Математическая Теория Игр и её Приложения, т.5, в.1, с. 45–60

УДК 519.837.3
ББК 22.18

АЛГОРИТМ НАХОЖДЕНИЯ РАВНОВЕСИЙ В ДИНАМИЧЕСКОЙ СЕТЕВОЙ ИГРЕ

Андрей П. Парфенов
Факультет прикладной математики – процессов управления
Санкт-Петербургский государственный университет
198504, Санкт-Петербург, Университетский просп., 35
e-mail: parf@bk.ru

Вводится обобщение игры формирования сети и многошаговой игры с одновременными ходами игроков. Это динамическая сетевая игра с двумя видами дуг: ребрами взаимодействия и динамическими дугами. Построен алгоритм нахождения равновесий по Нэшу в динамической сетевой игре, являющийся модификацией рекуррентного алгоритма для многошаговой игры и использующий свойства ребер взаимодействия. Дана оценка сложности алгоритма. Построен пример, показывающий преимущество модифицированного алгоритма по сравнению с обычным рекуррентным алгоритмом для многошаговой игры.

Ключевые слова: сетевые игры, динамические игры, позиционные игры, равновесия по Нэшу, алгоритмы оптимизации, сложность алгоритмов.

1. Параметрические сетевые игры

Игры формирования сетей – класс сетевых игр [1], в которых игроки образуют вершины графа, а параметры дуг графа определяются согласованными стратегиями игроков. Автором данной работы был введен [4] класс игр формирования сетей, обобщающий игры формирования неориентированных графов, ориентированных графов [1], игры формирования графов с нагруженными вершинами и дугами [2] и игры в нормальной форме:

Определение 1.1. Параметрическая сетевая игра – *это сетевая игра* $\Gamma_{S,U} = (N, ((S_i)_{i\in N}, (U_{(i,j)})_{i,j\in N}, (U_i(s_i))_{i\in N, s_i\in S_i}), G, g, (H_i)_{i\in N})$, *где*

- $N = \{1, \dots, n\}$ – *множество игроков;*
- S_i – *абстрактное множество* состояний i-го игрока*;*
- $U_{(i,j)}$ – *множество состояний связи между* i*-м и* j*-м игроком, являющееся нижней полурешеткой с нулем (т.е. для каждой пары* $a, b \in U_{(i,j)}$ *задается «инфимум»* $\inf(a, b) \in U_{(i,j)}$*);*
- $U_i(s_i)$ – *множество возможных наборов* $(u_{i1},\dots, u_{i,i-1}, u_{i,i+1},\dots, \dots, u_n), u_{ij} \in U_{(i,j)}$ *состояний связей* i*-го игрока с другими игроками;*
- *множество* X_i *стратегий* i*-го игрока – это множество наборов* $x_i = (s_i \in S_i,\ u_i = (u_{i1}, \dots u_{i,i-1}, u_{i,i+1}, \dots, u_{in}))$ *таких, что* $u_i \in U_i(s_i)$*;*
- $G(S, U)$ – *множество возможных сетей с состояниями вершин* $S = (S_i)_{i\in N}$ *и ребер* $U = (U_{(i,j)})_{i,j\in N}$*;*
- *матрица смежности* $g(x)$ *получившейся сети определяется по формуле*
$$g(x) = \inf(x, x^T),$$
где x *– матрица, строки которой – стратегии игроков (на диагонали стоят состояния игроков), а* $\inf$ *– поэлементная операция над матрицами (на диагонали полагаем для произвольного множества* $\inf(z, z) = z$*);*

- $H_i \colon G(S, U) \to \mathbb{R}$ – *произвольная функция выигрыша i-го игрока, определенная на множестве сетей.*

Можно интерпретировать параметрическую сетевую игру так: каждый игрок выбирает состояние своей вершины и предлагает некоторым другим игрокам построить ребра с определенными состояниями. При реализации ситуации $(s_i, u_i)_{i \in N}$ возникает граф (N, M) с нагруженными вершинами и дугами, где вершина i нагружена состоянием s_i i-го игрока. Если игрок i реализует стратегию $u_i(j)$, а игрок j – стратегию $u_j(i)$, то в графе появляется дуга (i, j), нагруженная значением $u_{ij} = \inf(u_i(j), u_j(i))$, тогда и только тогда, когда $\inf(u_i(j), u_j(i)) \neq 0$. В результате игроки получают выигрыши $H_i(g)$, зависящие от сети $g \in G(S, U)$.

Будем называть параметрическую сетевую игру *игрой согласия* (обобщение аналогичного понятия для обычной игры формирования графа [1]), если из $x_i \in X_i$ $(x_i = (u_{i1}, \ldots, u_{in}))$ и $x'_i \leq x_i$ (в смысле покомпонентного частичного порядка на полурешетках, элементы $x_{ii} = s_i$ считаем сравнимыми, только если они равны) следует $x'_i \in X_i$. То есть, если игрок i имеет право предложить игроку j образовать связь с большими параметрами, то имеет право предложить образовать связь и с меньшими параметрами.

Поскольку параметрическая сетевая игра – не что иное, как сетевая игра на мультиграфе, определения достижимой сети и сети, стабильной по Нэшу [1], для параметрической игры абсолютно аналогичны таковым для обычной сетевой игры координации. Также для параметрической игры согласия верны свойства [3]:

1. Если ситуация x является равновесием Нэша и приводит к результирующей сети g, то ситуация x', в которой $x' = x'^T = g$, также будет равновесием Нэша.

2. Сеть $g \in G(S, U)$ стабильна по Нэшу тогда и только тогда, когда ситуация, в которой $x' = x'^T = g$, является равновесием Нэша.

3. Сеть $g \in G(S, U)$ стабильна по Нэшу тогда и только тогда, когда любой из игроков не может выиграть от уменьшения (в смысле частичного порядка на нижней полурешетке) любого количества параметров своих входящих или исходящих связей.

4. Для любого игрока i существует такая максиминная ситуация, что для любой дуги (i,j) $(i,j) \notin M$ (игрок i изолирован от остальных) и для любой дуги $(j,k), j,k \neq i$, $u_j(k) = u_k(j)$ (остальные игроки тоже согласуют свои действия).

Эти свойства позволяют искать оптимальные решения в параметрических играх, перебирая не возможные стратегии игроков, а возможные сети, что существенно сокращает количество вариантов при переборе (количество сетей – порядка квадратного корня из количества ситуаций). Каждую сеть можно проверять на оптимальность двумя способами: проверяя все возможные отклонения каждого игрока, либо проверяя все возможные сети и определяя результатом каких отклонений они могут быть. Аналогично непараметрическим сетевым играм, в параметрических сетевых играх можно найти [3] стабильные сети (а значит, и равновесные выигрыши) за время порядка $O(|G(S,U)|(|S_1|\prod\limits_j |U_{(j,1)}| + \ldots + |S_n|\prod\limits_j |U_{(j,n)}|))$. Если мы нашли k стабильных сетей, можно найти все равновесия, перебирая в каждой стабильной сети порядка $\prod\limits_{(i,j)\in M} q(U_{(i,j)})$ отклонений, не меняющих значение в ребре сети, где $q(U_{(i,j)}) = \max\limits_{m \in U_{(i,j)}} |\{(a,b) \in$ $\in U_{(i,j)}^2 \mid \inf(a,b) = m\}|$. Например, если $U_{(i,j)}$ – линейно упорядоченное множество, то $q(U_{ij}) = 2|U_{(i,j)}| - 1$. Максимальное возможное количество пар элементов – очевидно, в полурешетке, в которой для любых двух неравных элементов $\inf(a,b) = 0$ – тогда $q(U_{(i,j)}) =$ $= |U_{(i,j)}|^2 - |U_{(i,j)}| + 1$. Соответственно, все равновесия можно найти за время $O(k\prod_{(i,j)\in M} q((U_{(i,j)})(|S_1|\prod\limits_j |U_{(j,1)}| + \ldots + |S_n|\prod\limits_j |U_{(j,n)}|))$. Максиминную стратегию можно найти за время порядка $O(|G(S,U)|)$.

2. Динамические сетевые игры

Определение динамической сетевой игры опирается на общее определение многошаговой игры, которое мы дадим в соответствии с [7] с незначительными модификациями:

Определение 2.1. *Многошаговая игра – это набор*

$$\Gamma = (N, X, T, x_0, (D, U, \pi), i, \{U^j\}_{j\in N}, \mathbb{S}, \{H^j\}_{j\in N}),$$

где

- $N = \{1, \dots, n\}$ – *множество игроков;*
- X – *пространство игры;*
- $T = \{t_1, t_2, \dots, t_m\}$ – *время игры, т.е. множество моментов времени;*
- $x_0 \in X$ – *начальная позиция;*
- (D, U, π) – *управляемая динамическая система с дискретным временем* T *на множестве состояний* X *и с управлениями из множеств* $U(x,t)$, *т.е.* $D\colon X \times T \to 2^X$ – *отображение, определяющее возможные состояния в следующий момент времени, а* $\pi\colon X \times T \times U \to X$ – *функция, сопоставляющая состоянию* $x(t)$ *и управлению* $u \in U(x(t), t)$ *состояние в следующий момент времени* $x(t+1) = \pi(x(t), t, u(x(t), t)) \in D(x(t), t)$*;*
- $i\colon X \times T \to 2^N$ – *функция* $i(x,t)$ *определяет игроков, делающих ход в состоянии* x *в момент времени* t*;*
- $U_j(x,t), j \in i(x,t)$ – *множество управлений* j*-го игрока в точке* (x,t) *такое, что* $U(x,t) = \prod\limits_{j \in i(x,t)} U_j(x,t)$*;*
- *Множество ситуаций* $\mathbb{S} = \prod\limits_{j \in N} \mathbb{S}_j$, *где* $\mathbb{S}_j$ – *подмножество множества стратегий* j*-го игрока, т.е. семейств отображений вида* $\{s_j\colon p(x_0, x) \to U_j(x, l(p))\}_{x \in X}$, *где* $p(x_0, x)$ – *множество траекторий, выходящих из начальной точки* x_0 *и заканчивающихся в точке* x*;* $l(p)$ – *длина траектории* p *(момент времени, в который она заканчивается);*
- $H^i(x_0, x_1, \dots, x_m) = h^i_1(x_1) + \ldots + h^i_m(x_m)$ – *интегральная функция выигрыша* i*-го игрока.*

Определение 2.2. *Динамическая сетевая игра – это многошаговая игра с одновременными ходами, в которой*

- *Пространство игры – это множество сетей* $G = (N, M)$, $M = M^p \cup M^d$, *с нагруженными вершинами и нагруженными дугами двух видов: ребрами взаимодействия* M^p *(определяющих* *связи между игроками) и динамическими дугами M^d (определяющих влияние одного игрока на другого);*

- *Время игры – это множество $T = \{0, 1, \ldots, m\}$, где 0 – условный момент времени, для которого задано фиксированное условное состояние x_0 (оно нужно, чтобы корректно определить множество управлений в момент $t = 1$);*

- *Вершины сети нагружены множествами состояний $S_i(t)_{i \in N, t \in T}$;*

- *Ребра взаимодействия $(i, j) \in M^p$ нагружены множествами $U^p_{(i,j)}(t),\ t \in T$ состояний связи i-го игрока с j-м (нижние полурешетки), а динамические дуги $(i, j) \in M^d$ – множествами $U^d_{(i,j)}(t),\ t \in T$ состояний влияния игрока i на игрока j (произвольные множества);*

- *Состояние сети $x(t)$ в момент времени t – это набор $x(t) = ((s_i(t))_{i \in N}, (\inf(u_{ij}(t), u_{ji}(t)))_{(i,j) \in M^p}, (u^d_{ij}(t))_{(i,j) \in M^d})$;*

- *Множество управлений $U_i(x, t)$ каждого игрока i в каждом состоянии $x(t)$ в каждый момент времени t состоит из управлений игрока вершинами, дугами взаимодействия и динамическими дугами и, строго говоря, определяет состояние в следующий момент времени:*

$$U_i(x, t) \subseteq S_i(t + 1) \times \prod_{j|(i,j) \in M^p} U^p_{(i,j)}(t + 1) \times \prod_{j|(i,j) \in M^p} U^d_{(i,j)}(t + 1); \tag{2.1}$$

- *Множество S_i состояний вершины игрока i зависит от состояний входящих в вершину динамических дуг и ребер взаимодействия в предыдущий момент времени (зависимость множества управлений в момент t от состояния в момент t):*

$$\begin{aligned} S_i(t + 1) = S_i(&\{u^d_{ji}(t)\}_{(j,i) \in M^d}, \\ &\{\inf(u_{ij}(t), u_{ji}(t)\}_{(i,j) \in M^p}, t + 1) \end{aligned} \tag{2.2}$$

- *Множество возможных состояний всех дуг для каждого игрока i зависит от состояния его вершины, из которой выходят* *эти дуги (т.е. множество управлений в каждый момент времени непрямоугольно – одни его компоненты определяют множество возможных состояний других компонент):*

$$((u_i^p(j))_{(i,j)\in M^p}, (u_{ij}^d)_{(i,j)\in M^d}) \in U^i(s_i(t), t).$$

Иначе говоря, игрок выбирает состояние каждой вершины, зависящее от ребер взаимодействия и динамических дуг, связанных с вершиной в предыдущий момент времени. После этого он выбирает состояния динамических дуг и предлагаемые взаимодействия с другими игроками, зависящие от выбранных им состояний вершин. Состояния ребер взаимодействия в каждый момент времени определяются, исходя из предложений игроков друг другу в этот момент времени.

Как обычно в многошаговой игре, *стратегия* игрока – это управление $u_i(x,t) \in U_i(x,t)$, заданное для каждой сети $x(t)$ и каждого момента времени t. Как обычно, каждый набор стратегий определяет *партию* – то есть, последовательность состояний сети, причем состояние каждой вершины l в момент времени t зависит только от состояния дуг, в l и определенных в момент времени $t-1$. Партию можно изобразить графом g_t, развернутым во времени – то есть, графом, в котором множество вершин $N \times T$ и множество динамических дуг типа $((a,t),(b,t+1))$, где $(a,b) \in M = M^p \cup M^d$. При этом, состояние каждой вершины зависит от состояний входящих в нее дуг (ребра M^p отличаются от дуг M^d тем, что они неориентированны и состояние каждого ребра – нижняя грань управлений игроков, которых оно соединяет).

Определение 2.3. *Динамическая игра на сети – это динамическая сетевая игра, в которой нет ребер взаимодействия между игроками: $M^p = \emptyset$.*

Таким образом, дуги в такой игре – это динамические дуги M^d. Сеть, развернутая во времени, g_t в такой игре – бесконтурная. Это значит, что можно определить некоторую допустимую партию игры, рассматривая поочередно вершины сначала в 1-ый момент времени, потом во 2-ой и т.д. до конца.

3. Решение динамических сетевых игр

Как и в любой конечной игре, в конечной сетевой игре существует максиминная (осторожная) стратегия любого игрока [5]. Оказывается, для динамической сетевой игры верно свойство максимина, аналогичное свойству максимина в статической игре:

Теорема 3.1. *В динамической сетевой игре согласия существует такая максиминная ситуация для i-го игрока, что в получившейся сети нет дуг из множества M^p, соединяющих i-го игрока с остальными. При этом остальные игроки согласуют свои интересы, т.е. для любого (x, t) параметры любой дуги $(j, k) \in M^p$ совпадают с соответствующими стратегиями $u_j(k, x, t)$, $u_k(j, x, t)$ игроков j, k.*

Доказательство проводится индукцией по времени, исходя из соответствующего утверждения для статической игры [4]. □

Таким образом, в динамической сетевой игре для каждого игрока i существует максиминная ситуация, в которой множество M^p дуг взаимодействия i с другими игроками пусто. Более того: для остальных игроков дуги M^p имеют такое свойство: для любой дуги $(i, j) \in M^p$ должно быть $u_i(j, x, t) = u_j(i, x, t)$, при этом состояние вершин $s_i(t+1), s_j(t+1)$ в следующий момент времени определяется состоянием двух дуг $u_i(j, x, t) = u_j(i, x, t)$. Это значит, что при нахождении максиминной стратегии множество M^p можно устранить и решать максиминную задачу оптимизации динамической сети. Получаем рекуррентный алгоритм нахождения максимина для произвольного игрока сложностью $O(xms^n u_d^n u_p^{n(n-1)/2})$, где x – количество возможных состояний сети, $s = \max\limits_i |S_i|$, $u_d = \max\limits_{i,s_i,t} |\prod\limits_{j=1,\ldots,n} U^d_{(i,j)}(s_i, t)|$, $u_p = \max\limits_{i,j,s_i} |U^p_{(i,j)}(s_i)|$. Заметим, что, используя стандартный алгоритм для многошаговой игры и игнорируя сетевую структуру игры, мы получили бы большую сложность алгоритма – $O(xms^n u_d^n u_p^{n^2})$.

Равновесие по Нэшу в динамической сетевой игре также (как и в статической) обладает свойством согласовывать стратегии игроков:

Теорема 3.2. *Для каждой равновесной по Нэшу ситуации, определяющей управление $u(x, t)$ в динамической сетевой игре, для каждой дуги $(i, j) \in M^p$ можно управлениям $u_i(j, x, t), u_j(i, x, t)$ присвоить* *одно значение* $u_{ij}(x,t) = \inf(u_i(j,x,t), u_j(i,x,t))$. *При этом параметры сети не изменятся, а ситуация останется равновесной.*

Доказательство аналогично доказательству утверждения для статической игры [4]. □

Если известны максиминные значения для всех подыгр в развернутой форме, то в многошаговой игре можно найти все ситуации равновесия так [4, 6]: ситуация, порождающая управления $u(x,t)$, ведущая к траектории p и выигрышам H, равновесна, если отклонение каждого игрока i от каждой позиции (x,t) ведет к подыгре в развернутой форме $\Gamma_{(x,t)}$, в которой его максиминный выигрыш не больше $h^i_t(p(t)) + \ldots + h^i_m(p(m))$. Получаем алгоритм нахождения всех равновесий, который состоит из m шагов и на каждом шаге работает следующим образом:

1. На i-м шаге, зная равновесные управления u' при всех $t > m-i$, перебираем все состояния $x \in X$ и всех игроков $j \in i(x, m-i)$. Для каждого игрока находим максимин – то есть верхнее значение $V_j(x, m-i)$ во вспомогательной антагонистической игре $\Gamma_j(x, m-i)$, используя алгоритм нахождения максиминов.

2. Для всех $x \in X$ для всех управлений $u^* \in U(x, m-i)$ «стыкуем» управление u^* с каждым управлением u', начиная с состояния $(x, m-i)$. Получаем управление u''. Для каждого такого управления для каждого игрока j находим выигрыш $H^j_x(p(u''))$ при применении управления u'' в подыгре в развернутой форме $\Gamma_{(x,t)}$.

3. Для всех $x \in X$ для всех управлений $u^* \in U(x, m-i)$ для всех игроков $j = 1, \ldots, n$ рассматриваем все возможные «отклоняющиеся» управления $u_j \in U_j(x, m-i)$ каждого игрока. Для каждого управления u_j определяем состояние $x' = \pi(x, m-i, u'||u_j)$ при применении управления $u'||u_j = ({u'}_1, \ldots, {u'}_{j-1}, u_j, {u'}_{j+1}, \ldots, \ldots, {u'}_n)$. Если при этом верхнее значение для j-го игрока $V_j(x', m-i+1) > H^j_x(p(u''))$, то управление u'' не является равновесным и дальше его не рассматриваем. Если же при всех возможных отклонениях верхнее значение в отклонившейся игре оказывается не больше выигрыша, включаем управление u''

в список равновесных.

4. Если множество равновесных для $t \geq m-i$ управлений u'' пусто, то равновесий не существует. Иначе переходим к следующему моменту времени.

Для нахождения абсолютных равновесий в динамической сетевой игре получаем алгоритм «фильтрованного перебора», который работает так:

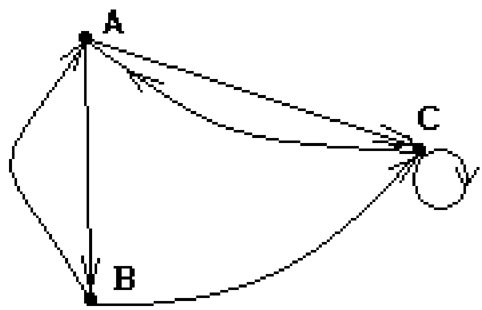
1. Находим все стабильные сети тем же алгоритмом, что и в общей динамической игре с позиционными стратегиями. При этом в каждой позиции (x,t) рассматриваем только те наборы ходов, в которых параметры каждой дуги из M^p оба игрока устанавливают равными.

2. Для каждой стабильной сети g в каждой позиции $x = g(1), \dots, \dots, g(m)$ в каждый момент времени $t = 1, \dots, m$ для каждого игрока i пытаемся добавить безответные предложения $u_i \in U^p_{(i,j)}$ в каждом ребре взаимодействия $(i,j) \in M^p$. Все такие комбинации предложений проверяем на равновесность путем добавления «ответных» предложений от игроков $j \mid (i,j) \in M^p$.

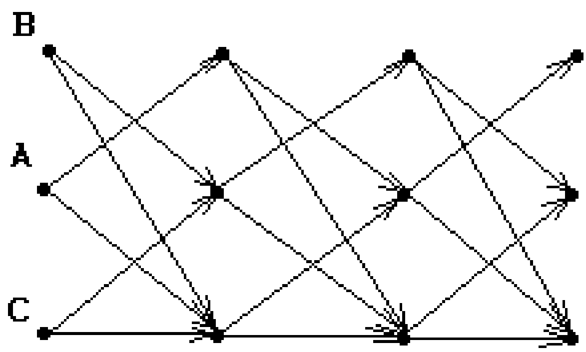
Сложность алгоритма нахождения абсолютного равновесия (всех абсолютных равновесий) – $O(xmns^{n+1}u_d^{n+1}u_p^{n(n+3)/2})$ $(O(xmns^{n+1}u_d^{n+1} u_p^{n(n+3)/2}v + nsu_d^n u_p^n q^{mn(n-1)/2}v))$, где $q = \max\limits_{m \in M} q(U_m)$. Исходя из сложности алгоритма нахождения максиминов, получаем сложность алгоритма для нахождения всех равновесий – $O(x^2m^2ns^{n+1}u_d^{n+1}u_p^{n(n+1)/2}v)$. Несложно видеть, что за счет дуг взаимодействия из U^p это существенно меньше, чем в общих алгоритмах для многошаговых игр.

4. Пример динамической сетевой игры

Рассмотрим динамическую сетевую игру, похожую на известную игру с мячом «картошка». Пусть имеется 3 игрока A, B, C, управляющих одноименными вершинами. Пусть между ними могут образовываться только динамические связи таких видов: $(A,B), (A,C), (B,A)$, $(B,C), (C,A), (C,C)$. Иначе говоря, это игра управления динамическим потоком в такой сети:



Пусть всего 4 момента времени $T = \{1, 2, 3, 4\}$, так что сеть, развернутая во времени, выглядит так:



Пусть каждая вершина может находиться всего в 2 состояниях: 0 («нет мяча») и 1 («есть мяч»), т.е. $S_i = \{0, 1\}$. Пусть, аналогично примеру с движущимся объектом, на каждом шаге мяч перемещается в другую вершину по дуге, т.е. $U_i^d(s_i(t), t) = \{(u_{i1}^d, u_{i2}^d, u_{i3}^d) \in \{0, 1\}^3 \mid u_{i1}^d + u_{i2}^d + u_{i3}^d = s_i(t)\}$ и $S_i(t+1) = \{\sum_{j=1}^{3} u_{ij}^d(t)\}$. Пусть выигрыш игрока равен количеству моментов времени, в которые у игрока был мяч, со знаком «-»: $f_i(s(t), t) = -s_i(t)$. То есть каждый игрок «перебрасывает мяч» другому игроку и стремится к тому, чтобы мяч был у него как можно реже.

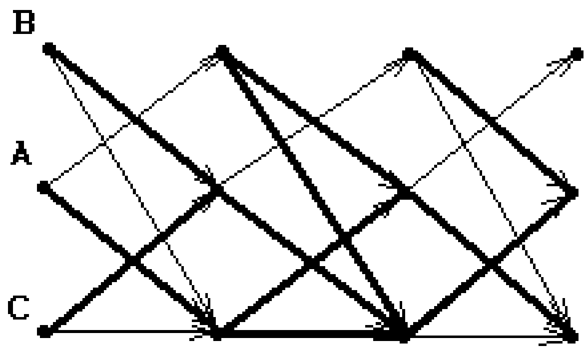
Таким образом, это динамическая игра на сети, в которой есть только дуги M^d, причем каждая дуга также может находиться только в двух состояниях $\{0, 1\}$. Поскольку игрок в каждый момент времени обязан передать кому-то мяч, игра не является игрой согласия. Сеть может находиться в 3 состояниях: $x_A = (1,0,0)$, $x_B = (0,1,0)$, $x_C = (0,0,1)$ (соответственно, мяч находится у игрока A, B, C). Это игра с постоянной суммой: $H_A + H_B + H_C = -4$.

Найдем сперва максиминные выигрыши игроков H_A^*, H_B^*, H_C^* во всех подыграх в развернутой форме (x,t), где $x \in \{x_A, x_B, x_C\}$, $t \in \{0,1,2,3\}$. Максимины легко посчитать «обратным ходом», и они таковы:

$$\begin{array}{lll}
t=3: & & \\
H_A^*(x_A,3)=-1, & H_B^*(x_A,3)=0, & H_C^*(x_A,3)=0, \\
H_A^*(x_B,3)=0, & H_B^*(x_B,3)=-1, & H_C^*(x_B,3)=0, \\
H_A^*(x_C,3)=0, & H_B^*(x_C,3)=0, & H_C^*(x_C,3)=-1. \\
t=2: & & \\
H_A^*(x_A,2)=-1, & H_B^*(x_A,2)=-1, & H_C^*(x_A,2)=-1, \\
H_A^*(x_B,2)=-1, & H_B^*(x_B,2)=-1, & H_C^*(x_B,2)=-1, \\
H_A^*(x_C,2)=-1, & H_B^*(x_C,2)=0, & H_C^*(x_C,2)=-1. \\
t=1: & & \\
H_A^*(x_A,1)=-2, & H_B^*(x_A,1)=-1, & H_C^*(x_A,1)=-1, \\
H_A^*(x_B,1)=-1, & H_B^*(x_B,1)=-1, & H_C^*(x_B,1)=-1, \\
H_A^*(x_C,1)=-1, & H_B^*(x_C,1)=-1, & H_C^*(x_C,1)=-2. \\
t=0: & & \\
H_A^*(x_A,0)=-2, & H_B^*(x_A,0)=-1, & H_C^*(x_A,0)=-2, \\
H_A^*(x_B,0)=-2, & H_B^*(x_B,0)=-2, & H_C^*(x_B,0)=-2, \\
H_A^*(x_C,0)=-2, & H_B^*(x_C,0)=-1, & H_C^*(x_C,0)=-2.
\end{array} \tag{4.1}$$

Очевидно, в каждой подыгре в развернутой форме $\Gamma_{(x,t)}$ равновесные выигрыши $H'_A(x,t), H'_B(x,t), H'_C(x,t)$ не меньше максиминных: $H'(x,t) \geq H^*(x,t)$. Из найденных значений и соотношения $H'_A(x,t) + H'_B(x,t) + H'_C(x,t) = 4-t$ видно, что максиминные выигрыши во многих случаях совпадают с равновесными – то есть, игрок не получает больше выигрыша, который ему гарантирован. Кроме того, максиминный выигрыш игрока часто не зависит от начальной позиции – следовательно, равновесий очень много. Однако, есть и подыгры в развернутой форме, в которых $H_A^*(x,t) + H_B^*(x,t) + H_C^*(x,t) < 4-t$, а следовательно, не всякая максиминная ситуация является равновесием.

Абсолютное равновесие также не единственно. Найдем (также неединственное, но с единственным вектором выигрышей) абсолютное равновесие «в тип-стратегиях» [5], то есть, в лексикографической игре, в которой $\overline{H}_A = (H_A, H_B, H_C)$, $\overline{H}_B = (H_B, H_C, H_A)$, $\overline{H}_C = (H_C, H_A, H_B)$. Из рекуррентных соотношений следуют такие равновесные ходы (они выделены жирными линиями на рисунке):



с такими выигрышами при разных начальных позициях x_A, x_B, x_C:

$$\begin{array}{lll} H'_A(x_A) = -2, & H'_B(x_A) = 0, & H'_C(x_A) = -2, \\ H'_A(x_B) = -2, & H'_B(x_B) = -1, & H'_C(x_B) = -1, \\ H'_A(x_C) = -2, & H'_B(x_C) = 0, & H'_C(x_C) = -2. \end{array} \tag{4.2}$$

Рассмотрим теперь модифицированную игру, в которой есть ребра взаимодействия M^p, причем $U_i^p = \{\emptyset, \{j\}, j \neq i\}$, т.е. каждый игрок может соединиться только с одним другим игроком либо ни с кем не соединяться. Таким образом, это игра согласия. Пусть теперь состояние каждой вершины состоит из 6 элементов: $S_i = \{0,1\} \times \times \{1,2,3\}$, где первая компонента вектора состояния определяет, есть ли у игрока мяч, а вторая – с каким игроком он соединился в предыдущий момент времени, либо i, если он ни с кем не соединился. То есть $S_i(t+1) = \{(\sum\limits_{j=1}^{3} u_{ij}^d(t), j \mid \inf(u_{(i,j)}^p(t), u_{(j,i)}^p(t))) = 1\}$. Пусть на каждом шаге проигрыш игрока, не соединившегося ни с кем, определяется так же, как и в предыдущем случае, а проигрыш соединившегося игрока равен $-2b$, если мяч у него, 0, если мяч у его соседа по коалиции, и $-b$, если мяч не попал ни к кому:

$$f_i((s^1(t), s^2(t)), t) = \begin{cases} -s_i^1(t), & s_i^2(t) = i, \\ -bs_i^1(t) - b(1 - s_j^1(t)) & s_i^2(t) = j \neq i. \end{cases} \quad (4.3)$$

Можно интерпретировать это так: плата за соединение игроков равна $-2b$, зато соединение защищает от мяча. Если мяч попал в одного из соединившихся игроков, то он отдает всю плату за соединение, иначе плата делится поровну.

Итак, по сравнению с предыдущей игрой количество состояний сети на каждом шаге увеличилось в 4 раза: к каждому состоянию x_i добавились состояния $x_i^{AB}, x_i^{BC}, x_i^{AC}$ с соответствующими соединениями игроков – например, $x_A^{BC} = ((1,1), (0,3), (0,2))$. Количество ходов на каждом шаге увеличилось в $3^3 = 27$ раз, поскольку каждый игрок выбирает, с кем ему соединяться. Кроме того, игра, благодаря возможности «кооперации», перестала быть игрой с постоянной суммой. Но, воспользовавшись свойствами игр согласия, мы можем решить эту игру почти так же просто, как предыдущую.

Действительно, максиминные стратегии игрока i, по теореме 3.1, определяются сетями, в которых игрок i не соединяется с другими игроками. Соединение же оставшихся игроков между собой можно не учитывать, поскольку оно не влияет на выигрыш игрока i. Таким образом, максиминные стратегии и выигрыши – те же, что и в предыдущей игре.

При поиске абсолютных равновесий можно воспользоваться теоремой 3.2 и искать стабильные сети, не рассматривая те ходы, в которых игроки образуют безответные соединения. Таким образом, на каждом шаге мы вместо $2 \cdot 27$ ходов рассматриваем всего $2 \cdot 4$ хода: без соединений, с соединением AB, с соединением BC и с соединением AC. По свойству игры согласия, найденные в предыдущей игре стабильные сети без соединений стабильны и в модифицированной игре, а соответствующие ситуации, к которым, возможно, добавлены безответные предложения – абсолютно равновесны. Проверим теперь на стабильность сети с соединениями, для каждого соединения проверяя, не выиграет ли игрок от его разрыва. В случае, если $b \geq 1/2$, стратегии без соединений всегда будут доминирующими, а сети с любыми соединениями – нестабильными. Если $b \leq 0$ – стратегии с соединениями всегда будут доминирующими, а сети с любыми соединениями – стабильными.

Рассмотрим теперь наиболее интересный случай с $0 < b < 1/2$. В этом случае на каждом шаге, на котором мяч попал к игроку i, стабильными, кроме сети без соединений, оказываются еще 2 сети, в которых i соединился с любым другим игроком (даже с тем, кто послал ему мяч!). Действительно, i от разрыва связи теряет, получая -1 вместо $-2b$, а другой игрок – не выигрывает, в обоих случаях получая 0. Опять же, к стратегиям, порождающим эти сети, можно добавить любые безответные предложения, сохранив равновесие. Сеть же, в которой соединяются игроки, не получившие мяч, всегда нестабильна, ибо каждому из них выгодно разорвать соединение, получив 0 вместо $-b$. В наиболее благоприятных для игроков ситуациях все равновесные проигрыши надо умножить на $2b < 1$.

СПИСОК ЛИТЕРАТУРЫ

1. Губко М. В. *Управление организационными системами с сетевым взаимодействием агентов* // Автоматика и телемеханика. 2004. №8. С. 115–123, № 9. С. 131–148.

2. Губко М.В. *Теоретико-игровая модель формирования торговой сети* // Управление большими системами. 2004. № 6. С. 56–83.

3. Малафеев О.А., Парфенов А.П. *Решение сетевых игр// Математическое и компьютерное моделирование социально-экономических систем на уровне многоагентного взаимодействия (введение в проблемы равновесия, устойчивости и надежности)*. СПб.: СПбГУ. 2006. С. 697–710.

4. Парфенов А.П. *Многошаговые сетевые игры управления потоками* // Вестн. С.-Петербург. ун-та. Сер. 10. 2009. Вып. 4. С. 200–212.

5. Петросян Л.А, Зенкевич Н.А., Семина Е.А. *Теория игр*. М.: Высшая школа, 1998.

6. Слобожанин Н.М. *Информация и управление в динамических играх*. СПб: изд.-во СПбГУ, 2002.

7. Basar T., Olsder G.J. *Dynamic noncooperative game theory.* London: Academic press inc., 1982.

THE ALGORITHM SEARCHING NASH EQUILIBRIA IN DYNAMIC NETWORK GAME

Andrey P. Parfyonov, Saint-Petersburg State University, Faculty of Applied Mathematics and Control Processes (parf@bk.ru).

Abstract: We carry out the assimilation of network formation game, which allows simultaneous actions. This is a dynamic network game with arcs of two kinds: interactional arches and dynamic arches. We have set an algorithm for retrieval of Nash equilibrium in a dynamic network game, which is a modification of the recurrent algorithm for multistage game, and uses the interactional arches' properties. We have estimated the complexity of the algorithm. An example is constructed, which shows the advantages of the modified algorithm in comparison to the standard recurrent algorithm for multistage game.

Keywords: network games, dynamic games, positional games, Nash equilibrium, optimization algorithms, algorithm complexity.