

4-19 패스 파인딩 (Path finding)

패스 파인딩 (path finding)



패스 파인딩 vs. 모션 플래닝

패스 파인딩

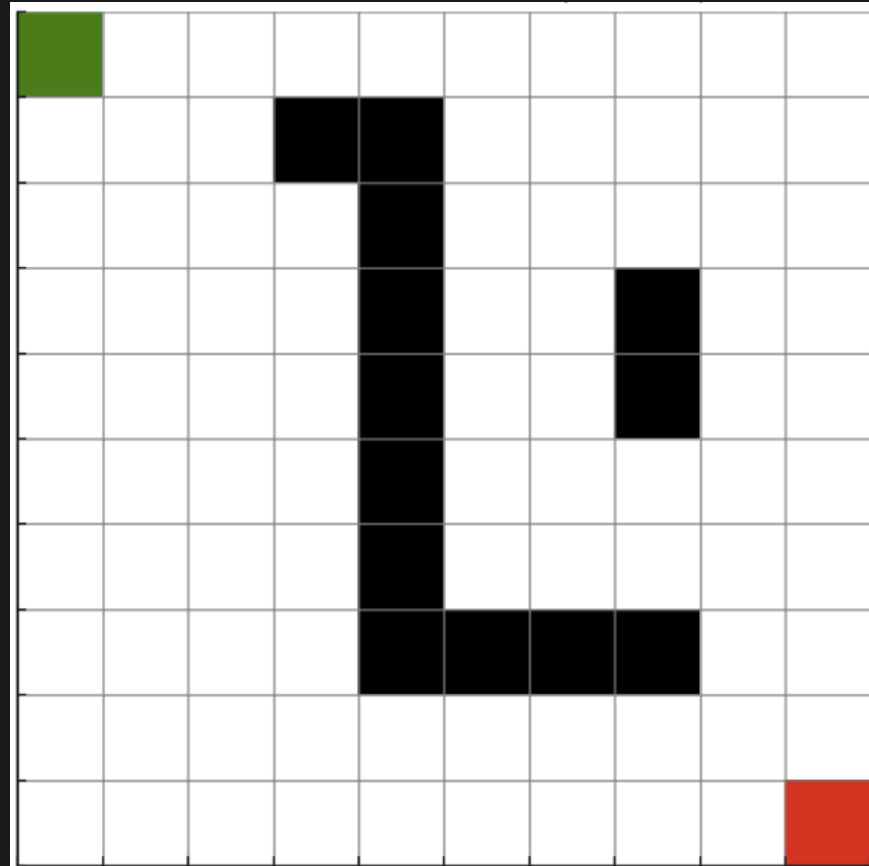
- 로봇이 시작 위치에서 목표 위치까지 최단 경로를 찾는 것
- 보통 단일 점(point)처럼 간주하거나, 로봇 크기/형상을 미리 축소하여 무시
- 속도나 가속도 같은 동역학(dynamics) 조건은 대부분 고려하지 않음
- 예시: 게임 캐릭터의 이동, 물류로봇 자동화

최단 경로 알고리즘

모션 플래닝

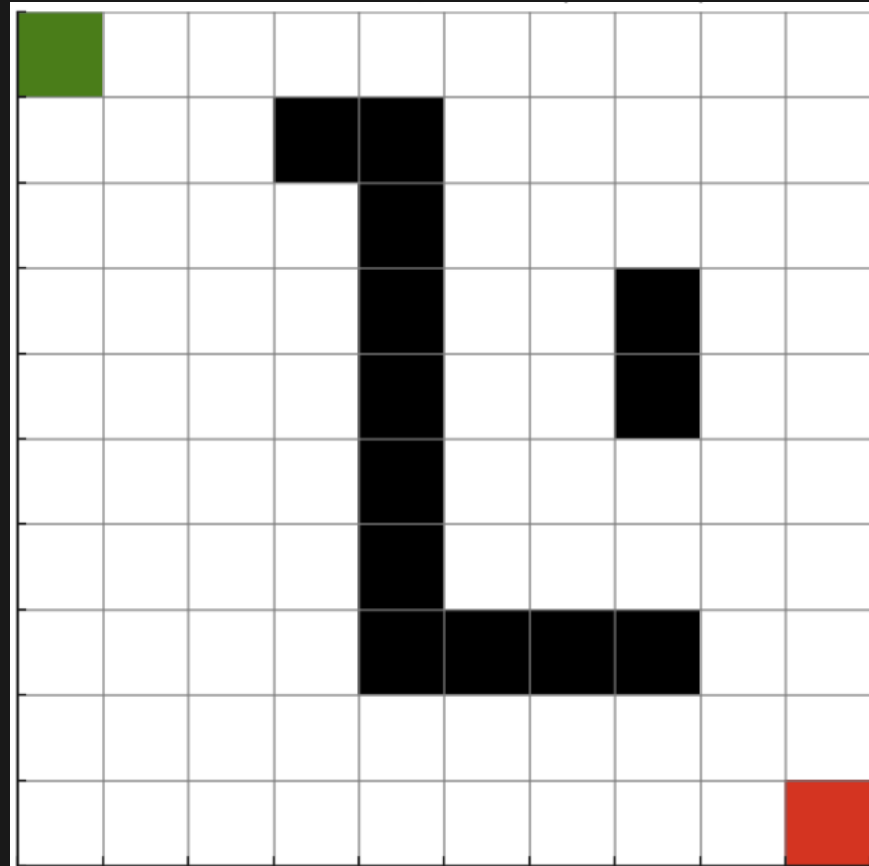
- 로봇이 실제 물리적·기하학적 제약 아래에서 장애물을 피하며 움직일 수 있는 연속적인 경로를 생성
- 로봇의 형상(shape), 크기(size), 관절(joint) 제약, 속도·가속도 같은 동역학(dynamics)까지 반영
- 예시: 로봇팔 플래닝

패스 파인딩 (path finding)



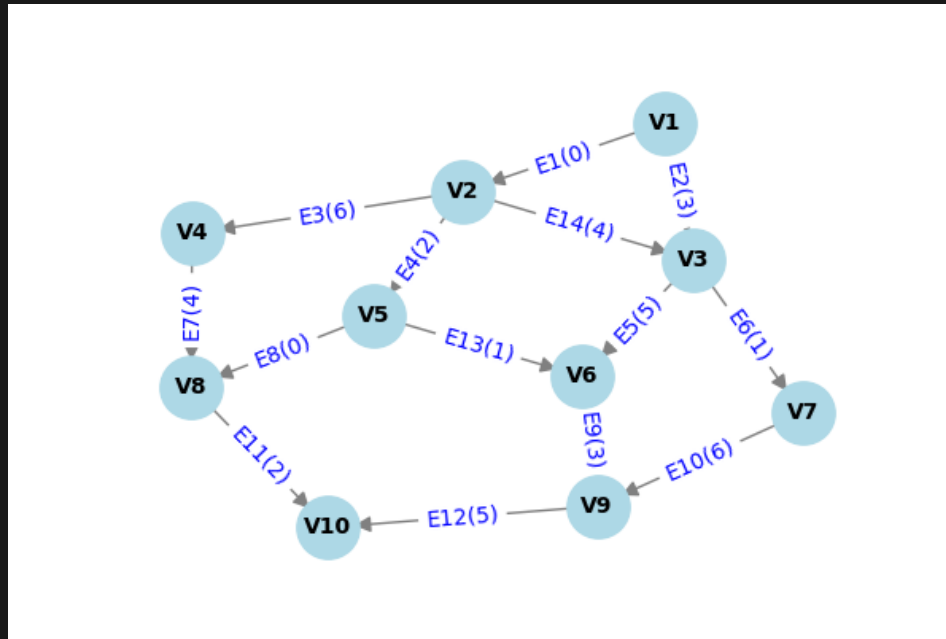
패스 파인딩 (path finding)

- 어떻게 솔루션을 찾을까?
→ 서치 알고리즘

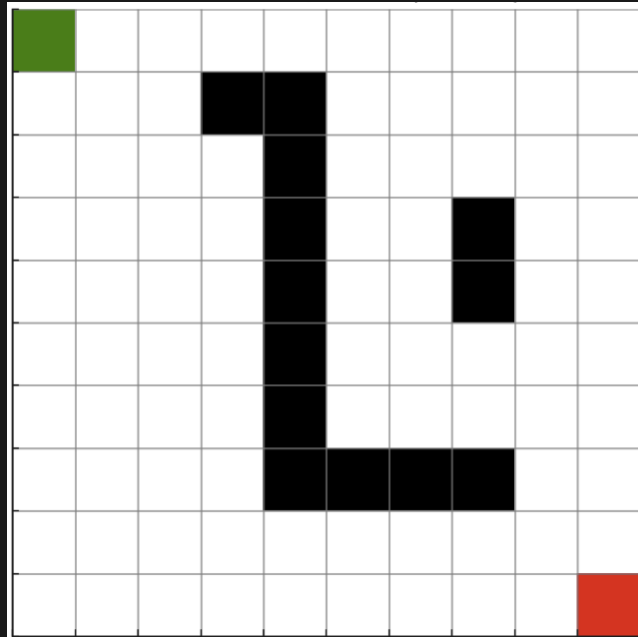
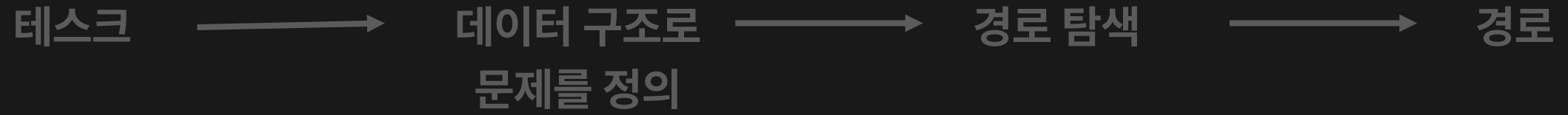


패스 파인딩 (path finding)

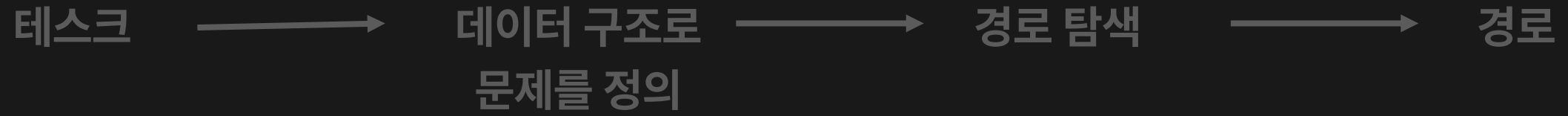
테스크 → 데이터 구조로 문제를 정의 → 경로 탐색 → 경로



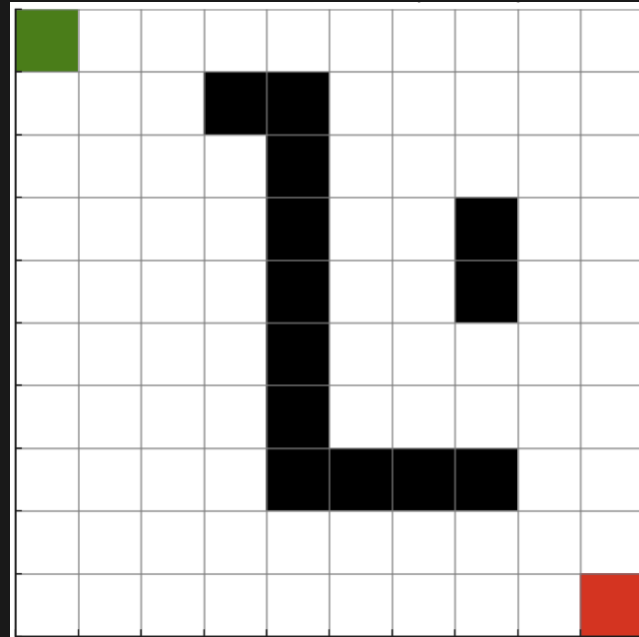
패스 파인딩 (path finding)



패스 파인딩 (path finding)



- V: 로봇의 위치
- E: 이동가능한 스텝 (가중치=1)



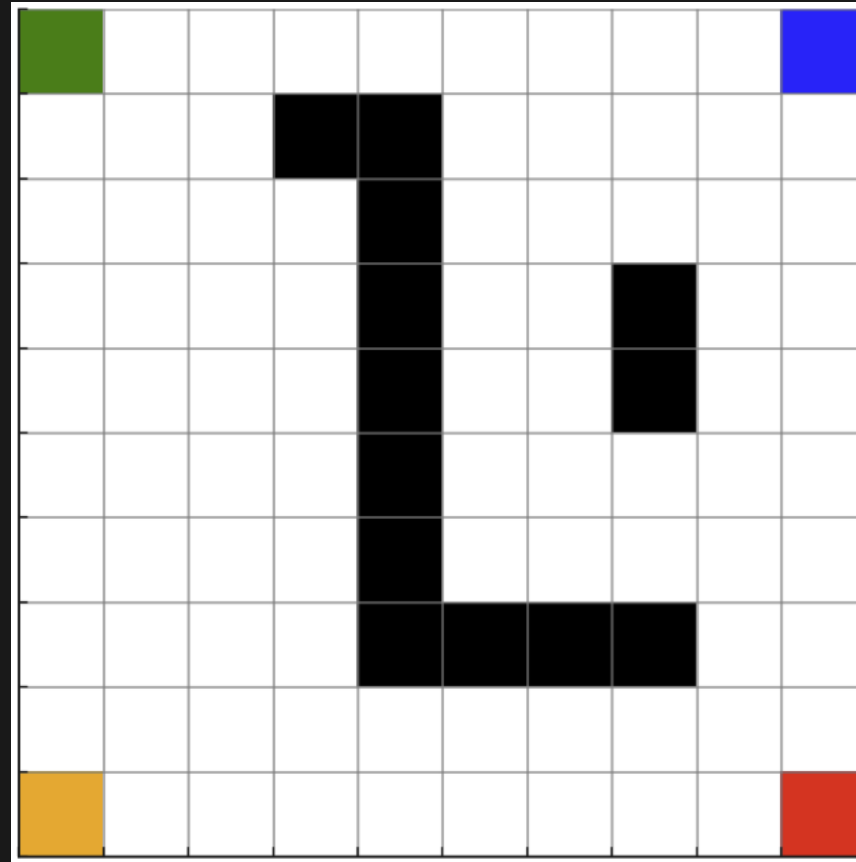
패스 파인딩 (path finding)

- 최단 경로 알고리즘
 - Dijkstra
 - A*
 - BFS
- 최단 경로 알고리즘이 아니라면?
 - DFS

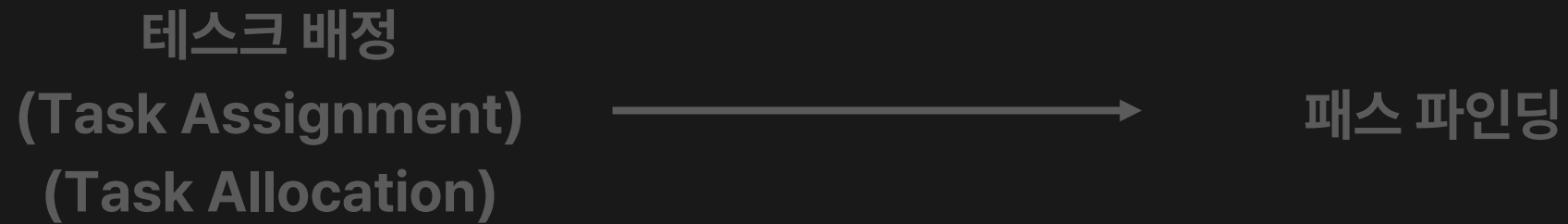
패스 파인딩 (path finding)



멀티 로봇 패스 파인딩 (multi-robot path finding)

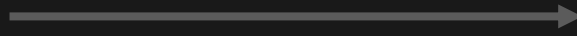


멀티 로봇 패스 파인딩 (multi-robot path finding)



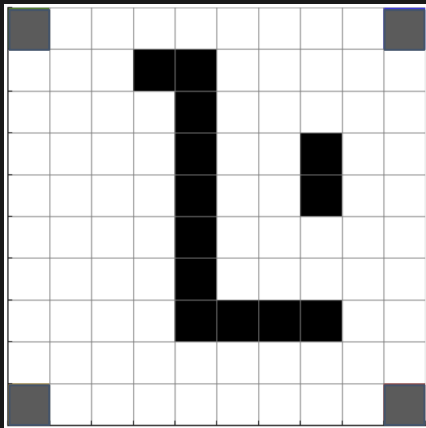
멀티 로봇 패스 파인딩 (multi-robot path finding)

테스크 배정
(Task Assignment)
(Task Allocation)

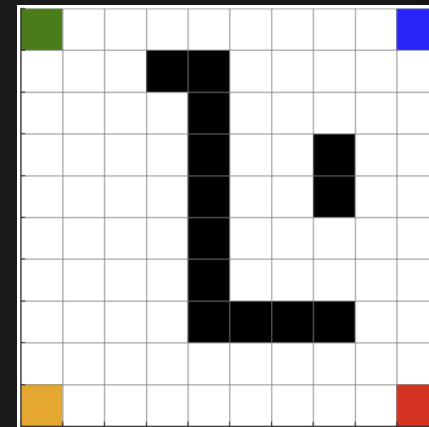


패스 파인딩

최적화 문제
(Linear Program)

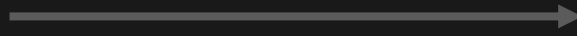


멀티로봇 경로탐색
알고리즘



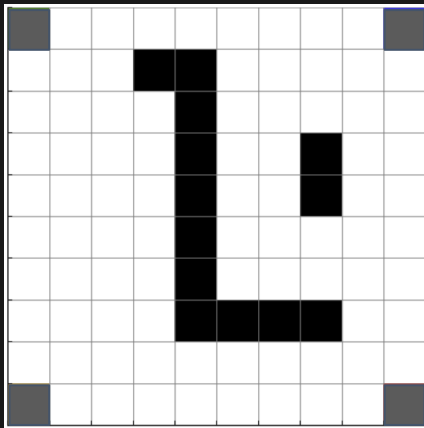
멀티 로봇 패스 파인딩 (multi-robot path finding)

테스크 배정
(Task Assignment)
(Task Allocation)

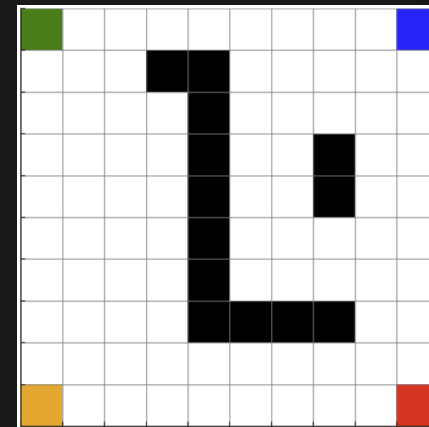


패스 파인딩

최적화 문제
(Linear Program)



멀티로봇 경로탐색
알고리즘



멀티 로봇 패스 파인딩 (multi-robot path finding)

$$\min_{x \in \mathcal{X}} f(x)$$

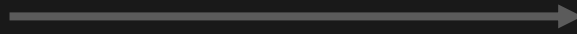
$$g_i(x) \leq 0, \quad h_j(x) = 0$$

2대의 모바일 로봇 (A, B, C, D) 이 4개의 픽업 (Pick-up) 작업(Task 1, 2, 3, 4)을 수행하는데 가장 짧은 방법은?

- 목적 함수 (objective function) $\min \sum_{i \in \{A, B, C, D\}} \sum_{j=1}^4 d_{i,j} x_{i,j}$
- 설계 변수 (design variables) $x_{i,j} = \begin{cases} 1, & \text{로봇 } i \text{가 Task } j \text{를 수행한다,} \\ 0, & \text{그렇지 않다,} \end{cases} \quad i \in \{A, B, C, D\}, j \in \{1, 2, 3, 4\}$
- 제약 조건 (constraints) $x_{i,j} \in \{0, 1\}, \quad \forall i \in \{A, B, C, D\}, j \in \{1, 2, 3, 4\}$

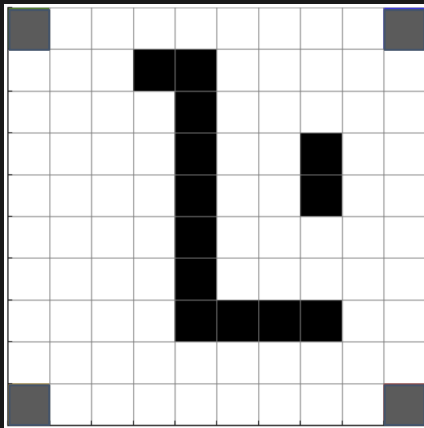
멀티 로봇 패스 파인딩 (multi-robot path finding)

테스크 배정
(Task Assignment)
(Task Allocation)

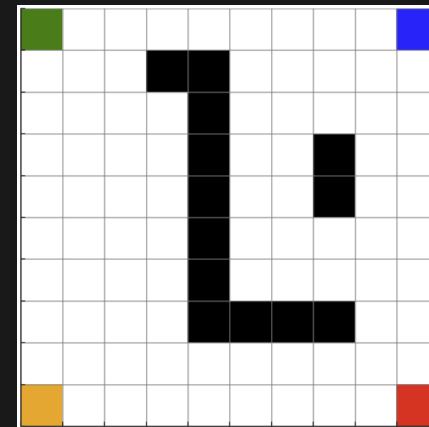


패스 파인딩

최적화 문제
(Linear Program)



멀티로봇 경로탐색
알고리즘



멀티 로봇 패스 파인딩 (multi-robot path finding)

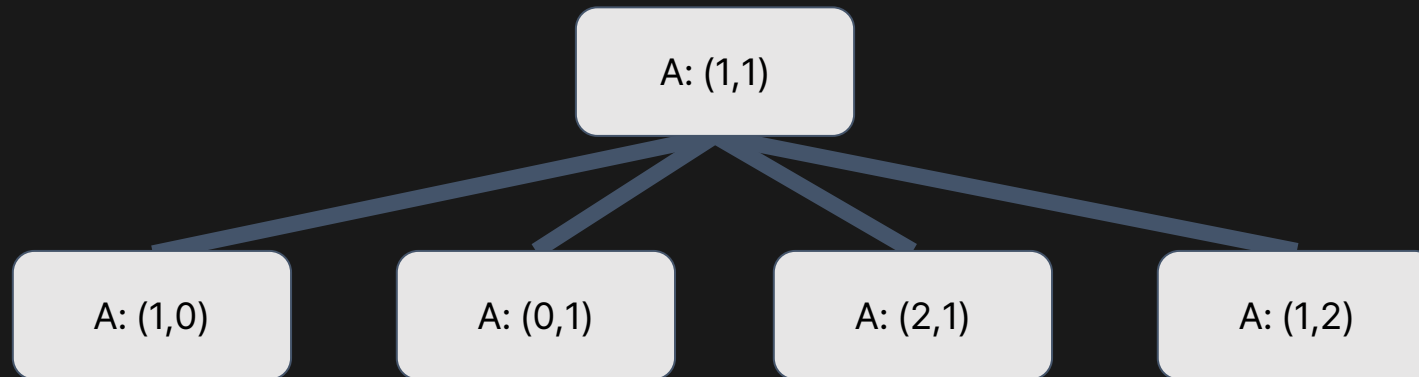
왜 멀티 로봇 상황에서 경로를 찾는 것이 어려운가?

→ 탐색의 경우의 수가 기하급수적으로 증가하기 때문

멀티 로봇 패스 파인딩 (multi-robot path finding)

왜 멀티 로봇 상황에서 경로를 찾는 것이 어려운가?

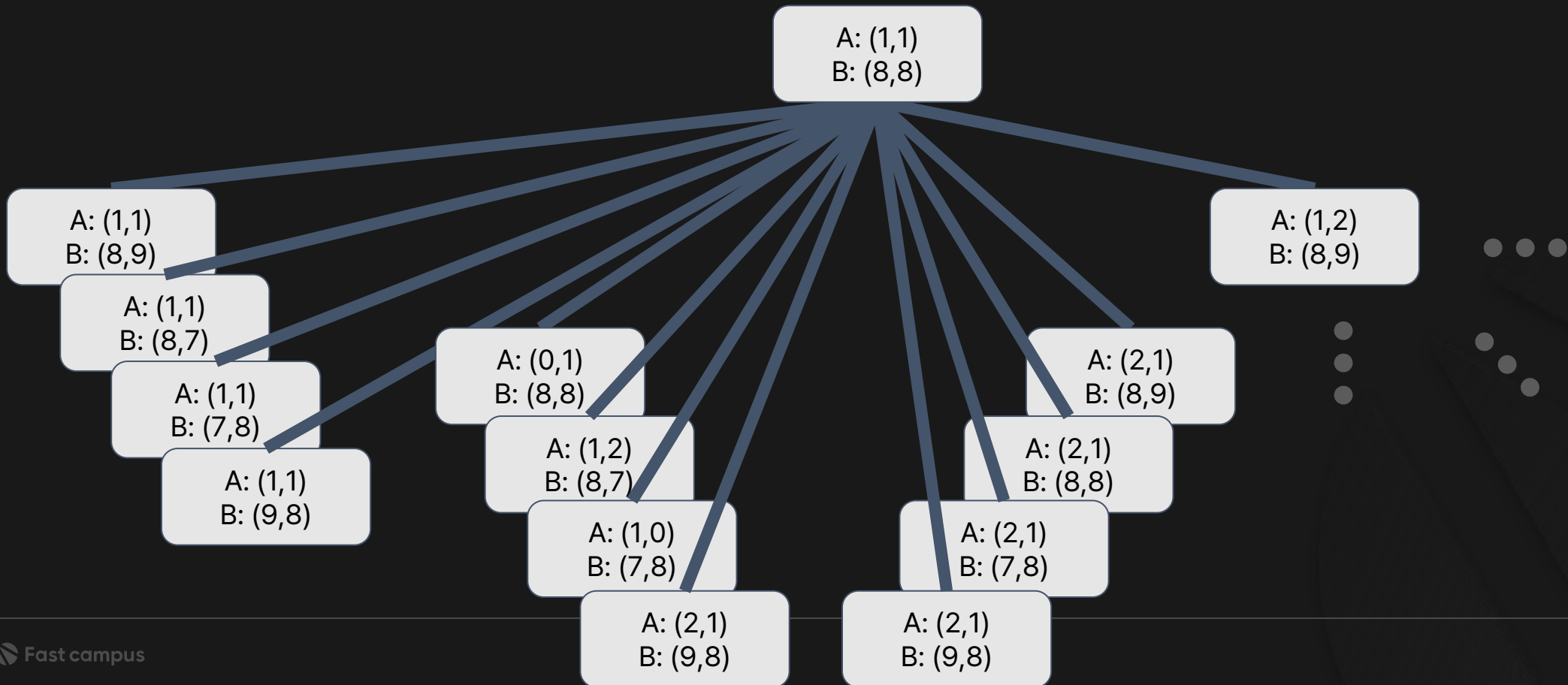
→ 탐색의 경우의 수가 기하급수적으로 증가하기 때문



멀티 로봇 패스 파인딩 (multi-robot path finding)

왜 멀티 로봇 상황에서 경로를 찾는 것이 어려운가?

→ 탐색의 경우의 수가 기하급수적으로 증가하기 때문

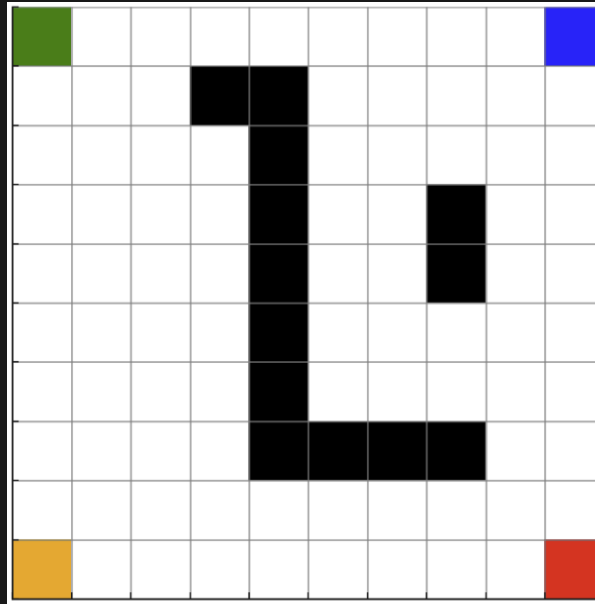


멀티 로봇 패스 파인딩 (multi-robot path finding)

왜 멀티 로봇 상황에서 경로를 찾는 것이 어려운가?

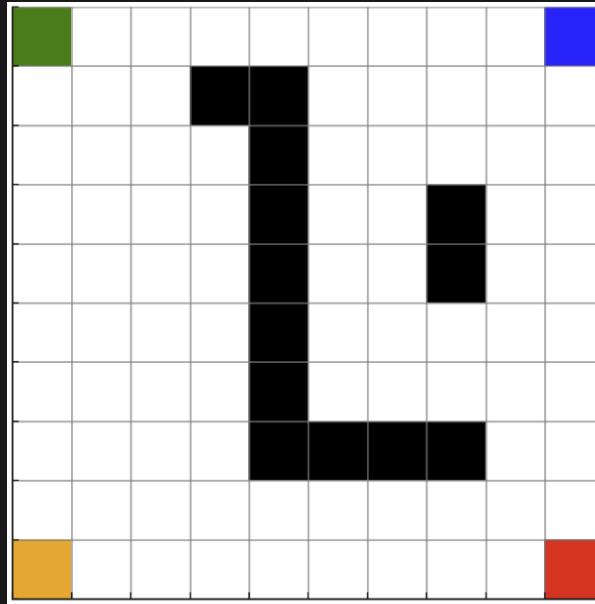
→ 탐색의 경우의 수가 기하급수적으로 증가하기 때문

→ 로봇 간 충돌을 고려해야 하기 때문



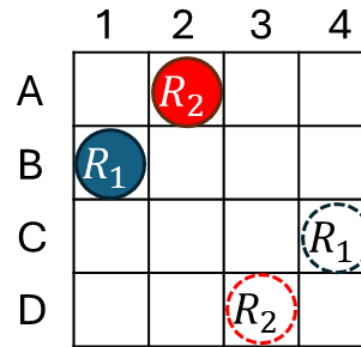
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



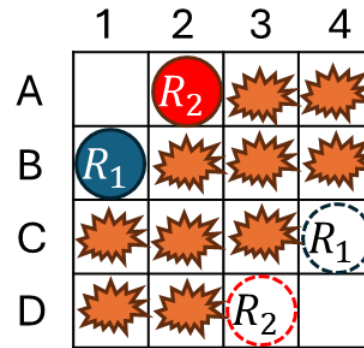
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



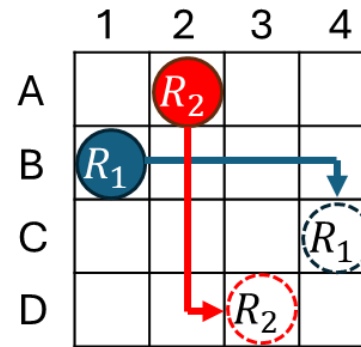
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)

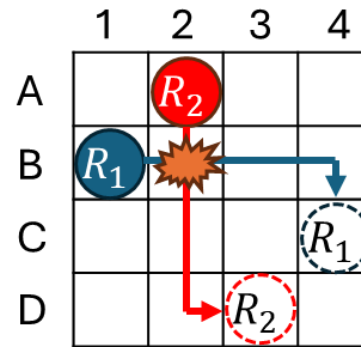


$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

cost: 4

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)

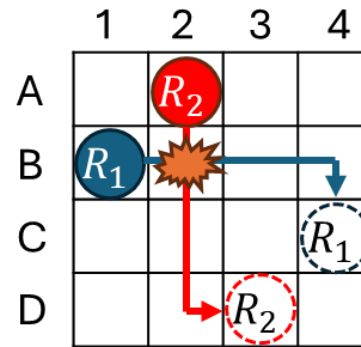


$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

cost: 4

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



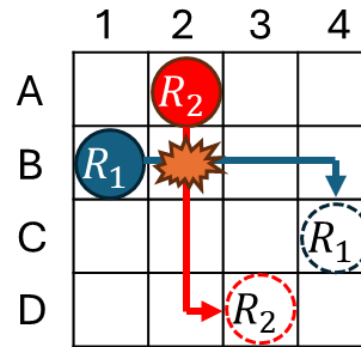
$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 4

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 4

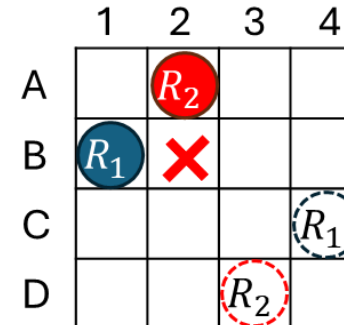
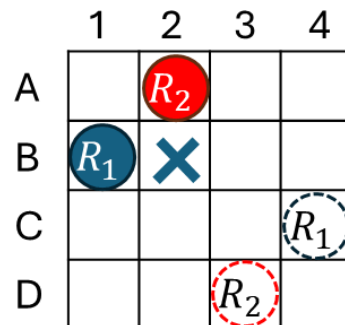
R_1 cannot stay at B2
at timestep 1

R_2 cannot stay at B2
at timestep 1

$R_1 - B1, B1, B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6



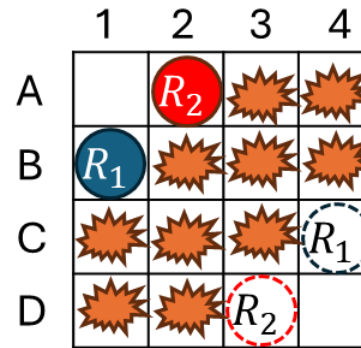
$R_1 - B1, B2, B3, B4, C4, C4, C4$
 $R_2 - A2, A2, A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6

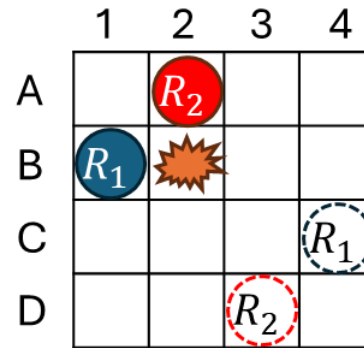
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



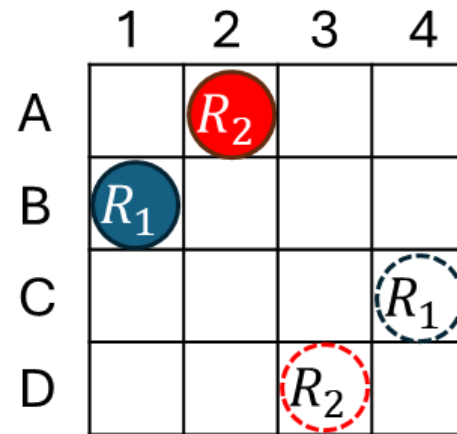
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



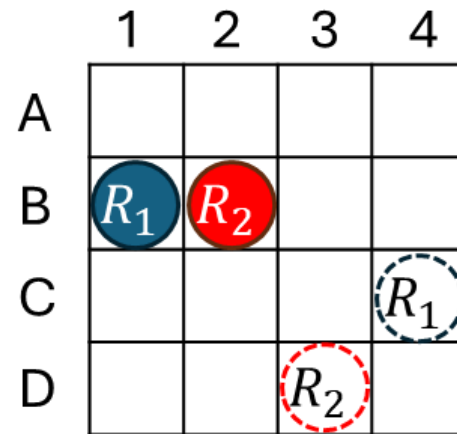
$R_1 - \mathbf{B1}, B1, B1, B2, B3, B4, C4$

$R_2 - \mathbf{A2}, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



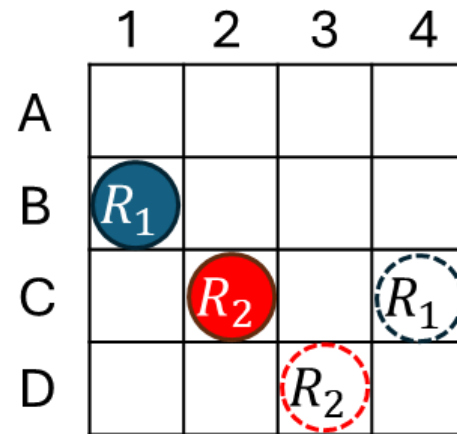
$R_1 - \mathbf{B1}, \mathbf{B1}, B1, B2, B3, B4, C4$

$R_2 - \mathbf{A2}, \mathbf{B2}, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



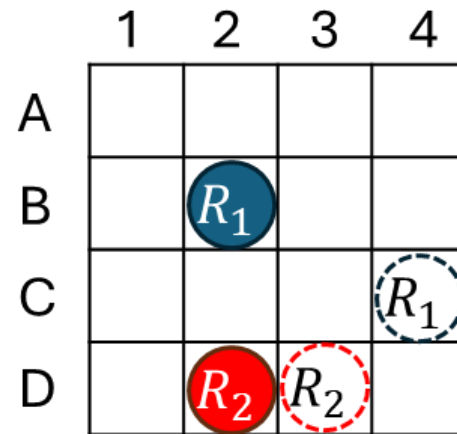
$R_1 - \mathbf{B1}, \mathbf{B1}, \mathbf{B1}, B2, B3, B4, C4$

$R_2 - \mathbf{A2}, \mathbf{B2}, \mathbf{C2}, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



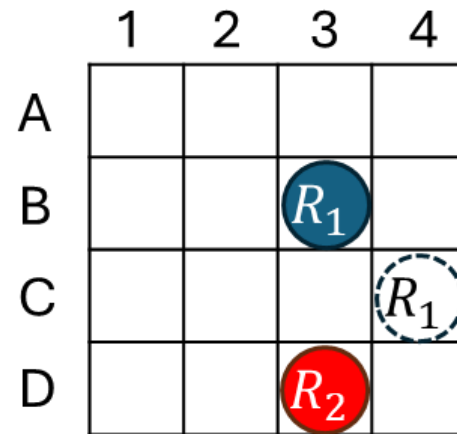
$R_1 - \mathbf{B1}, \mathbf{B1}, \mathbf{B1}, \mathbf{B2}, \mathbf{B3}, \mathbf{B4}, \mathbf{C4}$

$R_2 - \mathbf{A2}, \mathbf{B2}, \mathbf{C2}, \mathbf{D2}, \mathbf{D3}, \mathbf{D3}, \mathbf{D3}$

$\langle R_1, R_2, \mathbf{B2}, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



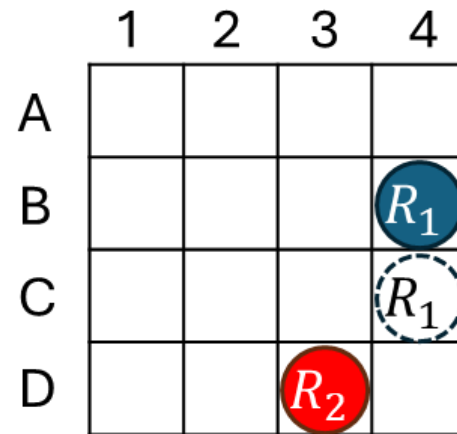
$R_1 - B1, B1, B1, B2, B3, B4, C4$

$R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



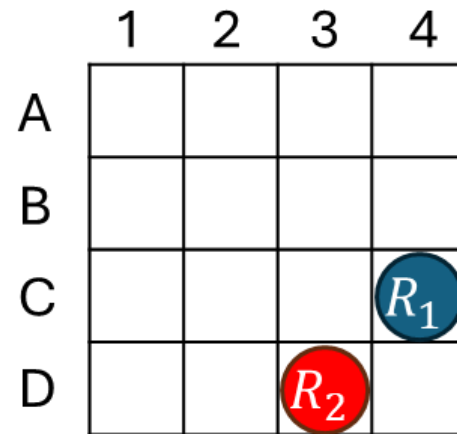
$R_1 - B1, B1, B1, B2, B3, B4, C4$

$R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



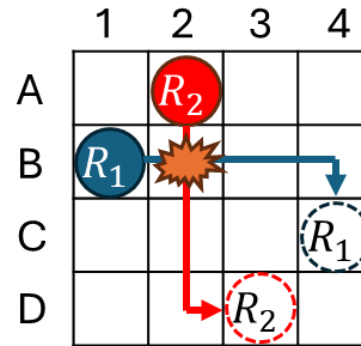
$R_1 - B1, B1, B1, B2, B3, B4, C4$

$R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 4

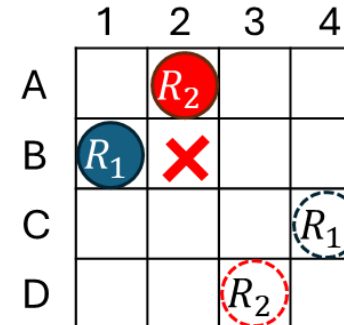
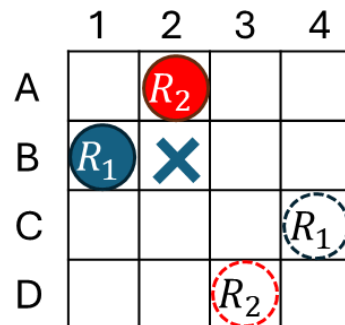
R_1 cannot stay at B2
at timestep 1

R_2 cannot stay at B2
at timestep 1

$R_1 - B1, B1, B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6



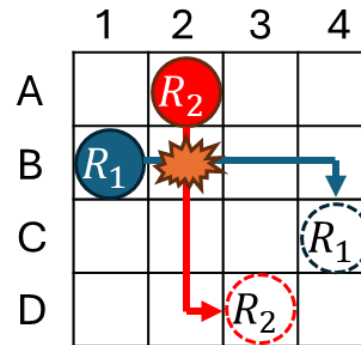
$R_1 - B1, B2, B3, B4, C4, C4, C4$
 $R_2 - A2, A2, A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

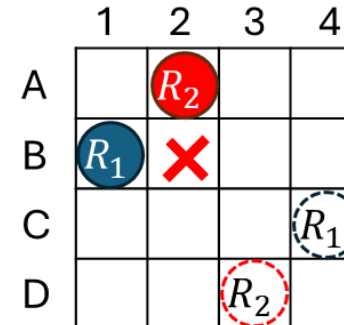
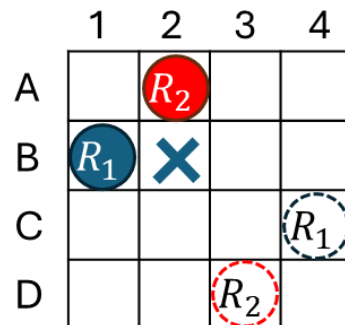
cost: 4

R_1 cannot stay at B2 at timestep 1
Conflict!
 R_2 cannot stay at B2 at timestep 1

$R_1 - B1, B1, B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6



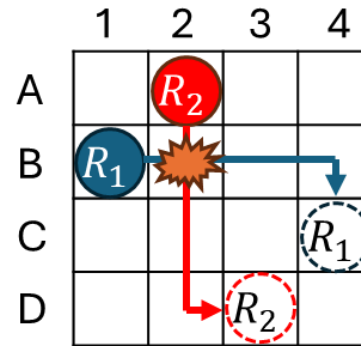
$R_1 - B1, B2, B3, B4, C4, C4, C4$
 $R_2 - A2, A2, A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



$R_1 - B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

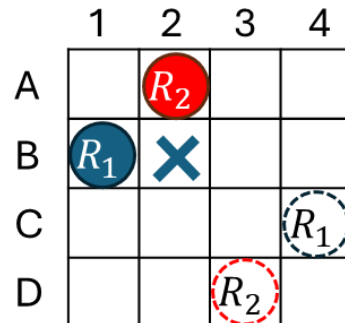
cost: 4

R_1 cannot stay at B2 at timestep 1
Conflict!
 R_2 cannot stay at B2 at timestep 1

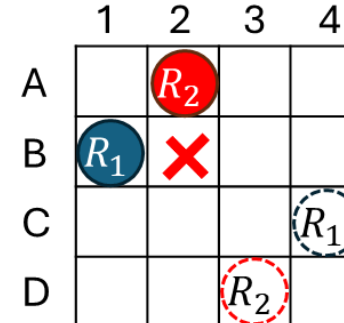
$R_1 - B1, B1, B1, B2, B3, B4, C4$
 $R_2 - A2, B2, C2, D2, D3, D3, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6



Conflict!



Conflict!

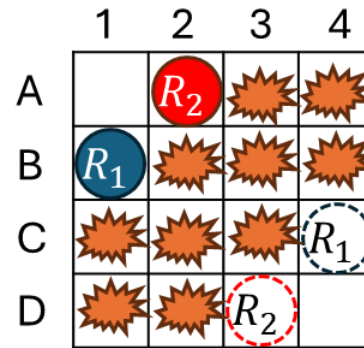
$R_1 - B1, B2, B3, B4, C4, C4, C4$
 $R_2 - A2, A2, A2, B2, C2, D2, D3$

$\langle R_1, R_2, B2, 1 \rangle$

cost: 6

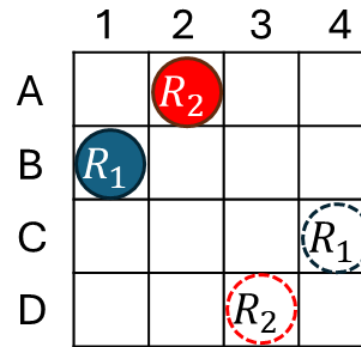
멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)



멀티 로봇 패스 파인딩 (multi-robot path finding)

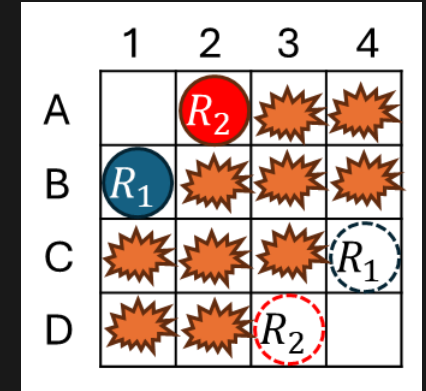
Conflict-Based Search (CBS)

- 이중 레벨 구조 (Two-Level Search)
 - High-level: Conflict Tree (CT) 탐색
 - Low-level: 각 로봇 별로 제약조건을 만족하는 독립적인 최단 경로 탐색

멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)

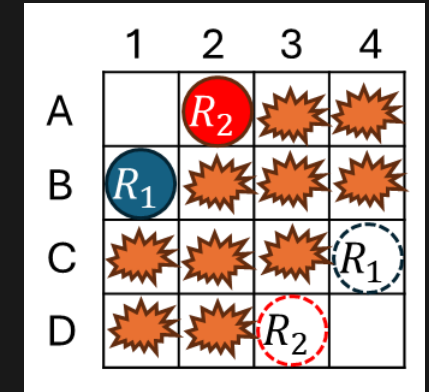
- 이중 레벨 구조 (Two-Level Search)
 - High-level: Conflict Tree (CT) 탐색
 - Low-level: 각 로봇 별로 제약조건을 만족하는 독립적인 최단 경로 탐색
- 완전성 (Completeness)
 - 해 (solution)가 존재한다면 유한 시간 내에 반드시 해를 찾을 수 있음



멀티 로봇 패스 파인딩 (multi-robot path finding)

Conflict-Based Search (CBS)

- 이중 레벨 구조 (Two-Level Search)
 - High-level: Conflict Tree (CT) 탐색
 - Low-level: 각 로봇 별로 제약조건을 만족하는 독립적인 최단 경로 탐색
- 완전성 (Completeness)
 - 해 (solution)가 존재한다면 유한 시간 내에 반드시 해를 찾을 수 있음
- 최적성 (Optimality)
 - 전체 경로 길이 (makespan)을 최소화하는 최적해를 보장
 - CT 에서 cost가 낮은 노드부터 탐색하도록 하여 최적 해를 보장
- 확장성: Meta-CBS, Windowed-CBS, Suboptimal-CBS, ...



강의 요약

01

패스 파인딩 vs. 모션 플래닝

02

멀티 로봇 패스 파인딩

- Task Allocation
- Multi-robot Path Finding

03

Conflict- Based Search (CBS)

- Two-level Search
- 완전성
- 최적성