

Сравнение оценочных функций при реализации эвристического поиска

Сабитов Эдем Шевкетович

КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ В.И. ВЕРНАДСКОГО
ТАВРИЧЕСКАЯ АКАДЕМИЯ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 601И)

e-mail: edem.sabitov@gmail.com

Рассматривается тестовая задача поиска в пространстве состояний, используемая для численного исследования эвристической силы различных эвристических оценок, используемых в алгоритме A^ , реализованном на языке Пролог.*

Эвристический поиск [1] является одним из методов поиска в пространстве состояний для решения интеллектуальных задач. Суть подобных задач состоит в том, что нам следует найти кратчайший путь перехода из начального состояния в целевое. Эвристический поиск использует оценочную функцию, с помощью которой можно сравнивать состояния в пространстве. Эта функция оценивает близость к целевому состоянию. Алгоритм A^* во взвешенном графе находит маршрут наименьшей стоимости от начальной вершины до выбранной конечной. Используемая при этом оценочная функция имеет вид:

$$f(n) = g(n) + h(n)$$

Здесь $g(n)$ соответствует расстоянию на графе от узла n до начального состояния, а $h(n)$ - оценка расстояния от n до узла, представляющего конечное (целевое) состояние. Чем меньше значение оценочной функции $f(n)$, тем «лучше», т.е. узел n лежит на более коротком пути от исходного состояния к целевому. Если $h(n)$ — нижняя оценка расстояния до целевого состояния, т.е. если $h(n)$ никогда не дает завышенной оценки расстояния, то алгоритм A^*

всегда отыщет оптимальный путь до цели при помощи оценочной функции $f(n)$. Алгоритм, обладающий таким свойством, называется разрешимым [1].

В качестве тестовой задачи рассматривается широко известная игра «8», облегчённый вариант более известной игры «15», отличающийся меньшими размерами игрового поля (3x3 вместо 4x4). На языке Пролог предложение, задающее вид целевого состояния, имеет следующий вид:

```
целевое([[1, 2, 3], [4, 5, 6], [7, 8, ' ']]).
```

Состояние задаётся списком, состоящим из трёх вложенных списков, каждый из которых соответствует строке игрового поля.

Поскольку у данной задачи пространство состояний состоит из двух частей, между которыми не существует переходов, начальное состояние имеет смысл задавать, отступив от целевого на заданное количество ходов [2].

Переходы в данной задаче соответствуют перемещениям свободной клетки. Поменять местами можно содержимое двух любых соседних клеток, если одна из них свободна. Например, переход, соответствующий обмену левой верхней клетки и клетки, соседней с ней по горизонтали, записывается в виде следующего предложения:

```
переход([[X11, X12, X13], L2, L3], [[X12, X11, X13], L2, L3], Name):-
X11=' ', Name = l; X12=' ', Name = r.
```

Здесь имя перехода в целях получения более компактного результата, сокращено до одной буквы (l — влево, r — вправо). Аналогичным образом задаются и другие варианты переходов.

Исходный код реализации алгоритма A* на Прологе задаётся следующим образом:

```
'A*' (Sol) :- start(Start), 'A*' (Start, Sol, OpenNum-ClosedNum), nl,
Vnum is OpenNum+ClosedNum, печать_реш(Sol), length(Sol, N1),
N is N1-1, write(длина=N), write(('перебрано' -Vnum, '
из них раскрыто'-ClosedNum, 'не раскрыто' -OpenNum)).
'A*' (Start, Sol, OpenNum-ClosedNum) :- h(Start, F),
a_star([v(Start, noparent, 0, F)], [], Sol-(OpenNum-ClosedNum)).
a_star(Opens, Closes, Sol-(OpenNum-ClosedNum)) :-
Opens=[v(V, Prev, _, _) | _], goal(V), !,
build_path(Prev, Closes, [V], Sol), length(Opens, OpenNum),
length(Closes, ClosedNum).
a_star([v(V, Prev, G, F) | Opens], Closes, Sol) :- findall(v(V1, V, G1, F1),
(next(V, V1, C), not(member(v(V1, _, _, _), Opens)),
not(member(V1, _, Closes))), G1 is G+C, h(V1, H1), F1 is G1+H1), Childs),
```

```
ord_insert_list(Childs,Opens,NewOpens),  
a_star(NewOpens,[V-Prev|Closes],Sol).  
a_star([],_,fail).
```

В качестве эвристических оценочных функций $h(n)$ для данной задачи рассматривается расстояние Хемминга (т.е. количество фишек, находящихся не на своих местах), манхеттенское расстояние (сумма разностей горизонтальной и вертикальной координат между текущим и целевым положениями фишки, просуммированная по всем фишкам), а также, для сравнения со слепыми методами поиска, функция, возвращающая тождественный нуль, превращающая алгоритм A^* в обычный поиск в ширину.

СПИСОК ЛИТЕРАТУРЫ

- [1] sapr-mgsu.narod.ru/biblio/ex-syst/Glava2/Index4.htm
- [2] Братко И. *Программирование на языке Пролог для искусственного интеллекта: Пер с англ.* – М.: Мир, – 1990.
- [3] Бондарев В. Н. Аде Ф. Г. *Искусственный интеллект: Учебное пособие для вузов.* – Севастополь: Изд-во СевНТУ, – 2002.