

УДК 004.021

АЛГОРИТМ КЛАСТЕРИЗАЦИИ КОЛЛЕКТИВА РОБОТОВ

В.В. Воробьев (*gatus86@mail.ru*)
НИЦ “Курчатовский институт”, Москва

Аннотация. В статье рассматривается вопрос применения механизмов роения для задач коллективной робототехники. Предлагается алгоритм, который инициализируется лидером роя и заключается в последовательном разделении роя на заданное количество кластеров и выборе лидера в каждом из полученных кластеров, используя только локальное взаимодействие. Процесс повторяется до тех пор, пока их число не станет равным заданному, при этом количество кластеров, получаемых за одну итерацию деления, зависит от количества соседей лидера¹.

Ключевые слова: статический рой, групповая робототехника, модели социального поведения, кластеризация

Введение

В настоящий момент широкий интерес в области робототехники представляют вопросы, касающиеся коллективной робототехники (swarm robotics). Возможное практическое применение систем, состоящих из множества роботов, огромно: разведка [Tan et al., 2013], патрулирование [Bina, 2013], [Portugal et al., 2013], и т.д. Особый интерес представляет появление подобных систем, состоящих из роботов, которые имеют относительно простую конструкцию [Rubenstein et al., 2009], что должно привести к простоте масштабирования системы, высокой отказоустойчивости и т.д. и появлению в такой системе эмерджентных свойств [Yogeswaran et al., 2010].

С другой стороны, система из простых роботов обладает рядом ограничений, одно из которых – локальный характер взаимодействия друг с другом. В свою очередь это приводит к некоторым особенностям реализации механизмов управления коллективом и передачи информации между его членами.

В основе реализации механизмов коллективного управления могут

¹ Работа выполнена при финансовой поддержке РФФ (проект № 16-11-00018) и РФФИ (проект № 15-01-07900 и проект №16-29-04412).

лежать как автоматные модели [Цетлин, 1969], [Варшавский и др., 1984], так и многоагентные системы, эволюционные модели и т.д. В этом ряду можно выделить концепцию, которая подразумевает создание систем групповой робототехники, используя модели социального поведения, которая достаточно подробно описана в [Карпов, 2016].

Одним из интересных механизмов социального поведения, который наблюдается у многих видов муравьев и пчел и может быть полезен в коллективной робототехнике, является *роение*. Роение – размножение пчелиной семьи путем отделения от материнской семьи половины пчел с маткой и трутнями [Кокорев и др., 2005]. Если рассматривать пчел, то причины, способствующие старту роения, являются большое число молодых особей, не занятых работой в улье, старая матка, плохая вентиляция гнезда и его перегрев, переполнение его печатным расплодом и т.д. [Тамбовцев и др., 2004].

Примером применения роения в робототехнике может являться ситуация, когда коллективу необходимо выполнить ряд параллельных действий. Действительно, наблюдая за пчелами или муравьями, можно легко увидеть, что у них есть функциональная дифференциация. Часть занимается разведкой, часть обустройством улья/муравейника и т.д. Ситуации, когда рой целиком занят одной задачей, если и встречаются, то крайне редко. Таким образом, существует необходимость в механизме разделения роя в качестве дополнения к алгоритму функциональной дифференциации.

Кроме того, рой роботов может работать в условиях, когда каналы локальной связи работают нестабильно или данную связь сознательно подавляет противник. В таком случае информация от периферийных узлов может доходить до лидера с потерями или теряться вовсе. При этом, чем дальше узел-отправитель от лидера, тем больше узлов ее ретранслирует, тем вероятнее ее потери или повреждения. Следовательно, имеет смысл не собираться в таких условиях в большие группы, а если рой уже обладает большим числом элементов – разделиться.

Вообще говоря, вопрос интеграции механизма роения в систему управления коллективом роботов не является новым. В данной области эта проблема называется *кластеризацией*.

В работах [Chen et al., 2012], [Gross et al., 2009] представлен механизм, основанный на гранулярной конвекции или на *эффекте бразильского ореха*. Основная идея работы – группировать роботов по принципу случайного движения вокруг общей точки притяжения и отталкивания между соседями. Очевидно, что во время выполнения данного алгоритма тратится дополнительная энергия на передвижение роботов, что может быть критично в ряде задач.

В работе [Caro et al., 2012] представлена интересная реализация

алгоритма кластеризации, основанная на механизме передачи жетонов (token'ов) между роботами. Наличие жетона определяет принадлежность к кластеру. Алгоритм содержит в себе элементы, используемые в бисекции и балансировки нагрузки в сетях.

Идеологическим продолжением данной работы является [Cruz et al., 2016], где предложена модификация предыдущего алгоритма, позволяющая разделять на произвольное количество кластеров.

Кроме того, в работе [Каляев и др, 2009] достаточно подробно описывается процесс кластеризации. Выделяется иерархическая кластеризация, причем иерархия может выстраиваться двумя путями: “сверху-вниз” и “снизу-вверх”. Если механизм иерархической кластеризации идет по пути объединения отдельных роботов в кластеры, то она называется ”снизу-вверх”. В случае если производится дробление большой группировки роботов на более мелкие, то это кластеризация “сверху-вниз”. Также описаны механизмы создания непересекающихся кластеров постоянного состава, как для гетерогенных, так и для гомогенных групп роботов. Рассматриваются проблемы перераспределения роботов между кластерами и динамической кластеризации – процесс инициализации кластеров и их роста.

В основном в работах рассматривается проблема кластеризации в децентрализованных системах [Caro et al., 2012], [Cruz et al., 2016]. Однако в ряде задач целесообразно использовать централизованные системы, например задачи, где необходима высокая степень координации и контроля. При этом централизованная система также может решать задачи, например, функциональной дифференциации, где так или иначе необходимо проводить кластеризацию.

В связи с этим имеет смысл использовать особенности таких систем для реализации механизма деления. К примеру, наличие лидера может существенно упростить данную задачу, так как именно он может иметь полномочия формировать кластеры под конкретную задачу.

Однако, как уже было сказано, есть существенное ограничение в виде локального характера взаимодействия между роботами, т.е. он может общаться только со своими соседями. Следовательно, даже лидер не знает структуры и топологии роя, которым он командует. Это приводит к некоторым особенностям реализации механизма кластеризации в такой системе.

В работе предлагается достаточно простой алгоритм кластеризации коллектива роботов, который обладает лидерами, где каждый робот использует локальное взаимодействие. Лидер формирует зачатки кластера, но так как структура и топология роя ему неизвестны, зачатками становятся его соседи, которые стараются присоединить к себе своих соседей и т.д. Таким образом, легко формируются кластеры, число

которых меньше, либо равно числу соседей лидера. В случае если кластеров должно быть больше, вместе с формированием их зачатков, по числу соседей лидера им передается информация о необходимом делении после формирования кластера. Затем процедура повторится уже для сформированной группы роботов. Более всего данная процедура напоминает иерархическую кластеризацию “сверху-вниз”, описанную в [Каляев и др, 2009]. В работе показана техническая реализация данной процедуры в условиях наличия лидера и локального взаимодействия.

Отдельно стоит отметить, что, не смотря на то, что понятия “коллектив роботов” и “рой роботов” различны [Карпов, 2016], это несколько не должно влиять на механизм роения, т.к. он должен работать как в коллективе, так и в рое роботов.

Кроме того, в работе используются термины “робот” и “узел”, которые являются синонимами в контексте данной задачи: подразумевается, что, в конечном итоге, алгоритм должен работать в коллективе роботов, а термин “узел” чаще используется при описании работы алгоритма в качестве некоторой абстракции. Аналогично обстоит дело с терминами “фрагмент”, “часть”, “кластер” и “кластеризация”, “роение”.

1 Постановка задачи

Рассмотрим задачу: существует N роботов, взаимодействие которых друг с другом носит локальный характер, т.е. каждый робот “знает” о существовании только себя и своих ближайших соседей. Общая структура роя неизвестна никому, однако, средствами локального взаимодействия имеется возможность собрать косвенные данные, например, число элементов в нем. В рое выбирается лидер, используя, например, алгоритм, описанный в [Карпов и др., 2015].

Кроме того, предполагается, что в процессе кластеризации топология роя не меняется, т.е. его можно представить в виде структуры под названием статический рой - полученная в некий момент времени схема соединения роботов по каналам связи [Карпов, 2013]. При этом совсем не обязательно допускать, чтобы роботы были неподвижны – важно чтобы сохранялась схема соединений друг с другом.

Таким образом, задача звучит следующим образом: необходимо провести процедуру роения в статическом рое, разделив ее на M заданных частей.

2 Алгоритм кластеризации

Решение этой задачи осуществляется поэтапно: лидер формирует зачатки кластера и, при необходимости, передает информацию о дальнейшей кластеризации. Затем в каждом кластере выбирается свой

лидер, который уже имеет информацию о том, надо ли делиться дальше. Если такая необходимость есть, то процесс повторяется.

В первую очередь необходимо чтобы в рой был лидер, который инициирует процесс кластеризации. Механизм выбора лидера, который использовался в этой работе базируется на локальной передаче сообщений от периферийных узлов в центр роя. Это позволяет подсчитать количество уникальных соседей узла, в том числе и тех, с которыми он связан через другие узлы. Такой механизм позволяет выбирать лидера, который располагается достаточно близко к топологическому центру роя, что обеспечивает приемлемое время, которое необходимо сообщению от любого периферийного узла, чтобы дойти до лидера.

Алгоритм кластеризации представляет собой несколько процедур деления/выбора лидера, количество которых зависит от количества частей, на которые необходимо поделить рой и количества соседей у лидера, которым он может отправить сообщение о начале роения. Алгоритм начинается с сообщения, которое отправляет лидер своим соседям. При этом, если количество соседей больше или равно количеству частей, на которые рой необходимо разделить, то лидер произвольно выбирает кому он отправит данное сообщение. Каждому из выбранных соседей сообщается уникальный номер части, на которые делится рой. В свою очередь данные узлы отправляют данное сообщение с уникальным номером своим соседям и т.д. Если узел уже принадлежит какой-либо части, то он не может принимать сообщения от частей с другим уникальным номером, что позволяет закончить работу алгоритма разделением роя на заданное количество частей.

В случае если разделить рой необходимо на части, количество которых больше, чем количество соседей лидера, то алгоритм работает так, как если бы они были равны. Однако недостающие разделения запоминаются соседями лидера, которые передадут их своему новому лидеру. В дальнейшем, когда рой разделится первый раз и в каждой из частей появится лидер, инициирует новое разделение и т.д.

Например, есть рой, состоящий из 10 особей, у лидера 3 соседа и надо разделить его на 8 частей. Сначала рой разделится на 3 части, в каждой из которых будет храниться информация о том, что необходимо делиться дальше. Первоначально, она будет храниться только у соседей лидера, но в процессе разделения ее будут передавать всем узлам формирующейся части роя. В данном случае два соседних узла будут знать, что из их фрагментов должно получиться по три части в каждой, а третьему соседу известно, что его фрагмент должен разделиться надвое. Когда в каждом фрагменте появится свой лидер, то он инициирует новое разделение, в соответствии с данной информацией. Псевдокод алгоритма представлен ниже.

Алгоритм K1(M). Кластеризация

A – элемент роя (узел).

N – количество соседей узла A.

C_i – i-й сосед узла.

M – количество частей, на которые необходимо разделить рой.

P_c – номер части роя, к которой принадлежит узел-сосед.

P_A – номер части роя, к которой принадлежит узел A.

l – целочисленный результат деления M на N.

k – остаток от деления M на N.

Mem_c – информация о необходимости дальнейшего деления, хранящаяся в узле C.

Mem_A – информация о необходимости дальнейшего деления, хранящаяся в узле A.

Процедура K1.

if $M > N$ **then** – Если число частей, на которые необходимо разделить больше числа соседей лидера.

l = M/N

k = M%N

end if

if A- лидер **then**

for i:=1 to N [step 1] **do** – Цикл по соседям лидера

if C_i еще не принял сообщение о роении **then**

C_i принимает запрос о роении.

$P_c = i$

if $M > N$ **and** i \neq N **then** – Если число частей, на которые необходимо разделить больше числа соседей и есть еще соседи, которым сообщение не передано.

$Mem_c = 1$

end if

if $M > N$ **and** i == N **then** – Если число частей, на которые необходимо разделить больше числа соседей и это последний сосед, которому еще не передали сообщение

$Mem_c = k$

end if

end if

end for

A – не лидер

end if

if A- принял сообщение **then**

for i:=1 to N [step 1] **do**

```

if  $C_i$  еще не принял сообщение о роении then
   $C_i$  принимает запрос о роении.
   $P_c = P_A$ 
   $Mem_c = Mem_A$ 
end if
end for
  Выбор лидера
end if

```

3 Вычислительные эксперименты

Алгоритм был реализован в виде программы на языке C++. Входными данными алгоритма является количество роботов, составляющих рой, который необходимо разделить и число кластеров, на которые надо его разделить. Их число варьировалось от 50 до 200 с шагом в 25. Таким образом, было проведено 7 симуляций, по 100 итераций в каждом, где в каждой итерации случайно создавалась своя конфигурация роя с заданным количеством роботов.

Затем осуществлялись выборы лидера, и он выбирал соседей, которые будут формировать кластер. Рой делится на четыре части (рис. 1). Во всех случаях кластеризация прошла успешно. Эксперимент показал, что время роения линейно зависит от максимального расстояния между лидером и любым периферийным узлом(1):

$$T=O(\max(S(L,P_1), S(L,P_2), \dots, S(L,P_n))), \quad (1)$$

где T – время роения, $S(x,y)$ – расстояние, между x и y , L -лидер, P_n -периферийный узел.

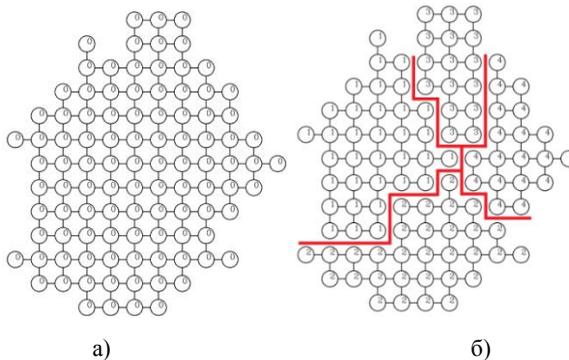


Рис. 1. а) – Рой до разделения, б) - Рой после разделения. Границы фрагментов отмечены линиями

Кроме того, установлено, что основной вклад во время работы алгоритма роения вносит механизм повторного выбора лидера в новом рое (рис. 2).

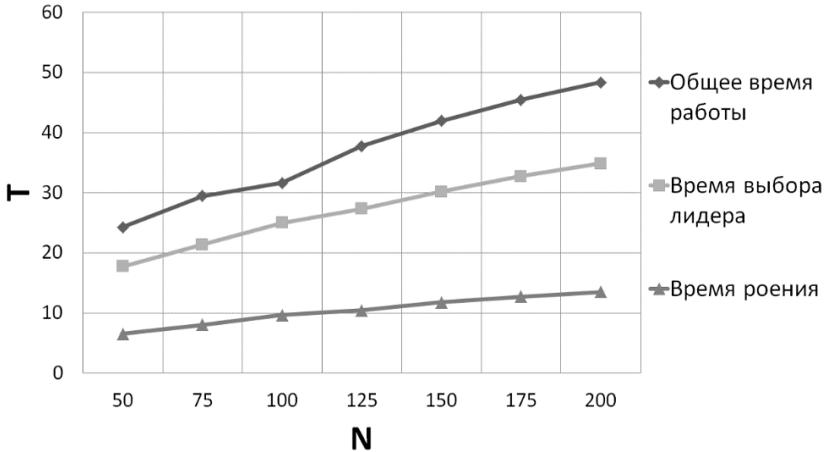


Рис. 2 Зависимость среднего времени выполнения (T) от количества роботов (N).

При разном начальном количестве роботов их число после роения в каждом кластере примерно одинаково для каждого кластера (рис. 3).

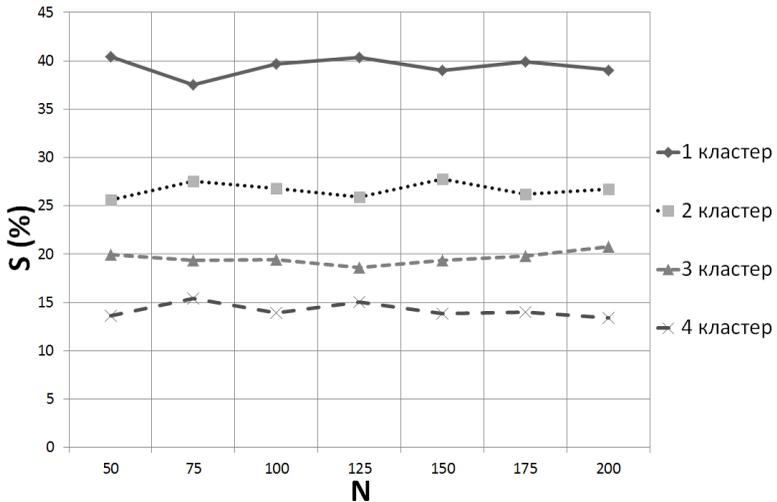


Рис. 3 Процентное соотношение числа роботов в каждом кластере S в зависимости от числа роботов N. 100% - число роботов в начале роения.

При разделении на небольшое количество фрагментов, время работы алгоритма зависит только от числа роботов в рое. Поэтому, для наиболее эффективной работы алгоритма необходимо, чтобы выполнялось неравенство (2):

$$N \geq M, \quad (2)$$

где N – количество соседей лидера, M – количество фрагментов

Заключение

Таким образом, был предложен алгоритм кластеризации, который может быть полезен для задач коллективной робототехники, в частности, для задач функциональной дифференциации. При этом стоит отметить, что данный алгоритм идеологически наиболее подходит для разделения роя на небольшое число фрагментов.

Это обуславливается тем, что алгоритму требуется повторно выбирать лидера в каждом фрагменте, что приводит к увеличению времени работы алгоритма при большом количестве фрагментов.

С другой стороны, с ростом числа кластеров накладные расходы на выбор лидера в каждом новом кластере должны сократиться, так как число роботов внутри них будет падать.

Кроме того, стоит отметить, что используемый алгоритм выбора лидера, также написанный автором, в процессе моделирования оказывал существенно влияние на результаты работы процедуры кластеризации. Данное влияние заключается не столько во времени, которое затрачивается на выбор лидера, сколько в том, какой именно узел будет выбран лидером. Если в самом начале лидер находился достаточно близко к центру роя, то после появления кластеров их лидеры не всегда обладали таким свойством, что часто приводило к ситуациям, когда кластеризация далее невозможна.

Несмотря на это, алгоритм проводит кластеризацию. Устранение данных недостатков может расширить границы его применимости.

Список литературы

- [Варшавский и др., 1984] Варшавский В.И., Поспелов Д.А. Оркестр играет без дирижера: размышления об эволюции технических систем и управления ими. – М.: Наука, 1984, –208с.
- [Каляев и др., 2009] Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов /. – М.: Физматлит, 2009.
- [Карпов, 2013] Карпов В.Э, Управление в статических роях. Постановка задачи. //VII-я Международная научно-практическая конференция "Интегрированные модели и мягкие вычисления в искусственном интеллекте" (20-22 мая 2013) Сб. научных трудов. В 3-т., М: Физматлит, 2013, Т.2, с.730-739
- [Карпов и др., 2015] Карпов В.Э., Карпова И.П., Leader election algorithms for static

- swarms //Biologically Inspired Cognitive Architectures №12, 2015, с.54-64
- [**Карпов, 2016**] Карпов В.Э., Модели социального поведения в групповой робототехнике. //Управление большими системами, М: ИПУ РАН, 2016, Выпуск 59, с.165-232, -388с.
- [**Кокорев, 2005**] Кокорев Н., Чернов Б., Роение медоносных пчел и противороевые приемы. – М.: ТИД Континент-Пресс, Континеталь-Книга, 2005. – 48с.
- [**Тамбовцев и др., 2004**] Тамбовцев К.А., Салагаев К.А., Пырялин Г.Л., Яковлева М.П., Ишмуратов Г.Ю. Особенности применения препарата «Апирой» // Пчеловодство. 2004. No 3. С. 13-14.
- [**Цетлин, 1969**] Цетлин М.Л., Исследования по теории автоматов и моделированию биологических систем. М.:Наука,1969. 316с.
- [**Bina, 2013**] Bina D., Effective cooperation and scalability in multi-robot teams for automatic patrolling of infrastructures, 2013.
- [**Caro et al., 2012**] Di Caro G.A., Ducatelle F., Gambardella L., A fully distributed communication-based approach for spatial clustering in robotic swarms, in: Proceedings of the Second Autonomous Robots and Multi-Robot Systems Workshop (ARMS), Affiliated With the 11th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), Valencia, Spain, 2012,
- [**Chen, et al, 2012**] Segregation in swarms of e-puck robots based on the Brazil nut effect, in: Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems, vol. 1., International Foundation for Autonomous Agents and Multi-Agent Systems, Richland, SC, 2012, pp. 163–170.
- [**Cruz et al., 2016**] Cruz, N.B., Nedjah, N., de Macedo Mourelle, L., 2016. Robust distributed spatial clustering for swarm robotic based systems. *Applied Soft Computing*, p.11.
- [**Gross et al., 2009**] Gross R., Magnenat S., Mondada F., Segregation in swarms of mobile robots based on the brazil nut effect, in: Proceedings of International Conference on Intelligent Robots and Systems – IROS 2009, IEEE/RSJ, 2009, pp. 4349–4356
- [**Portugal et al., 2013**] Portugal D., Rocha R., Multi-robot patrolling algorithms: examining performance and scalability. *Advanced Robotics*, 27(5), pp.325–336.
- [**Rubenstein et al., 2010**] Rubenstein M., Nagpal R., Kilobot: A Robotic Module for Demonstrating Behaviors in a Large Scale (2^{10} Units) Collective. Proceedings of the IEEE 2010 International Conference on Robotics and Automation Workshop, Modular Robotics: State of the Art, (D), pp.47–51.
- [**Tan et al., 2013**] Tan Y., Zheng Z., Research Advance in Swarm Robotics. *Defence Technology*, 9(1), pp.18–39.
- [**Yogeswaran et al., 2010**] Yogeswaran M., Ponnambalam S.G., Swarm Robotics: An Extensive Research Review. In *Advanced Knowledge Application in Practice*, In Tech. pp. 259–278.