УДК 004.832

Анализ сложности задач для классического домена планирования Blocks World

Трофимов И.В., Институт программных систем РАН, г. Переславль-Залесский, igor@warlock-98.botik.ru

В работе рассматривается вопрос сравнения задач автоматического планирования по фактору сложности. Предложен способ вычисления размеров поискового пространства для классического домена Blocks World; приведены результаты. Также рассматривается способ сравнения эффективности методов абстрагирования на примере двух таких методов для данного домена.

Введение

Известно, что задачи автоматического планирования сложны. Согласно исследованию, проведенному Байлендером [1], практически все задачи планирования по сложности относятся к классу NP. Однако, причисление задач планирования к этому классу оставляет открытыми следующие вопросы. Насколько сложна каждая конкретная задача планирования? Как сравнить две конкретные задачи планирования? Чтобы ответить на эти вопросы, нужна более точная (по сравнению с классами сложности) *мера сложности* для задач планирования. В качестве простейших таких мер можно предложить мощности множеств вершин и дуг в поисковом пространстве задачи, при условии, что эти множества конечны. Эти величины вытекают из устройства домена и не зависят от того, какая конкретная цель поставлена в задаче и каково начальное состояние.

В данной статье мы рассмотрим, каким образом можно произвести точный расчет количества состояний и количества действий для конкретных задач из известного классического домена Blocks World (мир кубиков). Для анализа был взят вариант домена с четырьмя действиями: pickup, putdown, stack, unstack [2]. В статье также рассматривается способ сравнения эффективности двух методов абстрагирования, используемых для сокращения пространства поиска в данном домене.

Расчет количества состояний для задач из мира кубиков

Все, что есть в мире кубиков — это сами кубики, стол и манипулятор. Кубики образуют башни различной высоты, в том числе и выстой в один кубик.

Обозначим N — количество кубиков в задаче планирования (задачу с N=1 исключим из рассмотрения). Обозначим *States* — множество всех состояний задачи. Каждому состоянию S из *States* сопоставим кортеж T, содержащий набор высот башен, имеющихся в данном состоянии. Компоненты кортежа упорядочиваются по возрастанию высоты башен. Количество компонент равно количеству башен в S, а сумма компонент равна N, если манипулятор пуст, и N–1, если манипулятор содержит кубик. Пример состояния и соответствующего ему кортежа приведен на рисунке 1.

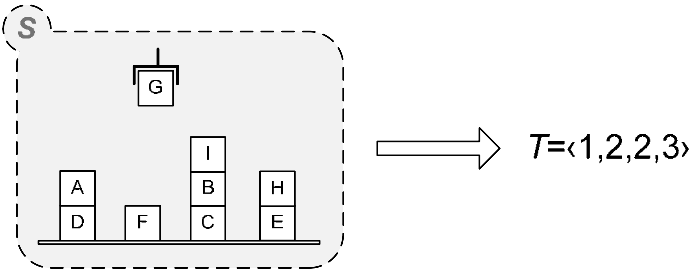

Рис.1. Пример состояния S и соответствующего ему кортежа T

Множество всех состояний, которым сопоставлен один и тот же кортеж будем называть **конфигурацией**. Другими словами, конфигурация — это совокупность всех состояний с одинаковым набором высот башен из кубиков. Графически конфигурацию можно изобразить так же, как и состояние, но удалив буквы с кубиков.

Обозначим *Confs* — множество всех конфигураций задачи. Для *Confs* справедливы следующие утверждения.

Утверждение 1. Конфигурации попарно не пересекаются.

$$\forall i, j\left(C_i \cap C_j = 0 \mid C_i \in Confs,\ C_j \in Confs,\ i \neq j\right) \tag{1}$$

Доказательство. Если бы пересечение какой-либо пары конфигураций было не пусто, то должно существовать состояние S', которое принадлежит каждой из этих конфигураций. А это означало бы, что состоянию S' сопоставлено два различных кортежа, что невозможно, так как одно состояние характеризуется только одним набором высот башен. □

Утверждение 2. Объединение всех конфигураций формирует множество состояний задачи.

$$\bigcup_{\forall i} C_i = States \tag{2}$$

Доказательство. С одной стороны, конфигурация может состоять только из состояний, принадлежащих *States*, — по определению. То есть не может быть состояния S', которое принадлежит C_i, но не принадлежит *States*. С другой стороны, каждое состояние из *States* принадлежит какой-то конфигурации из *Confs*, так как каждое состояние характеризуется некоторым кортежем *T*. Поэтому объединение всех конфигураций представляет собой множество состояний задачи. □

Из данных утверждений, а также из того факта, что всякая конфигурация является непустым множеством, следует, что *множество конфигураций является разбиением множества состояний задачи*, а сами конфигурации будут классами разбиения.

Из утверждений следует также, что количество состояний равно сумме мощностей конфигураций.

$$\sum_{\forall i} |C_i| = |States| \tag{3}$$

Отсюда видно, что для подсчета количества состояний необходимо определить, какие возможны конфигурации и мощность каждой из них.

Множеству конфигураций задачи с *N* кубиками при свободном манипуляторе соответствует множество разбиений числа *N* на слагаемые. Набор слагаемых моделирует башни из кубиков, а абсолютные величины слагаемых — высоты башен. Действительно, если диаграмму Юнга [3] развернуть на 90° против часовой стрелки, то мы получим набор башен из кубиков, описывающий некоторую конфигурацию.

Если мы рассматриваем конфигурацию с кубиком в манипуляторе, то следует отметить, что оставшиеся кубики образуют башни, соответствующие разбиению на слагаемые числа *N*–1.

Фактически, набор слагаемых (как элемент разбиения числа) соответствует «высотному» кортежу *T* при условии, что слагаемые упорядочены по возрастанию.

В качестве примера рассмотрим задачу с 5 кубиками. Множество разбиений числа 5 следующее:

{<5>, <1,4>, <2,3>, <1,1,3>, <1,2,2>, <1,1,1,2>, <1,1,1,1,1>}.

Это множество соответствует множеству конфигураций со свободным манипулятором (Рисунок 2)

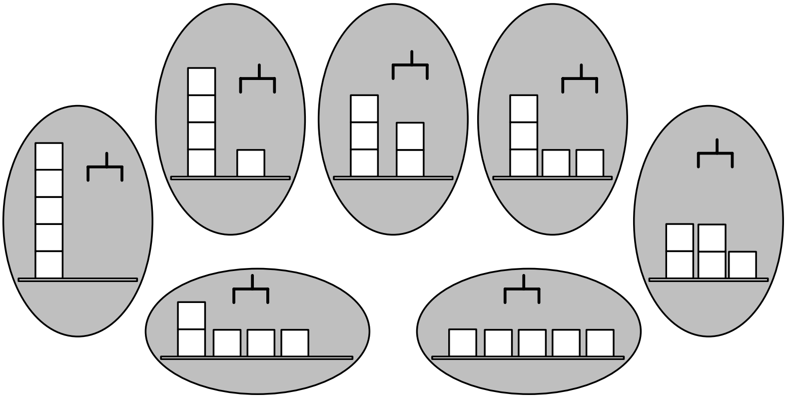

Рис.2. Конфигурации со свободным манипулятором для задач с пятью кубиками

Множество разбиений для 4 следующее:

{<4>, <1,3>, <2,2>, <1,1,2>, <1,1,1,1>}.

Это соответствует конфигурациям с кубиком в манипуляторе (Рисунок 3).

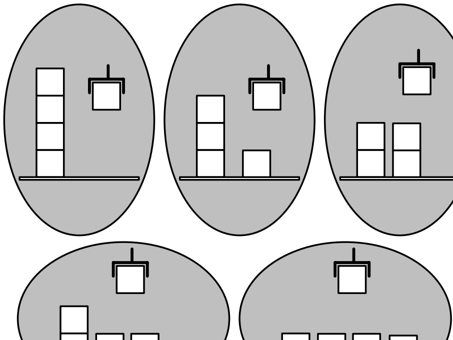

Рис.3. Конфигурации с кубиком в манипуляторе для задач с пятью кубиками

Всего для задачи с 5 кубиками получаем 12 различных конфигураций.

Теперь нужно установить, сколько состояний соответствует каждой конфигурации — мощность конфигурации.

В случае если в конфигурации нет башен одинаковой высоты, то конфигурации соответствует $N!$ состояний (множество перестановок кубиков в конфигурации). Следует отметить, что это утверждение распространяется как на конфигурации со свободным манипулятором, так и на конфигурации с кубиком в манипуляторе.

Когда в конфигурации присутствуют башни одинаковой высоты (далее, одновысотные башни), количество состояний будет меньше. Это происходит из-за того, что некоторые комбинации соответствуют одному и тому же состоянию. На рисунке 4 приведен пример двух различных комбинаторных перестановок, соответствующих одному состоянию.

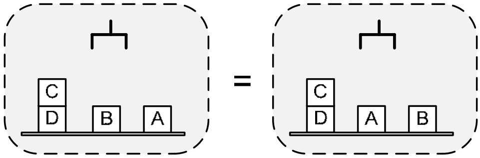

Рис.4. Состояние одно, но две комбинаторные перестановки

Принимая во внимание это обстоятельство, необходимо разделить N! на произведение факториалов от количества башен с одинаковой высотой. Например, при N=7 для конфигурации, соответствующей кортежу <1,1,1,2,2>, необходимо 7! разделить на 3!*2! (3 башни с высотой в один кубик и 2 башни с высотой в два кубика).

Действительно, если в некотором состоянии имеется несколько башен одинаковой высоты, то изменяя порядок их расположения на столе (друг относительно друга), мы будем получать различные комбинаторные перестановки, соответствующие все тому же состоянию. И все эти перестановки будут соответствовать одной конфигурации. Количество различных упорядочиваний одновысотных башен равно факториалу от количества этих башен.

Если наборов одновысотных башен несколько (как в случае конфигурации с T=<1,1,1, 2,2>), то соответствующие факториалы нужно перемножить.

Учитывая вышесказанное, не составит труда написать алгоритм для вычисления количества состояний в домене с N кубиками. Алгоритм порождения разбиений для заданного N хорошо известен [4]. Подсчет количества одинаковых слагаемых в элементе разбиения также не представляет сложности. Таким образом, мощность любой конфигурации, а следовательно, и сумма мощностей легко рассчитывается алгоритмически. В таблице 1 (колонка 2) приведены результаты вычисления количества состояний в домене Blocks World для различных N. Данная таблица является хорошей демонстрацией сложности задач из домена Blocks World и невозможности их решения алгоритмами, действующими по принципу грубой силы.

Таблица 1. Зависимость размеров поискового пространства от количества кубиков в домене Blocks World

N	Кол-во состояний	Кол-во действий	Усредненный коэффициент ветвления
2	5	8	1,6
3	22	42	~1,9
4	125	272	~2,2
5	866	2 090	~2,4
6	7 057	18 552	~2,6
7	65 990	186 578	~2,8
8	695 417	2 094 752	~3,0
9	8 145 730	25 951 122	~3,2
10	104 906 621	351 442 280	~3,35
11	1 472 425 142	5 162 569 082	~3,5
12	22 359 039 925	81 728 584 368	~3,65
13	365 088 510 242	1 386 695 814 842	~3,8
14	6 376 687 141 385	25 097 080 641 752	~3,94

Расчет количества действий для задач из мира кубиков

Расчет количества действий (или, другими словами, количества переходов в пространстве состояний) также будет опираться на понятие конфигурации.

В силу того, что в домене «мир кубиков» каждому действию соответствует обратное действие (pickup $\leftrightarrow$ putdown, stack $\leftrightarrow$ unstack) достаточно выяснить количество действий только для схем putdown и stack, а затем удвоить результат.

Количество действий putdown равно количеству состояний с кубиком в манипуляторе, так как всегда возможно положить кубик на стол. Метод расчета этой величины был приведен выше.

Количество действий stack в каждой конфигурации с кубиком в манипуляторе равно количеству состояний, соответствующей данной конфигурации, умноженному на количество башен в конфигурации (так как кубик можно установить на вершину любой из башен). Результаты по каждой такой конфигурации в сумме дают количество действий stack в задаче.

В таблице 1 (колонка 3) приведены результаты вычисления количества действий в зависимости от величины *N*.

Способ сравнения эффективности методов абстрагирования

Методы абстрагирования, на примере которых будет продемонстрирован способ сравнения, опираются на исключение из рассмотрения ряда действий, то есть по существу являются фокусировкой [5,6].

Метод 1. Значимый контекст рассуждений составляют:

- действия unstack для всех пар кубиков, которые в начальном состоянии находятся в отношении *on*;
- действия stack для всех пар кубиков, которые в цели находятся в отношении *on*;
- все действия, производные от схем pickup и putdown.

Остальные действия удаляются из рассмотрения. □

Метод 2. Пусть *GO* — множество кубиков, фигурирующих в формулировке цели. Обозначим *SO* — множество кубиков, которые в начальном состоянии находятся либо «непосредственно под» кубиками из *GO*, либо «непосредственно или опосредовано над» ними. Все действия, оперирующие кубиками *только* из *SO*, составляют значимый контекст рассуждений. Остальные действия можно абстрагировать. □

Метод абстрагирования тем эффективнее, чем в большей степени он упрощает поиск решения задачи. Чтобы иметь возможность сравнивать методы абстрагирования по эффективности, требуется некоторое количественное представление этой «степени упрощения». Будем называть данную количественную величину *коэффициентом абстрагирования* и обозначать символом E_a.

Коэффициент абстрагирования определим как отношение какой-либо количественной меры сложности задачи (D) после и до абстрагирования.

$$E_a = \frac{D_{abstract}}{D_{initial}} \tag{4}$$

Величина E_a принимает значения из диапазона [0; 1]. Чем меньше E_a, тем проще задача.

Что может выступать в качестве такой меры сложности? Простейший вариант такой меры — это размер пространства поиска (количество состояний или количество переходов между состояниями). Так как предложенные выше методы абстрагирования не оперируют состояниями в явном виде (хотя, разумеется, некоторые состояния могут стать недостижимыми после удаления действий), использовать количество состояний в поисковом пространстве как меру сложности неудобно. Поэтому в качестве меры сложности выберем усредненный коэффициент ветвления (отношение количества переходов к количеству состояний).

Усредненный коэффициент ветвления для исходных задач планирования в мире кубиков приведен в таблице 1 (колонка 4). Из таблицы видно, что коэффициент растет, хотя скорость роста замедляется.

Величина $D_{abstract}$ при выбранной мере сложности будет зависеть от конкретной решаемой задачи. Действительно, для одних задач может быть предпочтителен один метод, для других — другой. Для демонстрации рассмотрим следующую задачу (Рисунок 5).

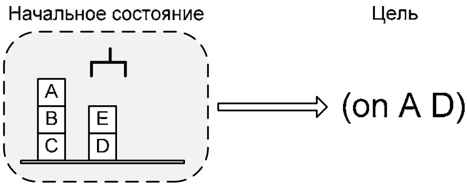

Рис.5. Демонстрационная задача из мира кубиков

Первый метод абстрагирования оставит в фокусе действия вида unstack(A,B), unstack(B,C), unstack(E,D), stack(A,D), а также весь ассортимент действий от схем pickup и putdown. Остальные действия исключаются из рассмотрения.

Второй метод исключает действия вида stack(C,x), stack(x,C), unstack(C,x), unstack(x,C), pickup(C), putdown(C).

В таблице 2 отражено влияние методов абстрагирования на сложность демонстрационной задачи.

Таблица 2. Влияние методов абстрагирования на сложность демонстрационной задачи

Количество действий до абстрагирования	2090
—//— после абстрагирования методом 1	866
—//— после абстрагирования методом 2	1400
$D_{initial}$	~2,41
$D_{abstract\_1}$	1
$D_{abstract\_2}$	~1,62

Из таблицы видно, что для данной конкретной задачи первый метод более предпочтителен, так как в большей степени сокращает пространство поиска.

К сожалению, предложенная мера сложности (усредненный коэффициент ветвления) также недостаточно точна. Например, $D_{abstract\_1} = 1$ не означает, что из каждой вершины пространства поиска исходит только одна дуга (и поиск решения осуществляется за линейное время). В действительности имеется масса состояний, из которых нет ни одной исходящей дуги, а в некоторых состояниях их более одной. Кроме того, многие состояния станут недостижимыми после осуществления абстрагирования. Фактически, усредненный коэффициент ветвления в достижимой части пространства поиска будет чуть больше единицы.

Тем не менее, данная мера позволяет оценивать степень влияния метода абстрагирования на сложность задачи и сравнивать различные методы абстрагирования.

Заключение

Для сравнения сложности задач планирования нужны точные меры сложности. В частности, это необходимо для оценки эффективности методов абстрагирования на конкретных задачах, а также для сравнения методов абстрагирования. Рассмотренные в данной работе меры, основанные на оценке размеров поискового пространства задачи, позволяют выполнять сравнение, но могут быть недостаточно точны. Более точные меры должны учитывать факторы цели и начального состояния. В частности, необходимо принимать во внимание частоту встречаемости цели в пространстве состояний и ее распределение. Так, например, в мире кубиков цель (arm-empty) всегда достижима за одно действие (если не является уже достигнутой в начальном состоянии). Сложность такой задачи не связана с размером пространства состояний, и абстрагирование на сложность не влияет.

В работе предложена мера сложности для задач планирования и описан способ ее вычисления в домене «мир кубиков». В данном домене вычисление таких параметров как количество состояний и переходов осуществить несложно. Однако более желательным был бы доменонезависимый и полностью автоматизированный способ вычисления этих параметров или иной меры сложности.

Работа выполнена при поддержке РФФИ (проект № 07-01-00763-а) и Президиума РАН (проект № 223)

Литература

1. *T. Bylander*. The Computational Complexity of Propositional STRIPS Planning. Artificial Intelligence 69(1-2):165-204, 1994.
2. *Estlin T.A.* Integrating Explanation-Based and Inductive Learning Techniques to Acquire Search-Control for Planning. Technical Report AI96-250, Department of Computer Sciences, University of Texas, Austin, TX, 1996.

3. *Эндрюс Г.* Теория разбиений. — М.:Наука. Главная редакция физико-математической литературы, 1982. — 256 с.
4. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика. — М.:Мир. 1980. — 476 с.
5. *Трофимов И.В*. Новые методы определения и выявления значимого контекста рассуждений. // Труды Второй международной конференции "Системный анализ и информационные технологии" САИТ-2007: В 2-х томах. — М.: ЛКИ, 2007, т.1, стр. 179—182.
6. http://ai-center.botik.ru/planning/workshops/ 2007_Trofimov_SignificantContextAsAbstraction.pdf