

Архитектуры и модели ИА реального времени (Конспект)

Ключевое требование к ИА, целенаправленно действующим в открытых динамических неопределенных мирах, является обеспечение режима реального времени (РВ). При этом трактовка режима РВ для данного класса систем существенно отличается от традиционной для информационных систем, поскольку ИА РВ представляют собой системы «ограниченной рациональности». Такие системы в общем случае не могут оптимально решать все стоящие перед ними задачи, а должны динамически определять имеющийся у них запас времени и максимально эффективно использовать его для выработки решений, направленных на минимизацию потерь и/или максимизацию полезности действий агента в соответствии с целью его функционирования.

Различным аспектам обеспечения РВ в системах ИИ посвящено достаточно много исследований. В системах ИИ, основанных на методах поиска в пространстве состояний, для обеспечения предсказуемости времени поиска используются следующие подходы:

- *сокращение* – использование предметных знаний для исключения частей пространства поиска, про которые известно, что они не содержат целевого состояния;
- *упорядочение* – рассмотрение в первую очередь сущностей (состояний или подзадач), которые с большей вероятностью ведут к решению;
- *аппроксимация* – сокращение точности рассматриваемого параметра;
- *отсечение* – управление во времени и пространстве максимальным просмотром вглубь при выборе следующего оператора.

В настоящее время в ИИ разработан ряд моделей и методов, которые могут быть использованы при построении ИА РВ:

- алгоритмы произвольного времени и их композиция;
- планирование процесса обдумывания;
- гибкие рассуждения на основе теории принятия решений;
- А*-поиск в реальном времени;
- множественные методы и приближенная обработка;
- планирование своевременных вычислений;
- ограниченные по времени рассуждения в условиях неопределенности.

Под *алгоритмами произвольного времени* (АПВ) понимаются алгоритмы, у которых качество формируемого решения зависит от выделяемого алгоритму времени. Предполагается, что качество результата может повышаться с увеличением времени выполнения до некоторого максимального уровня. Термин «*алгоритм произвольного времени*» (anytime algorithm) был введен в работе [**], однако следует отметить, что итеративные алгоритмы со свойствами АПВ изучались во многих областях таких, как численная аппроксимация, эвристический поиск, динамическое программирование, алгоритмы метода Монте-Карло, обработка запросов к БД и др.

Каждому АПВ ставится в соответствие *профиль производительности* (performance profile) – функция, отображающая выделенное алгоритму время (в некоторых случаях также качество входных данных) в уровень качества вырабатываемого алгоритмом решения. Такие алгоритмы могут быстро откликаться на изменения внешней ситуации и позволяют использовать гибкие механизмы управления ментальной активностью ИА для инкрементного повышения качества результата при выделении дополнительного времени. Эти особенности делают АПВ исключительно полезными в системах реального времени.

Неточные вычисления – подход к решению задач РВ, при котором задачи имеют обязательную и необязательную части. Обязательную часть составляют непрерываемые вычисления, которые должны быть выполнены полностью, чтобы гарантировать корректную работу системы. Необязательная часть представляет собой итеративно улучшаемый алгоритм, который повышает качество результата, генерируемого обязательной частью. На практике любой алгоритм всегда имеет обязательную часть, так как даже результат самого низкого качества требует определенных затрат времени. Однако во многих случаях он может быть вычислен таблично, т. е. с пренебрежимо малой задержкой.

В [**] рассмотрен подход, в соответствии с которым вычисления строятся как комбинации алгоритмов произвольного времени, решающих частные подзадачи. При этом результаты одних АПВ являются входами других, а профили их выполнения зависят от качества входных данных. Предполагается, что зависимости между задачами точно заданы. Необходимо распределить время между отдельными АПВ таким образом, чтобы максимизировать ожидаемую общую полезность вычислений. Zilberstein и

Russell называли составленные таким образом программы *контрактными* АПВ (contract anytime algorithms).

В общем случае контрактный АПВ возвращает лучший результат, если ему выделяется больше времени, причем он должен заранее объявить, сколько времени он рассчитывает получить и что он может не вернуть никакого ответа, если получит времени меньше запрошенного. Для более адекватного представления неопределенности в такой информации понятие профиля *производительности* алгоритма расширяется за счет введения трех типов профилей:

- *профиля распределения производительности* (ПРП) алгоритма – функции, отображающей время вычислений в распределение вероятностей качества результата;
- *профиля ожидаемой производительности* (ПОП) алгоритма – функции, отображающей время вычислений в ожидаемое качество результата;
- *интервального профиля производительности* (ИПП) алгоритма – функции, отображающей время вычислений в верхнюю и нижнюю границы качества результата.

Dean и Boddy предложили подход, названный *планированием процесса обдумывания* (deliberation scheduling). Планирование процесса обдумывания предполагает назначение задачам времени с целью максимизации общей полезности выполняемых агентом вычислений. Этот подход был развит применительно к задачам планирования действий агента в зависимости от наличия времени. Предполагается, что для решения задач используются АПВ, при этом время, доступное для реакции на события, варьируется от ситуации к ситуации. Согласно данному подходу, задача зависящего от времени планирования состоит из множества типов событий, множества типов действий, решающих процедур (представленных АПВ), определяющих действия для каждого типа событий, и функции полезности. Выделение большего времени процедуре, выбирающей действие для выполнения, будет приводить к выбору действий с более высокой полезностью (вплоть до некоторого максимального значения). Предполагается, что агент имеет знания о будущих событиях, включая типы этих событий и время их ожидаемого наступления.

Агент строит план вычислений, назначающий некоторое количество времени решающим процедурам. После наступления критического срока время решающим процедурам не назначается. Цель состоит в максимизации

суммарной полезности откликов на события. Предполагается, что пустой отклик имеет нулевую ценность (в противоположность наказанию за нарушение критического срока). При этих допущениях Dean и Boddy рассмотрели два алгоритма нахождения оптимальных планов, отличающиеся в первую очередь допущениями о профилях.

Первый алгоритм (названный DS-1) предполагает, что в каждый момент времени доступно возможное приращение профиля выполнения. Алгоритм строит план в обратном направлении от более поздних критических сроков к текущему моменту времени в фиксированных приращениях размера δ . Он просто назначает каждое приращение исполнимой задаче с большим увеличением выигрыша (дохода). После назначения каждой δ планировщик идет назад и выделяет каждой задаче непрерывный блок.

Второй алгоритм (названный DS-2) предполагает, что профили выполнения могут быть описаны в терминах некоторых известных функций. Этот алгоритм также выполняет планирование в обратном направлении от последнего критического срока к текущему моменту времени; в этом случае приращение представляет собой разность до предшествующего критического срока. В рамках каждого приращения время назначается каждой задаче в соответствии с решением множества линейных уравнений. Точная структура уравнений зависит от вида функции, описывающей профиль выполнения.

Таким образом, в этих работах за счет упрощающих допущений о независимости задач авторам удалось найти планирующий алгоритм, имеющий невысокую вычислительную сложность и предсказуемую производительность. Однако они не включили в свой анализ фактор стоимости планирования, хотя любое реальное приложение должно учитывать время на выполнение планирования.

Гибкие рассуждения на основе теории принятия решений.

Гибкие рассуждения на основе теории принятия решений основаны на использовании АПВ в различных задачах теории принятия решений. В теории принятия решений для определения наилучшего решения используются вероятностная информация о возможных результатах действий и знания о полезности этих результатов. Лучшим считается решение, обеспечивающее самое высокое значение полезности с максимальной вероятностью. Э. Хорвиц рассмотрел три основных аспекта этой проблемы:

- определение, как назначать значения полезности неполным частичным результатам, возвращаемым АПВ;

- нахождение АПВ для реализации рассуждений на основе теории решений о том, какие действия выполнять следующими;
- нахождение оптимального баланса между временем, затрачиваемым на выполнение метарассуждений, и временем, затрачиваемым на решение собственно задач.

А*-поиск в реальном времени

А-поиск в реальном времени* представляет собой алгоритм поиска в пространстве состояний с использованием подхода контрактного произвольного времени. Основная идея состоит в чередовании движения вглубь в направлении, которое в данный момент представляется лучшим, и улучшения текущего знания относительно лучшего направления движения. Текущее знание улучшается с использованием простого алгоритма, выполняющего поиск на фиксированную глубину. Глубина выбирается в зависимости от количества времени, допустимого для каждого движения. Это время, в свою очередь, определяется на основе эвристической оценки числа шагов, требуемых для достижения целевого состояния, и общего количества выделенного времени. Такой подход позволяет точно контролировать общее время поиска и время поиска на один просмотр вперед. Однако качество результата существенно зависит от точности эвристик, используемых для оценки расстояния до целевого состояния, и способности поиска ограниченной глубины рекомендовать шаги в направлении целевого состояния.

Множественные методы и приближенная обработка

Множественные методы и приближенная обработка вместо одного АПВ используют множество методов, обеспечивающих размен времени на качество. Методы могут иметь различные характеристики производительности в различных внешних ситуациях. Данный подход, назван *приближенной обработкой* (approximate processing). На рис. 3.9 приведен пример использования множества методов в задаче интерпретации сенсорных данных.

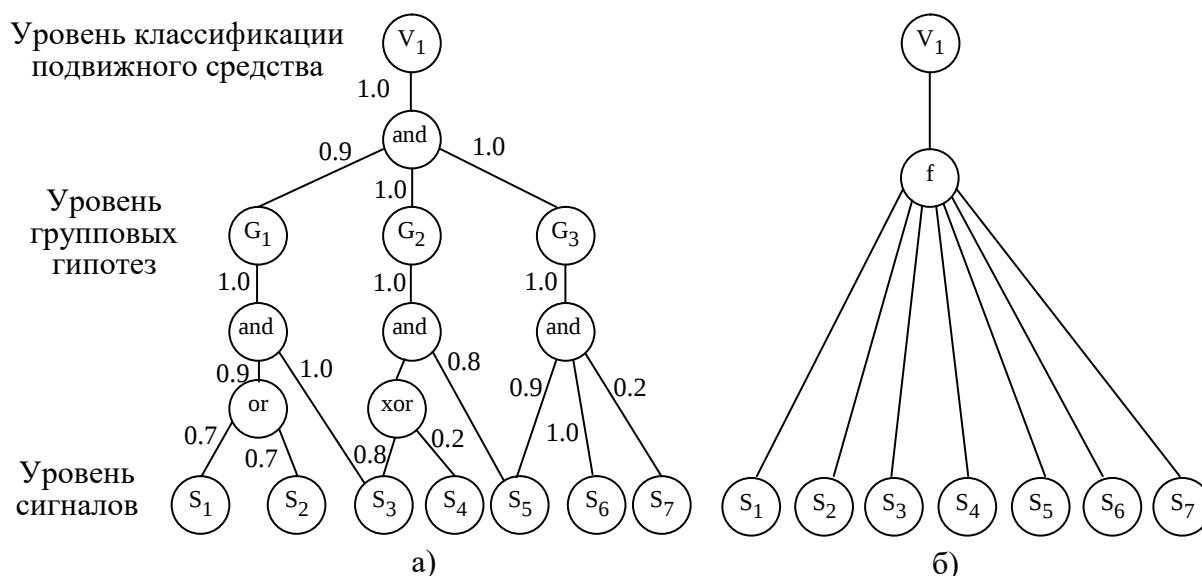


Рис. 3.9. Использование множественных методов в задаче идентификации объекта на основе сенсорных данных: а) полная грамматика; б) приближенная грамматика

Представлены две грамматики, описывающие характеристики гипотез, необходимые для идентификации конкретного типа подвижного средства. В случае а) сначала на основе низкоуровневых сенсорных данных строится ряд промежуточных гипотез $G_1 - G_3$, которые затем комбинируются для формирования гипотезы относительно движущегося средства. В случае б) промежуточный уровень исключен и гипотезы о классе подвижного средства строятся непосредственно на основе низкоуровневых сенсорных данных. Последний метод строит гипотезы быстрее, однако, с более низким качеством (точностью и определенностью) решения.

Подход на основе множественных методов имеет, по меньшей мере, два потенциальных преимущества по сравнению с АПВ:

1. Не требует существования итеративно улучшаемых алгоритмов. Ясно, что такие алгоритмы существуют для некоторых классов проблем, но существуют и задачи, которые будет трудно или невозможно решить с использованием АПВ. Трудность состоит в нахождении алгоритмов, ожидаемое качество которых улучшается предсказуемо и монотонно.

2. Различные методы могут не только обеспечивать размен времени решения на качество, но и реализовывать совершенно различные подходы к решению задачи. Эти подходы могут иметь значительно отличающиеся характеристики в зависимости от конкретной внешней обстановки. Таким образом, размен качества на время, реализуемый данным подходом, может

существенно отличаться в различных ситуациях. В большинстве работ по АПВ, напротив, принимается допущение, что доступен один АПВ, который, как ожидается, должен эффективно работать во всех возможных внешних ситуациях.

Приближенная обработка – подход к решению задач реального времени в ситуациях, где приемлемы удовлетворительные (а не только оптимальные) ответы и возможно использование приближенных данных, знаний и управляющих механизмов в различных комбинациях. Для успешного применения данного подхода необходимо иметь полезные аппроксимации с предсказуемым эффектом для рассматриваемого приложения, архитектуру решения задачи, которая позволяет представить эти приближения соответствующим образом и управляющие механизмы для принятия решений о том, какие приближения использовать. Важным аспектом приближенной обработки является наличие представлений, обеспечивающих их совместное использование, т. е. допускающих их смешивание и сопоставление (недопустимо наличие у каждого приближения своего представления, несовместимого с другими).

В первой работе по приближенной обработке [**] рассматривались систематические подходы к генерации приближений для существующих приложений. В этой работе рассмотрено несколько приближений для использования в задачах интерпретации сложных сигналов. Эти приближения классифицированы на *приближенные стратегии поиска*, *приближенные данные* и *приближенные знания*. Полезность этих аппроксимаций была доказана при создании распределенной среды мониторинга подвижных средств DVMT.

Планирование своевременных вычислений

Планирование своевременных вычислений (design-to-time real-time scheduling) является обобщением подхода приближенной обработки за счет использования множества методов для множества задач. Проблема состоит в построении плана решения задачи, использующего все доступные ресурсы для максимизации качества решения в рамках имеющегося времени [**]. Используется модель вычислительных задач TAEMS, в которой сложные задачи моделируются группой независимых задач, включающих, возможно, зависимые задачи. Отношение задача/подзадача внутри группы задач представляется ориентированным ациклическим графом и используется для вычисления качества задачи (т. е. качество задачи есть функция от качества

подзадач). Листьям этого графа соответствуют исполняемые методы, представляющие действительные вычисления, которые могут быть выполнены системой. Помимо отношений задача-подзадача, задачи могут иметь и другие взаимозависимости с задачами своей группы (например, выполнение одного метода разрешает выполнение другого или использование грубого приближения одним методом отрицательно влияет на эффективность метода, который использует его результаты). Влияние одних методов на качество и длительность других может иметь количественное выражение. Это аналогично влиянию качества входных данных на профиль эффективности в рассмотренном выше подходе к композиции АПВ.

В [**] рассматриваются задачи динамической классификации, в которых динамика изменений внешней среды сопоставима со временем решения задачи. Это означает, что некоторые данные, на основе которых решается

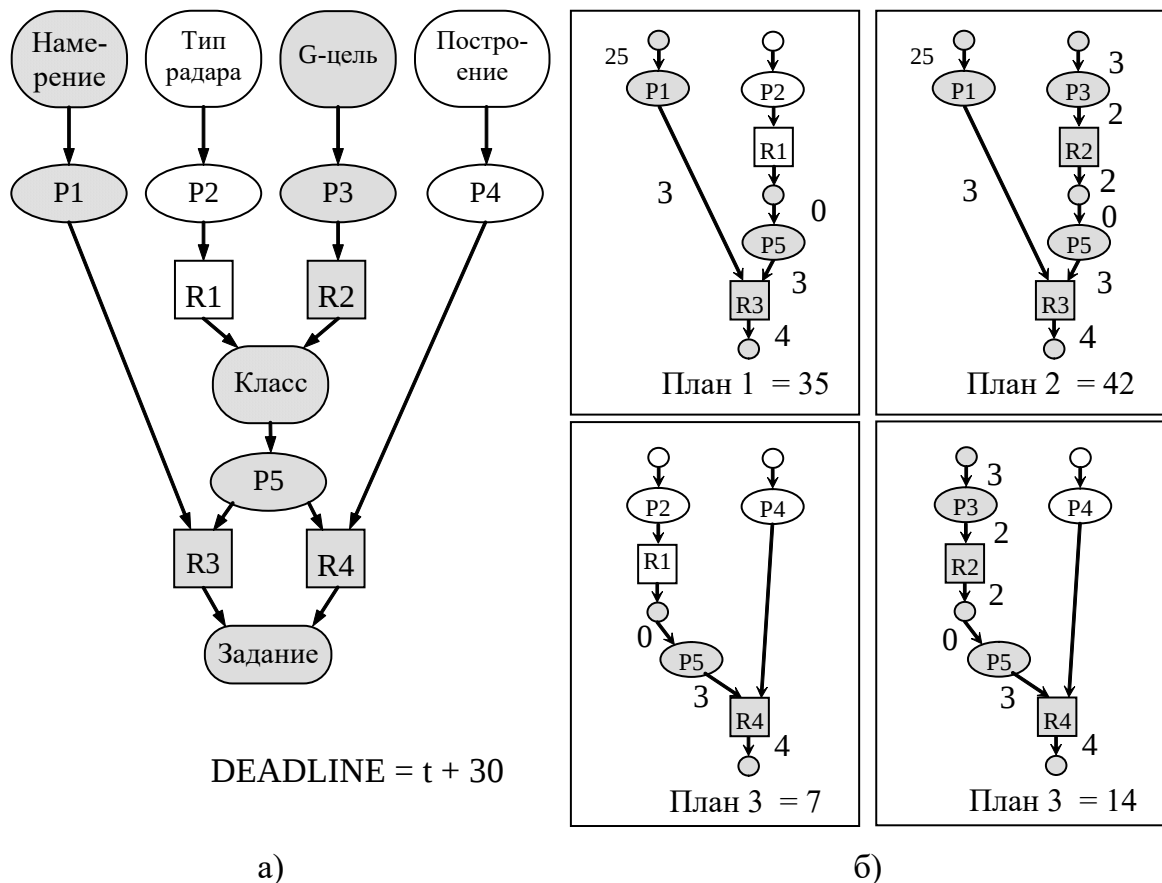


Рис. 3.10. Пример множества планов: а) полное множество планов; б) четыре индивидуальных плана с различными временами выполнения и ожидаемой полезностью

задача, могут измениться за время ее решения. Предложенный подход включает генерацию множества планов на этапе составления путем

предварительной оценки полезности планов и кэширования этой информации. На рис. 3.10 представлен пример множества планов.

На этапе исполнения, исходя из имеющегося для ответа на запрос времени, выбираются некоторые или все планы, способные обработать текущий запрос. Эти планы можно рассматривать как множество методов для достижения конкретной цели. Управляющий механизм выполняет планы в порядке увеличения ожидаемой полезности и может изменять порядок, основываясь на действительных значениях полезности, вырабатываемых более ранними планами. Предполагается, что запросы с более высоким приоритетом могут прерывать выполнение текущего запроса и это учитывается при выборе порядка выполнения планов.

Анализ известных методов построения интеллектуальных систем реального времени показывает, что к настоящему времени предложен ряд моделей и методов построения таких систем, ориентированных на обеспечение требований реального времени в частных задачах и компонентах систем ИИ, при игнорировании других задач и компонентов. Вместе с тем при построении интеллектуальных агентов задача обеспечения реального времени должна решаться комплексно с учетом взаимодействия всех уровней и подсистем агента.