

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕЛИНЕЙНЫХ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

С. В. Добрецов

Санкт-Петербургский государственный технический университет, Политехническая 29, Санкт-Петербург, 195251, РОССИЯ, тел. (812) 535-4739, e-mail sdobretsov@mail.ru

АННОТАЦИЯ

На основании экспериментальных данных проведен сравнительный анализ нелинейных алгоритмов планирования действий интеллектуальных агентов относительно наиболее важных с прикладной точки зрения показателей производительности. Предложен критерий оптимальности алгоритма планирования и продемонстрирована его практическая применимость.

1. ВВЕДЕНИЕ

Изучение проблемы построения планов действий являются важной составляющей теории решения задач и занимают одно из ведущих мест в области исследования интеллектуальных систем [1]. Однако подавляющее большинство опубликованных в рамках данной тематики работ посвящено созданию новых алгоритмов планирования, тогда как вопросы сравнения существующих алгоритмов и определения среди них наиболее оптимального рассматриваются крайне редко [10]. Между тем выбор адекватного алгоритма решения конкретной задачи планирования способен значительно повысить эффективность функционирования интеллектуального агента как на этапе построения плана действий, так и на этапе его выполнения.

2. ПЛАНИРОВАНИЕ ДЕЙСТВИЙ В ИСКУССТВЕННОМ ИНТЕЛЛЕКТЕ

Неформальное описание задачи планирования заключается в следующем: активный элемент (агент), находясь в некотором окружении (среде), работает над достижением определенной цели. В каждый момент времени среда находится в некотором состоянии, при этом агент может выполнять действия, изменяющие состояние среды. Задача планирования состоит в нахождении последовательности действий, которые позволяют агенту перевести систему из некоторого исходного состояния в заданное

целевое состояние. В общем случае целевое множество может состоять из нескольких состояний, достижение любого из них означает достижение цели. Возможна также ситуация, когда ни одно из этих состояний недостижимо, например, в силу того или иного поведения среды.

Формально под задачей планирования понимается двойка $\langle I, G \rangle$, где компонента I определяет множество начальных состояний, а компонента G задает желаемое поведение среды, например, в виде множества целевых состояний. Для представления задачи планирования используются два подхода:

1. Представление в пространстве состояний [8]: дана система агент-среда $M=(Q, A, \Gamma_M)$, где:

Q – множество наблюдаемых состояний;

A – множество действий;

$\Gamma_M: Q \times A \rightarrow Q$ – функция перехода, определяющая для каждого состояния $q \in Q$ и действия $a \in A$ следующее состояние $q' = \Gamma_M(q, a)$;

начальные условия $I \subset Q$ и множество целевых состояний $G \subset Q$.

В этом случае под решением (планом) понимается упорядоченное множество действий $P = \{a_1, \dots, a_n\}$, такое что суперпозиция функций перехода $\Gamma_M(\Gamma_M(\dots \Gamma_M(\Gamma_M(q_0, a_1) a_2) \dots, a_{n-1}), a_n) \in G$ при любом $q_0 \in I$.

2. Представление в пространстве частичных планов с использованием языка описания действий (Action Description Language, ADL) [6], в котором каждое возможное действие (оператор) $o \in A$ описывается через два набора формул логики предикатов первого порядка, называемых предусловиями $Pre(o)$ и постусловиями $Post(o)$ данного оператора. Для данного представления решением задачи планирования $\langle I, G \rangle$ называется

упорядоченное множество (цепочка) операторов $P = \{a_1, \dots, a_n\}$, $a_i \in A$, удовлетворяющая следующим условиям [9]:

1. S является выполнимым, то есть $I \vdash \text{Prec}(a_1)$, $a_{k-1}(a_{k-2} \dots (a_1(I))) \vdash \text{Prec}(a_k)$, $2 \leq k \leq n$
2. Последовательность состояний I , $a_1(I), \dots, a_n(a_{n-1} \dots (a_1(I)))$ удовлетворяет условиям G задачи планирования.

Решение P задачи планирования $\langle I, G \rangle$ называется минимальным, если любая последовательность операторов, получаемая удалением произвольного количества элементов из P , не является решением задачи планирования.

3. НЕЛИНЕЙНЫЕ АЛГОРИТМЫ ИХ ОБОБЩЕННОЕ ПРЕДСТАВЛЕНИЕ

Как было отмечено в [11], нелинейные алгоритмы являются одним из наиболее перспективных направлений исследований в области систем планирования. Применение универсального подхода, состоящего в постепенном добавлении ограничений, сужающих множество поиска, и отложенное упорядочивание шагов обеспечивают широкую область применения данного класса алгоритмов. В первую очередь это относится к неразрешимым с помощью линейных алгоритмов задачам многоцелевого планирования с аномалией Суссмана, заключающейся в необходимости чередования действий, направленных на достижение разных подцелей.

Между тем может быть показано, что большинство известных нелинейных алгоритмов планирования, может быть представлено в виде универсальной последовательности, при этом различия между алгоритмами оказываются локализованными на уровне детальной реализации процедур, соответствующих каждому шагу. В частности, в работе [9] была предложена следующая модель обобщенного нелинейного алгоритма планирования:

1. **Конструктор решений**, проверяющий выполнения одного из следующих условий для данной ветви поиска: решение задачи планирования найдено, решения не существует, необходима доработка текущего частичного плана. В первых двух

случаях выполнение алгоритма завершается.

1. **Модификация** частичного плана для достижения некоторого целевого условия, состоящая из следующих шагов:
 - Выбор промежуточной подцели в частичном плане P в соответствии с некоторой стратегией.
 - Достижение выбранной подцели с помощью некоторого (при необходимости, нового) шага s .
2. **Консервация** результатов путем введения дополнительных ограничений с целью предохранения накопленных на данном шаге наработок от последующих изменений.
3. **Оптимизация** текущего плана с целью выявления и разрешения конфликтов между шагами плана и набором ограничений.
4. **Проверка состоятельности** для отсека частичных планов, не имеющих корректного завершения.
5. **Рекурсивный вызов** алгоритма для модифицированного частичного плана.

В таблице 1 приведено сравнения трех основных алгоритмов нелинейного планирования – NONLIN [4], TWEAK [5] и SNLP [7] – на основе обобщенного алгоритма.

Таблица 1. Нелинейные алгоритмы.

Алгоритм	NONLIN	TWEAK	SNLP
Конструктор решений	Набор подцелей (SG)	На основе МКИ (MTC)	Набор подцелей (SG)
Выбор подцели	Произвольный (ARB)	На основе МКИ (MTC)	Произвольный (ARB)
Стратегия консервации	Одностороннее предохранение причинных связей (SGL)	Отсутствует (NONE)	Двустороннее предохранение причинных связей (DBL)
Стратегия оптимизации	Разрешение конфликтов (CFT)	Отсутствует (NONE)	Разрешение конфликтов (CFT)

В то время как NONLIN и SNLP используют концепцию интервальных ограничений на этапе консервации и процедуры обнаружения и разрешения конфликтов на шаге оптимизации, предоставляя свободу при выборе промежуточной подцели, TWEAK активно применяет модальный критерий истинности (пробное исполнение текущего плана), что позволяет обходиться без консервации и

оптимизации. С учетом зависимости между конструктором решений и процедурой выбора подцели, пространство алгоритмов планирования может быть определено как множество элементов вида $g - c - o$, где:

$g \in G = \{ MTC, ARB \}$ – способ выбора подцели (и соответствующий конструктор решений);

$c \in C = \{ NONE, SGL, DBL \}$ – используемая стратегия консервации;

$o \in O = \{ NONE, CFT, ORD \}$ – стратегия оптимизации.

Принимая во внимание естественные ограничения на совместимость компонент базиса, множество алгоритмов планирования Ω , получаемое на его основе, сводится к следующим пяти элементам:

1. ARB-SGL-CFT.
2. MTC-NONE-NONE.
3. ARB-DBL-CFT.
4. MTC-SGL-CFT.
5. MTC-DBL-CFT.

4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ПЛАНИРОВАНИЯ

В работе [3] был предложен структурированный подход к оценке и анализу алгоритмов планирования действий, используемых интеллектуальными агентами с выделением двух уровней показателей: универсальных, применимых ко всем алгоритмам планирования, и специфических, учитывающих характерные особенности конкретного класса алгоритмов. С учетом специфики нелинейного планирования следующее подмножество показателей представляет наибольший интерес с прикладной точки зрения:

1. Функциональность S , определяемая как вероятность нахождения алгоритмом минимального решения данной задачи планирования.
2. Использование вычислительных ресурсов в процессе решения задачи, в частности процессорное время T и оперативной памяти M .
3. Эффективность поиска, выполняемого алгоритмом планирования, выражаемая через общее количество ветвлений дерева поиска V и общее количество рекуррентных вызовов алгоритма R .

Экспериментальный анализ рассматриваемых алгоритмов был проведен на

основе результатов решения 200 задач планирования, полученных с помощью автоматизированной системы (генератора задач), позволяющей порождать произвольное количество пар <предметная область; задача планирования>, обладающих некоторым набором наперед заданных свойств, наиболее важными из которых являются:

1. Существование решения задачи планирования (хотя необходимо отметить, что эффективное определение задач, не имеющих решения, также представляет большой интерес).
2. Высокая репрезентативность многообразия задач планирования, определяемая наличием широкого диапазона значений показателей, характеризующих задачу планирования и среду функционирования агента [3].

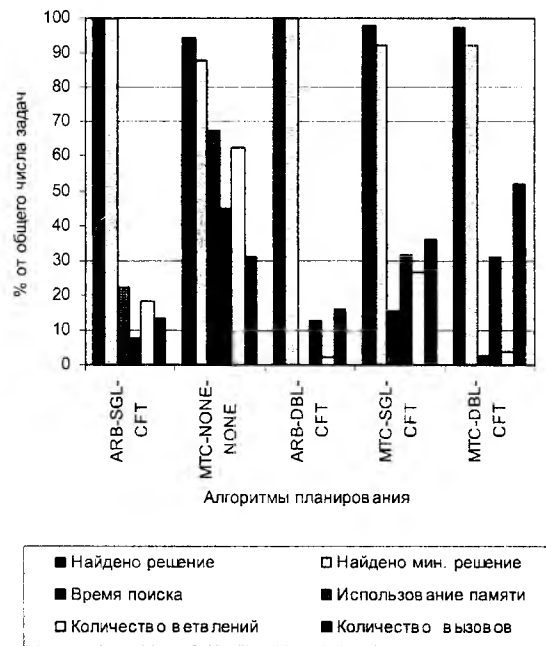


Рисунок. 1. Количество задач, в которых каждый алгоритм продемонстрировал наилучшую производительность относительно отдельно взятого показателя.

Алгоритмы планирования были программно реализованы с использованием режима поиска “в глубину” в соответствии с описанием обобщенного алгоритма, представленном в предыдущем разделе, в виде набора функциональных модулей, включающих в себя:

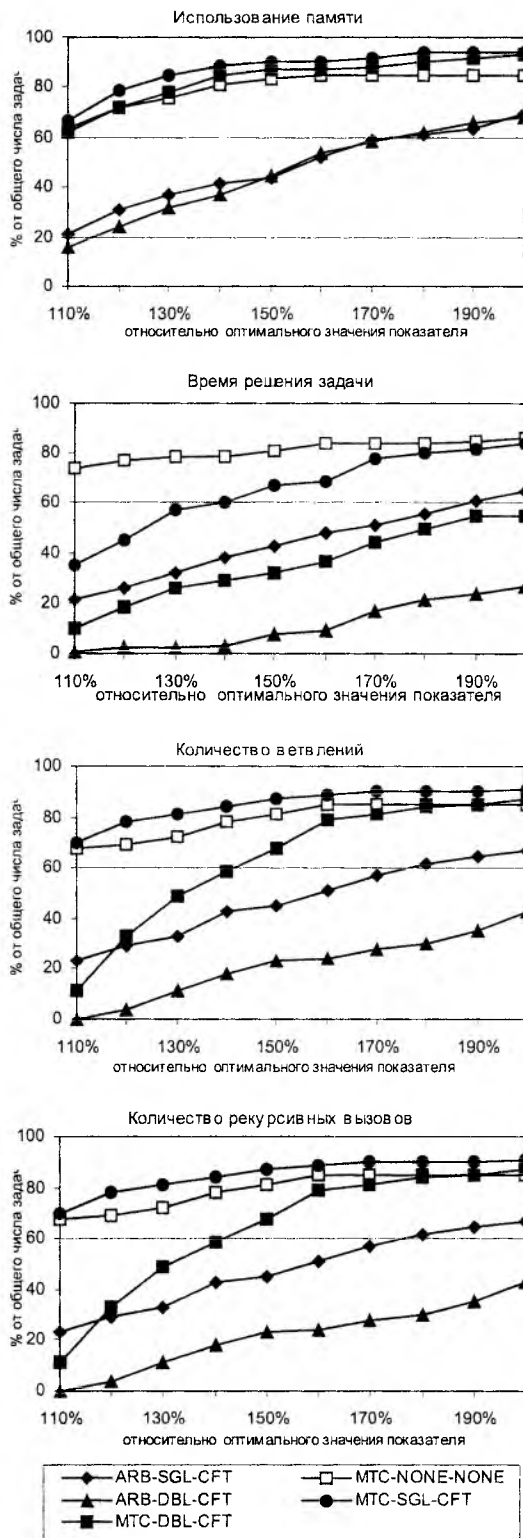


Рисунок. 2. Производительность алгоритмов планирования относительно оптимального значения каждого показателя.

1. Модуль обобщенного алгоритма планирования, независимый от детальной реализации каждого конкретного шага.
2. Отдельный модуль для каждого элемента базиса пространства алгоритмов планирования: выбора подцели/проверки условия окончания, а также процедур консервации и оптимизации.
3. Вспомогательный модуль, включающий сервисные и диагностические процедуры.

Модульная организация обеспечивает многоразовое использование каждой программной компоненты, отсутствие избыточности кода и возможность оперативной генерации требуемого алгоритма планирования.

Сравнительные результаты решения сгенерированных задач выделенными алгоритмами планирования представлены на рисунках 1 и 2. В первом случае для каждого алгоритма показана доля задач, при решении которых данный алгоритм оказался наилучшим относительно каждого из рассматриваемых показателей. Во втором случае для каждого алгоритма показана доля задач, при решении которых значение заданного показателя оказалась в определенных пределах от оптимального значения данного показателя, достигаемого на всем множестве алгоритмов.

На основании полученных результатов могут быть сделаны следующие выводы:

1. Ни один из рассматриваемых алгоритмов не является наилучшим в абсолютном смысле относительно совокупности рассматриваемых показателей.
2. Наблюдается большой разброс между наилучшим и наихудшим значением каждого из выделенных показателей на рассматриваемом множестве алгоритмов при решении практически каждой задачи планирования.
3. Алгоритмы, не использующие МКИ, обладают свойством полноты, то есть всегда находят решение задачи планирования, если оно существует.
4. В подавляющем большинстве задач алгоритмы, использующие МКИ, демонстрируют более высокую производительность, однако в некоторых случаях они не способны найти решение задачи планирования.

Для эффективного определения наиболее оптимального алгоритма планирования в

прикладных задачах может быть применен следующий подход:

1. Для оценки производительности алгоритма планирования $a \in \Omega$ используются относительные значения соответствующих показателей:

$$S'(a) = \frac{\max_{m \in \Omega} S(m) - S(a)}{\max_{m \in \Omega} S(m) - \min_{m \in \Omega} S(m)}$$

$$X'(a) = \frac{X(a)}{\min_{m \in \Omega} X(m)}, \quad X \in \{T, M, B, R\}$$

2. Оптимальность алгоритма определяется с помощью целевого функционала, состоящего из суммы относительных значений представленных показателей, умноженных на весовые коэффициенты, отражающие важность каждого показателя в конкретной задаче планирования:

$$F(I, a) = \alpha S'(I, a) + \beta T'(I, a) + \gamma M'(I, a) + \delta B'(I, a) + \varepsilon R'(I, a), \quad a \in \Omega$$

В качестве примера практического использования предложенного критерия оптимальности можно рассмотреть три наиболее распространенных типа задач планирования:

1. Автономное планирование: построение плана происходит независимо от текущего функционирования агента, в силу чего основными требованиями являются высокая вероятность нахождения решения при умеренном быстродействии и несущественных емкостных ограничениях.
2. Оперативное планирование: время построения плана ограничено из-за функциональных особенностей агента и постоянных изменений в среде или условиях задачи. Поскольку по истечении определенного промежутка времени текущая задача планирования перестает быть актуальной, основным показателем является скорость нахождения плана.
3. Распределенное планирование: имеется возможность параллельного поиска по нескольким ветвям дерева решений (многопроцессорный агент или мультиагентная система), поэтому основными показателями являются полнота

поиска и эффективность каждого вызова алгоритма.

Таблица 2. Пример выбора весовых коэффициентов в зависимости от типа задачи.

Тип задачи	α	β	γ	δ	ε
Автономное планирование	60	20	10	5	5
Оперативное планирование	20	50	20	5	5
Распределенное планирование	30	15	5	20	30

На рис. 3 изображена доля задач, в которых каждый из рассматриваемых алгоритмов оказался наилучшим для соответствующего типа задачи планирования в смысле предложенного критерия оптимальности с использованием весовых коэффициентов представленных в таблице 1.

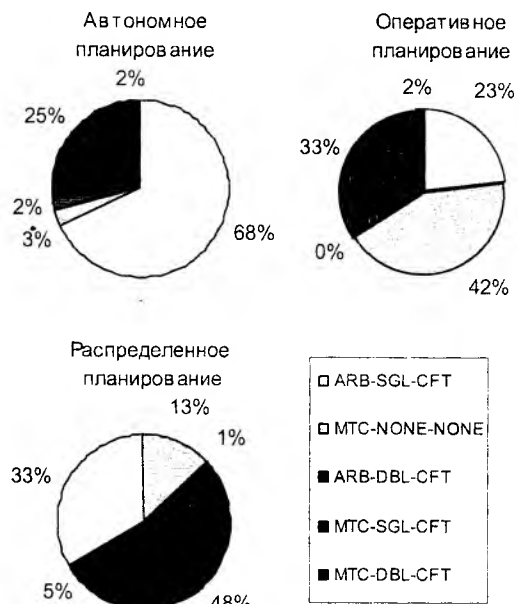


Рисунок 3. Оценка оптимальности алгоритма в зависимости от типа задачи планирования.

Принимая во внимание определенную степень свободы на этапе выбора весовых коэффициентов, можно сделать вывод о том, что алгоритмы ARB-SGL-CFT (автономное планирование), MTC-NONE-NONE (оперативное планирование) и MTC-SGL-CFT (оперативное и распределенное планирование) демонстрируют наилучшую суммарную производительность при решении

подавляющего большинства задач соответствующих типов.

ЗАКЛЮЧЕНИЕ

В рамках данной работы были получены следующие результаты, предоставляющие необходимый инструментарий для практического использования сравнительного анализа производительности алгоритмов планирования действий интеллектуальных агентов:

1. Выделен набор наиболее важных с прикладной точки зрения показателей производительности алгоритмов нелинейного планирования.
2. Получены экспериментальные результаты, свидетельствующие об отсутствии наилучшего в глобальном смысле алгоритма планирования.
3. Предложен критерий оптимальности алгоритма планирования и продемонстрирована его практическая применимость.

В заключение необходимо отметить, что полученные экспериментальные результаты также позволяют сделать вывод о необходимости рассмотрения задачи априорного определения наилучшего алгоритма решения для конкретной задачи планирования на основании следующих данных:

1. Характеристик среды функционирования агента.
2. Условий данной задачи планирования.
3. Относительной важности показателей производительности алгоритма, определяемой типом задачи планирования.

ЛИТЕРАТУРА

- [1]. Искусственный интеллект. Кн. 2. Модели и методы // Под ред. Д.А.Поспелова. М: Радио и связь, 1990.
- [2]. Добрецов С.В., Шестаков С.М. Планирование действий в искусственном интеллекте // ВАНТ СПбГТУ, №2 1998.
- [3]. Добрецов С. В. Анализ критериев оптимальности алгоритмов планирования действий интеллектуальных агентов // МНТК «Научно-технологические и интеллектуальные системы 2001», Москва, МГТУ им. Н. Э. Баумана, 21-22 марта 2001 г.
- [4]. Tate A. Generating Project Networks // Proceedings of IJCAI-77, Boston, MA, 1977. pp. 888-893.
- [5]. Chapman D. Planning for conjunctive goals // Artificial intelligence, Vol. 32, 1987. pp 333-337.
- [6]. Pednault E. P. D. ADL: Exploring the middle ground between STRIPS and the Situation Calculus // Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning. Toronto, Canada, 1989
- [7]. McAllister D, Rosenblitt D. Systematic Nonlinear Planning // Proceedings of 9th AAAI, 1991.
- [8]. Safra S., Tennenholtz M. On Planning while Learning // JAIR 9/1994
- [9]. Kambhampati S., Knoblock C., Yang Q. Planning as Refinement Search: A Unified framework for evaluating design tradeoffs in partial order planning // Artificial Intelligence. Special issue on Planning and Scheduling. Vol. 76. No. 1-2, September 1995. pp. 167-238.
- [10]. Weld D. Recent Advances in AI Planning // AI Magazine, 1999