

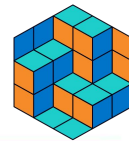


Алгоритмы конфликтно-ориентированного поиска для задачи многоагентного планирования: CBS, CBS+PC, CBS+H, CBS+DS

Им Евгений
Парамонов Антон
Эмдин Григорий



Постановка задачи



Вход:

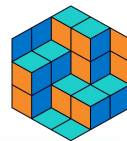
граф $G(V, E)$

агенты $a_i: s_i, f_i$

Выход:

$s_i \rightsquigarrow f_i \quad \forall i$

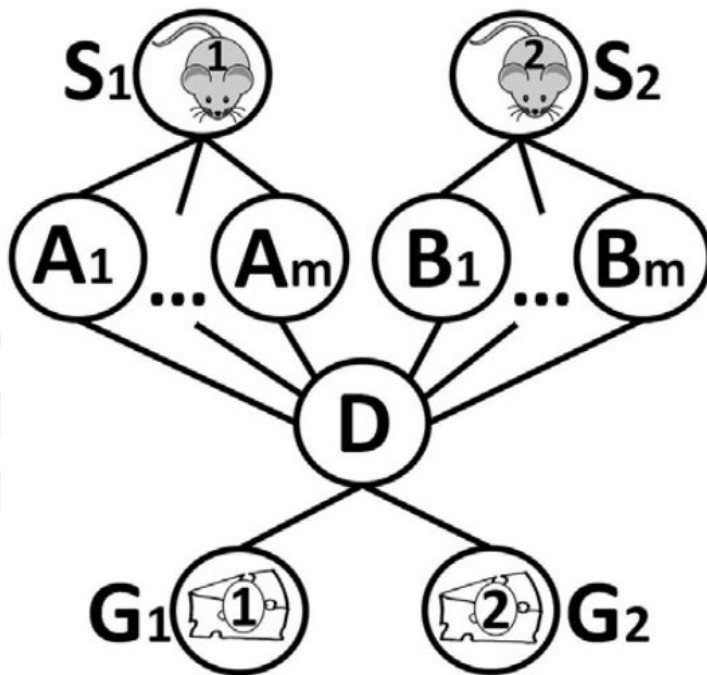




Конфликт:

(a_i, a_j, v, t)

— агенты a_i и a_j находятся в вершине v
в момент времени t





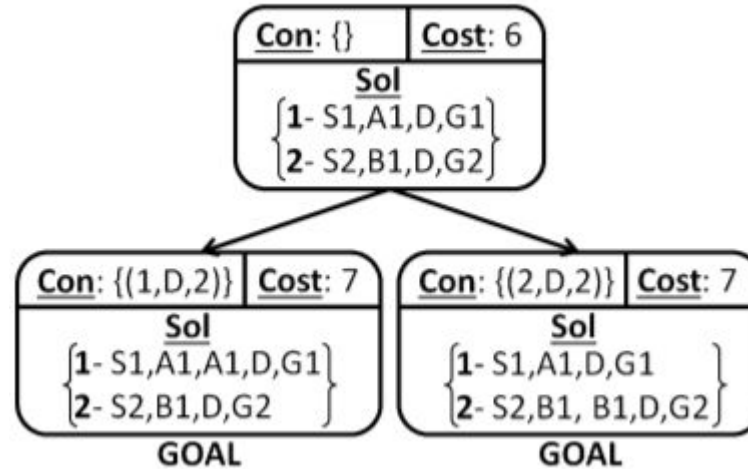
Алгоритм conflict based search (CBS)

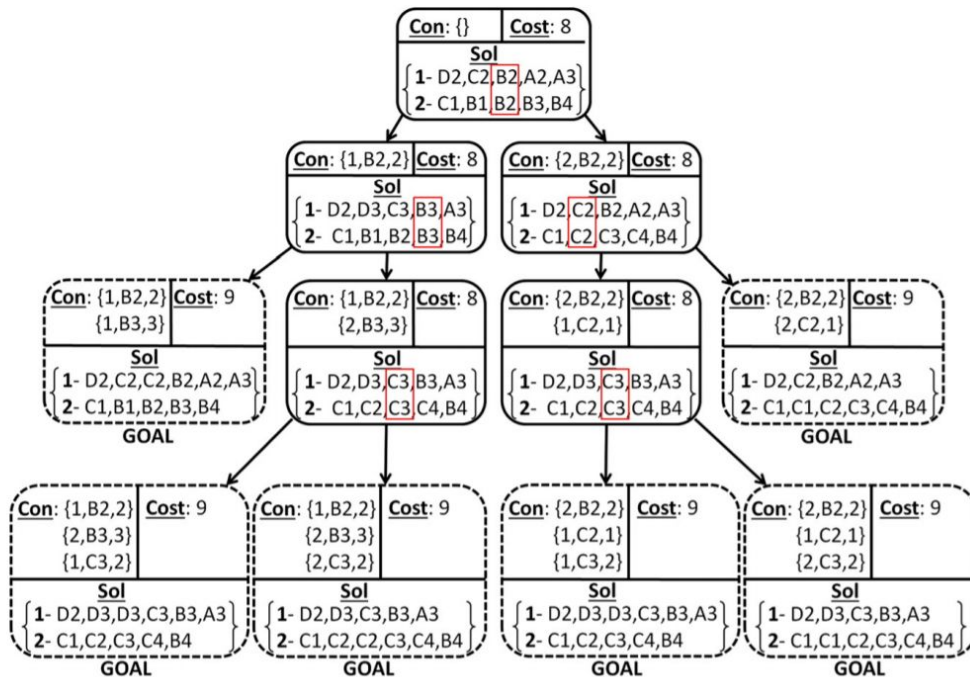
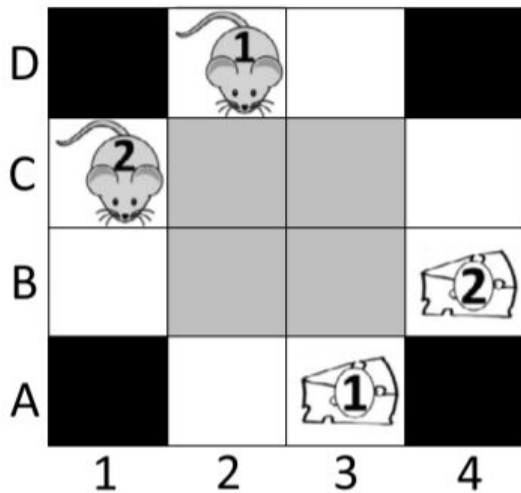
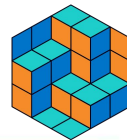


Ограничение:

(a, v, t)

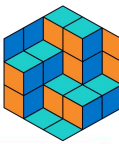
— запрет агенту a находится в вершине v
в момент времени t







```
Input: MAPF instance
Root.constraints =  $\emptyset$ 
Root.solution = find individual paths by the low level()
Root.cost = SIC(Root.solution)
insert Root to OPEN
while OPEN not empty do
     $P \leftarrow$  best node from OPEN // lowest solution cost
    Validate the paths in  $P$  until a conflict occurs.
    if  $P$  has no conflict then
         $\lfloor$  return  $P$ .solution //  $P$  is goal
     $C \leftarrow$  first conflict  $(a_i, a_j, v, t)$  in  $P$ 
    foreach agent  $a_i$  in  $C$  do
         $A \leftarrow$  new node
         $A$ .constraints  $\leftarrow P$ .constraints +  $(a_i, v, t)$ 
         $A$ .solution  $\leftarrow P$ .solution
        Update  $A$ .solution by invoking low level( $a_i$ )
         $A$ .cost = SIC( $A$ .solution)
        if  $A$ .cost  $< \infty$  //  $A$  solution was found then
             $\lfloor$  Insert  $A$  to OPEN
```

- CBS + Prioritization of conflicts (PC)

Input: MAPF instance

$Root.constraints = \emptyset$

$Root.solution = \text{find individual paths by the low level}()$

$Root.cost = SIC(Root.solution)$

insert $Root$ to OPEN

while OPEN *not empty* **do**

$P \leftarrow$ best node from OPEN // lowest solution cost

 Validate the paths in P until a conflict occurs.

if P has no conflict **then**

 | **return** $P.solution$ // P is goal

$C \leftarrow$ first conflict (a_i, a_j, v, t) in P $C \leftarrow \text{find-cardinal/semi-cardinal-conflict}(N)$ // (PC)

foreach agent a_i in C **do**

$A \leftarrow$ new node

$A.constraints \leftarrow P.constraints + (a_i, v, t)$

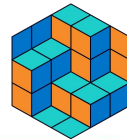
$A.solution \leftarrow P.solution$

 Update $A.solution$ by invoking $\text{low level}(a_i)$

$A.cost = SIC(A.solution)$

if $A.cost < \infty$ // A solution was found **then**

 | Insert A to OPEN

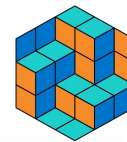


- CBS + H

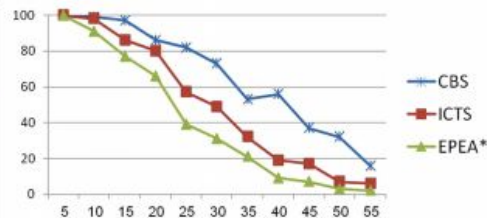
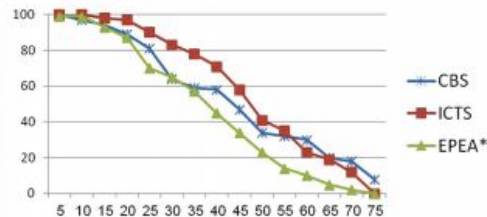
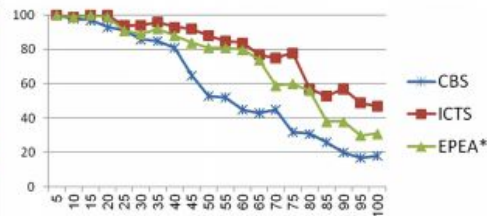
Эвристики:

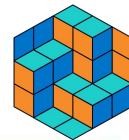
- h_1 : количество конфликтов
- h_2 : количество агентов с хотя бы одним конфликтом
- h_3 : количество пар конфликтующих агентов
- h_4 : размер вершинного покрытия для графа, вершинами в котором являются агенты, а ребрами соединены те из них, между которыми есть хотя бы один конфликт

- CBS + Disjoint Splitting (DS)



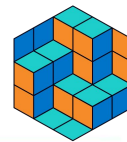
G. Sharon et al. / Artificial Intelligence 219 (2015) 40–66





1. Реализация классического CBS
2. Реализация системы тестирования алгоритма на бенчмарках
3. Тестирование CBS на корректность на маленьких собственных примерах
4. Тестирование CBS на корректность на больших бенчмарках
5. Сравнение нашей реализации CBS с реализацией в статье
6. Реализация CBS + H
7. Реализация CBS + PC
8. Реализация CBS + DS
9. Замеры времени работы, стоимостей решения и количества узлов дерева для расширений и самого CBS, отрисовка графиков, анализ результатов

Май 2021							^	v
Пн	Вт	Ср	Чт	Пт	Сб	Вс		
26	27	28	29	30	1	2		
3	4	5	6	7	8	9		
10	1 - 5	12	13	14	15	16		
17	6 - 8	19	20	21	22	23		
24	25	26	27	28	29	30		
31	9	2	3	4	5	6		



• Спасибо за внимание!

