

Некоторые простые алгоритмы управления мультиагентными системами¹

П. С. Щербаков, д. ф.-м. н.

Институт проблем управления РАН, Москва

sherba@ipu.ru

Рассматривается разработанный в [1] алгоритм эволюции совокупности точек на плоскости, приводящий ее в некоторую регулярную конфигурацию. Предлагаются обобщения алгоритма, изучаются некоторые новые свойства, обсуждается связь с методами управления формациями, намечаются новые и более простые алгоритмы такого типа.

Ключевые слова: мультиагентные системы, степенной метод, линейные алгоритмы.

1. Введение

В приложениях часто возникают задачи, которые можно грубо формализовать следующим образом. Имеется точки (на прямой, окружности, плоскости, в пространстве); каждая из них обладает некоторой информацией о других (о себе, соседях, некоторой реперной точке и т. д.). Задача заключается в построении алгоритма эволюции точек (иногда называемых агентами), который использует лишь эту информацию и приводит всю совокупность в некоторую заданную конфигурацию.

Популярность этой тематики в последние два-три десятилетия во многом объясняется развитием вычислительной техники и возможностью использования результатов в областях, на первый взгляд весьма далеких друг от друга: транспорт, военное дело, управление социальными группами и др.

Соответственно, и литература по данному вопросу обширна и разнородна, а терминология часто варьируется в зависимости от прикладной интерпретации формальной постановки задачи. Ключевыми словами в этой области являются: децентрализованное управление, групповое/кооперативное управление, управление формациями, самоорганизующиеся системы, мультиагентные системы. Одной из самых последних книг, предоставляющих содержательное введение в предмет, обсуждение типичных проблем, алгоритмов ре-

¹©П. С. Щербаков, 2010

шения, приложений, является [2], содержащая также и обширную библиографию.

Принципиальным в таких задачах является (i) неполнота информированности агентов о состоянии системы в целом и (ii) отсутствие единого управляющего органа.

Цель данной работы — привлечь внимание к одному из элегантных и неожиданных алгоритмов такого сорта, рассмотренному в [1], обсудить некоторые его новые свойства и обобщения, а также предложить альтернативные, более простые линейные алгоритмы.

2. Схема Ван Лоуна

В работе [1] исследовалась следующая задача, которую назовем схемой Ван Лоуна.

На плоскости даны n точек $p_i, i = 1, \dots, n$; каждой i -й точке доступна информация о своих координатах и координатах точки с номером $i + 1$; последняя n -я точка имеет информацию о себе и о точке с номером 1.

Предлагается следующий алгоритм управления движением совокупности $\{p_i\}_1^n$. Обозначим через x_i^k, y_i^k координаты i -й точки на k -м шаге алгоритма и введем “сборные” векторы $x^k = (x_1^k, \dots, x_n^k)^T$ и $y^k = (y_1^k, \dots, y_n^k)^T$. Шаг алгоритма состоит из двух этапов и устроен следующим образом:

$$\mathbf{A} : \begin{cases} \tilde{x}_i^{k+1} = \frac{1}{2}(x_i^k + x_{i+1}^k), & i = 1, \dots, n-1; & \tilde{x}_n^{k+1} = \frac{1}{2}(x_n^k + x_1^k), \\ \tilde{y}_i^{k+1} = \frac{1}{2}(y_i^k + y_{i+1}^k), & i = 1, \dots, n-1; & \tilde{y}_n^{k+1} = \frac{1}{2}(y_n^k + y_1^k); \end{cases}$$

$$\mathbf{B} : \quad x^{k+1} = \tilde{x}^{k+1} / \|\tilde{x}^{k+1}\|; \quad y^{k+1} = \tilde{y}^{k+1} / \|\tilde{y}^{k+1}\|.$$

Таким образом, на этапе **A** новое положение точки i есть среднее арифметическое ее самой и точки $i + 1$ (последняя точка усредняется с первой), а на этапе **B** происходит нормализация всего сборного вектора x и вектора y .

В работе [1] получен следующий неожиданный результат.

Теорема [1]. Пусть начальное расположение точек таково, что $\sum x_i^0 = 0, \sum y_i^0 = 0$. Тогда при $k \rightarrow \infty$ точки стремятся расположиться на эллипсе с центром в нуле и повернутом на $\pi/4$.

Матрица S предельного эллипса вычисляется следующим образом. Введем векторы

$$\tau = \frac{2\pi}{n} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{pmatrix}; \quad c = \cos(\tau); \quad s = \sin(\tau)$$

и составим 2×2 матрицу

$$A = \sqrt{\frac{2}{n}} \begin{pmatrix} \frac{c^T x^0}{\sqrt{(c^T x^0)^2 + (s^T x^0)^2}} & \frac{s^T x^0}{\sqrt{(c^T x^0)^2 + (s^T x^0)^2}} \\ \frac{c^T y^0}{\sqrt{(c^T y^0)^2 + (s^T y^0)^2}} & \frac{s^T y^0}{\sqrt{(c^T y^0)^2 + (s^T y^0)^2}} \end{pmatrix}. \quad (1)$$

Предельный эллипс имеет вид

$$\mathcal{E} = \{p \in \mathbb{R}^2 : p^T S^{-1} p = 1\}, \quad \text{где } S = (AA^T)^{1/2}. \quad (2)$$

На рис. 1 в качестве иллюстрации приведено некоторое начальное расположение $n = 10$ точек, случайно сгенерированных на единичном квадрате $[-1; 1]^2$, и их расположение после 40 шагов алгоритма.

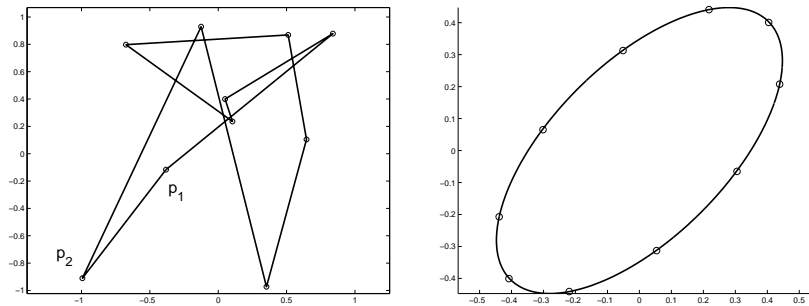


Рис. 1: Некоторое начальное расположение; расположение после $k = 40$ шагов ($n = 10$)

2.1. Обсуждение

Нетрудно видеть, что этап **A** можно записать в матрично-векторном виде, если ввести в рассмотрение матрицу

$$M = \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix} \in \mathbb{R}^{n \times n},$$

в которой все неотмеченные элементы — нули. Тогда шаг алгоритма запишется как

$$x^{k+1} = Mx^k / \|Mx^k\|; \quad y^{k+1} = My^k / \|My^k\|, \quad (3)$$

что является не чем иным, как независимым применением степенного метода для матрицы M при начальных условиях x^0 и y^0 , например, см. [3–5]. Это соображение лежит в основе подробного анализа алгоритма, проведенного в [1]; соответственно, и аппарат исследования опирается лишь на общие сведения из линейной алгебры.

В частности, в [1] показано, что скорость сходимости совокупности к предельному эллипсу зависит от начального расположения точек (близости x^0 , y^0 к некоторой двумерной плоскости) и от количества n точек, которое однозначно определяет матрицу M . Точнее, сходимость зависит от отношения 3-го и 2-го максимальных собственных значений матрицы M (которые выписываются в явном виде) и тем медленнее, чем больше n .

Под сходимостью понимается следующее. Для всякого $\varepsilon > 0$ найдется целое K такое, что для всех $k \geq K$ и для всех $i = 1, \dots, n$ выполнено $|(p_i^k)^T S^{-1} p_i^k - \varepsilon| \leq 1$, т. е. в любой момент времени k , начиная с некоторого, каждая точка p_i^k будет находиться близко к эллипсу. При этом, однако, у системы точек наблюдается нестационарное поведение: в процессе итераций точки “двигаются по эллипсу”, а не стремятся занять некоторые фиксированные предельные положения на нем. В [1] показано, что на четных (соответственно, нечетных) итерациях в совокупности расположение точек одинаково. Точнее, при всех четных (нечетных) k многоугольники с вершинами p_1^k, \dots, p_n^k совпадают друг с другом, причем если обозначить эти два расположения через \mathcal{P}_{even} и \mathcal{P}_{odd} , то $\mathcal{P}_{even} \neq \mathcal{P}_{odd}$.

Заметим, что x -координаты и y -координаты точек меняются по одинаковому закону, но независимо друг от друга. Кроме того, на каждом шаге координаты каждой точки нормируются на весь сборный вектор. В рамках теории мультиагентных систем такая нормировка означает значительную информированность агентов и слабо укладывается в схему распределенной информации, децентрализованного управления и т. д., поскольку для эволюции одной точки $p_i^k \rightarrow p_i^{k+1}$ требуется информация о всей популяции. Было бы интересно предложить какую-либо содержательную трактовку обсуждаемого алгоритма, тем более что в [1] не приводится никаких соображений по этому вопросу, и проведенный анализ чисто умозрителен.

Ниже предложим более простые алгоритмы, которые ближе по своим принимаемым предположениям к идеологии мультиагентных систем. Сейчас же приведем некоторые новые свойства и обобщения схемы Ван Лоуна.

2.2. Некоторые новые свойства и обобщения

Равномерность предельного расположения. В [1] показывается, что в процессе итераций точки выстраиваются на предельном эллипсе по порядку своих номеров. Кроме того, “регулярность” расположения точек на эллипсе заключается в следующем. При линейном преобразовании координат с матрицей S^{-1} эллипс \mathcal{E} перейдет в единичную окружность, причем точки $S^{-1}p_i^k$ будут располагаться на ней *равномерно*, на одинаковых расстояниях. Доказательство этого свойства предлагается читателю в качестве (несложного) упражнения.

Ошибки округления. Еще одно свойство алгоритма заключается в чувствительности к ошибкам округления в машинной арифметике.

Прежде всего заметим, что требование центрированности $\sum x_i^0 = \sum y_i^0 = 0$ является существенным (будем называть число $\frac{1}{n} \sum x_i$ центроидом вектора x). Это требование гарантирует, что в разложении начальных векторов x^0 и y^0 по собственным векторам e_1, \dots, e_n матрицы M коэффициент при e_1 (отвечающем максимальному собственному значению $\lambda_1 = 1$), равен нулю. Как известно (например, см. [3, 4]), в этих условиях степенной метод *не* сходится, в результате система точек и проявляет указанное поведение. В против-

ном случае степенной метод будет сходиться к вектору e_1 (так как кратность λ_1 равна единице). Можно показать, что e_1 имеет вид $e_1 = (a, \dots, a)^T$, где a — число, поскольку собственные векторы определены с точностью до скалярного множителя. Таким образом, если x^0 и y^0 не центрированы, то, в зависимости от нормы в (3), будет наблюдаться сходимость итераций к предельному вектору $(\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})^T$ для евклидовой нормы, к $(\frac{1}{n}, \dots, \frac{1}{n})^T$ для l_1 -нормы, к $(1, \dots, 1)^T$ для l_∞ -нормы и т. д. Наконец, нетрудно видеть, что знак предельного вектора (знак a) совпадает со знаком центроида начального вектора. Иными словами, вся совокупность точек p_i сходится к одной из точек $(\pm \frac{1}{\sqrt{n}}, \pm \frac{1}{\sqrt{n}})^T$ (для евклидовой нормы).

Ошибки округления при выполнении матрично-векторных операций приводят к тому же эффекту: после достаточно большого числа шагов алгоритма центроид изначально центрированного вектора перестает быть равным нулю, и итерации сходятся к собственному вектору e_1 . На рис. 2 показано изменение (изначально нулевой) величины центроида вектора x^k для системы точек из примера на рис. 1 (поведение центроида y^k аналогично) и, как следствие, сходимость совокупности p_i к единственной точке. Чтобы избежать

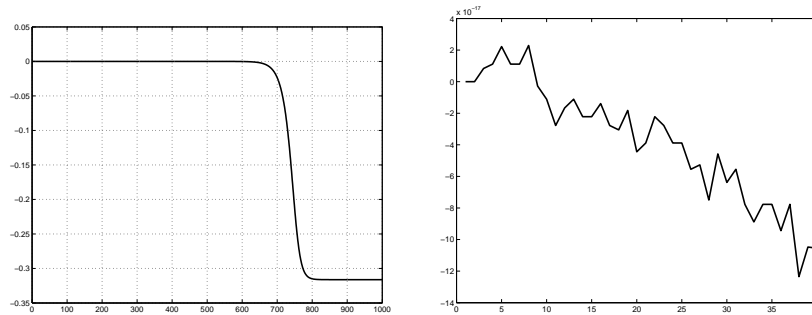


Рис. 2: Потеря центрированности и сходимость к e_1 (справа: увеличенный фрагмент первых 40 итераций).

этого эффекта и сохранить эллипсоидальное поведение, достаточно раз в несколько итераций проводить центрирование.

Сдвиг. Кратко рассмотрим следующую простейшую модификацию алгоритма:

$$x^{k+1} = Mx^k / \|Mx^k\| + \mathbf{1}u, \quad x^{k+1} = x^{k+1} - \mathbf{1} \frac{1}{n} \sum_{i=1}^n x_i^{k+1}; \quad (4)$$

$$y^{k+1} = My^k / \|My^k\| + \mathbf{1}v, \quad y^{k+1} = y^{k+1} - \mathbf{1} \frac{1}{n} \sum_{i=1}^n y_i^{k+1}; \quad (5)$$

где $u, v \in \mathbb{R}$ фиксированы, а $\mathbf{1} \in \mathbb{R}^n$ — вектор из единиц; кроме того, на каждом шаге производится центрирование. Внешние входы $\mathbf{1}u, \mathbf{1}v$ могут интерпретироваться как шум, управление, задающий сигнал и т. д. Очевидно, что добавление членов $\mathbf{1}u, \mathbf{1}v$ и центрирование — взаимно-обратные операции, поэтому траектории точек p_i остаются теми же самыми. Более содержательный пример алгоритма такого сорта будет рассмотрен ниже в разделе 3.

Обобщение на многомерный случай. Наиболее интересным представляется возможность обобщения схемы Ван Лоуна на случай, когда n точек $p_i^0 = \underbrace{(x_i^0, y_i^0, \dots, z_i^0)}_{d \text{ компонент}}^T, i = 1, \dots, n$, расположены не на плоскости, а в d -мерном пространстве.

В этом случае эволюция описывается d -кратным применением степенного метода:

$$\begin{aligned} x^{k+1} &= Mx^k / \|Mx^k\|; \\ y^{k+1} &= My^k / \|My^k\|; \\ &\vdots \\ z^{k+1} &= Mz^k / \|Mz^k\|, \end{aligned} \quad (6)$$

и на первый взгляд предельное расположение совершенно неочевидно. Однако ясно, что, как и в двумерном случае, все d сборных векторов x, y, \dots, z меняются независимо по одному и тому же алгоритму, что и ранее. Поэтому проекция предельного расположения на любую из координатных плоскостей является эллипсом, задаваемым матрицей типа (1). Эквивалентным каноническому описанию эллипса (2) является описание с помощью матрицы A как линейного преобразования окружности:

$$\mathcal{E} = \left\{ p \in \mathbb{R}^2 : p = A \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \right\},$$

когда φ пробегает $[0, 2\pi]$. Следовательно, в d -мерном случае матрица преобразования имеет вид

$$A_d = \sqrt{\frac{2}{n}} \begin{pmatrix} \frac{c^T x^0}{\sqrt{(c^T x^0)^2 + (s^T x^0)^2}} & \frac{s^T x^0}{\sqrt{(c^T x^0)^2 + (s^T x^0)^2}} \\ \frac{c^T y^0}{\sqrt{(c^T y^0)^2 + (s^T y^0)^2}} & \frac{s^T y^0}{\sqrt{(c^T y^0)^2 + (s^T y^0)^2}} \\ \vdots & \vdots \\ \frac{c^T z^0}{\sqrt{(c^T z^0)^2 + (s^T z^0)^2}} & \frac{s^T z^0}{\sqrt{(c^T z^0)^2 + (s^T z^0)^2}} \end{pmatrix} \in \mathbb{R}^{d \times 2}. \quad (7)$$

Соответственно, $S = (A_d A_d^T)^{1/2}$ является $d \times d$ -матрицей ранга 2, у которой $d - 2$ собственных значения нулевые и которая, таким образом, определяет двумерный (“плоский”) предельный эллипс в d -мерном пространстве точек!

3. Простые линейные алгоритмы

Как отмечалось выше, схема Ван Лоуна предполагает высокую информированность агентов; кроме того, алгоритм (3) нелинейный, и его непросто анализировать. В этом разделе приведем несколько простых алгоритмов в дискретном времени, требующих гораздо меньшей априорной информации; очень близкие алгоритмы для непрерывного времени рассматриваются в [6].

Отрезок. Рассмотрим простейшую одномерную схему. На прямой даны n точек; каждая x_i , $i = 2, \dots, n - 1$ имеет информацию о x_{i-1} и x_{i+1} ; точка x_1 имеет информацию о некоторой фиксированной точке a и об x_2 ; наконец, x_n — о точке x_{n-1} и фиксированной точке $b > a$. Задача состоит в том, чтобы расположить точки на отрезке $[a, b]$ на равных расстояниях друг от друга.

Предлагаемый алгоритм эволюции таков:

$$\begin{aligned} x_i^{k+1} &= (x_{i-1}^k + x_{i+1}^k)/2, \quad i = 2, \dots, n - 1; \\ x_1^{k+1} &= (a + x_2^k)/2; \quad x_n^{k+1} = (x_{n-1}^k + b)/2, \end{aligned} \quad (8)$$

т. е. новое положение точки равно среднему арифметическому двух ее соседей (считая точку a соседом для x_1 , и точку b соседом для x_n). Здесь, как и ранее, соседними называются точки, чьи номера отличаются на единицу.

Добавим к вектору состояний $x = (x_1, \dots, x_n)$ две крайние точки a и b и обозначим через $\tilde{x}^0 = (a, x_1^0, \dots, x_n^0, b)^T \in \mathbb{R}^{n+2}$ рас-

ширенный вектор начального расположения. Введя в рассмотрение матрицу

$$M = \begin{pmatrix} 1 & 0 & & & \\ 0.5 & 0 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.5 & 0 & 0.5 \\ & & & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+2) \times (n+2)}, \quad (9)$$

для расширенного вектора можем записать шаг алгоритма (8) как

$$\tilde{x}^{k+1} = M\tilde{x}^k,$$

при этом две искусственно введенные точки $\tilde{x}_1 = a$ и $\tilde{x}_{n+2} = b$ остаются фиксированными.

Алгоритм линеен по переменным \tilde{x} , и анализ его сходимости прост. В самом деле, поскольку $\tilde{x}^k = M^k \tilde{x}^0$, то остается понять поведение степеней матрицы M . Непосредственными несложными, но долгими выкладками можно показать, что при $k \rightarrow \infty$ имеем:

$$M^k \rightarrow \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \frac{n}{n+1} & \vdots & & \vdots & \frac{1}{n+1} \\ \frac{n-1}{n+1} & & & \frac{2}{n+1} & \\ \vdots & & & \vdots & \\ \frac{1}{n+1} & & & \frac{n}{n+1} & \\ 0 & \dots & & 0 & 1 \end{pmatrix} \doteq M_{lim}, \quad (10)$$

так что

$$M_{lim} \begin{pmatrix} a \\ x_1^0 \\ \vdots \\ \vdots \\ x_n^0 \\ b \end{pmatrix} = \begin{pmatrix} a \\ \frac{n}{n+1}a + \frac{1}{n+1}b \\ \frac{n-1}{n+1}a + \frac{2}{n+1}b \\ \vdots \\ \frac{1}{n+1}a + \frac{n}{n+1}b \\ b \end{pmatrix} = \begin{pmatrix} a + \frac{1}{n+1}(b-a) \\ a + \frac{2}{n+1}(b-a) \\ \vdots \\ a + \frac{n}{n+1}(b-a) \\ b \end{pmatrix}$$

для любого начального вектора x^0 . Таким образом, алгоритм обладает глобальной сходимостью к единственному предельному рас-

положению на отрезке $[a, b]$, при котором точки выстроены по порядку номеров, и расстояния $x_1 - a, x_2 - x_1, \dots, x_n - x_{n-1}, b - x_n$ одинаковы и равны $(b - a)/(n + 1)$.

Отметим, что алгоритм не предполагает знания количества n точек, но лишь координат концов отрезка и минимальную информированность агентов — о координатах двух соседей.

Если в качестве фиксированных концов отрезка брать не некоторые заданные a и b , а начальные положения первой и последней точек x_1^0 и x_n^0 , то алгоритм остается ровно тем же самым, с той лишь разницей, что матрица M будет иметь размерность $n \times n$, и при этом точки расположатся равномерно на отрезке $[x_1^0, x_n^0]$.

Отрезок: заданное отношение расстояний. Пусть теперь в предыдущей схеме (с закрепленными концами a и b) требуется расставить точки так, чтобы отношение расстояний между ними стало заданным: $\lambda_1 : \lambda_2 : \dots : \lambda_n : \lambda_{n+1}$. Тогда вместо среднего арифметического — как в алгоритме (8) — в качестве нового положения точки будем брать взвешенную сумму координат ее соседей:

$$\begin{aligned} x_i^{k+1} &= \frac{\lambda_{i+1}}{\lambda_i + \lambda_{i+1}} x_{i-1}^k + \frac{\lambda_i}{\lambda_i + \lambda_{i+1}} x_{i+1}^k, \quad i = 2, \dots, n-1; \\ x_1^{k+1} &= \frac{\lambda_2}{\lambda_1 + \lambda_2} a + \frac{\lambda_1}{\lambda_1 + \lambda_2} x_2^k; \\ x_n^{k+1} &= \frac{\lambda_{n+1}}{\lambda_n + \lambda_{n+1}} x_{n-1}^k + \frac{\lambda_n}{\lambda_n + \lambda_{n+1}} b. \end{aligned}$$

Как и ранее, вводя расширенный вектор $\tilde{x} = (a, x_1, \dots, x_n, b)^T \in \mathbb{R}^{n+2}$ и матрицу

$$M = \begin{pmatrix} 1 & 0 & & & & & \\ \frac{\lambda_2}{\lambda_1 + \lambda_2} & 0 & \frac{\lambda_1}{\lambda_1 + \lambda_2} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \frac{\lambda_{n+1}}{\lambda_n + \lambda_{n+1}} & 0 & \frac{\lambda_n}{\lambda_n + \lambda_{n+1}} & & \\ & & & 0 & 1 & & \end{pmatrix},$$

запишем алгоритм в том же виде $\tilde{x}^{k+1} = M\tilde{x}^k$, так что $\tilde{x}^k = M^k \tilde{x}^0$, и сходимость итераций к требуемому предельному расположению

заключается в весах $1/3$ и $2/3$. Они нужны для того, чтобы при “сшивании” отрезка обратно в окружность расстояние между первой и последней точками θ_1 и θ_n (т. е. вокруг “невидимой” точки шва) равнялось остальным; в случае равных весов 0.5 это расстояние оказалось бы вдвое бóльшим. Сходимость такого алгоритма к равномерному расположению точек на окружности анализируется так же, как и раньше.

Можно немного ослабить требования к начальному расположению: предполагая его произвольным и считая, что каждая точка знает свои координаты, достаточно произвести нормировку $p_i \rightarrow p_i/\|p_i\|$.

В рассмотренной схеме для окружности разрез можно производить по одной из точек набора $\theta_1, \dots, \theta_n$, выделяя ее как “ведущую”. Естественно в качестве такой точки выбрать первую.

В этом случае обозначим $\tilde{\theta} = (\theta_1, \dots, \theta_n, \theta_1 + 2\pi)^T \in \mathbb{R}^{n+1}$, введем матрицу

$$M = \begin{pmatrix} 1 & 0 & & & \\ 0.5 & 0 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.5 & 0 & 0.5 \\ & & & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}. \quad (11)$$

и запишем шаг алгоритма:

$$\tilde{\theta}^{k+1} = M\tilde{\theta}^k. \quad (12)$$

В таком алгоритме закрепленной точкой является θ_1 , а его анализ идентичен предыдущим.

Окружность: движение. По аналогии с (4)–(5), добавим к динамике (12)–(11) внешнее воздействие:

$$\tilde{\theta}^{k+1} = M\tilde{\theta}^k + \mathbf{1}\omega, \quad (13)$$

где $\omega \in \mathbb{R}$ — постоянная величина, а $\mathbf{1}$ — вектор из единиц. Нетрудно видеть, что в результате работы такого алгоритма точки стремятся выстроиться на равных углах и двигаться со скоростью ω . Действительно, поскольку $M\mathbf{1}\omega = \mathbf{1}\omega$, то имеем

$$\tilde{\theta}^k = M^k\tilde{\theta}^0 + \sum_{i=0}^{k-1} M^i\mathbf{1}\omega = M^k\tilde{\theta}^0 + k\mathbf{1}\omega,$$

поэтому $\tilde{\theta}^{k+1} - \tilde{\theta}^k = (M^{k+1} - M^k)\tilde{\theta}^0 + \mathbf{1}\omega$ и при достаточно больших k будет $\tilde{\theta}^{k+1} - \tilde{\theta}^k \approx \mathbf{1}\omega$, а векторы $\tilde{\theta}^k$ и $\tilde{\theta}^{k+1}$ задают равномерное расположение.

Для частного случая начального расположения $\theta_1^0 < \theta_2^0 < \dots < \theta_n^0$, исключающего “перепрыгивание” точек друг через друга, алгоритму (13) можно дать мультиагентную интерпретацию: автомобили, стоящие в круговом тоннеле по порядку номеров, должны выстроиться на равных расстояниях и двигаться со скоростью ω . Каждому агенту в такой схеме известны лишь требуемая скорость движения и координаты соседей, но не общее число агентов. Возможна модификация алгоритма, когда скорость известна лишь ведущему агенту θ_1 .

4. Заключение и дальнейшие исследования

В работе рассмотрен простой алгоритм управления расположением точек на плоскости и некоторые его модификации, которыми, разумеется, не исчерпываются все возможности схемы Ван Лоуна. Например, к любопытным анимационным картинкам эволюции приводит изменение знака правого нижнего элемента матрицы M .

Отметим, что схема Ван Лоуна может формулироваться в непрерывном времени; в этом случае вместо разностных уравнений (A) появятся соответствующие дифференциальные уравнения.

В проведенном исследовании схема Ван Лоуна послужила автору стимулом к изучению и разработке новых алгоритмов управления формациями из раздела 3. Дальнейшие исследования предполагается вести в нескольких направлениях. Прежде всего сюда относится создание и анализ линейных алгоритмов, например, типа (13), использующих минимум априорной информированности агентов и имеющих прозрачное предметное происхождение.

Алгоритмы раздела 3, в том числе и для случая окружности — одномерные. Большой интерес представляет разработка многомерных модификаций, например, приводящих трехмерные точки из произвольного начального расположения на сферу или в иную заданную трехмерную конфигурацию.

В рассмотренных схемах управляющий сигнал не присутствует в явном виде, и под управлением (по полному вектору состояний, как в схеме Ван Лоуна, или по части переменных, как в разделе 2) понимается собственно алгоритм, задающий эволюцию то-

чек. Представляется интересным исследовать альтернативные схемы, явно учитывающие наличие некоторого задающего внешнего воздействия.

Автор признателен Б.Т. Поляку за привлечение внимания к вопросам управления формациями, в частности, к работе [1], и за плодотворные обсуждения.

Список литературы

- [1] *Elmachtoub A.N., Van Loan C.F.* From random polygon to ellipse: an eigenanalysis // SIAM Review. 2010. Vol. **52**. No. 1. P. 151–170.
- [2] *Shoham Y., Leyton-Brown K.* Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. — Cambridge University Press. 2009.
- [3] *Амосов А.А., Дубинский Ю.А., Копченова Н.В.* Вычислительные методы для инженеров. — М: Высшая школа. 2008.
- [4] *Тыртмышников Е.Е.* Методы численного анализа. — М: Издательский центр “Академия”. 2007.
- [5] *Голуб Дж., Ван Лоан Ч.* Матричные вычисления. — М.: Мир, 1993.
- [6] *Петрикевич Я.И.* Линейные алгоритмы управления геометрическим расположением объектов в многоагентной системе // Управление большими системами. 2010.