

УПРАВЛЕНИЕ В СТАТИЧЕСКИХ РОЯХ. ПОСТАНОВКА ЗАДАЧИ

*Карпов В.Э., к.т.н., доцент
Московский институт электроники и математики НИУ ВШЭ
e-mail: vkarпов@hse.ru*

// «Интегрированные модели и мягкие вычисления в искусственном интеллекте». Сб. научных трудов VII-й Международной научно-практической конференции (Коломна, 20-22 мая 2013). В 3-х томах. Т.2. -М.: Физматлит, 2013, с.730-739

1. ВВЕДЕНИЕ

Одним из наиболее активно развивающихся направлений современной робототехники является т.н. групповая робототехника, объектами изучения которой являются множества взаимодействующих между собой роботов. Роботы могут образовывать разнообразные совокупности - рои, стаи, коллективы и проч. Отметим здесь, что основное отличие между этими групповыми образованиями заключается в степени информированности о других членах группы, об общих целях, задачах, мотивации и т.д. (см., например, [Каляев и др., 2009]). Если эта информированность ограничивается пониманием состояния лишь ближайших соседей, то мы имеем дело с роем. Если же информированность членов группы высока, то можно говорить об образовании коллектива роботов.

Судя по количеству работ и исследований, сегодня наиболее привлекательным направлением является роевая робототехника. Основным тезисом роевой робототехники является возможность реализации сложных форм поведения на основе множества (обычно – большого) простых технических устройств – роботов. При этом модели поведения в роевой робототехники базируются на том, что между представителями роя существует локальная связь, т.е. обмен информацией осуществляется лишь между ближайшими соседями. Типичными и наиболее распространенными задачами роевого поведения является отработка механизмов согласованного движения роя (см., например, [Brambilla a.o., 2012]). Если не принимать во внимание "экзотические" макромодели модели подобного движения (например, основанные на уравнениях гидродинамики), то наиболее часто используемыми являются модели локального взаимодействия. Например, множество роботов EPORO фирмы Nissan [Nissan, 2009]

успешно реализуют согласованное движение, основываясь на использовании "локальных" правил поведения членов стаи. В зависимости от окружения (степени близости соседей) роботы используют соответствующие простые правила вида "избегать столкновения", "удерживать дистанцию и направление движения", "стараться держаться вместе" и т.п.

Обычно локальное взаимодействие в роевых системах понимается как нечто само собой разумеющееся. Роботы каким-то образом обмениваются информацией с соседями; роботы наблюдают за поведением соседей; роботы распознают состояние своих соседей и проч. При этом, повторим, сложность решаемой роем задачи определяется прежде всего тем, как ведет себя рой в целом, причем в основном на уровне общего согласованного движения роя.

В настоящей работе предлагается, напротив, обратиться к проблеме информационного взаимодействия между членами роя.

2. ПОСТАНОВКА ЗАДАЧИ

Итак, сформулируем задачу следующим образом. Пусть имеется множество простых устройств – роботов. При этом роботы способны к непосредственному локальному взаимодействию между соседями. Вопрос заключается в том, можно ли сформулировать условия, при которых такая система будет способна решать действительно более сложные задачи как поведенческого уровня, так и уровня обработки информации, принятия решений и т.п. Иными словами, следует определить условия появления синергетических эффектов или эмерджентных свойств.

Роевые модели делятся на два основных класса – модели микро (microscopic)- и макро (macroscopic)- уровня. Первые основаны на использовании моделей поведения самих особей, вторые – на описании роя как единого целого. Например, на микро-уровне широко используются конечно-автоматные модели, а на макро-уровне – модели гидродинамики. Реже применяются гибридные модели, пытающиеся совместить оба этих подхода. Например, в [Berman a.o., 2007] описывается модель динамики состояния окружающей среды, которая определяет поведение членов роя (агентов). Приведенные в настоящей работе рассуждения большей частью относятся к микро-уровню, т.к. нас интересует прежде всего именно механизм локальной организации внутрироевого взаимодействия (вполне в духе локального подхода [Стефанюк, 2004]).

Будем полагать, что нашим основным объектом является множество неких "простых" устройств. Под простотой будем понимать

некую принципиальную ограниченность емкостных и вычислительных ресурсов.

Далее введем некоторое серьезное ограничение на коммуникационные возможности роботов. Будем считать, что каждый робот может общаться не более чем с некоторым количеством своих ближайших соседей. Для этого предположим, что роботы имеют фиксированное число коммуникационных портов – точек контакта, позволяющим роботам устанавливать каналы обмена информацией. Например, для организации связи роботы должны физически подключиться к коммуникационным портам друг друга так, как показано на Рис. 1.

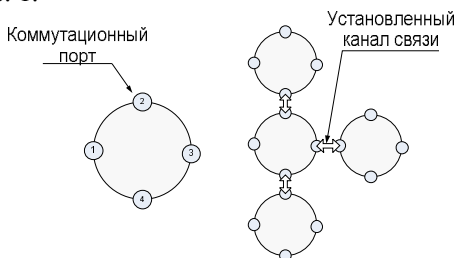


Рис. 1. Роботы, оснащенные четырьмя коммуникационными портами, и образование каналов связи

Вторым допущением (или ограничением) будем считать статический характер получаемых образований. Т.е. далее мы будем рассматривать статические структуры, представляющие собой полученные в некий момент времени схемы соединений роботов. Такую совокупность соединенных между собой роботов можно назвать статическим роём.

В каком-то смысле этот статический рой может рассматриваться как некая сеть. При этом надо помнить, что, в отличие от вычислительной сети, наша сеть должна быть способна к восприятию сигналов внешней среды, а также к совершению некоторых (например, двигательных) функций, т.е. оказывать влияние на внешний мир.

Итак, попробуем определить предварительный перечень вопросов, которые потребуются разрешить для того, чтобы получившаяся система смогла бы решать качественно более сложные задачи, т.е. быть способной к проявлению эмерджентных свойств. А именно, нас интересуют:

1. структура особи;
2. хранение данных (организация базы данных - БД);
3. формирование функциональной дифференциации узлов;

4. организация запросов к БД;
5. разрешение конфликтных ситуаций;
6. выработка решения (например, на основе логических выводов);
7. распределение вычислительных задач между узлами;
8. отработка согласованного действия.

3. ВОЗМОЖНЫЕ ПОДХОДЫ К РЕШЕНИЮ

Далее мы обсудим эти задачи более подробно и при этом попробуем предложить некоторые варианты их решений.

Структура особи. Обобщенная архитектура робота приведена на Рис.2. Она достаточно очевидна за исключением того, что память робота должна хранить как текущие факты (оперативная БД), так и содержать некие сведения, определяющие общие правила поведения робота – некий перечень мотиваций, целей управления, потребностей и проч. (постоянная БД).



Рис. 2. Архитектура особи

Организация БД. Все узлы сети хранят отдельные фрагменты общей базы данных – накопленный опыт, наблюдаемые факты и т.п. Очевидно, что такое фрагментарное хранение отлично от того, что принято в распределенных базах данных. Если в распределенных БД должна существовать некая структура, описывающая, в каких узлах сети хранятся те или иные фрагменты таблиц, то в рое такой карты нет. Это означает, что запросы на получение тех или иных данных должны распространяться по всей сети, причем получение запрашиваемой информации может сопровождаться неопределенными задержками, повторами и конфликтами.

Будем полагать, что информация в БД (или базе фактов) хранится в виде множества троек вида:

(<имя факта> <значение> <атрибуты>)

Именем факта может быть название параметра, измеряемого соответствующим датчиком, а атрибут будет описывать состояние узла, содержать временную метку и проч.

Функциональная дифференциация узлов. Из ограничений вычислительных возможностей узлов сети неизбежно следует необходимость организации иерархических структур.

При такой организации в каком-то смысле рой неизбежно превращается в коллектив. Речь идет о том, что все узлы должны обладать некоторой общей мотивацией (или перечнем потребностей), а также придерживаться ряда общих для всей группы узлов правил поведения. Иначе узлы просто не смогут договориться друг с другом. Дифференциация функций – сугубо динамическая, т.е. образуется исходя из текущей топологии группы.

Предположим, что особи-узлы могут в зависимости от текущей топологии образовывать одну из трех разновидностей, т.е. выполнять следующие функции: управляющие узлы, вычислители и сенсоры. Правила такой дифференциации могут выглядеть так. Узлы со свободными коммуникационными портами становятся сенсорными узлами (СУ) – их свободные каналы будут воспринимать информацию от датчиков (узлы С на Рис.3.). Соседи сенсорных узлов получают роль вычислителей (вычислительные узлы – ВУ, узлы В на Рис.3). Узлы с максимальной связностью, не соседствующие ни с ВУ, ни с СУ, становятся управляющими узлами (УУ, узлы А на Рис.3). Это, конечно, весьма условная, сугубо качественная схема.

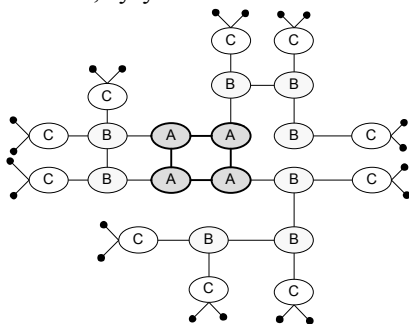


Рис. 3. Дифференциация узлов: С – сенсорные узлы, В – вычислительные узлы, А – управляющие узлы

Функции узлов также достаточно естественны: СУ занимаются сбором оперативной информации о состоянии окружающей среды, ВУ занимаются хранением управляющей информации и вычислением запросов, поступающих от УУ.

Организация запросов к БД. Элементарный запрос – это запрос на получение некоторого факта по его имени и, в общем случае, атрибуту. Таковым фактом может быть, например, значение некоторого наблюдаемого параметра (скажем, показание некоторого датчика). Здесь не хотелось бы останавливаться на технической стороне организации подобного рода запросов и ответов. Отметим лишь, что для роя больший интерес представляют сами механизмы доставки сообщений.

Существуют два пути решения задачи организации межузлового взаимодействия. Во-первых, для этого могут быть применимы хорошо известные сетевые протоколы или их аналоги. При таком подходе в сети будут распространяться пакеты следующего приблизительного формата:

<заголовок+адресная часть><маркер><имя параметра><значение>

Это "стандартное" решение сопряжено с рядом трудностей, среди которых основными являются необходимость синхронизации работы узлов, а также излишне большой трафик в сети с принципиально малой пропускной способностью.

Иной путь – это отправка безадресного запроса на получение параметра. При этом к пакету прикрепляется поле счетчика.

<маркер><имя параметра><значение><счетчик>

Значение счетчика увеличивается при прохождении очередного узла. Если значение счетчика достигает некоторого максимального значения, то пакет уничтожается. В противном случае узел ретранслирует пакет всем своим соседям. Такой счетчик играет роль фермента-теломеразы, отвечающего за старение организма.

Вычислительные эксперименты показывают, что подобная схема позволяет в асинхронном режиме гарантированно получать результаты запроса с детерминированной максимальной задержкой.

Разрешение конфликтных ситуаций. В статическом рое возможны два вида конфликтов: конфликты при распределении ролей и конфликты передачи данных. Конфликты распределения ролей возникают в ситуации, когда начинают образовываться несколько альтернативных управляющих центров. Или в том случае, если по формальным признакам нет оснований для формирования УУ и их роль должны брать на себя вычислительные или даже сенсорные узлы. Как будет вести себя рой в условиях "двоевластия" или, наоборот, "безвластия" – тема отдельного исследования. Конфликты, возникающие при передаче данных, в каком-то смысле разрешить легче. Например, можно ориентировать систему на работу в условиях неполноты информации. Иными словами, это означает, что на эти

коллизии мы просто не будем обращать внимание, воспринимая их как должное.

Выработка решения. Будем полагать, что все управление основано на множестве правил – импликаций вида

$$c_1 \& c_2 \& \dots c_n \rightarrow f$$

Пусть некий узел (УУ или ВУ) пытается отработать такое правило. Назовем такой узел иницирующим. Иницирующий узел будет поочередно отправлять в сеть запросы на определение истинности конъюнктов этого правила. Прочие узлы сети будут искать в своих БД интересующие инициатора данные, которые могут быть либо фактами, либо правыми частями неких хранящихся в узлах правил.

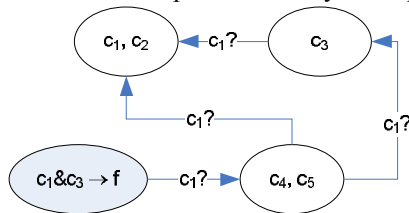


Рис. 4. Доказательство утверждения $c_1 \& c_3 \rightarrow f$ узлом-инициатором

Очевидно, что нецелесообразно реализовывать подобную схему в духе ПРОЛОГа, т.е. заставлять все узлы доказывать пришедшие к ним запросы. Будучи, конечно, универсальной, эта схема вывода приведет к тому, что все ресурсы сети (и главное – медленных коммуникационных каналов) будут заняты исключительно логическим выводом. Не спасет от вычислительного коллапса и использование STRIPS-подобных систем (см., например, [Нильсон, 1985] или [Братко, 2001]), в которых общий логический вывод частично заменен на более простую схему работы с данными.

Пожалуй, единственной возможностью избавиться от проблем логического вывода является переход к линейному выводу на основе системы продукций. Со всеми вытекающими достоинствами (простота) и недостатками (слабость контроля). Это означает, что конъюнктами должны являться имена фактов.

Интересно, что при такой организации основной недостаток продукционных схем – низкая производительность – становится неактуальным.

Открытым, однако, остается следующий вопрос: как поступить системе, если запрошенный конъюнкт не найден? Или если инициатор

запроса получил множество различных ответов на свой запрос. Это и есть упоминавшийся выше вывод в условиях неполноты. Очевидно, что такие радикальные варианты, как установка значения неопределенного конъюнкта в 0 или 1 вряд ли пригодны. Скорее всего здесь целесообразнее использовать понятие нечеткой импликации или вводить коэффициенты уверенности.

Распределение вычислительных задач между узлами.

Декомпозиция задачи в иерархической системе может быть осуществлена достаточно естественным способом. Рассмотрим следующий показательный пример. Предположим, что мы решаем такую сложную задачу, как управление некоторой "игрушечной" или модельной армией. Пусть все военнослужащие (будем дальше называть их агентами) одинаковы по своим вычислительным возможностям. Каждый из них воспринимает фиксированное число входных сигналов, обрабатывает их и реализует некоторое управляющее воздействие. Под управляющим воздействием может пониматься как эффекторная функция, так и отправка запроса или выдача ответа. При этом входные и выходные каналы агента связываются либо с другими агентами, либо с рецепторами. Предположим далее, что эта система образует строгую иерархию. Один из агентов назначается главнокомандующим. Его входные сигналы определяются тем, что поставляется ему непосредственными подчиненными. Им же он отдает приказы. Именно непосредственные подчиненные образуют среду обитания главнокомандующего. В самом низу иерархии находятся рядовые агенты. Их входные сигналы – это приказы непосредственных командиров и сигналы от рецепторов. Подчеркнем еще раз, что в этой иерархии все агенты одинаковы структурно, все они решают свои простые, одинаковые по сложности задачи (например, оптимизационные), но в целом система обладает явными эмерджентными свойствами, решая действительно сложные управленческие задачи.

Разумеется, рассмотренный пример относится к категории редко встречающихся задач - однородных, декомпозируемых на одинаковые по своей подструктуре подзадачи. В противном случае без специализации агентов обойтись сложно. Например, у рыжих лесных муравьев ([Длусский, 1967]) наблюдается явно выраженная профессиональная дифференциация (кастовость), зависящая от физических и даже психологических особенностей особей. Хотя, что очень интересно, каждый муравей универсален, т.е. при определенных условиях он может поменять свою "профессию".

Итак, если с иерархическими структурами распределение задач является более или менее проработанным вопросом, то открытым остается другой: каким образом заставить несколько узлов-агентов осуществлять согласованные вычисления без образования иерархий.

Отработка согласованного действия. Рой, приняв решение, должен вести себя как единое целое. Речь в первую очередь идет о движении роя. Уже говорилось о том, что согласованное движение – это одна из наиболее распространенных задач роевой робототехники. В ряде работ это согласованное движение и вовсе считается самым важным механизмом, к которому можно свести многие задачи роевого поведения ([Garnier a.o., 2005]).

Если структура "обычного" роя допускает неизбежные локальные вариации, то в статическом рое это уже неприемлемо. Связи должны быть постоянными. Или долговременными хотя бы для того, чтобы было время на переконфигурацию. Таким образом, крайне важно, чтобы структура роя была бы единым целым. А это уже сложный технический вопрос. Не говоря уже о том, что необходим эффективный механизм синхронизации движений членов роя.

4. ЗАКЛЮЧЕНИЕ

Итак, в данной работе был сформулирован ряд задач, которые старательно избегаются исследователями в области роевой робототехники. Задачи эти сложные, во многом претендующие на большую общность, однако без решения их вряд ли удастся достичь основной цели, декларируемой роевой (групповой, коллективной) робототехникой – создания действительно сложных по своему поведению систем, состоящих из простых элементов. Здесь самым важным, ключевым моментом является простота особи. Простота особи (ограниченность его когнитивных возможностей) – это наиболее привлекательная черта групповой робототехники. Но это же – самый сомнительный момент в той ситуации, когда мы говорим об ожидаемом сложном поведении всей системы, вплоть до реализации социального поведения. В [Necker a.o., 2012] авторы всерьез рассуждают об аналогах муравьев и их поведении (в основе – генетические алгоритмы), в [Garnier a.o., 2005] говорится о тараканах, а обзор, приведенный в [Brambilla a.o., 2012], всюду пестрит перечнем самых разнообразных насекомых и стайных животных. Дело же в том, что существуют серьезные основания полагать, что сложное социальное поведение (выходящее за рамки согласованного перемещения в пространстве) вряд ли может быть реализовано примитивными особями. Социальное насекомое – это очень сложный

объект, и примитивный робот вряд ли сможет считаться аналогом, скажем, муравья [Карпов, 2011].

Особенностью предложенного подхода (если перечень задач можно считать походом) является именно попытка определения путей решения задачи создания сложной особи из простых составных частей. Дело в том, что статический рой в каком-то смысле может считаться единым организмом. Причем экстенсивно сложным. Что будет, если таких организмов будет много, будут ли они организовывать некий сверх-рой, будут ли работать предложенные методы для такого сверхорганизма – это уже вопросы отдельного исследования.

Литература

- [Братко, 2001] Братко И. Алгоритмы искусственного интеллекта на языке PROLOG, 3-е издание. : Пер. с англ. -М.: Издательский дом "Вильямс", 2001.
- [Длусский, 1967] Длусский Г.М. Муравьи рода Формика –М.: Наука, 1967.
- [Карпов, 2011] Карпов В.Э. Коллективное поведение роботов. Желанное и действительное //Современная мехатроника. Сб. научн. трудов Всероссийской научной школы (г.Орехово-Зуево, 22-23 сентября 2011) - Орехово-Зуево, 2011. – 132 с. сс.35-51.
- [Каляев и др., 2009] Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного поведения в группах роботов. –М.: Физматлит, 2009. -280 с.
- [Нильсон, 1985] Нильсон Н. Принципы искусственного интеллекта. Под ред. В.Л. Стефанюка. Изд. «Радио и связь», Москва,1985.
- [Стефанюк, 2004] Стефанюк В.Л. Локальная организация интеллектуальных систем. -М.: Физматлит, 2004. – 328с.
- [Berman a.o., 2007] Berman S., Halasz A., Kumar V., Pratt S.: Bio-Inspired Group Behaviors for the Deployment of a Swarm of Robots to Multiple Destinations. In 2007 International Conference on Robotics and Automation. pp. 2318 - 2323, IEEE, 2007
- [Brambilla a.o., 2012] Brambilla M., Ferrante E., Birattari M., Dorigo M.: Swarm Robotics: A Review from the Swarm Engineering Perspective. Technical Report TR/IRIDIA/2012-014, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium, 2012.
- [Garnier a.o., 2005] Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: Collective decision-making by a group of cockroach-like robots. In: Proc. of Swarm Intelligence Symposium, SIS 2005. pp. 233-240. IEEE, Los Alamitos, 2005.

- [Hecker a.o., 2012] Hecker J., Letendre K., Stolleis K., Washington D., Moses M.: Formica ex Machina: Ant Swarm Foraging from Physical to Virtual and Back Again. ANTS, volume 7461 of Lecture Notes in Computer Science, pp. 252-259. Springer (2012)
- [Nissan, 2009] Nissan EPORO Robot Car "Goes to School" on Collision-free Driving by Mimicking Fish Behavior - Advanced Robotic Concept Debuts at CEATEC JAPAN 2009 – URL: http://www.nissan-global.com/EN/NEWS/2009/_STORY/091001-01-e.html