

КОМПЛЕКСЫ ПРОГРАММ | SOFTWARE PACKAGES

Научная статья | Original paper

УДК 007.5:004.89:62-50:681.5

Разработка вероятностной модели поведения многоагентной системы в трехмерном пространстве

Н.И. Левонович

Московский государственный психолого-педагогический университет,
Москва, Российская Федерация
✉ levonikitech@yandex.ru

Резюме

В статье рассматриваются теоретические аспекты управления многоагентной системой, на основе вероятностных марковских моделей. Система представляет собой набор агентов, с возможностью взаимодействовать между собой, для достижения группы целей. Поведение системы является непредсказуемым с точки зрения целей. Поведение агентов определяется с помощью алгоритма, который включает идентификацию параметров вероятностной модели, используя максимизацию целевых функций, представляющих индивидуальные и групповые вероятности достижения (пометки) целей.

Ключевые слова: многоагентные системы, контроль групп, марковские процессы, идентификация моделей, контроль поведения прикладных многоагентных систем

Для цитирования: Левонович, Н.И. (2025). Разработка вероятностной модели поведения многоагентной системы в трехмерном пространстве. *Моделирование и анализ данных*, 15(2), 152–164. <https://doi.org/10.17759/mda.2025150209>

Probabilistic model of behavior multi-agent system is three-dimensional space developing

N. I. Levonovich

Moscow State University of Psychology and Education, Moscow, Russian Federation
✉ levonikitech@yandex.ru



Abstract

The article discusses theoretical aspects of multi-agent systems based of probability Markov models. System consists of agents, that have ability to cooperate for achievement of goals. The behavior of system is unpredictable from point of goals' view. The behavior of agents Agent's behavior is determined with the aid of the algorithm that includes identification of the probabilistic model parameters using maximized objective functions representing individual and group probabilities for goals reaching (marking).

Keywords: multi-agent systems; group control; Markov process; model identification; behavior control for applied multi-agent systems

For citation: Levonovich, N.I. (2025). Probabilistic model of behavior multi-agent system is three-dimensional space developing. *Modelling and Data Analysis*, 15(2), 152—164. (In Russ.). <https://doi.org/10.17759/mda.2025150209>

Введение

В последнее время существенно актуализировались проблемы, связанные с управлением группой подвижных объектов, которым необходимо согласовывать свое поведение для достижения целей. Постановка и решение задач в данной сфере может использоваться для управления группами наземных и воздушных роботов и других мобильных систем.

Данные системы могут применяться для поиска в тайге, поиска и слежения за движущимися объектами на обширных пространствах.

Описание поведения системы

Агенты $L_m (m=0, \dots, M-1)$ движутся в трехмерном игровом пространстве, которое содержит набор целей $T_n (n=0, \dots, N-1)$, согласно некоторым правилам, пытаясь поразить цели. С целью определения положения агентов и целей вводится агент-наблюдатель, относительно которого производится позиционирование, к нему привязана относительная система координат. Пространство ограничено зоной действия агента-наблюдателя и разделено на кубы. Позиция агента и цели определяется с точностью до куба (i, j, k) , где i — индекс долготы ($i=(0, \dots, I-1)$), j — индекс широты ($j=(0, \dots, J-1)$), k — индекс высоты ($k=(0, \dots, K-1)$) (рисунок 1). Вероятность того, что агент L_m находится в ячейке (i, j, k) в момент времени t определяется функцией $p_{m,ijk}(t)$.

В дискретные моменты времени, разделенные интервалами дискретизации Δt , агент L_m может воздействовать на цель, и вероятно это воздействие будет успешно или претерпит воздействие со стороны цели и вероятно пораженным целью с вероятностями определяемые позициями агентов и целей. В параметрах системы задаются: пороговые вероятности для индивидуального p_i и коллективного p_m воздействия



на цель, максимальная разрешенная вероятность успешного воздействия цели на агента p_B , максимальная скорость агента $v_{\max}$. Число агентов, одновременно терпящих воздействие одной цели (не регулируется, один агента).

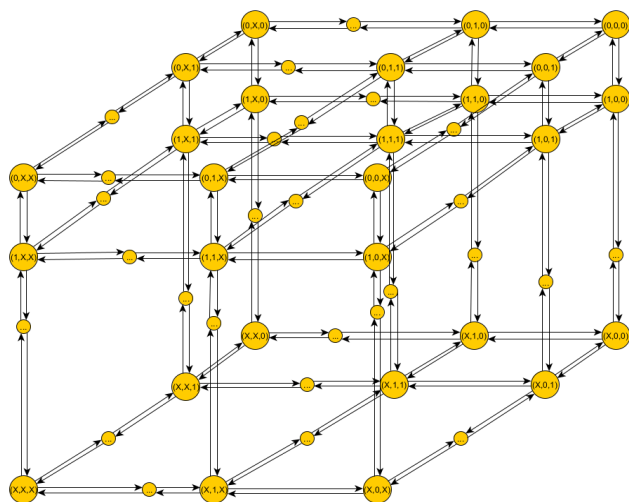


Рис. 1. Структура отношений между состояниями случайного марковского процесса, представляющая движение агентов по ячейкам игрового поля

Fig. 1. Structure of relationship between states of random Markov process representing agent movement through game field cells

В дискретный каждый момент времени все функционирующие агенты знают какой ячейке они находятся. В зависимости от игровой ситуации агенты могут получать или не получать информацию о положении других агентах.

Изменение положения агентов и целей на игровом поле, а также вероятностей взаимных поражений происходит за фиксированный интервал времени Δt называется *тактом игры*.

Концом игры является ситуация, когда на поле не осталось целей (все цели помечены), либо когда на поле не осталось агентов.

В рассматриваемой модели цели могут располагаться только на одном индексе высоты (наземные цели).

Предположим, что движение каждого агента по ячейкам анализируемого пространства описывается случайным марковским процессом с непрерывным временем и дискретными состояниями. Нахождение в состоянии такого марковского процесса означает пребывание в выделенной ячейке рассматриваемого пространства, которая имеет те же индексы. В таком случае переходы, подчиняющиеся пуассоновскому потоку случайных событий, возможны только между смежными ячейками, которые имеют общую грань.



Число переходов между смежными состояниями X , происходящий за любой интервал времени τ , начиная со времени t подчиняются закону Пуассона:

$$P_{t,\tau}(X = m) = \frac{a(t, \tau)^m}{m!} e^{-a(t, \tau)},$$

где $P_{t,\tau}(X = m)$ — вероятность m переходов за этот интервал, $a(t, \tau)$ среднее количество переходов, совершаемых за интервал τ с момента времени t . В дальнейшем рассматриваются только стационарные потоки $a(t, \tau) = \eta\tau$ и $\eta = \text{const}$

Гипотеза о пуассоновском распределении переходов является стандартной для рассматриваемой области, так как это распределение часто встречается на практике, так как следует из предельных теорем для потоков событий.

Поведение каждого агента в части перемещения определяется автономно. Динамика вероятностей пребывания m -того агента в состояниях марковского процесса определяется системой дифференциальных уравнений Колмогорова в матричной форме:

$$\frac{dp_m(t)}{dt} = M_m(\lambda_m)p_m(t)$$

где $p_m(t)$ — вероятности пребывания m -того агента в n -том состоянии процесса, λ_m — вектор интенсивностей перехода между смежными состояниями для m -того агента. M_m — матрица n -того порядка интенсивностей перехода между смежными состояниями m -того агента. Начальные условия $p_{m,i_0,j_0,k_0}(0) = 1, \{p_{m,ijk}(0) = 0\}_{i \neq i_0, j \neq j_0, k \neq k_0}$, где (i_0, j_0, k_0) — индекс ячейки в который m -тый агент находился в $t_0 = 0$.

Введём следующее обозначения для элементов вектора λ_m (рис. 2):

- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону увеличения первой координаты $\lambda_{m,i,j,k,\rightarrow}$;
- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону уменьшения первой координаты $\lambda_{m,i,j,k,\leftarrow}$;
- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону увеличения второй координаты $\lambda_{m,i,j,k,\odot}$;
- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону уменьшения второй координаты $\lambda_{m,i,j,k,\otimes}$;
- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону увеличения третьей координаты $\lambda_{m,i,j,k,\uparrow}$;
- Интенсивность переходов m -того агента в состоянии (i, j, k) в сторону уменьшения третьей координаты $\lambda_{m,i,j,k,\downarrow}$;

Расчет вероятностей $p_m(t)$ для всех агентов происходит синхронно в дискретные моменты времени с шагом Δt . Некоторые агенты могут оставаться в тех же ячейках.

Обозначим текущий момент времени как t_* , обозначим введем следующие обозначения для событий:



A_n — n -тая цель успешно помечена в результате воздействия;

B_{mn} — m -тый агент помечена в результате n -той цели;

D_{mn} — воздействие m -тым агентом на n -тую цель;

$H_{i_m j_m k_m m}$ — m -тый агент находится в ячейке (i_m, j_m, k_m)

$\tilde{H}_{i_m j_m k_m m}$ — в момент времени $t_* + \Delta t$ m -тый агент находится в ячейке (i_m, j_m, k_m) , которая смежна с ячейкой, в которой агент находился в момент времени t_* .

C_m — переход m -того агента из ячейки, в которой он был в момент времени t_* в одну из смежных ячеек.

$H_{i_n j_n k_n n}$ — n -тая цель находится в ячейке (i_n, j_n, k_n)

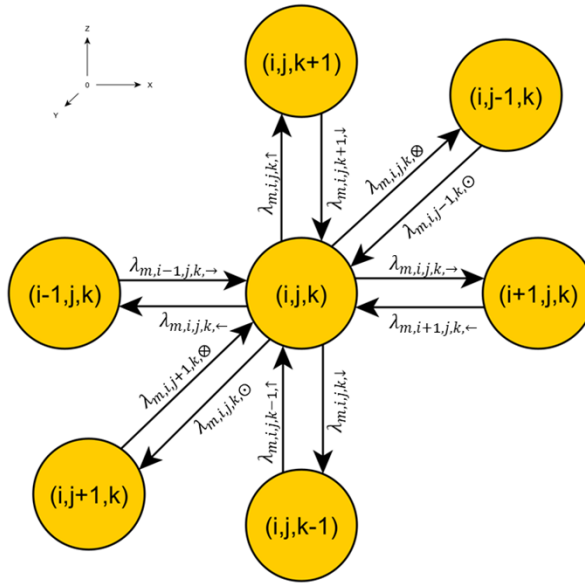


Рис. 2. Обозначение интенсивности переходов m -того агента в данном состоянии

Fig. 2. Designation of the intensity of transitions of the m -th agent in a given state

Вероятность успешной пометки n -той цели в результате воздействия m -тым агентом вычисляется согласно формуле полной вероятности:

$$p(A_n | D_{mn}) = \sum_{i,j,k} p(A_n | H_{i_m j_m k_m m} \cap H_{i_n j_n k_n n}) p(H_{i_m j_m k_m m}) p(H_{i_n j_n k_n n})$$

Вероятность успешной пометки n -той цели, находящейся в ячейке (i_n, j_n, k_n) m -тым агентом в результате воздействия из ячейки (i_m, j_m, k_m) в момент времени t_* задается относительной «картой осуществимости» f_a :



$$p(A_n | H_{i_m j_m k_m m} \cap H_{i_n j_n k_n n}) = f_a(i_m, j_m, k_m, i_n, j_n, k_n, t_*)$$

Вероятности $p(H_{i_m j_m k_m m})$ определяются из решения системы дифференциальных уравнений Колмогорова. Цель движется по функции.

Вероятность успешной пометки m -того агента, находящегося ячейке (i_m, j_m, k_m) , n -той целью находящейся в ячейке (i_n, j_n, k_n) в результате воздействия в момент времени t_* задается относительной «картой уязвимости» f_b :

$$p(A_n | H_{i_m j_m k_m m} \cap H_{i_n j_n k_n n}) = f_b(i_n, j_n, k_n, i_m, j_m, k_m, t_*)$$

Карты осуществимости и уязвимости пересчитываются каждый такт игры, чтобы отслеживать перемещения целей и агентов по игровому полю. Для решения множества прикладных проблем, целесообразно определить распределение вероятностей, которые задают две карты как логистические функций:

$$f_a(i_m, j_m, k_m, i_n, j_n, k_n, t_*) = c_a \left(\frac{e^{r_{a,d} d(i_m, j_m, k_m, i_n, j_n, k_n) + q_{a,d}}}{1 + e^{r_{a,d} d(i_m, j_m, k_m, i_n, j_n, k_n) + q_{a,d}}} \right)$$

$$f_b(i_m, j_m, k_m, i_n, j_n, k_n, t_*) = c_b \left(\frac{e^{r_{b,d} d(i_m, j_m, k_m, i_n, j_n, k_n) + q_{b,d}}}{1 + e^{r_{b,d} d(i_m, j_m, k_m, i_n, j_n, k_n) + q_{b,d}}} \right)$$

где $d(i_m, j_m, k_m, i_n, j_n, k_n)$ расстояние между ячейками (i_m, j_m, k_m) и (i_n, j_n, k_n) ; параметры $c_a, c_b, r_{a,d}, q_{a,d}, r_{b,d}, q_{b,d}$ идентифицируется по методу максимального правдоподобия согласно эмпирическим данным, для того чтобы обеспечить наивысшую вероятность попадания наблюдаемой цели и агентов в контрольную серию экспериментов.

Найдем закон распределения для $\tau_* > 0$, которое необходимо для перехода между состояниями процесса. Вероятность, что переход не случится — $P_\tau(X=0) = e^{-\eta\tau}$. Это значение эквивалентно вероятности события $\tau_* > \tau$: $P(\tau_* > \tau) = e^{-\eta\tau}$. Следовательно, $P(\tau_* \leq \tau) = 1 - P(\tau_* > \tau) = 1 - e^{-\eta\tau}$, где $F(\tau, \eta) = P(\tau_* \leq \tau)$ — функция распределения случайной величины τ_* . Это распределение имеет плотность $\rho(\tau) = \eta e^{-\eta\tau}$ и математическое ожидание $E = \int_0^\infty t \eta e^{-\eta t} dt = \frac{1}{\eta}$

Вероятность успешной пометки n -той цели в ходе групповой атаки оценивается согласно формуле сумме вероятностей, в случае воздействия на него всеми агентами одновременно:

$$p(A_n | D_{1n} + \dots + A_n | D_{mn}) = p(A_n | D_{1n}) + \dots + p(A_n | D_{mn}) - p(A_n | D_{1n} * A_n | D_{2n}) - \\ - p(A_n | D_{1n} * A_n | D_{3n}) - \dots + p(A_n | D_{1n} * A_n | D_{2n} * A_n | D_{3n}) + \dots \pm p(A_n | D_{1n} * A_n | D_{2n} * \dots * A_n | D_{mn})$$

События $A_n | D_{in}$ и $A_n | D_{jn}$ полагаются независимыми если $i \neq j$.



Построение пар атак

Введем понятие воздействия (потенциального воздействия), как пары агента и цели $d_{mn} = (l_m, t_n)$, $\mathbf{D}$ — множество воздействий, функционал $f_a^+(d_{mn}, t_*) = f_a(i_m, j_m, k_m, i_n, j_n, k_n, t_*)$ назовем значением воздействия.

Матрица значений воздействия $\mathbf{C}$, образуется из значений воздействий, которые стоят на пересечении строки (относящейся к агенту) и столбца (относящейся к цели).

$$\mathbf{C}(t_*) = c_{ij} = f_a^+(d_{ij}, t_*)$$

Построим граф $G = (W, E); W = L_m \cup \mathbf{D}$; $E = \{(l_m, d_{mn}) : l_m \in L_m; l_m \in d_{mn}\}$. Для ребер существует функционал разметки $g: E \rightarrow \mathbb{R}$, $g(e) = \sup f_a^+(d_{mn}, t_*) - f_a^+(d_{mn}, t_*) + \epsilon$, где $d_{mn} \in e$.

Для полученного графа с помощью венгерского алгоритма [1; 2] решается задача о назначениях получается распределение воздействий.

Определения:

- Паросочетанием M называется набор попарно несмежных ребер графа (иными словами, любой вершине графа должно быть инцидентно не более одного ребра из множества M). Мощностью паросочетания назовём количество ребер в нём. Наибольшим (или максимальным) паросочетанием назовём паросочетание, мощность которого максимальна среди всех возможных паросочетаний в данном графе. Все те вершины, у которых есть смежное ребро из паросочетания (т.е. которые имеют степень ровно один в подграфе, образованном M), назовём насыщенными этим паросочетанием.
- Полное паросочетание — паросочетание, в которое входят все вершины.
- Цепью длины k назовём некоторый простой путь (т.е. не содержащий повторяющихся вершин или ребер), содержащий k ребер.
- Чередующейся цепью (в двудольном графе, относительно некоторого паросочетания) назовём цепь, в которой ребра поочередно принадлежат/не принадлежат паросочетанию.
- Увеличивающей цепью (в двудольном графе, относительно некоторого паросочетания) назовём чередующуюся цепь, у которой начальная и конечная вершины не принадлежат паросочетанию.

Алгоритм решает следующую задачу пусть дан взвешенный полный двудольный граф с целыми весами ребер $K_{n,n}$, нужно найти в нем полное паросочетание минимального веса. Вес паросочетания определяется как сумма весов его ребер.

Функцию ϕ назовём потенциалом, если для любых вершин i и j выполняется условие:

$$\phi(i) + \phi(j) \leq g((i, j))$$

где $g((i, j))$ — стоимость ребра между i и j . Значением потенциала называется сумма потенциалов всех вершин.



Заметим, что стоимость любого совершенного паросочетания не может быть меньше значения любого потенциала. Венгерский алгоритм решает задачу оптимально: он находит одновременно и совершенное паросочетание, и потенциал, у которых стоимость паросочетания совпадает со значением потенциала. Это доказывает, что оба решения являются оптимальными.

Алгоритм работает с жёсткими рёбрами — теми, для которых выполняется равенство:

$$\phi(i) + \phi(j) = c(i, j)$$

Если обозначить подграф из таких рёбер как G_ϕ , то стоимость любого совершенного паросочетания в G_ϕ (при его существовании) в точности равна значению потенциала ϕ .

Алгоритм работает с матрицей весов графа.

Вспомогательный алгоритм (алгоритм Куна)

Алгоритм

1. Берем пустое паросочетание;
2. Пока в графе удается найти увеличивающую цепь, выполняется чередование паросочетание вдоль этой цепи, и повторять процесс поиска увеличивающей цепи.
 - а. Как только не удалось найти увеличивающую цепь, процесс поиска останавливается, текущее паросочетание максимально.

Алгоритм Куна [3] ищет любую увеличивающую цепь, с помощью обхода в глубину или в ширину. Алгоритм Куна просматривает все вершины графа по очереди, запуская из каждой обход, пытающийся найти увеличивающую цепь, начинающуюся в этой вершине.

Алгоритм последовательно обрабатывает все вершины v первой доли ($v = 1, \dots, n_1$):

- Если вершина v уже насыщена текущим паросочетанием (т. е. уже соединена с какой-то вершиной второй доли), то она пропускается.
- В противном случае алгоритм пытается насытить v путём поиска увеличивающей цепи, начинающейся в этой вершине.

Для поиска используется обход в глубину (DFS) (реже — в ширину, BFS):

1. Начинаем из ненасыщенной вершины v первой доли.
2. Перебираем все рёбра, исходящие из v . Пусть текущее ребро ведёт в вершину to второй доли.
 - Если to ненасыщенна, то цепь найдена — это просто ребро (v, to) .
 - Действие: добавляем (v, to) в паросочетание и завершаем поиск.
 - Если to уже насыщена ребром (p, to) , то продолжаем поиск из p .
 - Таким образом, мы пытаемся построить чередующуюся цепь вида $(v, to), (to, p), \dots$



3. Обход продолжается, пока либо не будет найдена увеличивающая цепь, либо не станет ясно, что такой цепи не существует.

Результат обхода:

- Если цепь найдена, то вершина v становится насыщенной, и мощность паросочетания увеличивается на 1.
- Если цепь не найдена, то вершина v остаётся ненасыщенной (и в текущем паросочетании её уже нельзя покрыть).

Завершение работы алгоритма.

После обработки всех вершин первой доли текущее паросочетание становится максимальным, и алгоритм завершает работу.

Алгоритм для равных долей

Алгоритм хранит в памяти потенциал ϕ (в виде массивов u и v) и ориентацию (направление) каждого жёсткого ребра. Эта ориентация обладает ключевым свойством: рёбра, направленные от T к S , формируют паросочетание, обозначаемое M . Ориентированный граф, состоящий из жёстких рёбер с заданной ориентацией, обозначается как $\overrightarrow{G}_\phi$.

Шаг 1. В начале алгоритма потенциал полагается равным нулю и паросочетание M полагается пустым.

Цикл. На каждом шаге алгоритм пытается увеличить мощность текущего паросочетания M на единицу, не изменяя потенциалы. Это делается в графе жёстких рёбер G_ϕ с использованием модифицированного алгоритма Куна для поиска максимального паросочетания в двудольных графах.

Если на текущем шаге цикла не удалось увеличить паросочетание, производится корректировка потенциалов, чтобы создать новые возможности для увеличения паросочетания:

1. Определим множества Z_1 и Z_2 и величину Δ :
 - Z_1 — посещённые вершины первой доли при обходе (поиске увеличивающей цепи).
 - Z_2 — посещённые вершины второй доли.
 - Δ вычисляется как:

$$\Delta = \min_{\substack{i \in Z_1 \\ j \notin Z_2}} \{c_{ij} - u_i - v_j\}.$$

Таким образом $\Delta > 0$, иначе существовало бы «жёсткое» ребро (i, j) , ведущее к противоречию с определением Z_1 и Z_2 .

2. Корректировка потенциалов:

- Для всех $i \in Z_1$: $u_i = u_i + \Delta$.
- Для всех $i \in Z_2$: $v_j = v_j - \Delta$.



Корректность потенциала сохраняется: для рёбер (i, j) , где $i \in Z_1$ и $j \notin Z_2$: $u_i + v_j \leq c_{ij}$ (по выбору Δ). Для остальных комбинаций i, j неравенство $u_i + v_j \leq c_{ij}$ либо не изменилось, либо усилилось.

Жёсткие рёбра паросочетания остаются: рёбра (i, j) паросочетания могли измениться только если $i \notin Z_1$ и $j \in Z_2$, но такие рёбра не входят в M (так как i не была посещена).

3. Рост достижимого множества.

После пересчёта все ранее достижимые вершины остаются достижимыми. Появится хотя бы одно новое жёсткое ребро (i, j) (где $i \in Z_1$, $j \notin Z_2$), делающее вершину j достижимой. Таким образом, $|Z_1| + |Z_2|$ строго увеличивается.

4. Конечность алгоритма

Поскольку размер $|Z_1| + |Z_2|$ не может превысить $n_1 + n_2$, число пересчётов потенциалов ограничено ($O(n)$). После каждого пересчёта либо находится увеличивающая цепь, либо прогресс гарантирован.

Завершение алгоритма. Алгоритм продолжает итеративно выполнять следующие шаги:

- Поиск увеличивающей цепи для текущего паросочетания M .
- Если цепь найдена — увеличение M на единицу.
- Если цепь не найдена — пересчёт потенциалов, расширяющий множество достижимых вершин.

Рано или поздно будет достигнут потенциал, при котором существует совершенное паросочетание M (если исходный граф его допускает). Это паросочетание и будет решением задачи.

Если говорить об асимптотике алгоритма, то она составляет $O(n^4)$, поскольку всего должно произойти n увеличений паросочетания, перед каждым из которых происходит не более n пересчётов потенциала, каждый из которых выполняется за время $O(n^2)$.

Модификация алгоритма для неравных долей с асимптотикой $O(n^3)$

Ключевая идея — вместо одновременного рассмотрения всей матрицы, алгоритм последовательно добавляет строки, на каждом шаге поддерживая максимальное паросочетание для уже обработанной части. Это позволяет:

1. Локализовать пересчёты потенциалов только для новых данных.
2. Сделать алгоритм пригодным для неравных долей.
3. Снизить асимптотику до $O(n^3)$ (или $O(n^2m)$ для прямоугольных матриц). Оптимизации для асимптотики $O(n^3)$:
 - Поддержка массива $minv$, для каждого столбца j хранится минимальное значение $c_{ij} - u_i - v_j$ по всем посещённым строкам Z_1 . Обновляется за $O(n)$ при добавлении новой строки в Z_1 .
 - Быстрый поиск $\Delta = \min_{j \in Z_2} \{minv_j\}$ вычисляется за $O(n)$.



- Итеративный обход Куна — после пересчёта потенциала обход продолжается с новыми жёсткими рёбрами, не перезапускаясь с нуля.

В результате выполнения венгерского алгоритма имеются пары оптимальных атак, с вероятностью поражения.

Алгоритм поведения системного агента

Поведение m -того агента ($m = 0, \dots, M-1$) определяется следующим алгоритмом.

Шаг 1. Указать начальные условия, которые определяются:

- Распределение агентов по ячейкам (i, j, k) игрового поля;
- «Карта осуществимости» представлена функцией $f_a(i_m, j_m, k_m, i_n, j_n, k_n, t_*)$
- «Карта уязвимости» представлена функцией $f_b(i_m, j_m, k_m, i_n, j_n, k_n, t_*)$
- Приблизительная оценка интервала семплирования.

Шаг 2. Для текущего распределения агентов и целей на текущий момент t_* :

- Для каждой цели рассчитать вероятность пометки агента. Для каждой цели выбрать первого по порядку агента, вероятность пометки, которого превышает порог, удалить агента пропорционально вероятности пометки.
- Выполнить вероятностную атаку согласно построенным парам атак с помощью алгоритма, представленного в предыдущем пункте статьи.

Шаг 3. Если как минимум одно из условий для завершения игры выполнено на момент t_* :

- Получение информации о пометке всех агентов.
- Получение информации о пометке всех целей.
- Тогда перейти к шагу 6, иначе перейти к шагу 4.

Шаг 4. Выполним идентификацию свободных параметров $\{\lambda_m\}_{m=0, \dots, M-1}$ марковского процесса, полагая $|\mathbf{v}_k| \leq v_{\max}$, где ограничение скорости задает ограничения значения компонент вектора $\{\lambda_m\}_{m=0, \dots, M-1}$, со средними значениями $\bar{\tau}_{\lambda^*} = \frac{1}{\lambda_*}$, $\bar{\tau}_{\mu^*} = \frac{1}{\mu_*}$ и $\bar{\tau}_{\nu^*} = \frac{1}{\nu_*}$ времени перехода. Если агентам доступна информация друг о друге, максимизировать целевую функцию игры, которая вычисляется по формуле (obj), и представляет собой сумму вероятностей успешного воздействия на цели (простую сумму, которая сама не является вероятностью) в момент времени $t_* + \Delta t$, принимая в расчет всех агентов.

$$obj = \sum_{n=0, \dots, N-1} p(A_n | D_{1n} + \dots + A_n | D_{mn})$$

В противном случае, задача оптимизации решается отдельно для каждого агента автономно, с индивидуальными целевыми функциями $iobj$, в момент времени $t_* + \Delta t$ (при условии равновероятного выбора цели).

$$iobj = \sum_{n=0, \dots, N-1} \frac{p(A_n | D_{mn})}{N} (m = 0, \dots, M-1)$$



Перейти к шагу 5.

Шаг 5. Для каждого агента выбрать ячейку игрового поля, смежную с ячейкой, в которой он находится в момент времени t_* используя «метод рулетки», с вероятностями выбора пропорциональными предсказанным байесовским оценкам

$$p(\tilde{H}_{i_m j_m k_m m} | C_k) = \frac{p(C_k | \tilde{H}_{i_m j_m k_m m}) p(\tilde{H}_{i_m j_m k_m m})}{p(C_k)} = \frac{p(\tilde{H}_{i_m j_m k_m m})}{p(C_k)} \text{ и вероятностям } p(\tilde{H}_{i_m j_m k_m m}),$$

рассчитанным для момента времени $t_* + \Delta t$ как результат предыдущего шага алгоритма, и переместить агента туда со скоростью, случайные компоненты, которой вычисляются на основе интенсивностей идентифицированных на шаге 4, если выполнены следующие ограничения:

$$|\mathbf{v}_m| \leq v_{\max}$$

$$p(B|H_{ijk}) \leq p_B$$

Если не выполнено условие 1, установить скорость перемещения $|\mathbf{v}_k|$ равной $v_{\max}$, если не выполнено второе условие, не перемещаться. На текущий такт игры, интервал Δt определяется как максимальное время необходимое для перемещения между центрами соседних ячеек:

$$\ddot{A}t = \max_{m \in \{0, \dots, M-1\}} \frac{\Delta l_m}{|\mathbf{v}_m|}, \text{ где } \Delta l_m = \sqrt{(\Delta l_{\lambda, m})^2 + (\Delta l_{\mu, m})^2 + (\Delta l_{\nu, m})^2}$$

Переходы агентов между состояниями синхронизированы по интервалу Δt который одинаков для всех объектов.

Перейти к следующему по порядку дискретному моменту времени $t_* + \Delta t$, и далее полагать его текущим моментом времени. Перейти к шагу 2.

Шаг 6. Завершить игру.

Задача идентификации решается по методу, предложенному в статье [4].

Выводы

Разработана вероятностная модель поведения прикладной многоагентной системы, которая представляет собой игровое взаимодействие между агентами и целями. Поведение агентов не детерминировано и таким образом непредсказуемо с точки зрения целей. Модель подразумевает применимость в условиях скоординированной или автономной работы агентов, на основе доступности агентам информации о положении друг друга.

Поведение агентов определяется вспомогательным алгоритмом, который включает:

- Идентификация параметров вероятностной модели, используя максимизацию целевой функции, которая учитывает вероятности поражения целей.
- Последовательное движение агентов по игровому полю со скоростями, случайные компоненты которых вычисляются, используя идентифицированные параметры модели.



- Атаку врага в случае превышения вероятности группового или индивидуального поражения цели.

Разработанная модель и алгоритм обеспечивают управление поведением соответствующих прикладных многоагентных систем.

Список источников / References

1. Венгерский алгоритм решения задачи о назначениях [Электронный ресурс] // Вики-конспекты ИТМО — URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%92%D0%B5%D0%BD%D0%B3%D0%B5%D1%80%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%B%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BE_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%D1%85 (дата обращения 05.05.2025)
2. The Hungarian algorithm for solving the assignment problem (In Russ.) [Electronic resource] // Wikiconcepts of ITMO — URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%92%D0%B5%D0%BD%D0%B3%D0%B5%D1%80%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%B%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BE_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%D1%85 (Accessed 05.05.2025)
3. Задача о назначениях. Венгерский алгоритм (алгоритм Куна) [Электронный ресурс] // MAXimal: algo — URL: http://e-maxx.ru/algo/assignment_hungary (дата обращения 05.05.2025)
4. The assignment problem. Hungarian algorithm (Kuhn's algorithm) (In Russ.) [Electronic resource] // MAXimal: algo — URL: http://e-maxx.ru/algo/assignment_hungary (Accessed 05.05.2025)
5. Алгоритм Куна нахождения наибольшего паросочетания [Электронный ресурс] // MAXimal: algo — URL: http://e-maxx.ru/algo/kuhn_matching (дата обращения 05.05.2025)
6. Kuhn's algorithm for finding the largest match (In Russ.) [Electronic resource] // MAXimal: algo — URL: http://e-maxx.ru/algo/kuhn_matching (Accessed 05.05.2025)
7. Kuravsky, L.S., Marmalyuk, P.A., Yuryev, G.A., Dumin, P.N., A numerical technique for the identification of discrete-state continuous-time Markov models, Appl. Math. Sci. 9:379-391, 2015. <http://dx.doi.org/10.12988/ams.2015.410882>.

Информация об авторах

Левонич Никита Ильич, лаборант-исследователь, лаборатория количественной психологии, Московский государственный психолого-педагогический университет (ФГБОУ ВО МГППУ), Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0002-8580-0490>, e-mail: levonikitech@yandex.ru

Information about the authors

Nikita I. Levonovich, Laboratory Research Assistant, Laboratory of Quantitative Psychology, Moscow State University of Psychology and Education, Moscow, Russian Federation, ORCID: <https://orcid.org/0000-0002-8580-0490>, e-mail: levonikitech@yandex.ru

Поступила в редакцию 25.05.2025

Поступила после рецензирования 26.05.2025

Принята к публикации 06.06.2025

Опубликована 30.06.2025

Received 2025.05.25

Revised 2025.05.26

Accepted 2025.06.06

Published 2025.06.30