

УДК 004.021, 007.52

АЛГОРИТМ ДИНАМИЧЕСКОГО ФОРМИРОВАНИЯ СТАИ

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

Аннотация. В работе рассматривается решение задачи выбора лидера в группе роботов. Особенностью предлагаемого решения является возможность определять лидера таким образом, чтобы он располагался близко к топологическому центру динамически формируемой группы роботов. При решении последующих задач, которые требуют интенсивного обмена данными между лидером и периферийными роботами, такая конфигурация обеспечивает минимизацию времени, которое для этого необходимо. В основе предлагаемого механизма лежит обмен сообщениями между каждым отдельным роботом и его локальными соседями.¹

Ключевые слова: групповая робототехника, выбор лидера, статический рой, локальное взаимодействие.

Введение

Задача выбора лидера относится к категории фундаментальных задач групповой робототехники [Dieudonné et al., 2010], о чем также говорит и большое число работ и подходов к решению этой проблемы.

Важность задачи выбора лидера обусловлена тем, что наличие лидера в группе роботов позволяет более эффективно управлять этой группой и контролировать ее. Иерархия также необходима при решении задач планирования, контроля над исполнением плана и т.д., которые неизбежно возникают при решении сложных групповых задач, например, сохранения энергетического гомеостаза группы, выполнения запросов оператора и т.п. При использовании механизмов социального поведения в основе системы управления роботами (см., например, [Карпов, 2016]) также, так или иначе, требуются механизмы, способные выделить доминантных особей в группе таких роботов.

Процесс выбора лидера описан применительно к сетевым структурам в [Lynch et al., 1993]. Представлены алгоритмы Ле-Ланна, его модификация –

¹ Работа выполнена при частичной финансовой поддержке РФФИ (проект № 16-29-04412 офи_м).

алгоритм Ченя-Робертса, Хиршберга-Синклера, Фредриксона-Линча и т.д. для кольцевых структур. В основном алгоритмы основаны на обмене уникальными идентификаторами. Например, в случае алгоритма Ле-Ланна каждый элемент сети отправляет свой маркер с отличительным признаком (весом). Когда маркер возвращается отправителю остальные маркеры также прошли через него, следовательно, отправитель сможет выбрать маркер с наименьшим или наибольшим весом.

Решение подобной задачи описывается в [Santoro, 2007], где представлены сразу несколько стратегий выбора лидера, которые, однако, также подходят более для вычислительных сетей, нежели для группы роботов. В работе рассматриваются различные топологии сети и алгоритмы выбора лидера для них. Кроме того, представлен и ряд “универсальных” алгоритмов, наиболее интересный из которых УО-УО. Принцип его работы заключается в обмене заранее заданными весами между вычислительными устройствами. Лидером становится устройство с наименьшим весом. При этом алгоритм обладает стадией подготовки, что не всегда удобно, особенно для групповой робототехники.

Еще одним решением является определение лидера с помощью оценки его расстояния до центра определенной окружности для группы анонимных роботов [Canera et al., 2007]. Робот становится лидером, если это расстояние для него минимально. Если таковых нет, то с некоторой вероятностью роботы с некоторой вероятностью двигаются к центру этой окружности. Расстояние тем больше, чем они дальше от центра. Также описан случай с 3-я роботами, когда лидером становится тот, чья разница углов, вычисляемых для всех соседей минимальна. Таким образом, роботы должны иметь возможность двигаться и иметь средство связи со всеми роботами группы, например, доску объявлений.

Похожим образом задача решается в [Flocchini et al., 2008], где лидер определяется в соответствии с заданным шаблоном – тот, кто был ближе к заданной шаблонной точке, тот и становится лидером. При этом роботы анонимны и не имеют общей координатной системы.

Другой подход к выбору лидера, который разрабатывался специально для групп роботов, описан в [Gan Chaudhuri et al., 2015]. В рассматриваемой модели роботы не имеют общей координатной системы и уникальных идентификаторов, группировка гомогенна. При этом роботы не могут опираться на информацию о вычислениях и сенсорных данных, сделанных ранее. Выбор лидера происходит по принципу близости робота к центру определенной формации, т.е. роботы сначала формируют некий паттерн, который одновременно является чем-то вроде глобальной системы координат, а затем определяются финальные позиции каждого робота. В конце они стараются достичь данных позиций. Тот робот, который

окажется ближе всего к центру описанной окружности данной формации и станет лидером.

Еще один механизм выбора лидера описан в [Karpov et al., 2015], суть которого, заключается в том, что робот определяет за кого проголосовали его соседи. В зависимости от веса кандидата, за которого голосует его сосед, робот может поменять свой выбор и проголосовать за того же кандидата. Особенностью такого подхода является то, что нет необходимости в ведении уникальных идентификаторов роботов, а также то, что они используют только локальное взаимодействие друг с другом. Сам процесс выбора лидера происходит в структуре, называемой статическим роём – некой фиксированной в определенный момент времени сети, состоящая из роботов, соединенных друг с другом по каналам связи [Карпов, 2013].

Специфика рассмотренных методов заключается в том, что лидером может стать любой робот, в том числе и периферийный, т.е. тот, кто находится на краю группы. В условиях, когда роботы общаются исключительно локально, т.е. только с ограниченным числом своих соседей, может возникнуть ситуация, когда время обмена данными с лидером существенно увеличивается. Поэтому важно чтобы лидером становился робот, который находится близко к центру группы. Подобный алгоритм, который позволяет так назначать лидера, был описан в [Воробьев, 2017]. Однако его особенностью является то, что группа роботов заранее сформирована в статический рой [Карпов, 2013], а начало процедуры выбора лидера инициируется сигналом извне, например, с помощью команды оператора. В реальных задачах это не всегда возможно, другими словами специфика решения такой задачи должна учитывать не только топологические характеристики группировки, т.е. взаимное расположение роботов, но и его динамику, т.е. изменение топологии группы в процессе появления новых роботов/отключения уже включенных в группу. Сложность заключается в том, что это можно оценить только в процессе обмена информацией друг с другом по локальным каналам связи.

Таким образом, можно отметить, что некоторые алгоритмы выбора лидера ориентируются в основном на сетевые структуры, где не учитывается возможность перемещения узлов сети, что важно при использовании в робототехнике. Ряд других методов выбирает лидера по степени геометрической близости к определенной точке или с помощью обмена весами. При этом на группу роботов накладываются существенные ограничения, например, они анонимны, не имеют общей координатной системы, могут общаться только локально. Однако не рассматривается вопрос дальнейшего обмена информацией между лидером и остальными членами группы, который особенно важен в случае локальности взаимодействия роботов друг с другом. В таком случае, если

характеристики канала связи между соседями не зависят от расстояния или ими можно пренебречь, важно, чтобы лидером стал робот, близкий к топологическому центру этой группы.

1 Постановка задачи

Предположим, что есть гомогенная группировка роботов численности N , каждый из которых описывается шестеркой $V=(\alpha, L, C, W, W_{my}, P)$, где $\alpha > 0$ – уникальный идентификатор робота, L – список его соседей, C – идентификатор робота, за которого голосует V , W – вес лидера, за которого голосует V , W_{my} – собственный вес, P – расстояние от лидера до данного робота. Группировка является роем (по классификации групп роботов, представленной в [Карпов, 2016] и в [Kernbach, 2013]). За некоторое конечное время необходимо, чтобы роботы самостоятельно организовали стаю, с лидером во главе для дальнейшего решения задачи информационного обмена, который будет достаточно близко располагаться к топологическому центру роя. Формально, близость к центру, гарантированность процесса выбора и единственность лидера определяются таким же образом, как и в [Воробьев, 2017], т.е. необходимо гарантированно перевести рой из начального состояния (1):

$$S_{swarm}=\{V_1, V_2, \dots, V_N\}, \forall V_i C_i=0, i=1, 2, \dots, N \quad (1)$$

в состояние (2):

$$S_{flock}=\{V_1, V_2, \dots, V_N\}, \exists! V_i C_i=\alpha_i, i=1, 2, \dots, N \text{ и} \quad (2)$$

$$\forall V_j C_j=\alpha_i, j=1, 2, \dots, N; j \neq i;$$

Предположим, что $M=\{0, \alpha_1, \alpha_2, \dots, \alpha_N\}$, тогда гарантированность процесса выбора можно описать следующим образом (3):

$$\forall N \exists S=\{S_{swarm}, S_1, \dots, S_P, S_{flock}\}, S_i=\{V_1, V_2, \dots, V_N\}, \quad (3)$$

$$\text{где } \forall V_j C_j=M_k,$$

$$i=1, 2, \dots, P; j=1, 2, \dots, N; k=1, 2, \dots, N+1$$

Если интерпретировать рой как граф G , где роботы являются вершинами, а связь с соседями - ребрами, то близость лидера к центру роя d определяется следующим образом (4):

$$d = \frac{s}{r(G)} \quad (4)$$

где r – радиус графа G , s – расстояние от центра графа до робота-лидера.

В общем и целом, постановка задачи схожа с постановкой, представленной в [Воробьев, 2017]. Отличием является то, что в общем случае роботы в начальный момент времени могут находиться далеко друг от друга и быть никак не связаны по каналам связи, т.е. $\forall V |L| = 0$.

2 Алгоритм решения

Начало процедуры выбора лидера определяется каждым роботом в группе самостоятельно. В этот момент он переключает локальную связь в пассивный режим, т.е. не передает в эфир никакие сообщения, и направляется в ЦА) – специальное место, которое геометрически является окружностью, радиусом $R/2$, где R – радиус действия локальной связи робота, координаты которого заранее известны всем роботам. Необходимость в нем возникает из-за того, что роботы заканчивают решение предыдущей задачи несинхронно, следовательно, может возникнуть ситуация, когда одновременно формируется несколько отдельных групп роботов, которые никогда не соединятся, так как при включении робота в группу он прекращает свое движение. Наличие единого “места голосования” позволяет нивелировать эту проблему.

Когда первый робот попадает в ЦА, он переводит локальную связь в активный режим и начинает процедуру выбора лидера. Роботы, которые затем будут приезжать в ЦА, сначала будут активироваться сигналами локальной связи первого робота. Перейдя в активный режим, они смогут активировать прибывающих пассивных членов группы. Это расширяет “зону активации” – объединение областей действия локальной связи активных роботов, попадание внутрь которой, активирует пассивного робота (см. Рис. 1).

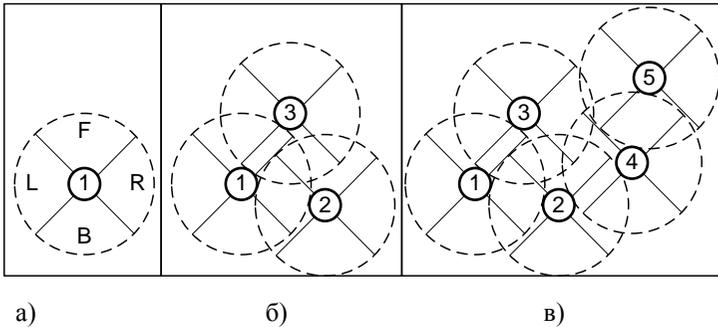


Рис. 1. а) - робот, приехавший в центр ЦА первым. Показаны сектора действия локальной связи б) - два робота, которые были активированы роботом №1 в) - два робота, которые были активированы вторично 2 и 4 роботами. Пунктир - зона действия локальной связи роботов.

Необходимость такого механизма обусловлена тем, что робот может обмениваться данными только с ограниченным числом своих ближайших соседей, т.е. исключается возможность связи “все-со-всеми” при числе роботов большим, чем число каналов локальной связи. Например, если это число равно 4, то он сможет обмениваться информацией только с 4-я

ближайшими соседями, по одному с каждой из сторон (см. Рис. а). Однако, используя соседей как ретрансляторы, робот может опосредованно взаимодействовать с остальными роботами.

Алгоритм выбора лидера представляет собой процесс обмена между роботами своими весами и изменения этих весов в зависимости от того, насколько далеко данный конкретный робот находится от самого удаленного от него робота. При этом играет роль не физическое расстояние между ними, а топологическое – число роботов между ними, так как принимается, что качество локальной связи и скорость передачи по ней данных не зависит от расстояния между роботами, в границах работоспособности локальной связи.

Фактически же, если рассматривать группу роботов как граф, куда произвольно добавляются вершины, а веса ребер одинаковы, то можно сказать, что эта процедура является процедурой динамического определения эксцентриситета всех его вершин. При этом лидером становится вершина с наименьшим эксцентриситетом, т.е. центр графа. Если встречаются вершины с одинаковым эксцентриситетом, то лидером становится вершина с наибольшим или наименьшим α .

При этом распределение весов роботов будет таково, что роботы с меньшими весами будут находиться ближе к топологическому центру роя, чем роботы с большими весами (см. Алгоритм 1).

Алгоритм 1. Динамическое определение лидера. При равенстве весов лидером становится робот с наибольшим α .

α – идентификатор робота V , α_L – идентификатор робота-кандидата, L – список соседей робота V , len_{max} – максимальная длина списка L , W – вес робота, за которого голосует V , W_{my} – собственный вес робота V , W_L – вес кандидата, W_{myL} – собственный вес соседа, P – расстояние до робота-кандидата, за которого голосует V , P_L – расстояние от робота-соседа, до кандидата в лидеры, C – идентификатор робота, за которого голосует V , $buffer$ – буфер входных сообщений робота V от соседей в списке L , S – “пройденный путь” сообщения, т.е. число роботов, его ретранслировавших, включая отправителя. С помощью него каждая вершина корректирует свой вес/эксцентриситет.

Начало

$W_{my} = 0$

если робот V не в стае **то**

цикл пока не конец $buffer$ **то**

если сообщение от маяка ЦА **то**

$C = \alpha$; $W = W_{my}$; $P = 0$ // V - лидер

иначе

$W_{my} = W_{my} + 1$

	если $((W_{my} < W_L)$ или $((W_{my} == W_L)$ и $(\alpha > \alpha_L))$)
то	$C = \alpha; W = W_{my}; P = 0 // V$ - лидер
	иначе
	$C = \alpha_L; W = W_L; P = P_{L+1} // V$ -
подчиняется	Отослать соседям $C, W, W_{my}, P, S = 0$
иначе	цикл пока не конец buffer то
	если buffer - сообщение от лидера то
	если $((W_{my} < W_L)$ или $((W_{my} == W_L)$ и $(\alpha > \alpha_L))$)
то	$C = \alpha; W = W_{my}; P = 0 // V$ - лидер
	иначе
	$C = \alpha_L; W = W_L; P = P_{L+1} // V$ -
подчиняется	Отослать соседям C, W, W_{my}, P
	иначе
	$S = S + 1$
	если $W_{my} > S$ то
	$W_{my} = W_{myL} + 1$
	Отослать соседям α_L, W_{my}, S, P
Конец	

3 Эксперименты

Как вычислительные так и реальные эксперименты использовали единый программный базис для получения экспериментальных данных. Его основой является система управления отдельным роботом, а также сопутствующие узлы, осуществляющие коммутацию портов локальной связи, сбор статистики и т.д., при создании которых использовался механизм узлов и топиков операционной системы ROS [ROS Kinetic, 2018]. Отличием является то, что в случае реальных экспериментов использовались навигационные данные (положение, угол поворота) реальных роботов на полигоне. В случае вычислительных экспериментов эти данные брались из конфигурационного файла.

Также для вычислительных экспериментов не моделировался процесс движения роботов. Это было сделано для того, чтобы нивелировать связанный с этим процессом фактор, который существенно влияет на итоговые показатели времени выполнения алгоритма. Иначе говоря, время, за которое будет выбран лидер для определенного числа роботов, безусловно зависит от того, насколько быстро они все достигнут ЦА.

Однако сам процесс выбора лидера не зависит от скорости прибытия роботов в ЦА. В связи с этим был предложен вариант вычислительных экспериментов, где роботы уже сформировали структуру, похожую на статический рой, однако все они находятся в пассивном режиме, т.е. $\forall V |L| = 0$. Затем произвольным образом инициируется один из этих роботов, как если бы он сам достиг ЦА, а он уже активирует своих соседей, те – своих и т.д., что имитирует их приближение к ЦА. При этом остальные роботы расположены вне ЦА, но в радиусе действия локальной связи инициированного робота.

3.1 Вычислительные эксперименты

При проведении вычислительных экспериментов использовалась возможность промоделировать работу алгоритма для существенного большего числа роботов $N=25,50,\dots,150$ с шагом в 25, чем в реальности. Для каждого значения N проводилось 100 вычислительных экспериментов с разной расстановкой роботов. Измерялось два параметра: T - время выполнения алгоритма, d - относительная близость лидера к центру графа этой группы.

На рис. 2 видна линейная зависимость $O(N)$ среднего и максимального времени выполнения алгоритма в зависимости от числа роботов N .

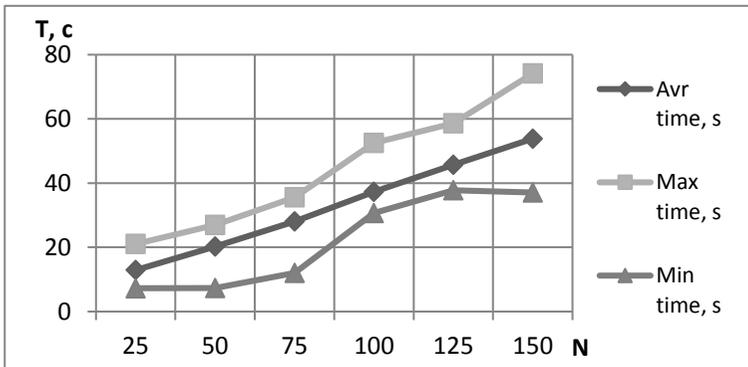


Рис. 2. Зависимость времени выполнения T алгоритма от числа роботов N

Вместе с тем, если сравнивать с алгоритмами, которые равновероятно выбирают лидера в группе, например, с [Karpov et al., 2015], то при одинаковой временной сложности в абсолютных значениях время выполнения предлагаемого алгоритма выше.

На Рис. 3, то можно увидеть, что среднее отношение расстояния от центра графа до робота-лидера к радиусу графа $Avg\ dev$ находится в пределах от 0.1 до 0.12, что говорит о том, что в среднем лидер отстоит от

центра графа не более чем на 10-12% радиуса графа, описывающего конфигурацию группы. Максимальное отклонение доходит до значения 0.63 для $N=75$, однако это связано непосредственно с формой группы. Дело в том, что при вычислении радиуса графа учитывается непосредственное расстояние между вершинами, а не топологическое. В связи с этим могут быть ситуации, когда вершина является радиусом графа, находясь с краю группы. Алгоритм же учитывает форму группы, что может давать существенные максимальные отклонения.

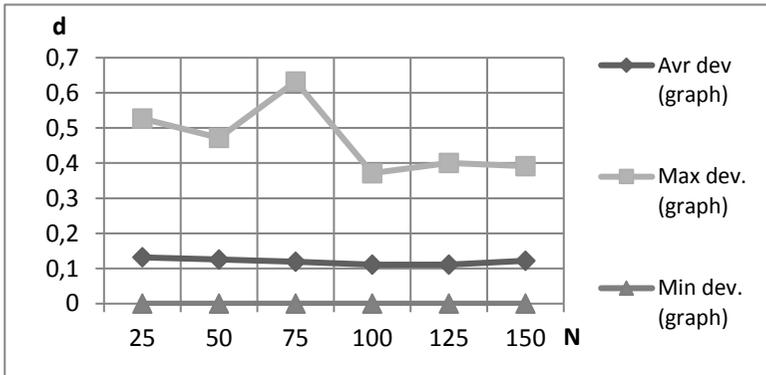


Рис. 3. Зависимость параметра d от числа роботов N

Если сравнивать полученные данные с данными, получаемыми от алгоритмов с равновероятным выбором лидера, то можно отметить, что первые существенно чаще выбирают лидером робота, который находится в центре группы или не более чем на расстоянии 1-2 роботов от него. Например, для $N=25$ – 95%, а для $N=150$ – 85%.

3.2 Эксперименты на реальной группировке роботов

Работа алгоритма была опробована на реальной группировке роботов YARP-2 на лабораторном полигоне. Использовалось 6 роботов в 20 экспериментах с различными расстановками. Как и было отмечено, основной вклад во время выбора лидера дает время, необходимое роботам, чтобы собраться в ЦА. В данном случае общее время работы $T \approx 10-11$ с. Оценка d для такого числа не показательна, однако в расстановках роботов, которые дают в дальнейшем четко выраженного робота в центре, именно он и становился лидером группы.

Заключение

В работе предложен алгоритм динамического выбора лидера в группе роботов с локальным взаимодействием, который позволяет учитывать топологию получаемой группы. В основе алгоритма лежит механизм обмена весами между роботами группы. Рассматривая схему соединений роботов по их каналам локальной связи в виде графа, отдельного робота можно интерпретировать как вершину графа, а его вес как эксцентриситет этой вершины. Таким образом, с помощью предлагаемого алгоритма лидером становится робот с минимальным эксцентриситетом, который считается динамически и локально для каждой вершины.

Эксперименты и на модели, и на реальной группировке роботов показали работоспособность предлагаемого механизма, приемлемую временную сложность порядка $O(N)$, где N – число роботов.

Список литературы

[**Gan Chaudhuri et al., 2015**] Gan Chaudhuri S., Mukhopadhyaya K. Leader election and gathering for asynchronous fat robots without common chirality // *J. Discret. Algorithms*. 2015. (33). P. 171–192.

[**Canepa et al., 2007**] Canepa D., Potop-Butucaru M. Stabilizing flocking via leader election in robot networks // *Proceedings of the 9th international conference on Stabilization, safety, and security of distributed systems*. 2007. P. 52–66.

[**Dieudonné et al., 2010**] Dieudonné Y., Petit F., Villain V. Leader election problem versus pattern formation problem // *Distribute computing*. 2010. P. 267–281.

[**Flocchini et al., 2008**] Flocchini P., Prencipe G., Santoro N., Widmayer P. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots // *Theor. Comput. Sci*. 2008. № 1–3(407). P. 412–447.

[**Karpov et al., 2015**] Karpov V., Karpova I. Leader election algorithms for static swarms // *Biol. Inspired Cogn. Archit*. 2015. (12). P. 54–64.

[**Kernbach, 2013**] Kernbach S // *Handbook of Collective Robotics: Fundamentals and Challenges*. 2013. 962 p.

[**Lynch et al., 1993**] Lynch N.A., Patt-Shamir B. *Distributed Algorithms* // 1993. 440 p.

[**Santoro, 2007**] Santoro N. *Design and analysis of distributed algorithms*. //2007. 589 p.

[**Воробьев, 2017**] Воробьев В.В. Алгоритмы выбора лидера и кластеризации в статическом рое роботов // *Мехатроника, автоматизация, управление*. 2017. № 3(18). С. 166–172.

[**Карпов, 2013**] Карпов В.Э. Управление в статических ряях. Постановка задачи // VII-я Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте». // 2013. С. 730–739.

[**Карпов, 2016**] Карпов В.Э. Модели социального поведения в групповой робототехнике // *Управление большими системами*. 2016. № 59. С. 165–232.

[**ROS Kinetic, 2018**] ROS Kinetic documentation. - <http://wiki.ros.org/kinetic>