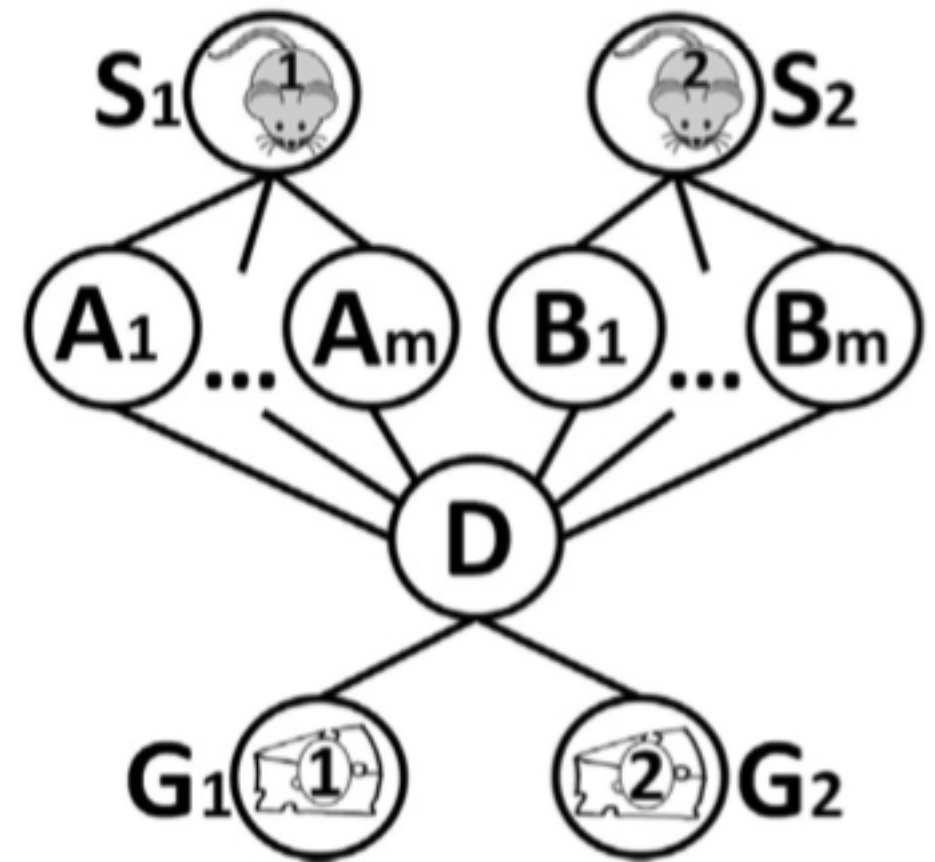


Алгоритмы конфликтно-ориентированного поиска для задачи многоагентного планирования (CBS, CBS+PC)

Макаренко Степан, M05-9056
Ушакова Анастасия, M05-9056

Задача многоагентного планирования

- Задача многоагентного планирования (Multi-agent pathfinding, MAPF) - обобщение задачи поиска пути в графе для $k > 1$ агентов.
- Имеется:
 - ориентированный граф $G(V, E)$
 - множество агентов (у каждого - стартовая и целевая позиции)
- Необходимо:
 - найти пути без столкновений
 - найти оптимальное решение



Ограничения

- В каждой вершине в текущий момент времени может находиться не более одного агента
- Агенты не могут за один и тот же промежуток времени проходить по одному и тому же ребру
- Конфликт - ситуация, когда ограничения нарушаются

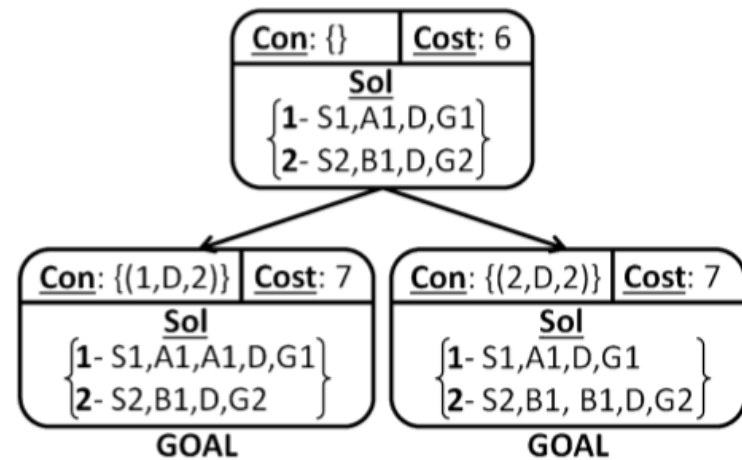
Вход и выход

- Вход:
 - ориентированный граф $G(V, E)$
 - k агентов a_1, a_2, \dots, a_k
 - стартовые и целевые точки $(start_1, goal_1) \dots (start_k, goal_k)$
 - время дискретно: t_0, t_1 и т.д
- Выход:
 - множество неконфликтующих путей для каждого агента

Конфликтно-ориентированный поиск (CBS)

- Путь - подразумевается путь для одного агента
- Решение - множество k путей для множества k агентов
- Ограничение - набор (a_i, v, t) - агенту a_i запрещено занимать вершину v в момент времени t
- Состоятельный путь для агента a_i - путь, который удовлетворяет всем ограничениям для этого агента
- Состоятельное решение - решение, которое состоит из состоятельных путей
- Конфликт - набор (a_i, a_j, v, t) , где агенты a_i и a_j занимают вершину v в момент времени t
- Решение допустимо, если все входящие в него пути не содержат конфликтов
- Состоятельное решение может быть недопустимо, если пути в нем все равно содержат конфликты
- 2 уровня:
 - Высокий (high level policy)
 - Низкий (low level policy)
 - A^*

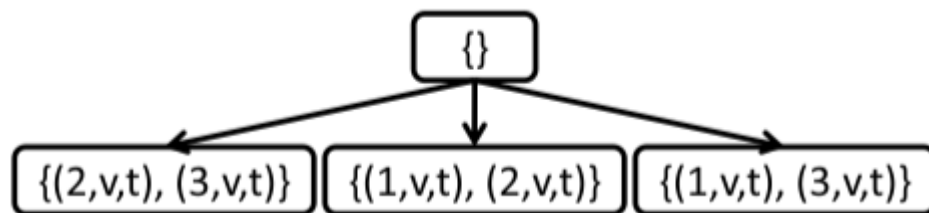
CBS: high level policy



Constraint tree (CT)

Вершина N:

- N.constraints
- N.solution
- N.cost



к потомков, если
конфликтуют к агентов

Input: MAPF instance

```

1 Root.constraints = ∅
2 Root.solution = find individual paths by the low level()
3 Root.cost = SIC(Root.solution)
4 insert Root to OPEN
5 while OPEN not empty do
6   P ← best node from OPEN // lowest solution cost
7   Validate the paths in P until a conflict occurs.
8   if P has no conflict then
9     return P.solution // P is goal
10  C ← first conflict (ai, aj, v, t) in P
11  if shouldMerge(ai, aj) // Optional, MA-CBS only then
12    a{i,j} = merge(ai, aj, v, t)
13    Update P.constraints(external constraints).
14    Update P.solution by invoking low level(a{i,j})
15    Update P.cost
16    if P.cost < ∞ // A solution was found then
17      Insert P to OPEN
18    continue // go back to the while statement
19  foreach agent ai in C do
20    A ← new node
21    A.constraints ← P.constraints + (ai, v, t)
22    A.solution ← P.solution
23    Update A.solution by invoking low level(ai)
24    A.cost = SIC(A.solution)
25    if A.cost < ∞ // A solution was found then
26      Insert A to OPEN
    
```

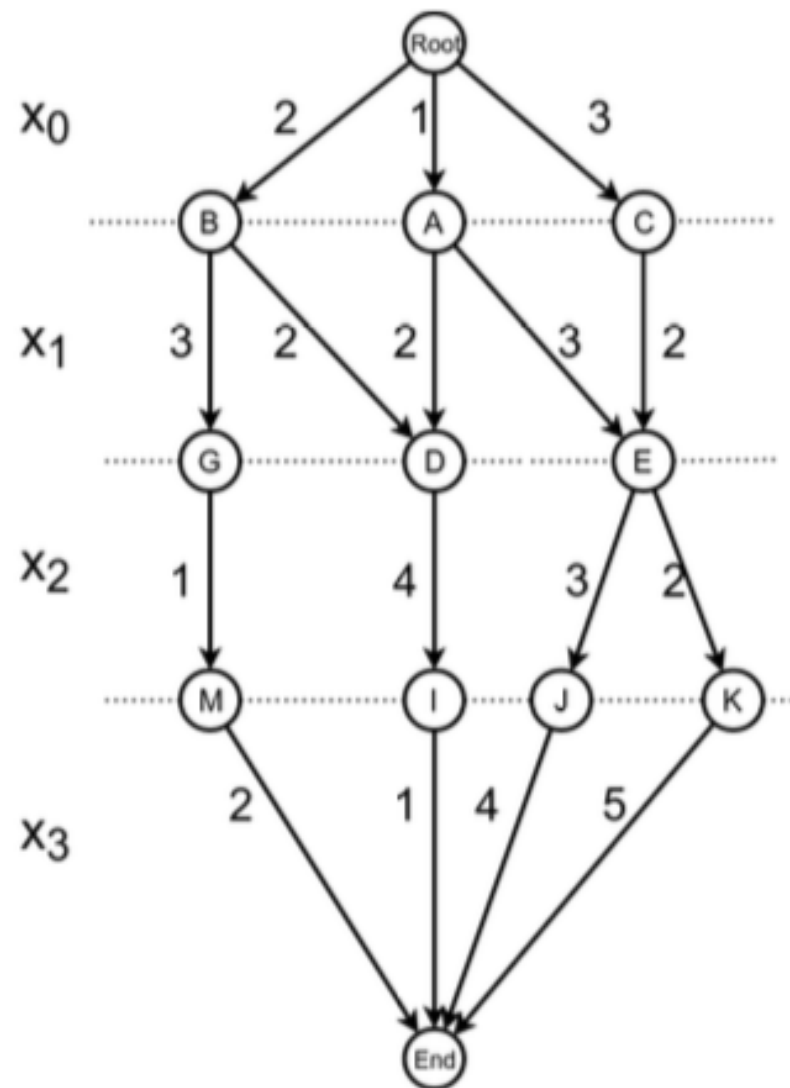
CBS с приоритизацией конфликтов (CBS+PC)

Проблема:

- CBS чувствителен к тому, на каком конфликте останавливать валидацию конфликтов
- Базовый CBS может «застрять» в области, где $N.cost$ схожи и порождать большое количество лишних вершин СТ
- 3 типа конфликтов:
 - Кардинальный
 - Полукардинальный
 - Некардинальный

Решение: **приоритизировать конфликты**

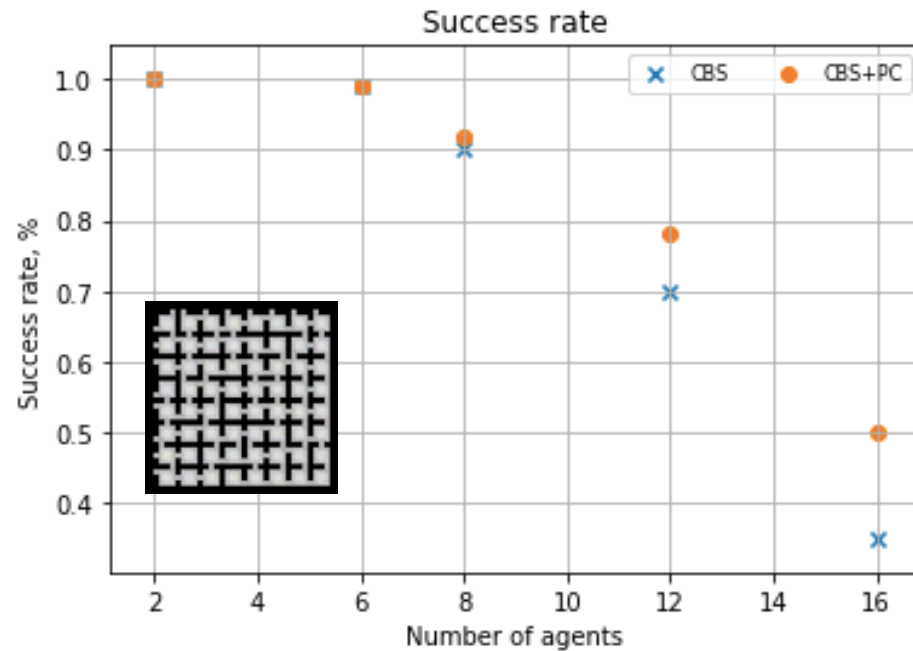
CBS с приоритизацией конфликтов (CBS+PC)



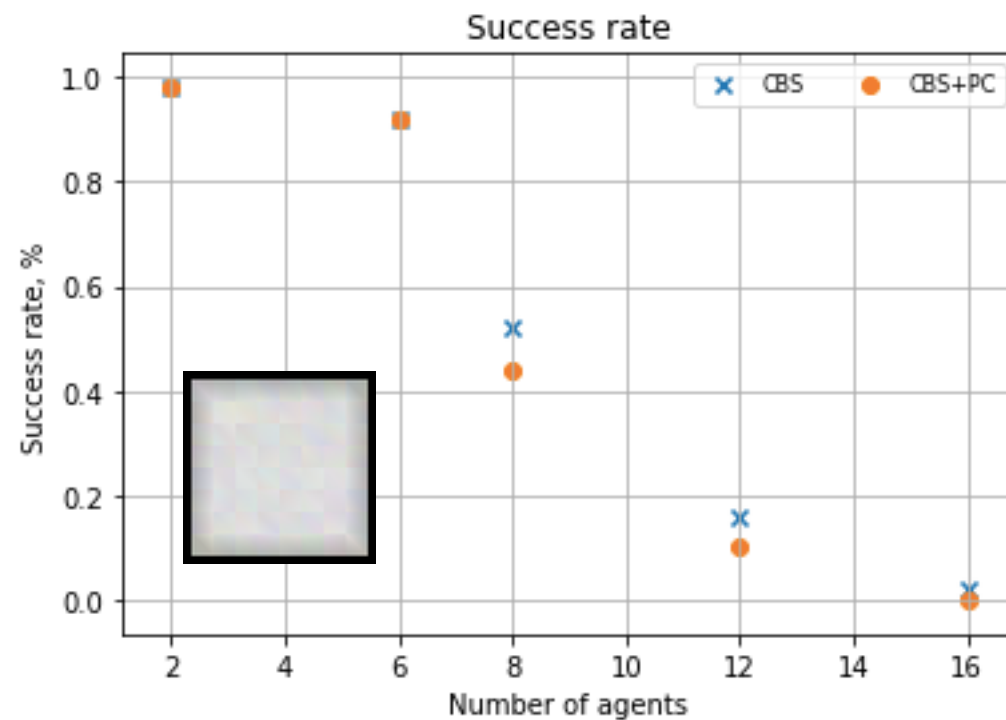
MDD (multi-value decision diagram) - хранит все возможные пути стоимости c для данного агента

- Если существует кардинальный конфликт (a_i, a_j, v, t) , то ширины диаграмм для каждого агента и их путей оптимальной длины в момент времени t равны 1
- Аналог MDD:
 - Ищутся все решения, стоимость которых равна оптимальному
 - Для каждого из таких решений получается его путь из узлов $(v_1, v_2 \dots)$.
 - Для каждого момента времени t создается множество из узлов
 - Для каждой найденной цепочки узлов узел добавляется в множество, соответствующее моменту времени
 - По размеру множества можно определить ширину
- Конфликты проверяются на кардинальность (для пары агентов)
- Порядок рассмотрения: кардинальный \rightarrow полукардинальный \rightarrow некардинальный

Эксперименты

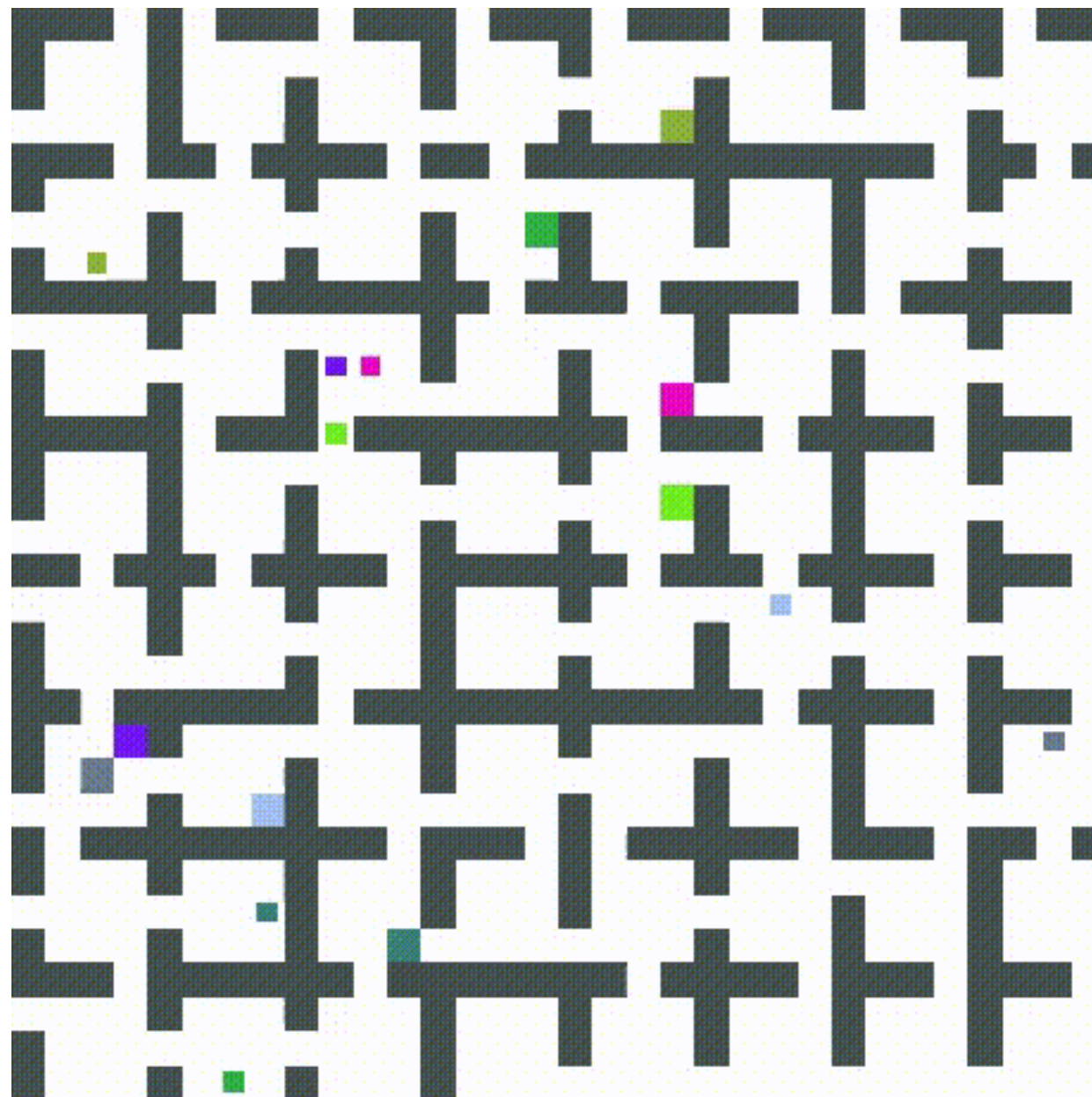


	Время, с (CBS)	Время, с (CBS+PC)	CTNodes (CBS)	CTNodes (CBS+PC)
2	0.038587	0.047440	1.440000	1.440000
4	1.289318	0.636122	6.414141	3.484848
8	2.362520	2.004964	53.888889	18.088889
12	19.550481	15.655129	263.782609	103.188406
16	50.341116	30.231400	1233.125000	251.093750



	Время, с (CBS)	Время, с (CBS+PC)	CTNodes (CBS)	CTNodes (CBS+PC)
2	0.000108	0.000141	1.081633	1.081633
4	0.005890	0.010502	8.000000	8.000000
8	0.005761	0.010414	9.454545	9.636364
12	0.012853	0.023692	19.000000	19.000000
16	NaN	NaN	NaN	NaN

Демонстрация работы



Спасибо за внимание!