

Планирование поведения

Киселёв Глеб Андреевич, ФИЦ ИУ РАН

About the Author

- Аспирант ФИЦ ИУ РАН
- Программист в Лаборатории 0-2 в Институте Системного Анализа
- Преподаватель ВШЭ (VBA, JS/HTML, Python)

Научные интересы:

- Мультиагентное планирование
- Робототехника
- Психологическиподобные системы хранения знаний



Git



Методы планирования поведения

Вступление

Виды
планирования

Планировщик
RuperPlan

Многоагентный
подход

Завершение

- В задаче планирования выделяют две основные составляющие:
 - Среда
 - Агент
- **Модель среды:** системы правил, логические методы, семантические сети, фреймовые структуры
- **Агент:** аппаратная или программная система, обладающая свойствами автономности, реактивности, активности, коммуникативности.

Методы планирования поведения

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- План - последовательность действий, формируемая агентом на основе общих целей, информации о текущем состоянии среды и динамике её изменения
- Сложность задачи планирования:
 - Причины изменения среды
 - Состояние среды
 - Достаточность источников данных
 - Характер изменения среды

Постановка задачи планирования

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Для процесса построения плана необходимы:
 - Описание задачи планирования
 - Алгоритм планирования
- Описание задачи планирования состоит из набора фактов, описывающих начальное и конечное состояние среды.
- Факт - замкнутая атомарная формула языка исчисления предикатов 1-го порядка.
- Состояние среды - множество фактов, истинных в текущий момент времени.

Планирование в пространстве состояний

Вступление

Виды планирования

Планировщик
RuperPlan

Многоагентный
подход

Завершение

Поиск плана в пространстве состояний может быть:

- Прямым
- Обратным

Первым планировщиком, осуществляющим планирование в пространстве состояний, является STRIPS (Stanford Research Institute Problem Solver)

Пример среды в терминах STRIPS:

$$S = \{ ATR(a), AT(B, b), AT(C, c), \forall u \forall x \forall y ((AT(u, x) \wedge (x \neq y)) \rightarrow \neg AT(u, y)) \}$$

Планирование в пространстве состояний

Вступление

Виды планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Действия агента описываются с помощью правил
- Пример правила:
 - Имя правила: Push (x, y, z)
 - Условие: $C(R) = \{ATR(y), AT(x, y)\}$
 - Список добавлений: $A(R) = \{ATR(y), AT(x, y)\}$
 - Список удалений: $D(R) = \{ATR(z), AT(x, z)\}$
- В приведённом примере правило Push(x, y, z) описывает действие робота по перемещению ящика 'x' из комнаты 'y' в комнату 'z'. Выполнение агентом действия сводится к применению правила. Применение правила модифицирует состояние 's'.

Планирование в пространстве СОСТОЯНИЙ

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

Алгоритм STRIPS:

STRIPS

вход: R, s, G

выход: plan

$s = s_0$

пока G не выполнимо в
 s

делать

- выбрать g из G
- выбрать r из R такое,
что g в $A(R)$
- **STRIPS**($R, s, C(R)$)
- применить R к s
- добавить R в plan

вернуть plan

Пример работы алгоритма
STRIPS:

- Дано:

Объекты:

кубики: A, B, C

Состояние описывается
предикатами:

- $ontable(x)$
- $clear(x)$
- $handempty$
- $holding(x)$
- $on(x, y)$

x, y - переменные.

Планирование в пространстве СОСТОЯНИЙ

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

Правила:

- R1: pick-up(x)
 - C(R1):ontable(x)&clear(x)&handempty
 - A(R1):holding(x)
 - D(R1):ontable(x)&clear(x)&handempty
- R2:put-down(x)
 - C(R2):holding(x)
 - A(R2):ontable(x)&clear(x)&handempty
 - D(R2):holding(x)
- R3:stack(x, y)
 - C(R3):holding(x)&clear(y)
 - A(R3):on(x, y)&clear(x)&handempty
 - D(R3):holding(x)&clear(y)
- R4:unstack(x, y)
 - C(R4):on(x, y)&clear(x)&handempty
 - A(R4):holding(x)&clear(y)
 - D(R4):on(x, y)&clear(x)&handempty

Планирование в пространстве СОСТОЯНИЙ

Вступление

Виды планирования

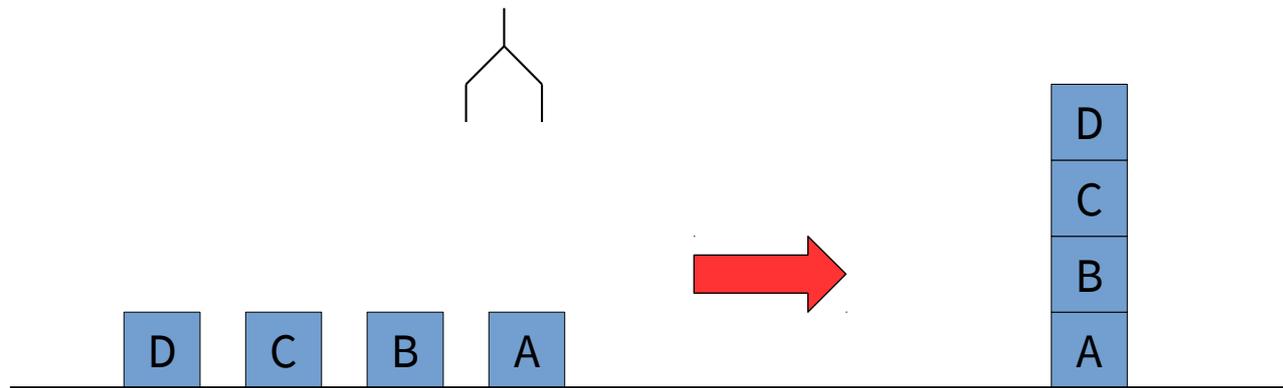
Планировщик
PuperPlan

Многоагентный
подход

Завершение

Пример плана:

*pick-up(b), stack(b, a), pick-up(c), stack(c, b),
pick-up(d), stack(d, c)*



Планирование на основе удовлетворения ограничений

Вступление

Виды
планирования

Планировщик
PugetPlan

Многоагентный
подход

Завершение

Задача удовлетворения ограничений - это тройка $\langle V, D, C \rangle$, где:

- $V = \{v_1, \dots, v_n\}$ - множество переменных.
- $D = \{D_1, \dots, D_n\}$ - множество доменов. Домен планирования - конечное множество, содержащее возможные значения переменных.
- $C = \{C_1, \dots, C_n\}$ - множество ограничений (отношений на множестве всех переменных).

Решением задачи является такая подстановка значений переменных, при котором удовлетворяются все ограничения.

Планирование на основе удовлетворения ограничений

Вступление

Виды планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Перед основной стадией работы Graphplan создаёт множество действий, осуществляя для каждого правила всевозможные варианты подстановки индивидов на места всех переменных. Имеется также специальный вид действия по-ор - "ничего не делать".
- Действия Acts - множество полностью конкретизированных правил, а также действие по-ор.
- Граф планирования PG - ориентированный ярусный граф с двумя типами узлов и с тремя типами рёбер.
- Узлы: 1) FN - множество узлов, ассоциированных с фактами F 2) AN - множество узлов, ассоциированных с действиями Acts.
- Множество узлов PG разбито на непересекающиеся подмножества $\langle FL_0, AL_0, FL_1, AL_1, \dots, AL_{n-1}, FL_n \rangle$, где FL - ярус, содержащий узлы-факты, AL - ярус, содержащий узлы-действия, FL_0 содержит узлы-факты, соответствующие фактам S_0
- ребро-предусловие - устанавливается между узлом-фактом fhf на некотором ярусе FL_i и узлом-действием anact на ярусе AL_i , если факт $f \in C(\text{act})$;
- ребро-добавление - устанавливается между узлом-действием an-act на некотором ярусе AL_i и узлом-фактом fhf на ярусе FL_{i+1} , если $f \in A(\text{act})$;
- ребро-удаление -- устанавливается между узлом-действием anact на некотором ярусе AL_i и узлом-фактом fnf на ярусе FL_{i+1} , если $f \in D(\text{act})$.

Планирование на основе удовлетворения ограничений

Вступление

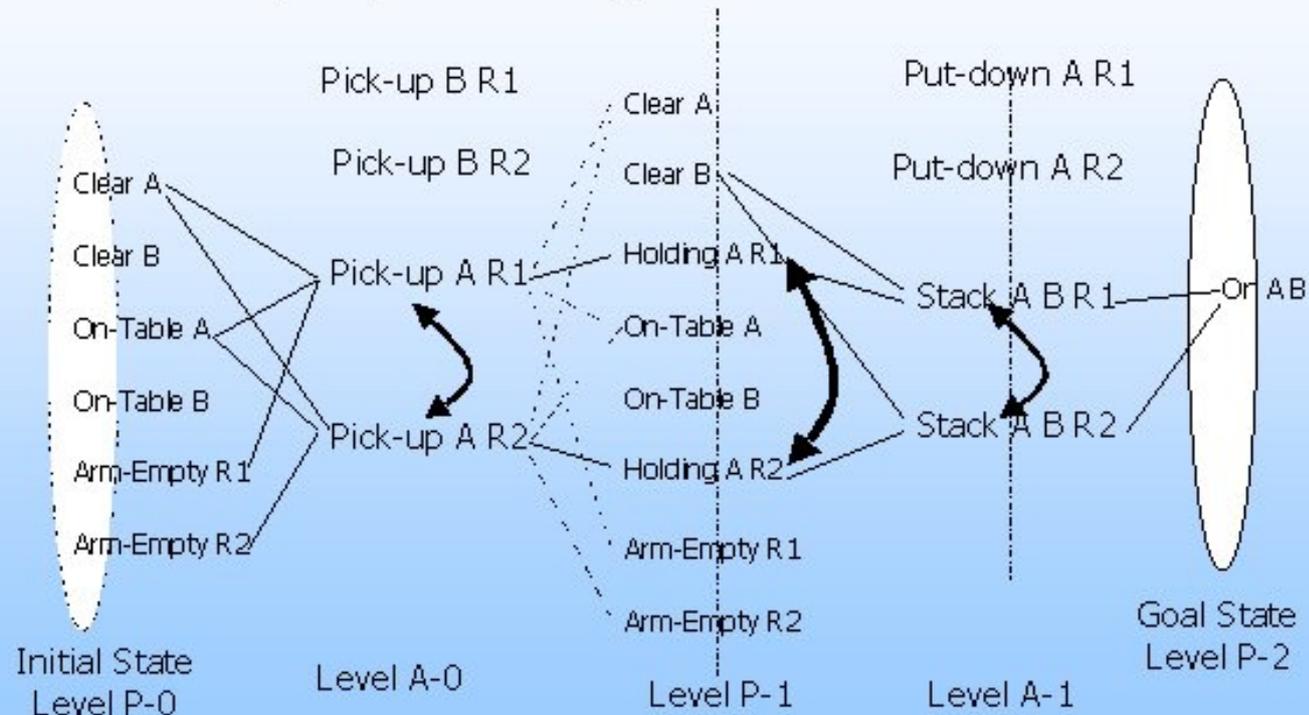
Виды планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

Graphplan Algorithm



Планирование на основе прецедентов

Вступление

Виды планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Постановка задачи планирования.
- Поиск похожих (сходство исходного состояния и целей) задач в библиотеке прецедентов.
- Модификация прецедента(замена целей и начальных условий).
- Адаптация некорректного (не достигающего цели) плана.
- Проверка полученного решения.
- Сохранение опыта текущей попытки планирования в библиотеке прецедентов.

Планирование на основе прецедентов

Вступление

Виды планирования

Планировщик
RuPerPlan

Многоагентный
подход

Завершение

Методы адаптации прецедентов:

- **Методы удовлетворения ограничений** - связывание переменных с конкретными объектами задачи без изменения структуры плана, либо замена объектов на новые.
- **Генеративная адаптация** - удовлетворение ещё не достигнутых целей методом, аналогичным порождению плана.
- **Разделение и слияние** - для каждой неудовлетворенной цели выполняется поиск решения независимо. Затем, решения объединяются.
- **Рекурсивная адаптация** - планирование по прецедентам для каждой из неудовлетворённых подцелей.
- **Адаптация по прецедентам** - использования адаптаций из библиотеки прецедентных адаптаций.
- **Эвристические методы адаптаций** - применение эвристических правил преобразования плана.

Планировщик Ruperplan

Вступление

Виды
планирования

Планировщик
RuperPlan

Многоагентный
подход

Завершение

- Albert-Ludwigs-Universität Freiburg
- Использует методы планирования в терминах STRIPS.
- Реализует выбор среди эвристик Astar search, weighted Astar search, greedy best first search, breadth first search, enforced hillclimbing search, iterative deepening search, sat solve.

Язык описания задачи и домена планирования PDDL.

Вступление

Виды планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Разработан Drew McDermott в 1998 году.
- Имеет множество версий: PDDL 1- PDDL3.1, MA-PDDL, PDDL+, NDDL, PPDDL и другие.
- Пример задачи и домена планирования на языке PDDL:

Домен планирования	Задача планирования
<pre>(define (domain domain-name) (:requirements :typing) (:types agent object1 - object) (:predicates (predicate-1 ?ob1 — object1 ?ag — agent) (predicate-2 ?ob1 — object1) (predicate-3 ?ag — agent)) (:action action-name :parameters (?ob1 — object1 ?ag — agent) :precondition (and predicate-1 ?ob1 ?ag)) :effect (and (not (predicate-1 ?ob1 ?ag)) (predicate-2 ?ob1) (predicate-3 ?ag — agent)))</pre>	<pre>(define (problem BLOCKS-4-0) (:domain blocks) (:objects a - object1 b - object1 ag1 — agent ag2 - agent) (:init (predicate-1 a ag1) (predicate-1 b ag2)) (:goal (and (predicate-2 a) (predicate-3 ag1) (predicate-2 b) (predicate-3 ag2)))</pre>

Astar поиск

Вступление

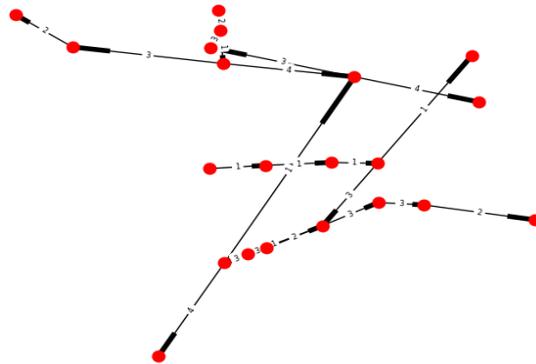
Виды планирования

Планировщик
HyperPlan

Многоагентный
подход

Завершение

- В процессе работы алгоритма рассчитывается функция $f(v) = g(v) + h(v)$, где $g(v)$ - выполненное количество действий для достижения текущего состояния v , $h(v)$ - эвристическая оценка количества действий до достижения целевой ситуации.
- Граф состояний для задачи 4 кубиков и башни:



- Используемые эвристики:
 - FF(Fast Forward) – не учитывая del эффекты действий
 - Deikstra

BFS - ПОИСК В ШИРИНУ

Вступление

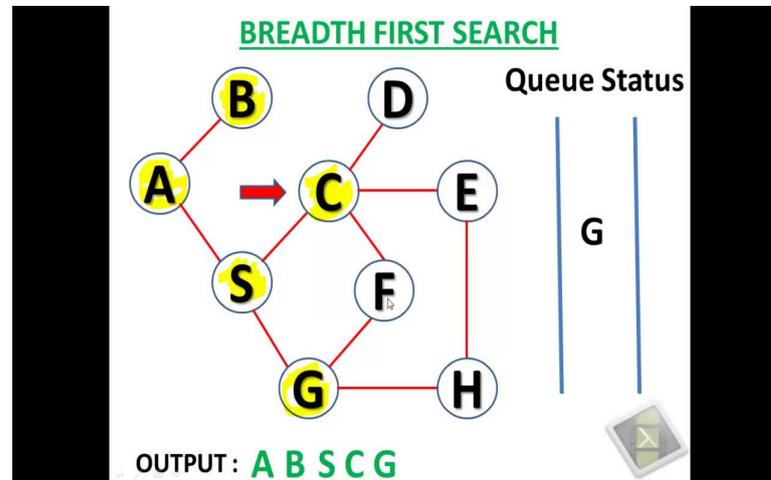
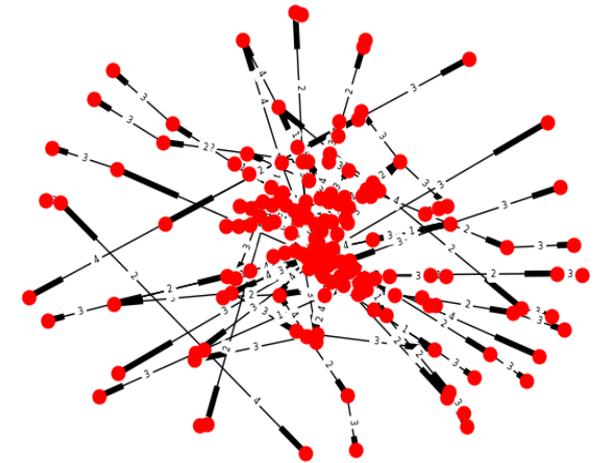
Виды планирования

Планировщик RuperPlan

Многоагентный подход

Завершение

```
BFS(start_node, goal_node) {  
  for(all nodes i) visited[i] = false; // изначально список посещённых узлов пуст  
  queue.push(start_node); // начиная с узла-источника  
  visited[start_node] = true;  
  while(! queue.empty() ) { // пока очередь не пуста  
    node = queue.pop(); // извлечь первый элемент в очереди  
    if(node == goal_node) {  
      return true; // проверить, не является ли текущий узел целевым  
    }  
    foreach(child in expand(node)) { // все преемники текущего узла, ...  
      if(visited[child] == false) { // ... которые ещё не были посещены ...  
        queue.push(child); // ... добавить в конец очереди...  
        visited[child] = true; // ... и пометить как посещённые  
      }  
    }  
  }  
  return false; // Целевой узел недостижим  
}
```



Enforced hillclimbing search

Вступление

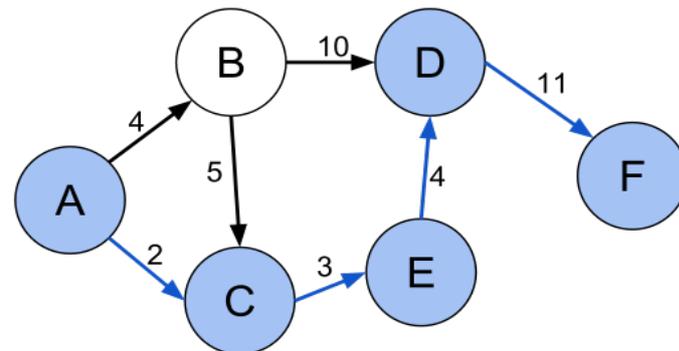
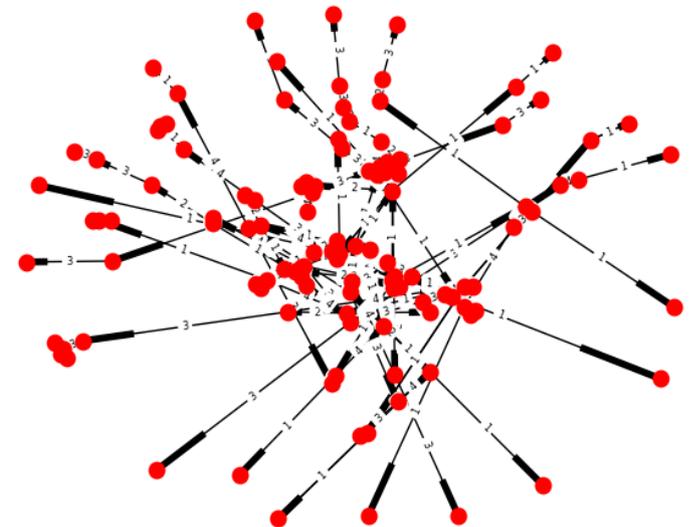
Виды планирования

Планировщик RuperPlan

Многоагентный подход

Завершение

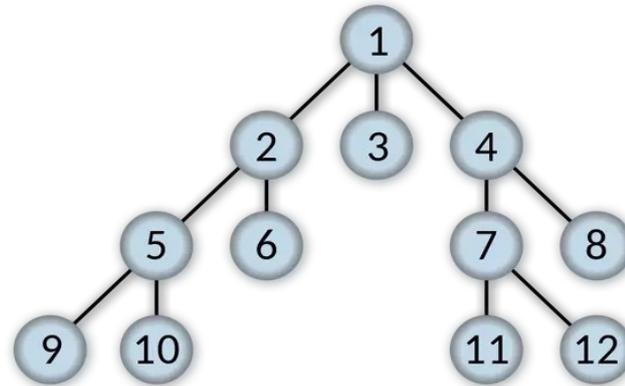
```
1: Procedure: EHCSearch
2: open_list = [initial_state];
3: best_heuristic = heuristic value of initial_state;
4: while open_list not empty do
5:   current_state = pop state from head of open_list;
6:   successors = the list of states visible from current_state;
7:   while successors is not empty do
8:     next_state = remove a state from successors;
9:     h = heuristic value of next_state;
10:    if next_state is a goal state then
11:      return next_state;
12:    end if
13:    if h better than best_heuristic then
14:      clear successors;
15:      clear open_list;
16:      best_heuristic = h;
17:    end if
18:    place next_state at back of open_list;
19:  end while
20: end while
```



Iterative deepening search

Вступление

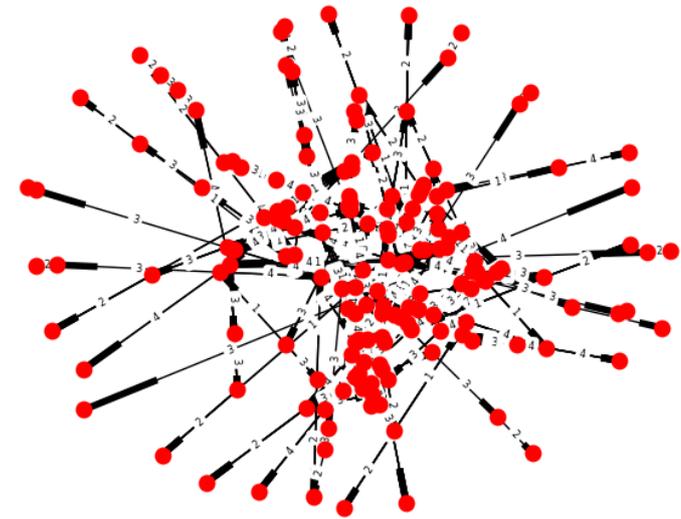
Виды планирования



Планировщик
PuperPlan

Многоагентный
подход

Завершение



МАС

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

- Рои, стаи, группы
- Нехватка ресурсов, наличие совместных целей, культурные/социальные нормы
- Многоагентные алгоритмы планирования
- Алгоритмы коммуникаций
- Способы рефлексии

Многоагентные алгоритмы планирования

Вступление

Виды
планирования

Планировщик
PuperPlan

Многоагентный
подход

Завершение

Основываясь на статье Granatyr MAC разделяют на:

- MAC основанная на численном подходе
- MAC основанная на когнитивном подходе
- MAC на гибридном подходе

Численный подход к планированию

Вступление

Виды планирования

Планировщик
RuperPlan

Многоагентный
подход

Завершение

Основан на численном агрегировании прошлых взаимодействий и представляет набор субъективных вероятностей того, что агенты будут правильно выполнять поставленную задачу. Большинство численных подходов используют статистические методы для вычисления доверия, байесовские вероятности, нечеткие множества, распределение вероятностей и функции Демпстера-Шафера. В рамках численного подхода упрощается процесс расширения системы новыми агентами и унифицируются правила построения взаимоотношений между агентами, но теряется возможность использования интеллектуальных агентов в качестве помощников при реализации многоагентного процесса планирования, участниками которого могут выступать как люди.

Основные работы:

- MAFS/Secure MAFS
- Алгоритм описанный Meng Guo
- Алгоритм описанный Jones E. G.

Algorithm 1 MAFS for agent φ_i

```
1: Insert  $I$  into open list
2: while TRUE do
3:   for all messages  $m$  in message queue do
4:     process-message( $m$ )
5:    $s \leftarrow \text{extract-min}(\text{open list})$ 
6:   expand( $s$ )
```

Algorithm 2 process-message($m = \langle s, g_{\varphi_j}(s), h_{\varphi_j}(s) \rangle$)

```
1: if  $s$  is not in open or closed list or  $g_{\varphi_i}(s) > g_{\varphi_j}(s)$  then
2:   add  $s$  to open list and calculate  $h_{\varphi_i}(s)$ 
3:    $g_{\varphi_i}(s) \leftarrow g_{\varphi_j}(s)$ 
4:    $h_{\varphi_i}(s) \leftarrow \max(h_{\varphi_i}(s), h_{\varphi_j}(s))$ 
```

Algorithm 3 expand(s)

```
1: move  $s$  to closed list
2: if  $s$  is a goal state then
3:   broadcast  $s$  to all agents
4:   if  $s$  has been broadcasted by all agents then
5:     return  $s$  as the solution
6: if the last action leading to  $s$  was public then
7:   for all agents  $\varphi_j \in \Phi$  with a public action whose public preconditions are satisfied in  $s$  do
8:     send  $s$  to  $\varphi_j$ 
9:   apply  $\varphi_i$ 's successor operators to  $s$ 
10: for all successors  $s'$  do
11:   update  $g_{\varphi_i}(s')$  and calculate  $h_{\varphi_i}(s')$ 
12:   if  $s'$  is not in closed list or  $f_{\varphi_i}(s')$  is now smaller than it was when  $s'$  was moved to closed list then
13:     move  $s'$  to open list
```

КОГНИТИВНЫЙ ПОДХОД К планированию

Вступление

Виды
планирования

Планировщик
PuperPlan

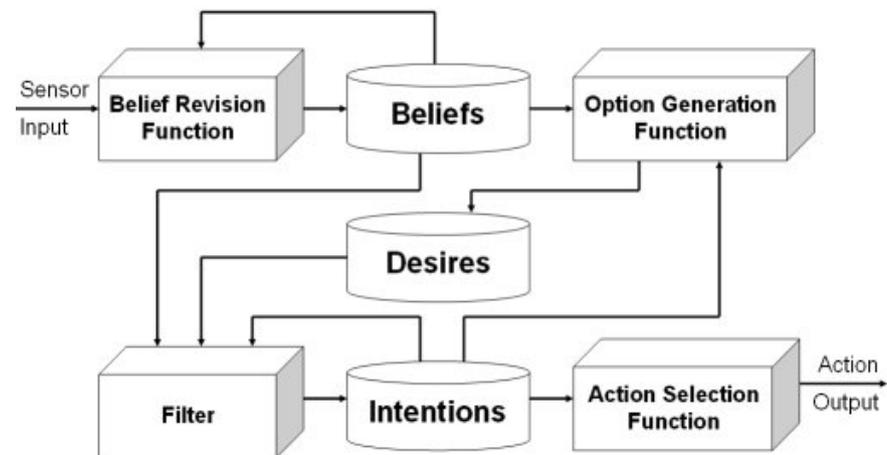
Многоагентный
подход

Завершение

В рамках многоагентного когнитивного планирования агенты формируют коалиции для совершения требуемых действий, основываясь на своих убеждениях и предпочтениях. Убеждения агента могут формироваться на основе взаимодействий с другим агентом, социальных норм, принятых в обществе, в котором находится агент, либо на основе психологического состояния самого агента. Как утверждалось ранее доверие является немаловажным фактором для построения многоагентных взаимоотношений.

Основные работы:

- BDI
- Алгоритм описанный Carter J.
- Алгоритм описанный Roy D.



Гибридный подход к планированию

Вступление

Виды
планирования

Планировщик
HyperPlan

Многоагентный
подход

Завершение

- Используют когнитивную архитектуру
- Имеют унифицированный способ добавления новых агентов
- Имеют унифицированный протокол коммуникаций агентов
- Имеют общий психологически правдоподобный способ представления знаний среди агентов (и искусственных, и естественных)

Thank You!

P.S. 7 раз спланируй, 1 раз дойди до цели...