

УДК 004.8

Планирование в средах с большим количеством объектов*

И.В. Трофимов¹

Автоматическое планирование — задача высокой вычислительной сложности. Классические планировщики не способны решать эту задачу в предметных областях с большим количеством объектов. Как правило, исследователи пытаются решить эту проблему при помощи эвристик. В этой работе представлен метод, позволяющий существенно сократить пространство поиска за счет выделения значимого, с точки зрения конкретной задачи, контекста в рамках описания предметной области.

Введение

В работах [Bylander, 1991] [Bylander, 1994] [Erol et al., 1991], посвященных теории сложности задачи планирования, было показано, что даже при очень сильных ограничениях задача интеллектуального планирования имеет неполиномиальную сложность. В общем случае задача планирования является EXPSPACE-полной (для STRIPS-формализма [Fikes et al., 1971]). Например, если мы воспользуемся планированием в пространстве состояний, то величина пространства поиска равна 2^n , где n — количество независимых и изменяемых высказываний о мире, выраженных атомарными формулами.

В то же время, практический интерес представляют как раз задачи с большим значением n . В этой работе рассматривается метод, который позволяет значительно сузить пространство поиска и, таким образом, сделать решаемыми довольно сложные задачи.

1. Об эвристиках в планировании

Очевидно, что при таких размерах пространства поиска не имеет смысла применять полный перебор для решения задачи планирования. Единственным разумным решением является сосредоточение усилий на

* Работа выполнена при финансовой поддержке РФФИ (проект № 03-01-00853а)

¹ 152020, г. Переславль-Залесский, м. Ботик, ИПС РАН, igor@warlock-98.botik.ru

разработке сложных эвристик, позволяющих в тех или иных предметных областях сокращать пространство поиска до приемлемых размеров.

Большинство современных планировщиков используют эвристики, основанные на выборе наиболее перспективных ветвей поиска. Для этого они используют планирование от текущего узла дерева поиска вперед на небольшую глубину (локальный поиск) и выбирают дальше ту ветвь, которая в наибольшей степени приближает планировщик к цели. Однако такой подход опасен попаданием в локальный экстремум. В качестве примеров таких планировщиков можно привести современные системы HSP [Bonet et al., 1998] и FF [Hoffmann et al., 2001].

Еще одним перспективным методом, позволяющим избежать полного перебора, является настройка системы на решение определенного класса задач в данной предметной области. К настоящему моменту выполнено множество попыток применить обучение для настройки планировщика на предметную область, начиная от макрооператоров для STRIPS [Fikes et al., 1972] и заканчивая современными подходами, пользующимися методологией EBL (explanation-based learning) [Minton et al., 1990] и CBR (case-based reasoning) [Rousu, 1997] [Spalazzi, 2001]. В этих системах основное внимание уделяется *"методам" решения задач*. Планировщики обучаются типичным способам решения конкретных проблем в рамках данной предметной области. В дальнейшем эти "методы" (наборы действий) могут быть использованы для решения аналогичных задач, но, например, оперирующих другими объектами.

Однако целью обучения могут быть не только методы решения. Например, можно выделять в рамках заданной предметной области **значимый контекст рассуждений** — набор объектов, которые необходимо принимать во внимание при решении заданной задачи. В этом случае обученная система довольно жестко привязывается к набору объектов, фигурирующих в предметной области. Но при этом очень сильно сужается пространство поиска. К тому же, такой подход позволяет оперировать специфичными свойствами объектов в данной конкретной предметной области.

Причиной того, что такой вид обучения не получил должного внимания, может оказаться исторически сложившаяся форма разбиения предметной области на домен планирования и задачу планирования. В STRIPS и его современных, более развитых аналогах (например, PDDL [Ghallab et al., 1998] [Fox et al., 2003]) доменом планирования считается множество возможных схем действий (в PDDL в описание домена входит большее число сущностей). Множество объектов предметной области является элементом задачи. Это дает определенную гибкость в том смысле, что мы одно и то же устройство (представленное набором

действий) можем использовать в разных средах (представленных набором объектов и связями между ними). Но есть и недостаток. Мы теряем возможность манипулировать знаниями об объектах в каждой конкретной среде.

2. О целесообразности применения значимого контекста рассуждений

Рассмотрим примеры, когда выделение и использование значимого контекста рассуждений может оказаться полезным. Существует множество достаточно статичных предметных областей, в которых знания об объектах играют ключевую роль при принятии решений. В частности, примером может послужить транспортная задача.

Пусть у нас есть множество маршрутов транспортных средств. Количество этих маршрутов может быть довольно большим. Кроме того, сами транспортные средства могут быть разнотипны. Будем решать задачи перемещения людей из одного месторасположения в другое.

Мы можем описать действия транспортировки в нотации PDDL, например, следующим образом:

<pre>(:action fly :parameters (?from, ?to – location, ?who – person) :precondition (and (at ?who ?from)(flight ?from ?to)(not (= ?from ?to))) :effect (and (not (at ?who ?from)) (at ?who ?to)))</pre>
<pre>(:action train_transportation :parameters (?from, ?to – location, ?who – person) :precondition (and (at ?who ?from) (train_route ?from ?to) (not (= ?from ?to))) :effect (and (not (at ?who ?from)) (at ?who ?to)))</pre>
<pre>(:action taxi_transportation :parameters (?from, ?to – location, ?who – person) :precondition (and (at ?who ?from) (not (far_from ?from ?to)) (not (= ?from ?to))) :effect (and (not (at ?who ?from)) (at ?who ?to)))</pre>

Этот список может быть продолжен. Здесь первое действие соответствует перелету на самолете, второе — поездке на поезде, третье — не такси. Предикат `at` — отражает местоположение объекта (человека), `flight` и `train_route` — наличие рейса на самолете и поезде соответственно, `far_from` — расположение объектов на большом расстоянии.

Кроме описания действий, в предметной области описывается множество рейсов транспортных средств и множество географических точек (с качественным указанием расстояний), между которыми выполняются рейсы. Еще раз отметим, что эти сущности достаточно

статичны. Их все следовало бы описывать в поле ":constants" описания домена в PDDL нотации. Но, к сожалению, язык не допускает описания отношений в поле констант. Множество персон в данном случае, динамический аспект. Их разумно описывать как элемент задачи.

Теперь рассмотрим задачу переезда персоны X из одного города в другой. Пусть X находится на железнодорожном вокзале города Ярославль, и ему требуется попасть в Нью-йоркский аэропорт им. Кеннеди. Типичным решением здесь будет: "на поезде до Москвы", далее "на такси в аэропорт Шереметьево" и "на самолете до Нью-Йорка".

Как будет действовать планировщик. Если мы пойдем от цели (at X, New-York_Kennedy_Airport), то при данном наборе действий у нас есть множество способов попасть в целевой аэропорт. Например, мы можем приехать на такси из близлежащего населенного пункта или прилететь на самолете из Зимбабве. Однако с практической точки зрения при поездке из Ярославля нет смысла рассматривать все эти направления. В 99% случаев нас будет интересовать только аэроперелет Москва — Нью-Йорк.

В аналогичную ситуацию мы попадаем, если воспользуемся прогрессией в качестве стратегии поиска планировщика. Из Ярославля можно заехать, например, в близлежащую деревню, что не приблизит нас к цели.

При большом количестве пересадок эвристический локальный поиск может не спасти. Однако может помочь обучение. Если мы будем учиться только методам, что и реализовано в большинстве систем, использующих CBR, то мы получим набор типичных решений задачи транспортировки без привязки к предметной области. Т.е. мы будем обладать множеством последовательностей действий, которые встречались в обучающей выборке. Но, даже обладая правильным "абстрактным планом" (а его еще надо найти среди множества прецедентов) вида

train_transportation(?x, ?y, ?person), fly(?y, ?z, ?person),

мы можем получить множество нетипичных (как правило, нерелевантных) решений, таких как "поездом до Владивостока, а там в Нью-Йорк".

Этот пример отчетливо демонстрирует неспособность современных методов планирования справляться с задачами такого рода. Таким образом, возникает необходимость искать какой-то новый способ ограничения пространства поиска.

Одним из возможных путей видится введение понятия релевантных сущностей с точки зрения задачи. Сузив контекст рассуждений до релевантного с точки зрения конкретной задачи, мы можем в дальнейшем быстро получить "хорошее" решение при помощи локального поиска или CBR. В примере с поездкой из Ярославля в Нью-Йорк нет смысла

рассматривать маршруты на Зимбабве. Зато Москва является релевантным объектом с точки зрения данной задачи.

3. Технология

Каким образом можно сузить контекст рассуждений? Целесообразным кажется использовать подход, предлагаемый CBR. Однако в качестве прецедентов нужно хранить не методы решения задачи, а множество наиболее релевантных объектов и действий с точки зрения данной задачи. Для примера с поездкой из Ярославля в Нью-Йорк релевантными будут объекты: Ярославский ж.-д. вокзал, Ярославский вокзал в Москве, аэропорт Шереметьево, Нью-йоркский аэропорт Кеннеди; и действия: поездка на поезде, поездка на такси, полет на самолете (но никак не плавание на яхте).

Методология CBR предполагает хранение прецедентов в виде индексированного списка. Индекс отображает постановку задачи в решение. Как поступать, если мы сталкиваемся с новой задачей, которой еще нет в индексе? Обычно в индексе ищется наиболее похожая задача и соответствующее ей решение используется в качестве основы для построения нового решения. Т.к. мы оперируем множествами релевантных сущностей, мы можем взять целый набор похожих задач (превышающих некоторый порог схожести) и считать релевантным контекстом объединение соответствующих множеств.

Следующий вопрос, который следует рассмотреть, это способ построения библиотеки прецедентов. В этих целях можно воспользоваться методами машинного обучения. Безусловно, допустимо обучение по обучающей выборке. Кроме того, сама методология CBR предполагает постепенное накопление опыта. По мере получения новых решений, можно сохранять их, как отдельные элементы библиотеки прецедентов или модифицировать (например, обобщать или дополнять) уже имеющиеся элементы. Видимо, имеет смысл применить оба подхода: обучающую выборку для начального заполнения библиотеки прецедентов и стандартную методологию CBR для уточнения и поддержания ее в актуальном виде.

После того, как принято решение сохранить полученный план в качестве прецедента, нужно определить, какие сущности должны войти в контекст, соответствующий решенной задаче. Безусловно, туда должны попасть все действия и "статичные" объекты, упомянутые в плане. Нужно ли включать в контекст какие-то еще объекты? Видимо это зависит от предметной области и наших требований к планировщику. Если мы хотим получать существенно новые решения, то возможно имеет смысл пополнить этот контекст еще какими-то объектами. Принятие решения по

этому вопросу можно, например, возложить на эвристики, сети ассоциаций или другие методы.

Выводы

Планирование, опирающееся на выявление значимого контекста рассуждений, может дать существенный выигрыш в скорости поиска и качестве построенных планов в предметных областях со статичной структурой. Более того, иногда такой подход будет даже единственным возможным решением. Возможно, в ряде предметных областей этот метод сможет даже стать ключом для перехода на прикладной уровень.

Список литературы

[Bonet et al., 1998] Bonet B., Geffner H. HSP: Heuristic Search Planner. // Entry at the AIPS-98 Planning Competition. 1998.

[Bylander, 1991] Bylander T. Complexity Results for Planning. // IJCAI'91. 1991.

[Bylander, 1994] Bylander T. The Computational Complexity of Propositional STRIPS Planning. // Artificial Intelligence. 1994. № 69 (1-2).

[Erol et al., 1991] Erol K., Nau D. S., Subrahmanian V. S. Complexity, decidability and undecidability results for domain-independent planning: A detailed analysis. // Tech. Rep. CS-TR-2797, UMIACS-TR-91-154, SRC-TR-91-96. 1991.

[Fikes et al., 1971] Fikes R.E., Nilsson N.J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. // IJCAI'71. 1971.

[Fikes et al., 1972] Fikes R.E., Hart P.E., Nilsson N.J. Learning and Executing Generalized Robot Plans. // Artificial Intelligence. 1972. № 3 (1-3).

[Fox et al., 2003] Fox M., Long D. PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. // Journal of Artificial Intelligence Research. 2003. № 20.

[Ghallab et al., 1998] Ghallab M., Howe A., Knobloch C., McDermott D., Ram A., Veloso M., Weld D., Wilkins D. PDDL - The Planning Domain Definition Language (version 1.2). // AIPS'98. 1998.

[Hoffmann et al., 2001] J. Hoffmann, B. Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. // Journal of Artificial Intelligence Research. 2001. № 14.

[Minton et al., 1990] Minton S., Carbonell J., Knobloch C., Kuokka D.R., Etzioni O., Gil Y. Explanation-Based Learning: A Problem-Solving Perspective. // Artificial Intelligence. 1989. № 40.

[Rousu, 1997] Rousu J. Case-based Planning. // In Seminar on Knowledge Engineering, Helsinki University of Technology. 1997.

[Spalazzi, 2001] Spalazzi L. A Survey on Case-Based Planning. // Artificial Intelligence Review. 2001. № 16.