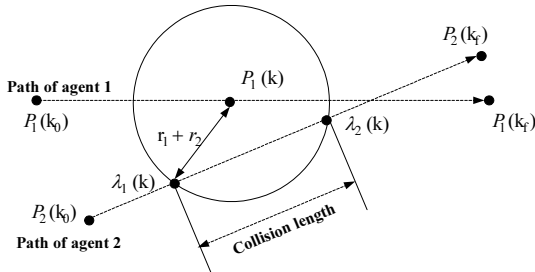


그림 2. 두 개체의 주행 경로와 충돌 길이

샘플링 시간 k에서 두 점 $p_1(k)$ 과 $p_2(k)$ 사이의 거리가 agent 1의 반경 (r_1+r_2)보다 크면 두 개체는 충돌하지 않는다. 이 무 충돌 조건을 수식으로 나타낸다면 다음과 같다.

$$(r_1 + r_2)^2 \leq \|p_1(k) - p_2\|^2 \tag{4}$$

식(3)을 이용하여 식(4)를 정리하면 식(5)와 같은 λ 의 2차 부등식이 된다.

$$\lambda^2 \|p_2(k_f) - p_2(k_0)\|^2 - 2\lambda(p_1(k) - p_2(k_0))^T \cdot (p_2(k_f) - p_2(k_0)) + \|p_1(k) - p_2(k_0)\|^2 - (r_1 + r_2)^2 \geq 0 \tag{5}$$

agent 1의 동작 계획은 변경되지 않는다고 가정하였기에, 모든 샘플링 시간 k에서의 $p_1(k)$ 는 미리 결정된다. 식(5)의 판별식이 음수일 경우 두 개체는 경로 특성상 충돌하지 않게 되는 것이고, 판별식이 0이거나 양인 경우에는 개체간 충돌이 발생할 가능성이 있다. 따라서, 경로 상 충돌 가능성이 있는 두 개체의 무 충돌 조건은 식(5)의 좌변 항이 음수가 되는 λ 의 영역이 0과 1사이 영역과 교집합이 없어야 한다는 것으로 표현된다. 두 개체 사이의 충돌은 그림 3을 이용하여 추상화시킬 수 있다. 그림 3에서 수직 축은 agent 2의 정규화된 이동 거리(Normalized traveled length)를 나타내고, 수평 축은 개체의 제어 시간이다. 매 샘플링 시간 k마다 검사된 중첩 경로들은 '충돌영역'(Collision region)'을 형성한다. agent 2의 TLVSTC(Traveled length versus servo time curve)가 충돌영역을 지나간다면 두 개체가 현재 동작 계획대로 이동할 경우 충돌할 것이다.

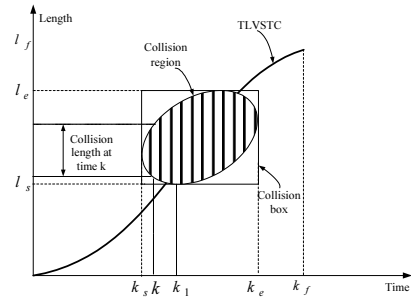


그림 3. TLVSTC와 충돌 영역

2.3. 시간 지연에 의한 충돌 회피

TLVSTC가 충돌영역을 회피하기 위한 방법으로 출발 시간 지연 방법이 있다[24]. 출발 시간 지연 방법에서는 수학적으로 경계를 나타내기 어려운 충돌영역 대신 이 영역을 둘러싸고 있는 충돌상자(collision box)를 사용한다. 충돌상자의 수평 경계는 그림 3의 agent 2 경로와 중첩되는 영역으로 agent 1이 진입하는 샘플링 시간 k_s 와 agent 1이 중첩 영역을 벗어나는 시간인 k_e 로 표현되고 수직 경계는 충돌영역의 최소 및 최대 값인 l_s, l_e 로 표현된다. 출발 시간 지연 기법은 충돌 회피를 위하여 그림 4에 나타난 바와 같이 k_e 와 k_1 사이의 차이만큼 agent 2의 출발 시간을 지연시킨다. 결과적으로 agent 2의 목표점 도달 시간은 식(6)과 같다.

$$k_f^1 = k_f + (k_e - k_1) \tag{6}$$

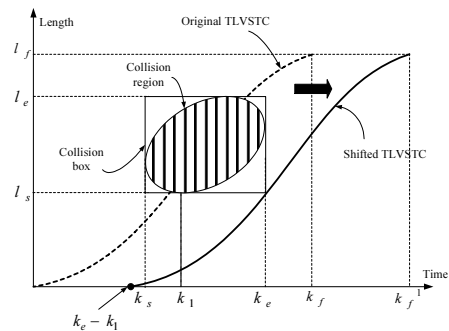


그림 4. 시간 지연(Time Delay)에 의한 충돌 회피

2.4. 임의 형상의 주행 경로에서의 충돌 회피

2.4.1. 확장 충돌 맵

초기 충돌 맵은 직선 주행 경로를 가지는 두 개체의 충돌 회피에 사용되었다. 이 기법을 임의 형상의 주행 경로를 가지는 두 개체의 충돌 회피로 확장시킨 확장 충돌 맵은 개체의 주행 경로를 경로 점(path node)들의 집합으로

로 표현하고, 이 점들의 집합에 대하여 충돌 예측 및 충돌 회피 계산을 수행하였다.

그림 5는 두 개체의 경로 점에서의 잠재적인 충돌을 예측하는 방법을 나타내고 있다. 두 경로 점 P_m 과 P_n 사이의 거리 d_{mn} 가 두 개체의 반경의 합 r 보다 작으면, 두 경로 점 사이에는 잠재적인 충돌 가능성이 존재한다.

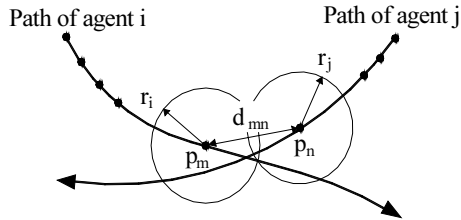


그림 5. 두 개체의 경로 점들을 이용한 잠재적 충돌 발견

잠재적인 충돌 가능성이 있는 경로 점들은 확장 충돌 맵에서 ‘충돌 점(collision point)’으로 나타난다. 확장 충돌 맵에서의 충돌 정보는 상위 우선 순위의 샘플링 시간과 하위 우선 순위 개체의 정규화된 이동 거리로 표현된다. 따라서, 그림 6에서의 충돌 점 $cp(t_n, l_m)$ 은 개체 i가 출발점으로부터 경로 점 P_m 까지의 정규화된 이동 거리와 개체 j가 P_n 에 도달하였을 때의 샘플링 시간으로 표현된다. 충돌 영역은 두 개체들의 경로 점들의 집합들 사이의 충돌 점들의 집합으로써 표현된다.

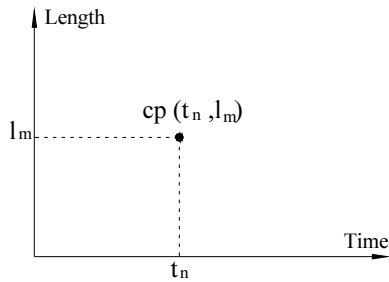


그림 6. 그림 5의 경로 점 P_m 과 P_n 에 상응하는 충돌 맵 상의 충돌 점

표 1. 그림 5와 그림 6에서의 변수

기호	의 미
r_i, r_j	개체 i와 개체 j의 반경
P_m, P_n	각 개체들의 경로 점m과 경로 점 n $1 \leq m \leq M, 1 \leq n \leq N$ M, N : 각 개체들의 경로 점의 총 수
d_{mn}	경로 점 P_m 와 P_n 사이의 거리
$cp(t_n, l_m)$	경로 점 P_m and P_n 에 상응하는 충돌 점

2.4.2. 경로 세그먼트 기반 충돌 회피

확장 충돌 맵은 주행 경로가 복잡할 경우에 계산할 경로 점들의 증가로 인하여 실시간 계산이 불가능하다. 이 문제점을 보완하기 위하여 충돌상자의 근사화 방법을 이용한다²⁶.

그림 7에서 agent i의 경로 점들은 agent j의 경로 점들에 대하여 세 개의 경로 그룹 $PS_{i,k-1}, PS_{i,k}, PS_{i,k+1}$ 로 나뉘고 있다. $PS_{i,k-1}$ 과 $PS_{i,k+1}$ 에 속한 경로 점들은 agent j의 어떤 경로 점과도 충돌하지 않는다. $PS_{i,k}$ 에 속한 경로 점들에 대하여는 agent j의 경로 점 일부와의 충돌 가능성을 내포하고 있기에 충돌 맵에서 충돌영역으로 표현된다. 충돌 회피를 위하여 상위 개체가 충돌 세그먼트에 존재하고 있는 동안에는 하위 개체가 충돌 관련 경로 점들을 지나가지 않도록 하위 개체의 출발을 지연시킨다.

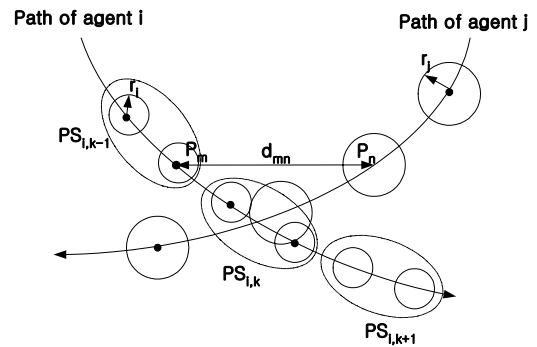


그림 7. 개체 i의 경로 세그먼트

2.5. 3대 이상의 다개체 충돌 회피

다개체의 주행 경로 교차점을 식(7)과 같이 나타낸다.

$$I_{ij}^k ; i > j \quad (7)$$

여기서, i와 j는 개체의 우선 순위를 의미한다. 상대적으로 낮은 j는 상위 개체의 번호를 나타낸다. 그리고, k는 개체 i가 개체 j와 가지는 교차점들 중에서 시작점을 기준으로 주행 경로를 따라 배열하였을 때 k번째 있는 것이다. 그림 8에서는 agent 3가 agent 1과 2개의 교차점 I_{31}^1 와 I_{31}^2 을 가지고 있다. agent 2는 agent 1과 교차점 I_{21}^1 을, 그리고 agent 3는 agent 2와 교차점 I_{32}^1 을 가지고 있다. 교차점 I_{ij}^k 에 해당되는 충돌 맵 상의 충돌영역은 R_{ij}^k 과 같이 표현한다. 각 개체에 한 개의 2D 충돌 맵이 주어지고 이 맵 위에 해당 개체보다 우선 순위가 높은 모든 개체들과의 충돌 가능성인 충돌영역들을 표현한다.

그러나 3 대 이상의 개체들에 대한 무 충돌 동작 계획은 단순한 2D 확장 충돌 맵의 조합으로 표현되지 않는다.

그림 9의 (a)에서 개체 2는 개체 1과의 충돌을 회피하기 위하여 출발을 지연시켰다. 이 결과 그림9의 (b)에서와 같이 개체 3의 충돌 맵에 정의되어 있는 충돌영역 R_{32}^1 가 옮겨지고 개체 2와 개체 3 사이의 충돌이 예상된다. 따라서, 개체 3는 변경된 충돌영역과의 충돌을 고려하여야 한다. 따라서, 3 대 이상의 무 충돌 주행을 위하여 상위 우선 순위 개체로부터 최하위 우선 순위의 개체까지 순차적으로 자신보다 높은 모든 상위 우선 순위 개체들과의 충돌영역들의 생성, 충돌 회피를 위한 지연 시간 도출 과정을 수행한다.

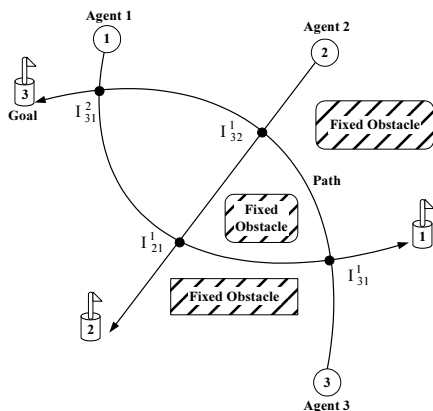
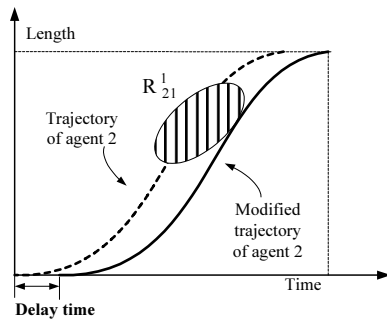
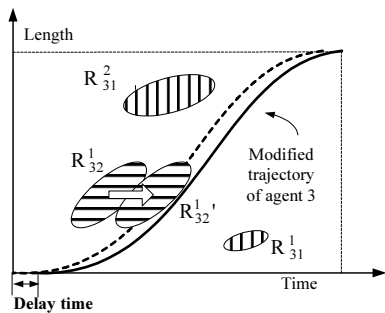


그림 8. 교차점을 가지는 3 개체들



(a) 개체 2의 충돌 회피



(b) 개체 3의 충돌 회피

그림 9. 그림 8에서의 개체 2와 개체 3의 충돌 회피

3. 다개체 주행 계획을 위한 네트워크 모델

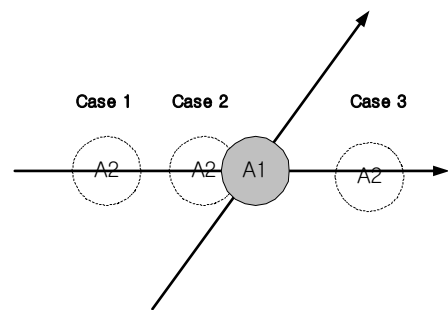
주행 경로가 복잡하게 얽혀 있는 3대 이상의 개체에 대한 실시간 무 충돌 주행 계획을 위하여 충돌상자의 특성 변수를 정의하고 무 충돌 주행 계획 과정을 네트워크 모델을 이용하여 구성한다.

3.1. 충돌 상자의 특성 변수

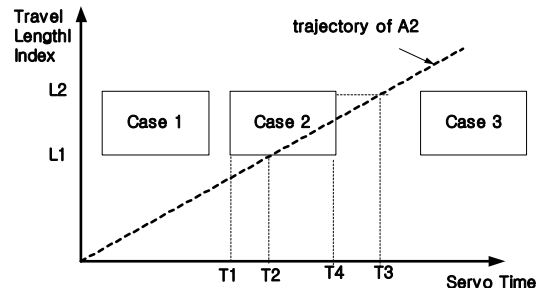
충돌상자에서 실제 충돌이 발생할 것인지를 판단할 수 있는 충돌상자 특성 변수를 정의한다.

3.1.1. 충돌상자의 특성 변수와 무 충돌 주행 조건

그림 10의 (a)에서 상위 개체(A1)와 하위 개체(A2)는 교차 점을 가지고 있다. A1과 A2의 교차점에서의 위치 관계는 다음과 같다. 첫째, A1이 교차 점을 포함하는 경로 세그먼트를 지나간 이후에야 A2가 교차 영역에 진입하는 경우(case1)이다. 둘째, 두 개체가 동시에 교차 영역에 존재하는 경우(case2)이다. 셋째, A1이 교차 점에 도달하기 전에 A2가 경로 점을 포함한 경로 세그먼트를 이미 지나간 경우(case 3)이다. L_1 은 출발점으로부터 A1과의 교차 영역을 이루고 있는 경로 점들까지의 이동 거리에 대한 최소 값이고, L_2 는 최대 이동 거리이다. 그림 10의 (b)에서는 충돌 맵에 세 가지 경우를 표시하였다. $T_k(k=1,2,3,4)$ 에 대한 설명은 표 2에 제시되었다.



(a) Collision Relations



(b) States of Collision Box

그림 10. 충돌 상자의 상태들

충돌은 M과 D가 모두 양일 때일 발생하므로, 개체는 자신보다 상위 개체와 관계된 모든 충돌상자들에 대하여 M과 D가 동시에 양이 되지 않도록 출발 시간 또는 속도를 조정하여야 한다. 그리고, M과 D는 두 개체의 우선 순위가 바뀌었을 경우 표2의 정의에 의하여 D와 M으로 서로 바뀌지는 대칭 관계가 있다.

표 2. 충돌 상자의 특성 변수

변수	의미
T_1	A1이 교차 점을 포함한 경로 세그먼트의 첫 경로 점에 도착하는 시간
T_2	A2가 교차 점을 포함한 경로 세그먼트의 첫 경로 점에 도착하는 시간
T_3	A2가 교차 점을 포함한 경로 세그먼트에서 벗어나는 시간
T_4	A1이 교차 점을 포함한 경로 세그먼트에서 벗어나는 시간
M	충돌상자의 무 충돌 상태를 유지하는 A2의 최대 출발 지연 시간. $T_3 - T_1$.
D	충돌이 예상된 충돌상자의 충돌 회피를 위한 최소 출발 지연 시간. $T_4 - T_2$.
T_{1d}	A1의 출발 지연 시간
T_{2d}	A2의 출발 지연 시간

표 3. 충돌 상자의 특성 변수와 충돌 여부

시간 변수 관계	M	D	충돌 가능성
$T_1 < T_4 < T_2 < T_3$	+	-	없음
$T_1 < T_2 < T_3 < T_4$	+	+	있음
$T_2 < T_1 < T_3 < T_4$			
$T_2 < T_1 < T_4 < T_3$			
$T_2 < T_3 < T_1 < T_4$	-	+	없음

3.1.2. 출발 시간 지연에 의한 특성 변수의 변화

그림 10의 A2의 출발 시간을 T_{2d} 만큼 지연시키는 경우에 A1의 동작 계획과 관련이 있는 T_1 과 T_4 은 변화가 없고 A2의 TLVSTC 곡선과 관련이 있는 T_2 와 T_3 는 출발 시간 지연만 발생하였기에 $T_2' = T_2 + T_{2d}$, $T_3' = T_3 + T_{2d}$ 로 변경된다. 그리고 M과 D는 $M' = M + T_{2d}$, $D' = D - T_{2d}$ 로 변경된다. 반대로 상위 개체 A1이 A1보다 상위인 개체와의 충돌 회피를 위하여 출발 시간을 지연(T_{1d})되는 경우, A2의 충돌 맵에 정의된 A1과 관련된 충돌 상자 변수 (M,D)는 M은 T_{2d} 감소하고 D는 T_{2d} 만큼 증가한다. 그리고 출발 지연을 이용하여 충돌 회피를 하는 경우에는 M과 D의 합이 일정하기에, (M,D)를 $(k_0 \cdot D, D)$ 로 정의하여

충돌 예측 및 충돌 회피를 위한 출발 시간을 계산할 수 있다. 여기서 k_0 는 충돌 회피 알고리즘을 적용하기 전에 각 충돌 상자에 정의된 시간으로 두 개체가 충돌 영역에 머무르는 시간의 합이다.

3.2. (M,D) 네트워크 모델과 충돌 회피

3.2.1. (M,D) 네트워크 모델

우선 순위가 정해진 개체들의 충돌 관계를 M과 D를 이용한 네트워크로 표현한다. 각 노드는 출발 시간 지연량, 순수 주행 시간을 특성 변수로 갖는다. 호의 방향은 상위 개체에서 하위 개체로 향하고 호의 비용은 충돌상자 특성 변수(M과 D)로 정의된다.

예를 들어 그림 11에서 개체 속도가 1m/sec, 2m/sec, 1m/sec이고 개체 반경이 5m라고 가정하자. 계산의 편의를 위해 개체들의 가감속 시간은 0이고, agent3는 경유 점 P_6 과 P_7 에서 감속 없이 방향 전환이 가능하다고 가정한다. 개체들의 우선 순위가 {1,2,3}일 때, (M,D) 네트워크 모델은 그림12와 같다.

그림 12에서 M_{ij}^k 과 D_{ij}^k 는 agent i와 상위 우선 순위 개체 agent j와의 k번째 충돌 상자의 특성변수들이다.

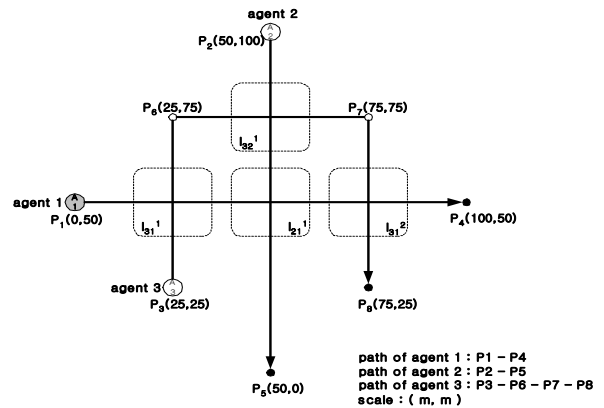


그림 11. 3대의 개체에 대한 주행 경로 정보

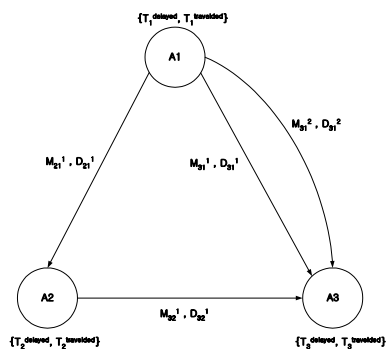


그림 12. 그림 11에 대한 (M, D) 네트워크 모델

3.2.2. 출발 시간 지연에 의한 충돌 회피

그림 13과 같이 상위 우선 순위 개체들로부터 유입되는 호들 중에서 충돌 가능성이 있는 호(M과 D가 모두 양인 호)들의 최대 D를 출발 시간의 지연(T_i^d)으로 결정한다. 노드의 특성 값이 변경된 후 호의 비용은 유입 호에 대하여는 M은 T_i^d 만큼 증가하고 D는 T_i^d 만큼 감소시킨다. 반대로 하위 순위 개체로 향하는 호에 대해서는 M은 감소시키고 D는 증가시킨다.

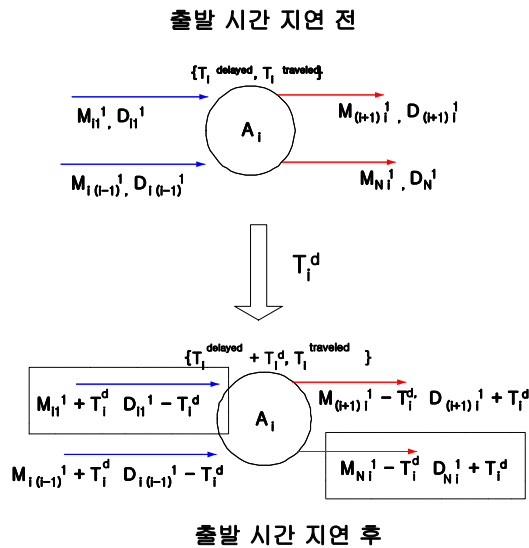


그림 13. (M, D) 네트워크 모델 변경

그림 14의 (a)는 그림11의 개체들에 대한 충돌 회피 전 (M,D) 네트워크이다. (a)에서 M과 D가 모두 양인 호가 존재하기 때문에 개체 간 충돌 발생이 예상된다. (b)에서는 agent 2의 충돌 회피를 위한 출발 지연 시간을 계산한다. agent 2로 들어 오는 호들 중 충돌 가능성이 있는 호들의 최대 D값은 20이므로 이 값을 agent 2의 출발 지연 시간으로 결정한다. (c)는 agent2에 대하여 충돌 회피를 위한 시간 지연의 결과이다. agent 1과 agent 2 사이의 (M,D) 값과 agent 2의 시간 지연이 바뀌었다. 그리고, 이 변화에 의하여 agent 2와 agent 3 사이에 정의된 호의 (M,D)값이 변경되었다. 마지막으로 agent 3에 대하여 충돌 회피를 위한 시간 지연을 계산하기 위하여 (c)에 나타난 agent 3로 오는 모든 호에 대하여 (M,D)들을 조사하여 D의 최대값 27.5를 시간 지연 량으로 결정하여 나타낸 것이 (d)이다. (d)에서 모든 호의 (M,D)가 충돌 가능성이 없기 때문에 모든 개체들의 무 충돌 주행 해가 도출되었고 충돌 회피 알고리즘은 종료한다. 각 개체들의 도착 시간은 각각 100sec, 120sec, 102.5sec으로 3 개체에 대한 최종 도착 시간은 120sec이다.

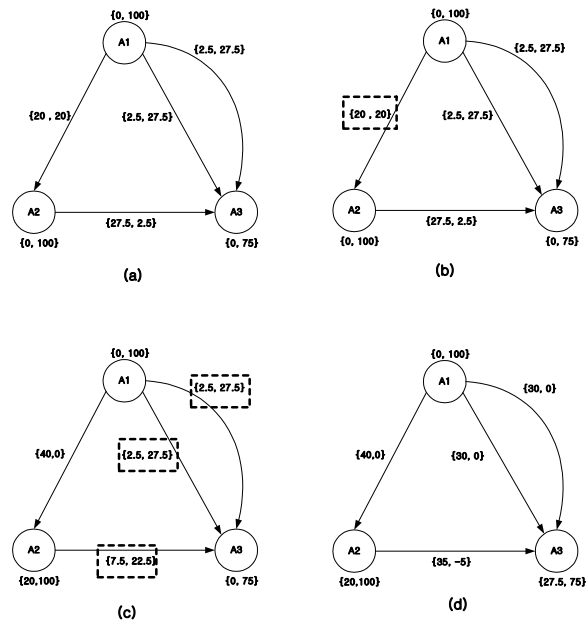


그림 14. 그림 11의 3 개체에 대한 (M,D) 네트워크 모델 기반 충돌 회피 과정

3.3 최적 우선 순위 결정 문제

3.3.1 목적 함수의 정의

본 연구는 N 개체 중에서 가장 늦게 목표점에 도착하는 개체의 도착 시간을 최소화 시키는 것을 목적으로 한다. 이를 수식으로 정리하면 식(8)과 같다.

$$\text{목적 함수}(J) = \max(T_1^{\text{completed}}, \dots, T_N^{\text{completed}})$$

$$\text{제한조건(무충돌 주행 조건)}: M_{ij}^k * D_{ij}^k \leq 0,$$

$$\forall k \text{ and } (i,j) \tag{8}$$

여기서, i와 j는 개체 번호이고 $T_i^{\text{completed}}$ 는 agent i의 도착 시간이고, M_{ij}^k 과 D_{ij}^k 는 agent i와 상위 우선 순위 개체 agent j와의 k번째 충돌 상자의 특성변수이다.

식(8)에서의 agent i의 도착 시간 $T_i^{\text{completed}}$ 는 식(9)와 같이 정의된다.

$$T_i^{\text{completed}} = T_i^{\text{traveled}} + T_i^{\text{delayed}} \tag{9}$$

T_i^{traveled} 는 agent i가 이동한 시간으로서 출발 시간 지연에 의한 충돌 회피 과정에서 변경되지 않는다. 그리고 T_i^{delayed} 는 충돌 회피에 의한 값으로서 상위 순위 개체에 따라 변경된다.

3.3.2. 동적 외관원 문제로의 전환

다개체 최적 주행을 위한 우선 순위 결정 문제는 작업의 시작과 완료를 교점 0으로 나타내게 하고 호의 비용(C)를 식(10)과 같이 정의하면 외관원 문제로 변경된다.

$$C_{0i} = T_i^{\text{traveled}}, C_{j0} = 0$$

$$C_{ij} = \max(0, T_i^{\text{completed}} - T_j^{\text{completed}}) \quad (10)$$

여기서, i, j 는 개체 번호이다. 그런데, 다개체 최적 주행을 위한 우선 순위 분배 문제에서 $T_i^{\text{completed}}$ 는 충돌 회피를 위한 시간 지연 T_i^{delayed} 를 포함하고 있다. 따라서, $T_i^{\text{completed}}$ 는 agent j 가 우선 순위 리스트의 위치에 따라 변동되고 호의 비용 C_{ij} 도 agent i 와 agent j 의 우선 순위 리스트의 위치에 따라 변동된다. 따라서, 최적 우선 순위 결정 문제는 동적 외판원 문제(Dynamic Traveling Salesman Problem, DTSP)이다.

하나의 우선 순위 순서에 대한 (M,D) 네트워크 모델이 주어지면, (M,D)의 대칭적인 특징을 이용하여 다른 순위 순서에 대한 (M,D)네트워크로의 전환이 가능하다. 예를 들어 그림 11의 예에서 가능한 우선 순위는 총 6가지로 각 우선 순위에 대한 (M,D) 네트워크 모델을 그림 15에 나타내었다. 그림 15에서 최적 주행을 제공하는 우선 순위는 총 4가지 {1-2-3}, {1-3-2}, {2-1-3}, {2-3-1} 이고 최종 도착 개체의 도착 시간은 120sec이다.

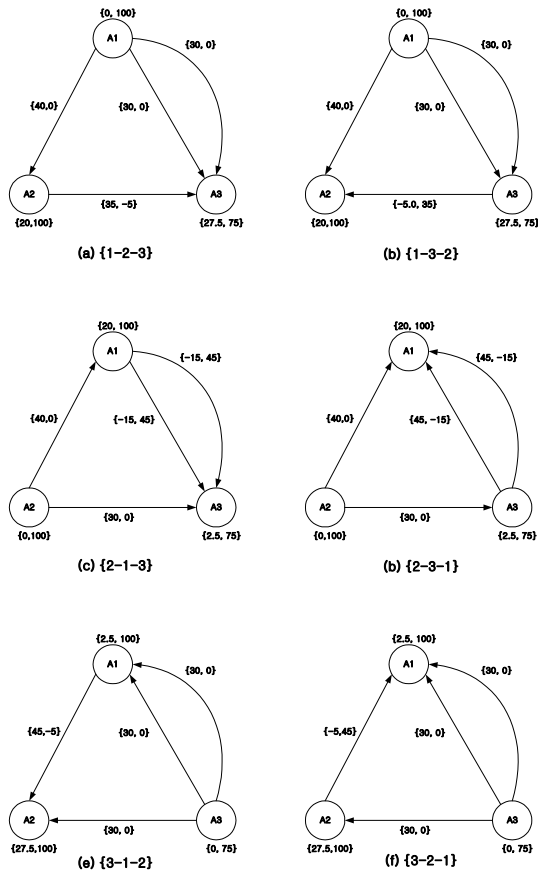


그림 15. 그림 11의 개체들의 가능한 모든 우선 순위에 따른 (M,D) 네트워크 모델

4. 최적 우선 순위 결정 기법

외판원 문제는 현재까지 다항식 알고리즘이 알려져 있지 않은 NP-hard 문제이다^[26, 34]. 해결 방법으로 국소 탐색 기법이나 신경망이나 유전자 알고리즘(Genetic Algorithm, GA)과 같은 발견적 기법들이 제시되었으나, 이 기법들은 도시 수가 적고, 도시 간 이동 비용이 대칭적이며 고정된 대칭 외판원 문제(Symmetric TSP, STSP)와 같은 제한된 경우에만 최적 해를 제공하고 있다. 그 외에 도시 수가 많거나 비대칭 외판원 문제(Asymmetric TSP, ATSP)인 경우, 그리고 DTSP인 경우에는 여러 기법들을 조합하여 사용한다^[29].

본 논문의 최적 우선 순위 결정 방법은 다음과 같다. 첫째, 주행 해의 성능에 영향을 주지 않는 개체들을 발견하여 제거해 줌으로써 우선 순위 탐색 공간을 축소한다. 둘째, 축소된 우선 순위 공간에 대하여 모든 우선 순위를 고려하여 최적 우선 순위를 결정한다. 만일 축소된 공간에 대한 전역 탐색이 실시간으로 힘든 경우에 한하여 임의적인 탐색 기법을 적용한다.

4.1. 검색 공간의 축소

식(8)의 목적 함수의 최적 해를 도출하기 위해 우선 순위 공간을 축소하는 방법으로 태스크 동작 계획에 사용되는 Bottom Level Earliest Deadline First (EDF) Scheduler를 사용한다^[33]. 이 기법을 다개체 주행 계획에 사용하기 위한 목적 함수는 식(11)과 같다.

$$J = \min(S_1, \dots, S_N)$$

$$S_i = E_i - T_i^{\text{completed}} \quad (11)$$

여기서, S_i 는 agent i 의 여유 시간으로 agent i 에 주어진 목표점 도착 시간에 대한 한계 시간(E_i)에서 agent i 의 도착 시간을 뺀 값이다.

Bottom Level EDF Scheduler를 사용하기 위하여 식(11)의 모든 개체의 한계 시간(E_i)을 주행 시간들의 최대값, 즉 $\max(T_1^{\text{traveled}}, \dots, T_N^{\text{traveled}})$ 로 정의한다. 그리고, T_i^{delayed} 는 확장 충돌맵에 의하여 만들어진 (M,D) 네트워크에서 긴급 개체(Urgent Group, UG)에 속한 개체와 agent i 사이에 연결된 모든 호의 비용(지연 시간과 여유시간)의 합으로 정한다. 예로서, 그림 16은 5개체 대하여 우선 순위가 {0,1,2,3,4,5}인 (M,D) 네트워크를 표시한 것이다. 이 그림에서 교차점이 없는 개체들 사이의 호는 생략하였다. 표 4는 그림 16의 5개체에 대한 Bottom EDF scheduler 알고리즘을 적용한 결과를 나타내었고, 최적 우선 순위는 {2-1-5-3-4}이다.

표 4. 그림 16의 개체들에 대한 Bottom level EDF Scheduler 적용 결과

횟수	$T_i^{completed}/S_i$					UG	E_i
	A1	A2	A3	A4	A5		
1	74/ -15	80/ -25	56/ -1	<u>53/</u> <u>2</u>	59/ -4	{1,2,3,4,5}	55
2	63/ -4	80/ -25	<u>49/</u> <u>6</u>	-	59/ -4	{1,2,3,5}	55
3	56/ -1	72/ -17	-	-	<u>51/</u> <u>4</u>	{1,2,5}	55
4	<u>47/</u> <u>8</u>	68/ -13	-	-	-	{1,2}	55
5	-	<u>55/0</u>	-	-	-	{2}	55

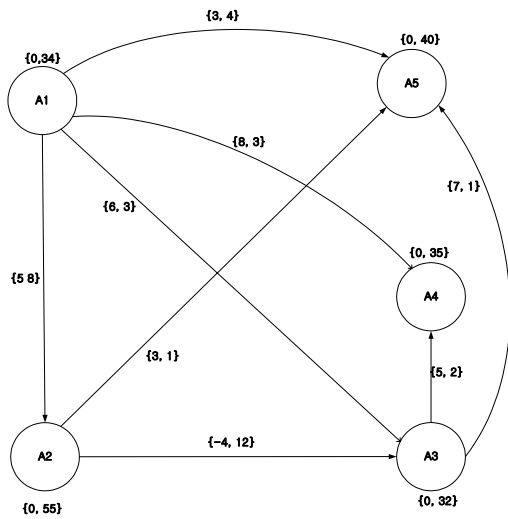


그림 16. 우선 순위가 {1-2-3-4-5}인 5개체들의 (M,D) 네트워크 모델

4.2. 축소된 공간에서의 최적 우선 순위 결정

본 논문에서는 DTSP에서의 국소 영역 최적 해를 구하는 방법으로 3-optimal 기법을 사용하고, 전역 영역 최적 해를 구하는 방법으로 GA 방법을 사용한다. 국소 탐색 기법으로 최단 거리 근접 기법이나 k-optimal 기법, 교점 교환 알고리즘 등이 있는데, 교점 교환 알고리즘의 경우 ATSP의 경우 성능이 좋지 않다고 알려져 있다^[26]. 이에 본 논문에서는 3-optimal 기법을 사용한다. 그리고 국소 최적 해 들에서 전역 최적 우선 순위를 결정하는 기법으로 GA를 사용한다.

4.2.1. 국소 영역 최적 해의 도출

Line이 개발한 k-optimal은 TSP에 대한 발견적 알고리즘의 하나이다^[34]. 임의의 해밀턴 순환로 t에서 k(k=2,3,...)개의 호를 제거하고, 순환로 t에 없었던 k개의 호로 대체하여 새로운 해밀턴 순환로를 만든다. 이 새로 만들어진

순환로를 원래의 순환로 t의 이웃(neighborhood)이라 하고, $N_k(t)$ 라고 표현한다. 임의의 순환로 t에는 많은 수의 이웃이 존재한다. 이 이웃들을 순차적으로 조사하여 개선된 순환로를 구한다. TSP의 문제에서 k가 3일 때가 계산 시간과 탐색 성능에서 최적이라고 보고되었다^[34]. 이에 본 논문에서는 3-optimal 기법을 사용한다.

4.2.2. 전역 영역 최적 해의 도출

본 논문에서는 전역 영역의 최적 해를 구하기 위하여 GA기법을 사용한다. DTSP문제에 GA를 효과적으로 적용하기 위해서 교배와 돌연변이 규칙에 다개체 최적 주행을 위한 우선 순위 결정 문제의 특성을 고려한다. 본 논문에서는 교배 연산자로서 거리 보존 교배 연산자(distance preserving crossover, DPX)를 사용하였다. DPX는 Boese의 TPS 탐색 공간 분석을 기초로 개발된 것으로 GA가 전역 영역 최적 해를 얻을 가능성을 높이는 탐색 영역에서의 점프를 수행하기 위하여, 두 부모 사이 거리와 생성된 자식과 한 쪽 부모와의 거리가 같도록 자식을 생성하는 방법이 DPX이다^[26].

본 논문에서는 조각난 연결 리스트를 연결하는 DPX의 탐욕적 재 연결(greedy reconnection-DTSP) 과정에서 agent i와 연결할 개체로서 충돌 상자의 특성 변수인 M과 D의 합이 가장 큰 개체로 선정한다.

돌연 변이 과정에서 제거되는 edge 수를 4와 7사이의 값 임의로 선택할 경우 ATSP에서 전역 영역 최적 해를 찾을 확률이 고정된 수를 사용하는 것보다 높다고 알려져 있다^[26]. 본 논문에서도 제거되는 edge 수를 4와 7사이에서 임의로 결정하였다. edge가 제거되어 조각난 우선 순위 경로를 탐욕적 재 연결 기법을 이용하여 조합한 후에는 국소 영역 최적 우선 순위를 구하기 위하여 3-optimal 알고리즘을 적용한다.

마지막으로 교환, 변이, 국소 최적화를 k번 반복하여 얻은 유전자 풀에서의 최적 값(다개체 주행 완료 시간의 최소값)의 변화율이 종료 계수보다 작으면 유전자 풀이 성숙되었다고 판단한다. 종료 계수는 개체 주행 시간($T_i^{traveled}$)의 분포와 시간 지연($T_i^{delayed}$)의 분포를 고려하여 선택한다.

5. 모의 실험

5.1. 시뮬레이터

본 논문에서는 우선 순위 결정 방법으로 다개체 주행 특성을 이용하여 최적 우선 순위의 탐색 영역을 축소하는 기법과 축소된 영역에서 전역 최적 해를 찾기 위한 유전자 알고리즘을 사용하였다. 제시된 기법들의 성능을

검증하기 위하여 Matlab을 이용하여 시뮬레이터를 개발하였다.

시뮬레이터의 입력으로 개체 수(N)과 개체 간 충돌 점 수(P)을 입력 받고, 주행 시간($T_i^{traveled}$)과 (M,D) 네트워크를 위한 P개의 충돌 점들을 총 $N \cdot C_2$ 의 연결 가능한 호들 중에서 분배한다. 분배된 연결 호들에 대해서 충돌 특성인 M과 D를 선택하는 과정에서는 M+D을 사용자가 원하는 분포로 생성하게 하였다. 개체 동작 환경은 100m x 100m의 정사각형이고, 모든 개체는 반경이 0.5m, 최대 이동 속도 1m/sec, 가속도는 편의를 위해 순시적으로 멈춘 상태에서 최대 속도로 가속할 수 있다고 가정하였다. 그리고 개체는 벽으로부터 최소 5m이상 떨어진 영역 안에서 균일 분포(uniformly distributed)되어 있다고 가정한다. 개체가 직교 좌표의 길(맨하탄 시티)로 이동한다고 가정하면, 개체의 평균 주행 시간은 평균이 60인 균일 분포를 갖는다. 이에 각 체의 주행 시간을 [40, 80] 사이의 균일 분포로 정하였다. 그리고, 두 개체가 교차점을 중심으로 충돌 경로 세그먼트에 머무르는 시간의 합을 [4, 20] 사이의 균일 분포를 갖는다고 가정하였다. 이것은 개체의 속도가 1m/sec이고 직교 좌표 이동 방식을 따르기 때문에 최소 값을 4로 정하였다. 그리고 최대 값인 20인 경우에는 각 개체의 주행 시간이 80인 것을 감안할 때, 두 개체의 경로가 많이 겹치고 있는 상황이다. 마지막으로 지연 시간(D)는 두 개체가 충돌 경로 세그먼트에 머무르는 시간의 합의 분포를 고려하여 [-8, 8] 사이의 균일 분포를 갖는다고 가정하였다.

GA에서의 한 루트 안에서의 교배 횟수를 10번으로, 초기 우선 순위 리스트를 임의적으로 개체 수만큼 설정해 주었고, 돌연 변이의 확률은 0.2로 설정해 주었다. 그리고, 국소 지역의 최적 해를 구하는 3-Optimal 기법에서 최대 반복 횟수를 (개체 수*(개체 수-1))로 제한하였다. 유전자 교체 과정에서 형질이 좋은 우수 유전자를 보호하기 위한 확률은 0.2로 설정하였다.

5.2. 실험결과

그림 15에서와 같이 충돌 맵 기반 다개체 동작 계획의 결과는 우선 순위에 상관없이 항상 무 충돌 해를 제시한다. 그리고 우선 순위가 결정되었을 경우 무 충돌 주행 해의 도출 과정은 단순한 산술 계산의 순차적인 조합으로 이루어졌기 때문에 계산 시간이 매우 짧다. 실제로 우선 순위가 결정된 3개체의 무 충돌 주행 해의 도출 시간은 0.1msec이하였다. 이에 본 논문에서의 모의 실험 결과는 무 충돌 최적 주행 해의 도출 시간만을 제시한다.

5.2.1. 검색 공간의 축소

Bottom-Level EDF Scheduler를 적용한 결과는 그림 17과 같다. 실험은 총 개체 수를 20, 30, 40, 50으로 변경하고 각 개체 수의 3배수의 교차 점을 가지도록 하였다. 그림 17은 각 단계에서 100번씩 수행한 결과의 평균이다. X축은 교차점 수를 총 개체수로 나눈 값이고, Y축은 긴급 개체 수를 총 개체수로 나누어 정규화한 값이다. 그림 17에서 개체 수에 상관없이 정규화된 교차점 수가 2(한 개체당 평균 교차점 수)인 근방에서 긴급 개체 수가 전체 개체 수의 0.5정도 이다. 결국 본 논문의 탐색 영역 축소 기법이 개체가 평균적으로 다른 개체와 2번 만나는 상황에서 유효 개체를 전체 개체의 절반으로 줄인다는 것을 확인되었다. 그리고 교차 점에 두 개체가 머문 시간이 적은 경우에는 UG에 남는 개체 수의 비율은 더 적을 것으로 예상된다.

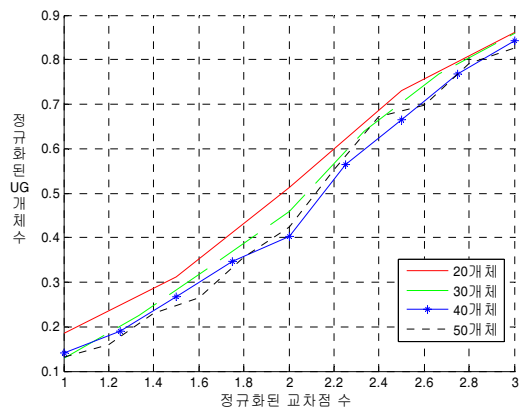


그림 17. Bottom-Level EDF scheduler 적용 결과

5.2.2. 축소된 공간에서의 최적 우선 순위 탐색

축소된 검색 공간에서의 최적 해를 긴급 개체 그룹에 개체 수가 30개인 경우 100개를 선택하여 GA/3-Optimal 기법을 수행하였다. 유전자 풀의 성숙도에 대한 판별 지수를 주행 시간의 최대값을 고려하여 0.0001로 정하였고, 각 판단 단계에서 20번의 교배 및 돌연 변이 과정을 수행한다.

100번을 수행한 결과는 그림 18에 나타난 바와 같이 반복 횟수의 평균은 2.47이고 표준 편차는 0.96이다. 그리고, 도출된 최적 해를 각 단계에서의 최대 주행 시간으로 나눈 값의 평균이 1.05이고 표준 편차는 0.03이다. 1.05의 평균은 최대 주행 시간이 79초와 80초 사이의 값이기 때문에, 실제 4초 정도 지연된 시간이다. 따라서, 본 논문에서 제시된 방법이 최적 해를 제공하고 있다고 판단된다.

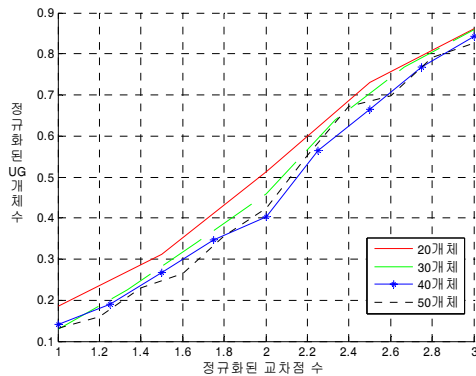


그림 18. GA/3-Optimal 적용 결과

6. 결 론

본 논문의 목적은 현실적인 다개체의 무 충돌 동작 계획의 개발이었다. 이를 위하여 우선 두 개체의 무 충돌 주행 계획에 적용되던 충돌 맵 기법을 네트워크 모델 형식을 이용하여 3대 이상의 개체에 적용되도록 하였다. 그리고 개체 수가 증가함에 따라 빠르게 팽창하는 우선 순위 공간에서 합리적인 시간 내에 최적 우선 순위를 구하는 방법을 제시하였다. 모의 실험 결과를 통하여 제시된 기법이 실제 환경에 사용 가능할 만큼 검색 시간이 짧다는 것을 확인하였다.

다만 개체 수가 많고 개체들 간 주행 경로의 상호 연관성이 높은 경우 실시간으로 우선 순위를 결정하기에는 긴급개체의 수가 여전히 많았다. 따라서 충돌상자들과 개체 주행 경로의 상관 관계를 고려한 우선 순위 결정 방법이 연구될 필요가 있다.

참 고 문 헌

- [1] J.C.Latombe, "Robot Motion Planning", Kluwer academic publishers, 1991.
- [2] K.Fujimura, "Motion Planning in Dynamic Environment", Springer-Verlag, NewYork, 1991.
- [3] M.B.Dias and A.Stentz, "Opportunistic Optimization for Market-Based Multirobot Control", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2002.
- [4] J.H.Reif, "Complexity of the Mover's Problem and Generalizations", Proc. of the 20th IEEE symp. on Foundations of Computer Science, pp.412-427, 1979.
- [5] J.F.Canny, "The Complexity of Robot Motion Planning", MIT Press, 1988.
- [6] S.Akella and S.Hutchinson, "Coordinating the Motions of Multiple Robots with Specified Trajectories", Proc. of the 2002 IEEE Int. Conf. on Robotics & Automation, Washington DC, May, 2002.
- [7] R.R.Murphy, "Introduction to AI Robotics", The MIT Press, 2000.
- [8] 고낙용, 서동진, "다중 이동 로봇의 중앙 감시에 의한 충돌 회피 동작 조정 방법", 전기공학회지, vol. 32D, no. 4, pp. 223-232, 2003.
- [9] Sanchez G. and Latombe J.C, "Using PRM Planner to Compare Centralized and Decoupled Planning for Multiple-robot Systems", Proc. of IEEE Int. Conf. on Robotics and Automation, 2002.
- [10] Y.K.Hwang and N.Ahuja, "Gross Motion Planning - A Survey", ACM Computing Surveys, vol. 24, no. 3, pp. 219-291, September, 1992.
- [11] J.Barraquand, L.Kavraki, J.C.Latombe, T.Y.Li and P.Raghavan, "A Random Sampling Scheme for Path Planning," Int. Journal of Robotic Research, vol. 16, no. 6, pp. 759-774, Dec, 1997.
- [12] J.Barraquand and J.Latombe, "Robot Motion Planning : A Distributed Representation Approach", Int. Journal of Robotics Research, vol. 10, pp. 628-649, 1991.
- [13] L.Kavraki, P.Svestka, J.Latombe, and M.Overmars, "Probabilistic Roadmap for Fast Path Planning in High-Dimensional Configuration Spaces", IEEE Trans. on Robotics and Automation, vol. 12, pp. 566-580, 1996.
- [14] T.Y.Li and H.C.Chou, "Motion Planning for a Crowd of Robots", Proc. of IEEE Int. Conf. on Robotics & Automations, September, 2003.
- [15] K.S.Barber, T.H.Liu, and S.Ramaswamy, "Conflict Detection During Plan Integration for Multi-Agent Systems", IEEE Transactions on Systems, Man, and Cybernetics, vol. 31, no. 4, pp. 616-627, August, 2001.
- [16] M.M.Quottrup, T.Bak, and R.Izadi-Zamanabadi, "Multi-Robot Planning : A Timed Automata Approach", Proc. of IEEE Int. Conf. on Robotics and Automation, April, 2004.
- [17] K.Azarm and G.Schmit, "Conflict-free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation", Proc. of IEEE Int. Conf. on Robotics and Automation, April, 1997.
- [18] Guang Song, Thomas S., and Amato N.M., "A General Framework for PRM Motion Planning", Proc. of IEEE Int. Conf. on Robotics and Automation, 2003.
- [19] J.H.Lee and H.Hashimoto, "Intelligent space", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 2, pp. 1358-1363, Nov. 2000.
- [20] H.Norihiro, K.Kiyoshi, M.Kehji, and S.Yasuyuki, "Collaborative Capturing of Experiences with Ubiquitous Sensors and Communication Robots", Proc. of IEEE Int. Conf. on Robotics & Automation, pp. 4166-4171,

- September, 2003.
- [21] M.Erdmann and T.Lozano-Perez, "On Multiple Moving Object", Proc. of IEEE Int. Conf. on Robotics and Automation, vol. 3, pp. 1419-1424, 1986.
- [22] Haoxum Chen, Luh P.B., and Lei Fang, "A Time Window Based Approach for Job Scheduling", Proc. of IEEE Int. Conf. on Robotics and Automation, 2001.
- [23] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun, "Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems", Proc. of IEEE Int. Conf. on Robotics and Automation, 2001.
- [24] B.H.Lee and C.S.G.Lee, "Collision-Free Motion Planning of Two Robots", IEEE Transactions on Systems, Man, and Cybernetics, vol. 17, no 1, pp. 21-31, January/February, 1987.
- [25] C.Chang, M.J.Chung, and B.H.Lee, "Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time", IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 3, pp. 517-522, March, 1994.
- [26] Bernd Freisleben and Peter Merz, "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems", IEEE, pp.616-621, 1996.
- [27] Gang Feng and Christos Douligeris, "Using Hopfield Networks to Solve Traveling Salesman Problems Based on Stable State Analysis Technique", IEEE, pp.521-526, 2000.
- [28] Yoshikane Takahashi, "A Mathematical Framework for Solving Dynamic Optimization Problems with Adaptive Networks", IEEE Transactions on Systems, Man, and Cybernetics, vol. 28, no. 3, pp. 140-143, 1998.
- [29] Christos Dimopoulos and Ali M.S. Zalzal, "Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions, and Comparisons", IEEE Trans. on Evolutionary Computation, vol. 4, no.2, pp93-113.
- [30] Huai-Juang Tasi, Jinn-Moon Yang, Yuan-Fang Tsai, and Cheng-Yan Kao, "An Evolutionary Algorithm for Large Traveling Salesman Problems", IEEE Transactions on Systems, Man, and Cybernetics, vol. 34, no. 4, pp. 1718-1729, August, 2004.
- [31] Xue-Song Yan, Li-Shan Kang, Zhi-Hui Cai, and Huili, "An Approach to Dynamic Traveling Salesman Problem", Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 2004, pp.2418-2420.
- [32] Hui-Dong Jin, Kwong-sak Leung, Man-Leung Wong, and Zong-Nen Xu, "An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling

Salesman Problem", IEEE Transactions on Systems, Man, and Cybernetics, vol. 33, no. 6, pp. 140-143, 2003.

- [33] J.Park, M.Ryu, and S. Hong, "Fair Real-Time Resource Allocation for Internet End System's QoS Support", Lecture Notes in Computer Science, vol. 2713, pp.764-769, 2003.

- [34] 강맹규, 네트워크와 알고리즘, 박영사



지 상 훈

- 1995 서울대학교 제어계측공학과 (학사)
 1997 서울대학교 전기공학부 (석사)
 2002~현재 서울대학교 전기컴퓨터공학부 박사과정

관심분야: Multi-agent system, Ubiquitous robot



정 연 수

- 1993 해군사관학교 해양물리학과 (학사)
 1997 Univ. of Colorado 전기컴퓨터공학부 (석사)
 2005~현재 서울대학교 전기컴퓨터공학부 박사과정

관심분야: Multi-AUVs, Sensor based robot control



이 범 희

- 1978 서울대학교 전자공학과 (학사)
 1980 서울대학교 전자공학과 (석사)
 1985 Univ. of Michigan Computer, Information & Control Eng. (박사)

1987~현재 서울대학교 전기컴퓨터공학부 교수
 관심분야: Multi-agent system, Flexible automatic system