

УДК 519.876 + 519.71 + 51-77

ББК 78.34

УПРАВЛЕНИЕ ФОРМАЦИЯМИ: СХЕМА ВАН ЛОУНА И ДРУГИЕ АЛГОРИТМЫ ¹

Щербаков П. С. ²

*(Учреждение Российской академии наук Институт проблем
управления им. В. А. Трапезникова РАН, Москва)*

Рассматривается разработанный в [5] алгоритм эволюции совокупности точек на плоскости, приводящий её 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: управление формациями, степенной метод, линейные алгоритмы.

Введение

В приложениях часто возникают задачи, которые можно грубо формализовать следующим образом. Имеются 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

¹ Автор признателен Б. Т. Поляку за привлечение внимания к вопросам управления формациями, в частности, к работе [5], и за плодотворные обсуждения.

² Павел Сергеевич Щербаков, доктор физико-математических наук (sherba@ipu.ru).

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 решения, приложений, является [6], 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 [5], обсудить некоторые его новые свойства и обобщения, а также предложить альтернативные, более простые линейные алгоритмы.

1. Схема Ван Лоуна

В работе [5] исследовалась следующая задача, которую назовем схемой Ван Лоуна.

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

Предлагается следующий алгоритм управления движением совокупности $\{p_i\}_1^n$. Обозначим через $p_i^k = (x_i^k, y_i^k)$ координаты i -ой точки на k -ом шаге алгоритма и введем «сборные» векторы $x^k = (x_1^k, \dots, x_n^k)^\top$ и $y^k = (y_1^k, \dots, y_n^k)^\top$. Шаг алгоритма состоит

из двух этапов и устроен следующим образом:

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

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

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 [5] получен следующий неожиданный результат.

Теорема 1 [5]. Пусть начальное расположение точек таково, что $\sum x_i^0 = \sum y_i^0 = 0$. Тогда при $k \rightarrow \infty$ они стремятся расположиться на эллипсе с центром в нуле и повернутом на $\pi/4$. Матрица S этого предельного эллипса вычисляется следующим образом. Введем векторы

$$c = \begin{pmatrix} \cos(0) \\ \cos(2\pi/n) \\ \cos(4\pi/n) \\ \vdots \\ \cos(2\pi(n-1)/n) \end{pmatrix}; \quad s = \begin{pmatrix} \sin(0) \\ \sin(2\pi/n) \\ \sin(4\pi/n) \\ \vdots \\ \sin(2\pi(n-1)/n) \end{pmatrix}$$

и составим 2×2 матрицу

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

Предельный эллипс имеет вид

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

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.

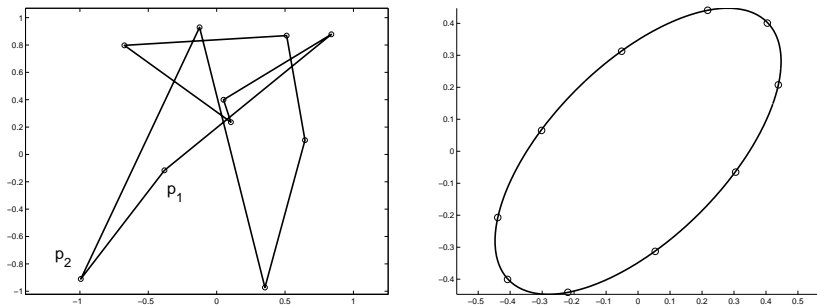


Рис. 1. Некоторое начальное расположение; расположение после $k = 40$ шагов ($n = 10$)

1.1. 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

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

что является не чем иным, как независимым применением степенного метода для матрицы M при начальных условиях x^0 и y^0 , например, см. [1, 3, 2]. Это соображение лежит в основе подробного анализа алгоритма, проведенного в [5]; соответственно, и аппарат исследования опирается лишь на общие сведения из линейной алгебры.

В частности, в [5] 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 $|(p_i^k)^\top S^{-1} p_i^k - \varepsilon| \leq 1$, 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 [5] 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 [5] 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.

1.2. НЕКОТОРЫЕ НОВЫЕ СВОЙСТВА И ОБОБЩЕНИЯ

Равномерность предельного расположения. В [5] 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 чем перейти к описанию этого явления, обсудим некоторые свойства степенного метода (3).

Заметим, что требование центрированности $\sum x_i^0 = \sum y_i^0 = 0$ является существенным в Теореме 1 (ниже 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 (который отвечает максимальному собственному значению λ_1), равен нулю. Как известно (например, см. [1, 3]), в этом случае степенной метод (3) *не* сходится к фиксированному вектору; как следствие, система точек $\{p_i\}_1^n$ в схеме Ван Лоуна и проявляет указанное поведение.

Из стандартного анализа степенного метода следует, что при отсутствии центрированности каждый из начальных векторов x^0 , y^0 имеет ненулевую компоненту в направлении e_1 , и степенной метод *будет* сходиться, причем к вектору e_1 , — при дополнительном условии, что кратность доминирующего собственного значения λ_1 равна единице.

Можно показать, что для матрицы M доминирующее собственное значение равно единице и имеет кратность 1, а соответствующий ему собственный вектор 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 используемой векторной нормы, будет наблюдаться сходимость итераций (3) к предельному вектору $e_1 = (\frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}})^T$ для евклидовой нормы ($\|e_1\|_2 = 1$), к $e_1 = (\frac{1}{n}, \dots, \frac{1}{n})^T$ для l_1 -нормы ($\|e_1\|_1 = 1$), к $e_1 = (1, \dots, 1)^T$ для l_∞ -нормы ($\|e_1\|_\infty = 1$) 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^k к единственной точке.

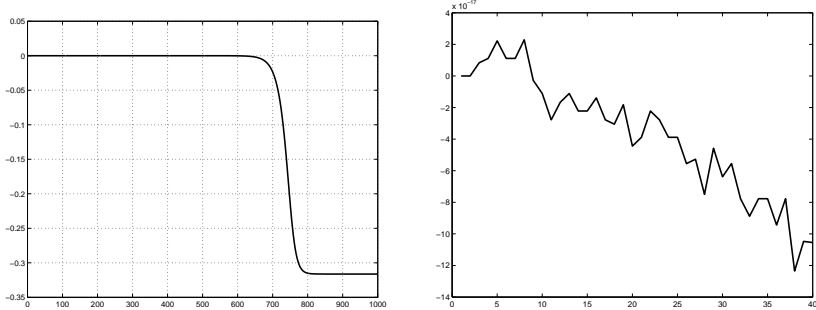


Рис. 2. Потеря центрированности и сходимость к e_1 (справа: увеличенный фрагмент первых 40 итераций)

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

Сдвиг. Кратко рассмотрим следующую простейшую модифи-

кацию алгоритма:

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

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

где $u, v \in \mathbb{R}$ фиксированы, а $\mathbf{1} \in \mathbb{R}^n$ — вектор из единиц; т. е. к динамике системы добавлен постоянный внешний вход и, кроме того, на каждом шаге производится центрирование. 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 2.

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

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

$$(6) \quad x^{k+1} = \frac{Mx^k}{\|Mx^k\|}; \quad y^{k+1} = \frac{My^k}{\|My^k\|}, \dots, \quad z^{k+1} = \frac{Mz^k}{\|Mz^k\|},$$

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 каждую из координатных плоскостей является эллипсом, задаваемым матрицей типа (1). Эквивалентным каноническому описанию эллипса (2) является описание с помощью матрицы A (1) 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 мат-

рица преобразования имеет вид

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

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!

2. 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 [4].

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:

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

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.

так что

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

для любого начального вектора x^0 . Таким образом, алгоритм обладает глобальной сходимостью к единственному предельному расположению 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 $(b - a)/(n + 1)$.

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:

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

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 обеспечивается сходимостью степеней матрицы M :

$$M^k \rightarrow \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \frac{\lambda_1 + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_{n+1}} & \vdots & & \vdots & \frac{\lambda_1}{\lambda_1 + \dots + \lambda_{n+1}} \\ \frac{\lambda_1 + \dots + \lambda_{n-1}}{\lambda_1 + \dots + \lambda_{n+1}} & & & \frac{\lambda_1 + \lambda_2}{\lambda_1 + \dots + \lambda_{n+1}} & \\ \vdots & & & \vdots & \\ \frac{\lambda_1}{\lambda_1 + \dots + \lambda_{n+1}} & & & \frac{\lambda_1 + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_{n+1}} & \\ 0 & \dots & 0 & & 1 \end{pmatrix} \doteq M_{\text{lim}}.$$

Окружность. Описанный выше алгоритм (8) легко модифицируется для следующей ситуации. На окружности с центром в нуле даны n пронумерованных точек p_i , положение которых определяется их углами $\theta_i \in (0, 2\pi)$. Каждой точке известны углы соседей (по номерам) и положение центра окружности. Задача — та же: расставить точки равномерно на окружности.

Один из возможных алгоритмов очевиден: «разрезать» окружность в некоторой произвольной точке $\theta_c \in [0, 2\pi]$ и применить алгоритм для отрезка.

Действительно, обозначим $\tilde{\theta} = (\theta_c, \theta_1, \dots, \theta_n, \theta_c + 2\pi)^T \in$

введем матрицу

$$(12) \quad M = \begin{pmatrix} 1 & 0 & & & \\ 0,5 & 0 & 0,5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0,5 & 0 & 0,5 \\ & & & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}.$$

и запишем шаг алгоритма:

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

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

Окружность: движение. По аналогии с (4)–(5), добавим к динамике (12)–(13) внешнее воздействие:

$$(14) \quad \tilde{\theta}^{k+1} = M\tilde{\theta}^k + \mathbf{1}\omega,$$

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

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

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 (14) 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 .

3. 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 (А) появятся соответствующие дифференциальные уравнения.

В проведенном исследовании схема Ван Лоуна послужила автору стимулом к изучению и разработке новых алгоритмов управления формациями из раздела 2. Дальнейшие исследования предполагается вести в нескольких направлениях. Прежде всего сюда относится создание и анализ линейных алгоритмов, например, типа (14), использующих минимум априорной информированности агентов и имеющих прозрачное предметное происхождение.

Алгоритмы раздела 2, 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 точек. Представляется интересным исследовать альтернативные схемы, явно учитывающие наличие некоторого задающего внешнего воздействия.

Литература

1. АМОСОВ А.А., ДУБИНСКИЙ Ю.А., КОПЧЕНОВА Н.В., *Вычислительные методы для инженеров.* – М:

- Высшая школа, 2008.
2. ГОЛУБ ДЖ., ВАН ЛОАН Ч., *Матричные вычисления*. – М.: Мир, 1993.
 3. ТЫРТЫШНИКОВ Е. Е., *Методы численного анализа*. – М: Издательский центр “Академия”, 2007.
 4. ПЕТРИКЕВИЧ Я. И., *Линейные алгоритмы управления геометрическим расположением объектов в многоагентной системе // Управление большими системами. Специальный выпуск 30.1 «Сетевые модели в управлении»*. М.: ИПУ РАН, 2010. С. 665–680.
 5. ELMACHTOUB A. N., VAN LOAN C. F., *From Random Polygon to Ellipse: An Eigenanalysis // SIAM Review*. – 2010. – Vol. 52, No. 1. – P. 151–170.
 6. SHOHAM Y., LEYTON-BROWN K., *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. – Cambridge University Press, 2009.

FORMATION CONTROL: THE VAN LOAN SETUP AND OTHER ALGORITHMS

Pavel Shcherbakov, Institute of Control Sciences of RAS, Moscow,
Doctor of Science (sherba@ipu.ru).

Abstract: The subject of this note is the algorithm of evolution of a point set on the plane devised in [5], which drives the whole system to a certain regular configuration. Generalizations of the algorithm are analyzed, certain new properties are studied, the connection to formation control methods is discussed, and new simpler algorithms of this sort are proposed.

Keywords: formation control, power iterations, linear algorithms.

*Статья представлена к публикации
членом редакционной коллегии Б. Т. Поляком*