

## КОМПЬЮТЕРНОЕ ОБЕСПЕЧЕНИЕ И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

### COMPUTER ENGINEERING AND SOFTWARE

Научная статья  
УДК 004.81  
<https://doi.org/10.24143/2072-9502-2025-2-49-59>  
EDN MDRTXD

#### **Метод первоначального формирования и последующей адаптации архитектуры многоагентной системы, основанный на применении генетического алгоритма**

**Юрий Владимирович Дубенко<sup>1</sup>, Евгений Евгеньевич Дышкант<sup>2✉</sup>,  
Владислав Анатольевич Демидов<sup>3</sup>**

<sup>1, 3</sup>Кубанский государственный технологический университет,  
Краснодар, Россия

<sup>2</sup>Армавирский механико-технологический институт (филиал)  
ФГБОУ ВО «Кубанский государственный технологический университет»,  
Армавир, Россия, [ed0802@yandex.ru](mailto:ed0802@yandex.ru)✉

**Аннотация.** Архитектура многоагентной системы (МАС) представляет собой совокупность ролей и связей агентов, а также множества базовых правил, определяющих их генезис. Целью данной работы является разработка метода, обладающего следующими характеристиками: возможность модификации базовых правил, определяющих условия установления и разрыва вертикальных и горизонтальных связей между агентами; возможность автоматической адаптации архитектуры (в процессе решения поставленной задачи) к изменениям значений показателей окружающей среды, а также в количественном и качественном составе множества агентов; возможность моделирования более трех архитектур МАС. Разработанный метод первоначального формирования и последующей адаптации архитектуры МАС полностью удовлетворяет обозначенным характеристикам. Определение оптимального множества базовых параметров МАС производится с помощью «генетического алгоритма» путем обучения на виртуальной модели окружающей среды. Существует возможность моделирования архитектур МАС следующих типов: «коалиция», «команда», «федерация», «конгрегация». Проведенная экспериментальная оценка подтвердила эффективность разработанного метода по показателю «общее время выполнения поставленной задачи» в сравнении со следующими подходами: применение «генетического алгоритма» к «хромосоме», содержащей множество вертикальных связей между агентами (изменение архитектуры МАС возможно лишь на новой итерации); множество вертикальных связей фиксировано и на каждой новой итерации остается неизменным. Преимущество предложенного подхода в сравнении с аналогами заключается в возможности автоматической адаптации архитектуры МАС в процессе решения поставленной задачи (а не с началом новой итерации), с адаптацией к вариациям множества агентов либо к новым значениям параметров окружающей среды.

**Ключевые слова:** многоагентная система, архитектура, вертикальная связь, горизонтальная связь, агент, адаптация, генетический алгоритм

**Для цитирования:** Дубенко Ю. В., Дышкант Е. Е., Демидов В. А. Метод первоначального формирования и последующей адаптации архитектуры многоагентной системы, основанный на применении генетического алгоритма // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2025. № 2. С. 49–59. <https://doi.org/10.24143/2072-9502-2025-2-49-59>. EDN MDRTXD.

Original article

## The method of initial formation and subsequent adaptation of a multi-agent system architecture based on the application of a genetic algorithm

Iurii V. Dubenko<sup>1</sup>, Evgenii E. Dyshkant<sup>2✉</sup>, Vladislav A. Demidov<sup>3</sup>

<sup>1,3</sup>Kuban State Technological University,  
Krasnodar, Russia

<sup>2</sup>Armavir Institute of Mechanical and Technological (branch) FSBEI HE "Kuban State Technological University",  
Armavir, Russia, [ed0802@yandex.ru](mailto:ed0802@yandex.ru)

**Abstract.** The architecture of a multi-agent system (MAS) is a set of roles and relationships of agents, as well as a set of basic rules that determine their genesis. The objective of this work is to develop a method with the following characteristics: the ability to modify the basic rules that determine the conditions for establishing and breaking vertical and horizontal links between agents; the ability to automatically adapt the architecture (in the process of solving the problem) to changes in the values of environmental indicators, as well as in the quantitative and qualitative composition of the set of agents; the ability to model more than three MAS architectures. The developed method of initial formation and subsequent adaptation of the architecture of a multi-agent system fully satisfies the specified characteristics. The optimal set of basic parameters of a MAS is determined using a "genetic algorithm" by training on a virtual model of the environment. It is possible to model the following types of MAS architectures: "coalition", "team", "federation", "congregation". The conducted experimental evaluation confirmed the efficiency of the developed method in terms of the "total time to complete the assigned task" indicator in comparison with the following approaches: application of the "genetic algorithm" to the "chromosome" containing a set of vertical connections between agents (the architecture of the MAS can be changed only at a new iteration); the set of vertical connections is fixed and remains unchanged at each new iteration. The advantage of the proposed approach in comparison with its analogues is the ability to automatically adapt the architecture of the MAS in the process of solving the assigned task (and not at the beginning of a new iteration), adapting to variations in the set of agents or to new values of the environmental parameters.

**Keywords:** multi-agent system, architecture, vertical communication, horizontal communication, agent, adaptation, genetic algorithm

**For citation:** Dubenko Iu. V., Dyshkant E. E., Demidov V. A. The method of initial formation and subsequent adaptation of a multi-agent system architecture based on the application of a genetic algorithm. *Vestnik of Astrakhan State Technical University. Series: Management, computer science and informatics*. 2025;2:49-59. (In Russ.). <https://doi.org/10.24143/2072-9502-2025-2-49-59>. EDN MDRTXD.

### Введение

Архитектура многоагентной системы (МАС) представляет совокупность ролей и связей агентов, а также множества базовых правил, определяющих их генезис [1]. Очевидно, что от выбора архитектуры и ее способности к адаптации к изменениям окружающей среды в существенной мере зависит производительность МАС. Существующие подходы к формированию архитектуры МАС могут быть классифицированы следующим образом:

1) по глубине модификации (под модификацией будем понимать изменение параметров архитектуры МАС с целью ее адаптации к текущим условиям окружающей среды):

а) модификация множества связей между агентами в зависимости от состава множества агентов, поставленной задачи, текущих параметров окружаю-

щей среды [2–6]. При этом корректировке подвергается именно древовидная структура, описывающая множество связей агентов МАС, а базовые правила, описывающие, например, условия возникновения или разрыва этих связей, остаются неизменными;

б) модификация множества базовых правил, описывающих фундаментальные условия функционирования и взаимодействия агентов МАС [7, 8]. При этом установление и разрыв связей между агентами осуществляется в автоматическом режиме в зависимости от их состояния, параметров окружающей среды, условий поставленной задачи;

2) по уровню адаптивности:

а) статическая архитектура [2, 4, 5]. Модификация архитектуры производится по результатам выполнения поставленной задачи после останова процедуры функционирования МАС;

б) динамическая архитектура [3, 6–8]. Связи между агентами МАС могут изменяться в процессе решения поставленной задачи;

3) по степени масштабируемости:

а) наличие возможности интеграции новых агентов в процессе работы МАС [3, 7];

б) отсутствие возможности приема новых агентов в процессе функционирования МАС [2, 4, 6, 8];

4) по числу моделируемых архитектур (например, «команда» – предусматривается наличие горизонтальных и вертикальных связей между агентами, максимизация группового выигрыша; «коалиция» – наличие только горизонтальных связей, максимизация индивидуального, а не группового выигрыша; «федерация» – члены группы взаимодействуют только с агентом-координатором, т. е. разрешены только вертикальные связи; «конгрегация» – формирование долговременных структур, остающихся неизменными в процессе решения поставленной задачи [5]):

а) одна [2–6];

б) две [7, 8];

в) три и более (в настоящее время отсутствуют работы, удовлетворяющие данному критерию).

Очевидно, что повышение вариативности при выборе архитектур МАС будет способствовать большей гибкости при адаптации к различным условиям окружающей среды. Таким образом, *целью данной работы* является разработка метода первоначального формирования и последующей адаптации архитектуры МАС, обладающего следующими характеристиками: по глубине модификации – модификация множества базовых правил, описывающих фундаментальные условия функционирования и взаимодействия агентов МАС; по уровню адаптивности – динамическая архитектура; по степени масштабируемости – наличие возможности интеграции новых агентов в процессе работы МАС; по числу моделируемых архитектур – три и более.

### Постановка задачи и методы

Дано:  $S$  – МАС, функционирующая в двух режимах: обучения (цель – изучение окружающей среды), основной (цель – максимизация производительности);  $X = \{A_\tau, T_\tau\}$  – множество входных параметров, где  $T_\tau$  – параметры поставленной задачи  $T$  в момент времени  $\tau$ ;  $A_\tau$  – множество интеллектуальных агентов МАС в момент времени  $\tau$ ,  $A_\tau = \{a_i\}, i = \overline{1, |A_\tau|}$ , при этом агенты гетерогенны по техническим характеристикам (параметры вычислительной подсистемы, подсистемы машинного зрения, коммуникационной подсистемы, постоянного запоминающего устройства и т. д.);  $\tau \in \{\tau_0, \tau_1 = \tau_0 + \Delta\tau, \dots, \tau_N\}$ , где  $\Delta\tau$  – шаг дискретизации параметра времени,  $\tau_0$  – время

начала выполнения задачи  $T$ ,  $\tau_N$  – время завершения выполнения задачи  $T$ ;  $Z = \{O_{S_\tau}\}$  – множество внутренних параметров  $S$ , описывающих ее архитектуру  $O_{S_\tau}$  в момент времени  $\tau$ ;  $Y$  – множество выходных параметров, описывающих результат выполнения множеством агентов  $A$  поставленной задачи  $T$ ;  $E = \{E_\tau\}$  – множество параметров окружающей среды, где  $E_\tau$  – значения параметров окружающей среды в момент времени  $\tau$ ;  $Q = \{t, R_p\}$  – множество показателей эффективности системы, где  $t$  – общее время выполнения поставленной задачи  $T$  (глобальный параметр эффективности, фиксируется в момент времени  $\tau_N$ ),  $R_p$  – усредненное вознаграждение по всем траекториям  $p_{a_i}$  [9], формируемым агентами, определяемое по формуле

$$R_p = \frac{\sum_{i=1}^{|A_\tau|} \left( \sum_{k=1}^{N_{p_{a_i}}} \gamma(k) r_k \right)}{|A_\tau|},$$

где под траекторией  $p_{a_i}$  понимается множество кортежей вида

$$\langle S_{k-1}, A_k, S_k, r_k \rangle, \quad (1)$$

где  $A_k$  – примитивное действие, выполнение которого агентом  $a_i$  привело к переходу из состояния  $S_{k-1}$  в состояние  $S_k$ ;  $r_k$  – полученное при этом вознаграждение,  $r_k \in [-1, 1]$ ;  $N_{p_{a_i}}$  – число кортежей, составляющих траекторию агента  $a_i$ ;  $\gamma$  – «коэффициент обесценивания»,  $\gamma(k) \in [0, 1]$  [9].

Таким образом, необходимо разработать метод  $M$ , предназначенный для первоначального формирования (на момент времени  $\tau_0$ ) и последующей модификации (для  $\tau > \tau_0 \wedge \tau \leq \tau_N$ ) архитектуры МАС:

$$M : \{S, X, Z, Y, E, Q\} \rightarrow \{\Delta t | \Delta t > 0\} \quad (2)$$

при условии

$$\left( (A_\tau - A_{\tau-1} > \delta_A \vee E_\tau - E_{\tau-1} > \delta_E) \wedge R_p \downarrow \right) \rightarrow \Delta O_{S_\tau} \neq \emptyset,$$

где  $\delta_A$  – характеристика изменения состава множества агентов  $A$  на интервале  $(\tau-1, \tau)$ ;  $\delta_E$  – характеристика изменения параметров окружающей среды  $E$  на интервале  $(\tau-1, \tau)$ ;  $\Delta O_{S_\tau}$  – модифицирующее воздействие на  $O_{S_\tau}$  с целью компенсации отклонения  $\delta_R$ ;  $\Delta t = t^b - t^a$ ,  $M$ ;  $t^b$  – значение показателя  $t \in Q$  до применения метода  $M$ ;  $t^a$  – значение показателя  $t \in Q$  после применения метода  $M$ .

При этом агенты  $a_i \in A$  имеют следующие характеристики:

1. Число «бонусов»  $B_{a_i}$  – оценка успешности последних действий агента в рамках выполнения поставленной задачи  $T$ :

$$B_{a_i} = \sum_{\tau=\tau_0}^{\tau_{now}} r_{\tau}^{T_k} k_{obl}^B(\tau), \quad (3)$$

где  $\tau_{now}$  – текущий момент времени;  $r_{\tau}^{T_k}$  – вознаграждение агента за решение некоторой подзадачи  $T_k$  в момент времени  $\tau$  ( $T_k \in T$ ,  $k = \overline{1, N_T}$ ,  $N_T$  – число подзадач, выделенных из  $T$ ),  $r_{\tau}^{T_k} \in [-1, 1]$ ;  $k_{obl}^B(\tau)$  – функция, реализующая снижение вклада вознаграждений, полученных ранее, в величину  $B_{a_i}$ ,

$$k_{obl}^B(\tau) = \begin{cases} \frac{1}{1 + e^{-\beta_{obl}^B \tau}} & \text{при } \tau > \tau_0, \\ 0 & \text{при } \tau = \tau_0 \end{cases}, \quad (4)$$

где  $\beta_{obl}^B \geq 0$  – коэффициент, определяющий скорость «забывания» вознаграждений, полученных ранее. Данная характеристика является индикатором как технического состояния, так и востребованности агента (если агент обладает небольшим числом «бонусов», то он либо имеет технические неполадки и не справляется с заданием, либо оно ему не было назначено и он является лишним). Каждый агент стремится к максимизации данного показателя, т. е.

$$B_{a_i} \rightarrow \max.$$

*Примечание:* агент, длительное время имеющий нулевой показатель «бонусов», исключается из МАС и отзывается на базу.

2. «Престиж»  $\varepsilon_{a_i}$  – показатель успешности действий агента с начала выполнения поставленной задачи (чем больше «бонусов» имел агент в каждый момент времени  $t$ , тем он обладает большим «престижем»):

$$\varepsilon_{a_i} = \frac{\sum_{t=\tau_0}^{\tau_{now}} B_{a_i,t} k_{obl}^{\varepsilon}(t)}{\tau_{now}}, \quad (5)$$

где учитываются только значения  $B_{a_i,t}$  при  $k_{obl}^B|_{\tau \in [\tau_0, t]}$ ,  $k_{obl}^{\varepsilon}(t)$  определяется по формуле, аналогичной (4),  $\beta_{obl}^{\varepsilon} \geq 0$ . «Престиж» является индикатором стабильности успешных действий агента.

3. Территориальные ограничения:

а) «личная зона» – окрестность агента, представляющая окружность с радиусом  $R_{private}$  (очерчивает границы, нарушение которых иными объек-

тами влечет существенный риск для физической целостности агента);

б) «социальная зона» – окружность с радиусом  $R_{social}$  ( $R_{private} < R_{social}$ ), в пределах которой допускаются формирование агентами горизонтальных связей;

в) «публичная зона» – окружность с радиусом  $R_{public}$  ( $R_{private} < R_{public}$ ), в пределах которой допускается формирование агентами вертикальных связей.

4. Множество связей  $R_{a_i}$ , при этом под связью понимается наличие коммуникационного взаимодействия между двумя агентами и выделяются следующие типы связей:

а) *горизонтальные связи* – коммуникационное взаимодействие двух равноправных агентов, которое может заключаться в следующем: репликация опыта (под опытом понимается множество коротких (1)); арбитраж «коллизий» (спорных ситуаций, например, когда два агента планируют переход в одну и ту же область окружающей среды, что может повлечь их столкновение);

б) *вертикальные связи* – коммуникационное взаимодействие двух неравноправных агентов (*агент-подчиненный* и доминирующий над ним *агент-координатор*): трансляция агенту-подчиненному подзадач  $T_k$ , по результатам выполнения которых ему назначается вознаграждение  $r_{\tau}^{T_k}$ ; арбитраж «коллизий» между агентами-подчиненными.

При этом для множества связей  $R_{a_i}$  могут быть установлены следующие ограничения (распространяются на всех агентов множества  $A_i$ ):

1)  $P_{hor}$  – вероятность установления горизонтальных связей (за исключением арбитража коллизий);

2)  $P_{vert}$  – вероятность установления вертикальных связей;

3) применяемые правила арбитража «коллизий» (ситуации, при которых действия, планируемые к выполнению, влекут попадание агентов в личные зоны друг друга) для равноправных агентов (может быть разрешено только одно из перечисленных)  $R_{coll}$ :

а) учет «престижа» агентов, непосредственно участвующих в «коллизии»:

– решение выносится в пользу агента, имеющего больший «престиж» (стимуляция «расслоения» агентов по числу «бонусов») ( $R_{more \varepsilon}^{eq}$ );

– решение выносится в пользу агента, имеющего меньший «престиж» (попытка выровнять показатели производительности агентов) ( $R_{less \varepsilon}^{eq}$ );

б) учет «престижа» агентов вышестоящего уровня, с которыми участники «коллизии» установили вертикальные связи:

– решение выносится в пользу агента-подчиненного, имеющего вертикальную связь с агентом-

координатором, обладающим более высоким «престижем» ( $R_{more\ \epsilon}^{uneq}$ );

– решение выносится в пользу агента-подчиненного, имеющего вертикальную связь с агентом-координатором, имеющим меньший «престиж» ( $R_{less\ \epsilon}^{uneq}$ ).

*Примечание:* при арбитраже коллизий между неравноправными агентами (агент-подчиненный и агент-координатор) решение всегда выносится в пользу доминирующего агента.

При установлении вертикальных связей между агентами  $a_i, a_j$  ( $j = \overline{1, |A_i|}$ ) применяются следующие правила:

а) *свободный агент*  $a_i$  (т. е. агент, непосредственно подчиненный лицу, принимающему решения) получает возможность установления вертикальных связей с другими свободными агентами (например,  $a_j$ ), находящимися в его публичной зоне, в случае выполнения следующего условия:

$$\frac{\sum_{t=\tau_{now}-\tau_{obs}}^{\tau_{now}} \epsilon_{a_i t}}{\tau_{obs}} > \epsilon_{threshold}, \quad (6)$$

где  $\tau_{obs}$  – величина временного интервала, на котором учитывается показатель «престижа»;  $\epsilon_{threshold}$  – некоторое пороговое значение «престижа»,  $\epsilon_{threshold} > 0$ ;

б) на вероятность установления вертикальной связи также оказывает влияние расстояние между потенциальными агентом-координатором и агентом-подчиненным:

$$P_{vert\_total} = P_{vert} k_{vert}^S, \quad (7)$$

где  $P_{vert}$  – вероятность установления вертикальной связи между агентами  $a_i, a_j$  при расстоянии между ними  $S(a_i, a_j) = R_{private}$ ;  $k_{vert}^S$  – поправочный коэффициент, позволяющий изменять величину  $P_{vert}$  в зависимости от  $S(a_i, a_j) \in (R_{private}, R_{public}]$ ,  $k_{vert}^S : S(a_i, a_j) \rightarrow [0, 1]$ ,  $k_{vert}^S$  монотонно убывает на интервале  $(R_{private}, R_{public})$  и определяется по формуле, аналогичной (4), при  $\beta_{vert}^S < 0$ ;

в) вероятность разрыва агентом-подчиненным вертикальной связи и перехода в разряд свободных агентов (данный механизм подробно описывается в работе [3]):

$$P_{transfer} = k_B k_{transfer}^S, \quad (8)$$

где  $k_B : B_{a_i} \rightarrow [0, 1]$ ,  $k_B$  монотонно убывает на интер-

вале  $[0, B_{sup}]$  и определяется по формуле, аналогичной (4), при  $\beta_B < 0$ ,  $B_{sup}$  – величина «бонусов» агента-починенного, при достижении которой он теряет желание разорвать вертикальную связь с агентом-координатором;  $k_{transfer}^S : S(a_i, a_j) \rightarrow [0, 1]$ ,  $k_{transfer}^S$  монотонно возрастает на интервале  $[R_{private}, R_{public}]$  и определяется по формуле, аналогичной (4), при  $\beta_{transfer}^S > 0$ ;

г) факторы, стимулирующие свободных агентов к созданию вертикальных связей. При определении числа «бонусов»  $B_{a_i}$  (см. (3)) учитывается подкрепление, полученное за реализацию подзадачи  $T_k \in T$ . Таким образом, если задача  $T$  будет назначена некоторому свободному агенту  $a_j$ , он получит вознаграждение  $r_{\tau}^T$  непосредственно за ее решение ( $r_{\tau}^T = 1$ ). Однако та же самая задача  $T$  может быть назначена агенту-координатору  $a_i$ , который выполнит ее декомпозицию на подзадачи  $T_k$ , часть из них, например числом  $N_{a_j}$  ( $N_{a_j} < N_T$ ), может быть назначена для реализации агенту  $a_j$  (в данном случае имеет роль агента-подчиненного), который суммарно может получить вознаграждение  $r_{\tau}^{T_{k, \dots, T_{k+N_{a_j}-1}} \max} = \sum_{q=k}^{k+N_{a_j}-1} r_{\tau}^{T_q}$ , т. е. при  $N_{a_j} > 1$   $r_{\tau}^{T_{k, \dots, T_{k+N_{a_j}-1}} \max} > r_{\tau}^T$ . Следовательно, за выполнение подзадач  $T_{k, \dots, T_{k+N_{a_j}-1}$  агент-подчиненный может получить больше «бонусов», чем за выполнение задачи  $T$  получит свободный агент.

*Примечание:* данный механизм будет стимулировать свободных агентов образовывать вертикальные связи. Однако если МАС требуется сохранить в децентрализованном виде, может быть установлен запрет на создание вертикальных связей путем обнуления показателя  $P_{vert}$ .

Следующие правила применяются при установлении горизонтальных связей между агентами  $a_i, a_j$ :

а) агенты, расположенные в пределах «социальной» зоны друг друга, обмениваются информацией о намерениях (о планируемых действиях) либо полученным опытом. При этом агенты обязаны принимать отправленные в их адрес кортежи опыта (1);

б) вероятность обмена опытом между агентами (характеризует желание агента передавать свой опыт другим) определяется по формуле

$$P_{hor\_total} = P_{hor} k_{hor}^S, \quad (9)$$

где  $P_{hor}$  – вероятность установления горизонтальной связи между агентами  $a_i, a_j$ ; с целью обмена опытом при расстоянии между ними  $S(a_i, a_j) = R_{private}$ ;  $k_{hor}^S$  – поправочный коэффициент, позволяющий изменять величину  $P_{hor}$  в зависимости от  $S(a_i, a_j) \in (R_{private}, R_{social}]$ , определяется по формуле, аналогичной (4), при этом  $\beta_{hor}^S < 0$ .

$$H_{O_s} = \{ \beta_{obl}^B, \beta_{obl}^E, \beta_{vert}^S, \beta_{hor}^S, \beta_{transfer}^S, B_{sup}, \beta_B, R_{private}, R_{social}, R_{public}, P_{hor}, P_{vert}, \tau_{obs}, \varepsilon_{threshold}, R_{coll} \}, \quad (10)$$

где  $\beta_{obl}^B$  – см. (4);  $\beta_{obl}^E$  – см. (5);  $\tau_{obs}, \varepsilon_{threshold}$  – см. (6);  $P_{vert}, \beta_{vert}^S$  – см. (7);  $B_{sup}, \beta_B, \beta_{transfer}^S$  – см. (8);  $P_{hor}, \beta_{hor}^S$  – см. (9);  $R_{private}, R_{social}, R_{public}$  – радиусы личной, социальной, публичной зон агента соответственно;  $R_{coll}$  – применяемое правило арбитража «коллизий»,  $R_{coll} \in \{ R_{more \varepsilon}^{eq}, R_{less \varepsilon}^{eq}, R_{more \varepsilon}^{uneq}, R_{less \varepsilon}^{uneq} \}$ .

Очевидно, что

$$\forall A, T, E: H_{O_s}^{learn} \neq H_{O_s}^{main},$$

где  $H_{O_s}^{main}, H_{O_s}^{learn}$  – соответственно, вид множества (10) в основном режиме и режиме обучения [7].

На основании (1) может быть определена ценность выбора действия  $A_k$  в состоянии  $S_{k-1}$ , которая может быть выражена в значении Q (Q-Learning) [9]. В целях побуждения агентов к исследованию окружающей среды вероятность выбора агентом пары

$$\langle S_{k-1}, A_k \rangle: Q \rightarrow \max \quad (11)$$

может быть обозначена как  $\pi \in [0, 1]$ , при этом по мере обучения МАС она может возрастать:

$$\pi = \begin{cases} \pi_{min} & \text{при } R_p \leq R_{\pi_{min}}, \\ \frac{1}{1 + e^{-\beta_{\pi} R_p}} & \text{при } R_p \in (R_{\pi_{min}}, R_{plearn}), \end{cases} \quad (12)$$

где  $\pi_{min}$  – минимальная вероятность выполнения условия (11) при выборе действия;  $R_{\pi_{min}}$  – максимальное значение усредненного вознаграждения по всем траекториям  $p_{a_i}$ , при котором  $\pi = \pi_{min}$ ;  $R_{plearn}$  – величина показателя  $R_p$ , при которой активируется режим обучения либо происходит переход в основной режим  $\beta_{\pi} > 0$ . Дополним множество (10) параметрами из (12):

$$H_{O_s}^{learn} = \{ H_{O_s}^{main}, \pi_{min}, R_{\pi_{min}}, R_{plearn}, \beta_{\pi} \}, \quad (13)$$

*Примечание:* параметр  $P_{hor}$  позволяет регулировать интенсивность информационного потока с целью обмена опытом. Его значение может быть увеличено при функционировании МАС в режиме обучения (с целью интенсификации обмена опытом между агентами для ускорения обучения).

Таким образом, архитектура МАС  $O_s$  может быть описана следующим множеством:

при этом будем считать, что  $H_{O_s}^{main} = H_{O_s}$ , где  $H_{O_s}$  – см. (10).

Ниже представлен алгоритм функционирования рассматриваемой МАС:

1) формирование  $O_s^{learn}$  – архитектура МАС в режиме обучения, соответствующая  $H_{O_s}^{learn}$ ;

2) МАС работает в режиме обучения, пока

$$R_p < R_{plearn}, \quad (14)$$

в ином случае активируется основной режим;

3) формирование  $O_s^{main}$  – архитектура МАС в основном режиме, соответствующая  $H_{O_s}^{main}$ ;

4) МАС работает в основном режиме, пока

$$R_p > R_{plearn}, \quad (15)$$

иначе снова происходит переход в режим обучения;

5) шаги 1–4 повторяются, пока не будет решена поставленная задача  $T$ .

Следует отметить, что выражение  $O_s^{learn} = H_{O_s}^{learn} \wedge O_s^{main} = H_{O_s}^{main}$  не является истинным.

Множества (10), (13) лишь задают базовые правила формирования и функционирования МАС. При этом построенная на их основе архитектура может иметь большое множество вариаций, зависящих как от состава множества агентов  $A$ , так и от текущих параметров окружающей среды  $E$ , а также условий задачи  $T$ . Таким образом, связь множеств (10), (13) и архитектуры МАС может быть описана следующим выражением:

$$O_s^{learn} \Leftrightarrow H_{O_s}^{learn} \wedge O_s^{main} \Leftrightarrow H_{O_s}^{main}.$$

При переходе между режимами происходит смена базовых правил формирования и функционирования МАС, описанных в (10), (13), что возможно в случае достижения целей обучения МАС либо при изменениях состава множества агентов

или параметров окружающей среды, влекущих существенное падение величины среднего вознаграждения  $R_p$ , получаемого агентами (т. е. при ложности условий (14) или (15)). Таким образом, будем расценивать условие, представленное в формуле (2), как применяемое в рамках одного режима работы МАС при истинности условий (14) или (15). В этом случае модифицирующее воздействие  $\Delta O_{st}$  вырабатывается автоматически в рамках одного режима и заключается в формировании (или разрыве) вертикальных или горизонтальных связей между агентами.

Для определения значений множеств  $H_{O_s}^{learn}$ ,  $H_{O_s}^{main}$ , позволяющих оптимизировать работу МАС по глобальному показателю эффективности  $t$ , применим «генетический алгоритм» со следующими параметрами:

- 1) «хромосома» объединяет множества (10), (13);
- 2) функция приспособленности:  $t \rightarrow \min$  (общее время выполнения поставленной задачи  $T$ );

3) применяемые «генетические операторы»: «скрещивание» методом рулетки с использованием «видовых стратегий» (препятствование «скрещиванию» между особями с близкими «хромосомами»), «равномерная мутация» [10].

### Результаты

*Цель эксперимента:* в результате выполнения компьютерного моделирования доказать эффективность применения предложенного метода первоначального формирования и последующей адаптации архитектуры МАС, основанного на применении генетического алгоритма.

*Описание решаемой задачи:* достижение агентами в лабиринте всех сгенерированных целевых состояний.

*Описание внешней среды:* виртуальный лабиринт, сформированный в среде Microsoft Unity, размерностью  $150 \times 150$  (фрагмент представлен на рис. 1).

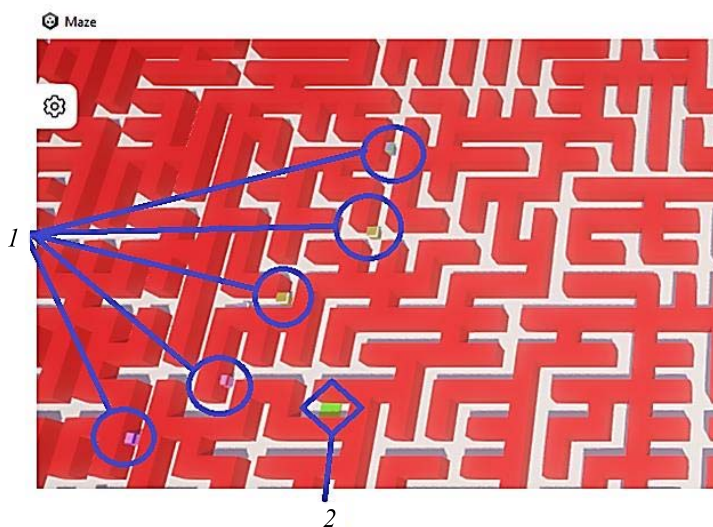


Рис. 1. Фрагмент виртуального лабиринта: 1 – агенты; 2 – целевое состояние

Fig. 1. A fragment of the virtual maze: 1 – agents; 2 – target state

Под единицей размерности понимается клетка, которая может иметь одно из следующих состояний: проходимая, непроходимая (в этом случае соответствует квадратному блоку, применяемому для формирования стен). В лабиринте не допускаются замкнутые контуры, т. е. все проходимые клетки должны быть достижимы для агентов. Структура лабиринта меняется случайным образом (изменение типа одной случайной клетки происходит с периодичностью в 5 с, при этом периодически допускается одновременное изменение состояния на противоположное для 30 % клеток лабиринта). Одновременно на карте отображаются только

5 целевых состояний, при этом обнаруженное агентами целевое состояние заменяется на новое. Общее число целей, генерируемых за все время выполнения задачи, составляет 100. Координаты целевых состояний определяются случайным образом. Периодически выполняется ограничение мобильности (обездвижение) случайных агентов (на 10–30 с), при этом одновременно может быть обездвижено не более 20 % агентов. Версии окружающей среды, включая все флуктуации в структуре лабиринта и мобильности агентов, абсолютно одинаковы для обоих тестируемых методов (как для предложенного в данной работе, так и для его аналога [2]).



Число, типы применяемых агентов: всего 40 ед. Виртуальные агенты, имитирующие мобильных роботов на колесной транспортной платформе, оснащенных видеокамерой [3], генерируются в трех комплектах (в зависимости от возможностей вычислительной системы): агенты, имеющие наиболее производительную вычислительную систему (потенциальные агенты-координаторы) – 3 ед.; агенты, имеющие вычислительную систему со средней производительностью – 12 ед. (могут быть как агентами-координаторами, так и агентами-подчиненными); агенты, имеющие вычислительную систему с низкой производительностью – 25 ед. (потенциальные агенты-подчиненные, вероятность выдвижения такого агента в агенты-координаторы очень низкая). Моделирование различных уровней производительности вычислительной системы осуществлялось путем применения задержки при выполнении алгоритмов функционирования агентов.

*Механизм выбора стартовых положений агентов:* координаты стартовых положений генерируются случайным образом на каждой итерации.

*Механизм выбора целевых состояний:* координаты целевых состояний генерируются случайным образом на каждой итерации.

*Критерий, позволяющий утверждать об успешном решении задачи:* обнаружение всех целевых состояний.

*Число итераций (попыток):* 300. При этом итерация считается выполненной в случае достижения

критерия решения задачи.

*Имитация загрузки вычислительной системы агента-координатора:* при назначении задач, а также при расчете подкреплений на каждого агента-подчиненного генерировалась задержка в 0,3 с [3].

*Имитация задержки в канале связи в зависимости от расстояния:* агент-подчиненный получал информационные сообщения от агента-координатора с задержкой из расчета 0,2 с на 5 клеток прямого расстояния между ними [3].

*Критерий оценки эффективности метода:* время выполнения одной итерации.

*Количество «хромосом», генерируемых на каждой итерации:* 100.

Результаты, полученные в ходе эксперимента, представлены на рис. 2 (оценка выполнялась по показателю  $t$  – общее время выполнения поставленной задачи  $T$ ), при этом показаны результаты для лучшей «хромосомы» в популяции, приняты следующие условные обозначения: «Аналог [2]» – алгоритм, описанный в [2] («генетический алгоритм», «хромосома» содержит множество вертикальных связей между агентами); «Связи агентов – статичны» – алгоритм с вертикальными связями в статике (5 агентов-координаторов, 35 агентов-подчиненных, по 8 агентов в подчинении у одного агента-координатора), тип архитектуры МАС – «конгрегация» [1]; во всех рассматриваемых алгоритмах функционирование агентов основано на парадигме «обучения с подкреплением».

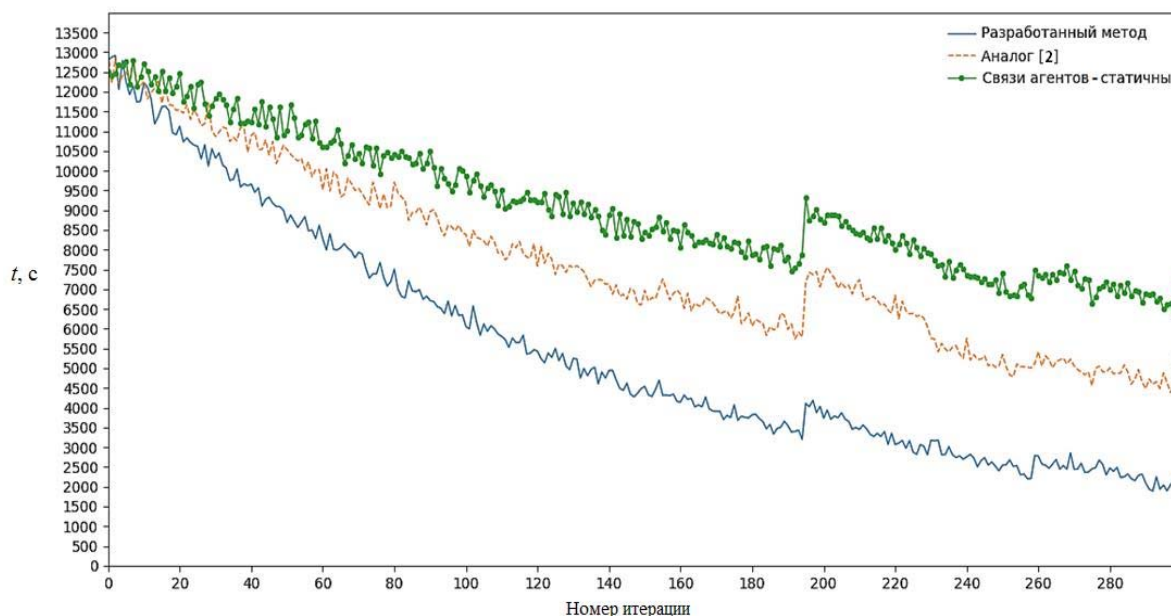


Рис. 2. Результаты эксперимента

Fig. 2. Experimental results



### Обсуждение

Как видно на рис. 2, методы, предусматривающие последовательную адаптацию множества связей агентов к условиям окружающей среды, показали лучшие результаты, чем метод, предполагающий неизменную структуру этого множества, для которого уменьшение показателя «общее время выполнения поставленной задачи» происходило лишь за счет повышения степени изученности окружающей среды на каждой новой итерации. В частности, метод, описанный в [2], уже на 100-й итерации показал результат, лучший на 13 %. Аналогичный результат для разработанного метода составил 37 %, что объясняется его гибкостью – в отличие от [2] архитектура МАС может автоматически изменяться в процессе решения поставленной задачи (а не с началом новой итерации), адаптируясь к вариациям множества агентов либо к новым значениям параметров окружающей среды.

Также была достигнута цель данной работы, а именно разработка метода первоначального формирования и последующей адаптации архитектуры МАС, обладающего следующими характеристиками:

1) по глубине модификации – модификация множества базовых правил, описывающих фундаментальные условия функционирования и взаимодействия агентов МАС. Имеется возможность модификации показателей, определяющих вероятность возникновения или разрыва вертикальных и горизонтальных связей различных видов, в том числе в зависимости от показателей работы самих агентов (число «бонусов», уровень «престижа»). Базовые параметры архитектур МАС задаются в «хромосомах». «Хромосома», содержащая оптимальные параметры архитектуры МАС для окружающей среды конкретного типа, определяется с помощью «генетического алгоритма» в процессе обучения на виртуальной модели окружающей среды («модель мира»);

2) по уровню адаптивности – динамическая архитектура. Агенты могут автоматически устанавливать и разрывать вертикальные и горизонтальные связи в процессе решения задачи;

3) по степени масштабируемости – наличие возможности интеграции новых агентов в процессе работы МАС. Разработан механизм, позволяющий выполнять перестроение множества связей в случае выхода агента из строя (включая агента-координатора – в этом случае агенты-подчиненные, не получая подзадачи для выполнения, просто

разорвут с ним вертикальные связи), в случае добавления в МАС новых агентов (новый агент сначала получает роль «свободный агент», далее, в зависимости от базовых параметров МАС, например, может установить вертикальную связь с агентом-координатором);

4) по числу моделируемых архитектур – 3 и более. Ниже приводятся достаточные значения параметров «хромосом», определяющих следующие типы архитектур (приведены согласно [1]):

а) «коалиция» (наличие только горизонтальных связей, максимизация индивидуального выигрыша): при  $P_{vert} = 0$ ,  $R_{social} = R_{public}$  (см. (10)) (запрет вертикальных связей);

б) «команда» (наличие горизонтальных и вертикальных связей, максимизация группового выигрыша): при  $P_{hor} > 0$ ,  $P_{vert} > 0$  (см. (10)).

в) «федерация» (члены группы взаимодействуют только с агентом-координатором): при  $P_{hor} = 0$ ,  $R_{social} = R_{private}$  (запрет горизонтальных связей),  $R_{coll} = \{R_{more \ \varepsilon}^{uneq} | R_{less \ \varepsilon}^{uneq}\}$  (агенты-подчиненные не обмениваются напрямую информацией с целью арбитража «коллизий») (см. (10)).

г) «конгрегация» (формирование долговременных структур): при  $B_{sup} = 0$  (вероятность разрыва вертикальных связей всегда будет равна нулю) (см. (10)).

### Заключение

В результате выполнения данной работы был разработан метод первоначального формирования и последующей адаптации архитектуры много-агентной системы (МАС), основанный на применении генетического алгоритма, обладающий следующими характеристиками: по глубине модификации – модификация множества базовых правил, описывающих фундаментальные условия функционирования и взаимодействия агентов МАС; по уровню адаптивности – динамическая архитектура МАС; по степени масштабируемости – наличие возможности интеграции новых агентов в процессе работы МАС; по числу моделируемых архитектур – 3 и более. Разработанный метод может быть использован при управлении МАС, применяемых в таких областях, как мониторинг состояния инфраструктурных объектов, складская логистика и доставка грузов, выполнение сельскохозяйственных работ (сбор урожая, внесение удобрений, обработка сельскохозяйственных культур от вредителей).

### Список источников

1. Horling B., Lesser V. A Survey of Multi-Agent Organizational Paradigms // *The Knowledge Engineering Review*. 2005. N. 19 (04). P. 281–316.  
2. Ling Yu, Zhiqi Shen, Chunyan Miao, Victor Lesser V. Genetic Algorithm Aided Optimization of Hierarchical Multi-Agent System Organization // *10th International Confer-*

*ence on Autonomous Agents and Multiagent Systems (AAMAS 2011) (Taipei, Taiwan, May 2–6, 2011)*. V. 1–3. URL: [https://www.researchgate.net/publication/221456103\\_Genetic\\_algorithm\\_aided\\_optimization\\_of\\_hierarchical\\_multiagent\\_system\\_organization](https://www.researchgate.net/publication/221456103_Genetic_algorithm_aided_optimization_of_hierarchical_multiagent_system_organization) (дата обращения: 22.09.2023).

3. Дубенко Ю. В., Дышкант Е. Е., Обозовский А. А. Метод оптимизации организационной структуры централизованных многоагентных систем в автоматическом режиме // Вестн. Астрахан. гос. техн. ун-та. Сер.: Управление, вычислительная техника и информатика. 2024. № 1. С. 46–55.

4. Bistaffa F., Farinelli F., Cerquides J., Rodríguez-Aguilar J., Ramchurn S. D. Anytime Coalition Structure Generation on Synergy Graphs. URL: [https://www.researchgate.net/publication/269092245\\_Anytime\\_coalition\\_structure\\_generation\\_on\\_synergy\\_graphs](https://www.researchgate.net/publication/269092245_Anytime_coalition_structure_generation_on_synergy_graphs) (дата обращения: 22.09.2023).

5. Rahwan T., Michalak T. P. Coalition Structure Generation on Graphs. URL: <https://doi.org/10.48550/arXiv.1410.6516> (дата обращения: 24.09.2023).

6. Ghrieb N., Mokhati F., Tahar G. Maintaining Organizational Multi-agent Systems: A Reorganization-based Preventive Approach. URL: <https://www.scitepress.org/Link.aspx?doi=10.5220/0010314803840389> (дата обращения: 04.10.2024).

7. Дубенко Ю. В., Дышкант Е. Е., Обозовский А. А. Метод автоматической реорганизации структуры многоагентных систем, основанный на применении алгоритма симулированного закалывания // Изв. высш. учеб. заведений. Поволж. регион. Техн. науки. 2024. № 1 (69). С. 5–18.

8. Mahani M. N. Strategic Structural Reorganization in Multi-agent Systems Inspired by Social Organization Theory: Dissertation, Degree Level - Ph.D., Discipline - Electrical Engineering & Computer Science. University of Kansas, 2012. 105 p.

9. Саттон Р. С., Барто Э. Дж. Обучение с подкреплением: введение: практ. рук. / пер. с англ. А. А. Слинкина. М.: ДМК Пресс, 2020. 552 с.

10. Дэн Саймон. Алгоритмы эволюционной оптимизации / пер. с англ. А. В. Логунова. М.: ДМК Пресс, 2020. 1002 с.

## References

1. Horling B., Lesser V. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 2005, no. 19 (04), pp. 281–316.

2. Ling Yu, Zhiqi Shen, Chunyan Miao, Victor Lesser V. Genetic Algorithm Aided Optimization of Hierarchical Multi-Agent System Organization. *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011) (Taipei, Taiwan, May 2–6, 2011)*. Vol. 1-3. Available at: [https://www.researchgate.net/publication/221456103\\_Genetic\\_algorithm\\_aided\\_optimization\\_of\\_hierarchical\\_multiagent\\_system\\_organization](https://www.researchgate.net/publication/221456103_Genetic_algorithm_aided_optimization_of_hierarchical_multiagent_system_organization) (accessed: 22.09.2023).

3. Dubenko Iu. V., Dyshkant E. E., Obozovskii A. A. Metod optimizatsii organizatsionnoi struktury tsentralizovannykh mnogoagentnykh sistem v avtomaticheskom rezhime [A method for optimizing the organizational structure of centralized multi-agent systems in automatic mode]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naia tekhnika i informatika*, 2024, no. 1, pp. 46–55.

4. Bistaffa F., Farinelli F., Cerquides J., Rodríguez-Aguilar J., Ramchurn S. D. Anytime Coalition Structure Generation on Synergy Graphs. Available at: [https://www.researchgate.net/publication/269092245\\_Anytime\\_coalition\\_structure\\_generation\\_on\\_synergy\\_graphs](https://www.researchgate.net/publication/269092245_Anytime_coalition_structure_generation_on_synergy_graphs) (accessed: 22.09.2023).

5. Rahwan T., Michalak T. P. Coalition Structure Generation on Graphs. Available at: <https://doi.org/10.48550/arXiv.1410.6516> (accessed: 24.09.2023).

6. Ghrieb N., Mokhati F., Tahar G. *Maintaining Organizational Multi-agent Systems: A Reorganization-based Preventive Approach*. Available at: <https://www.scitepress.org/Link.aspx?doi=10.5220/0010314803840389> (accessed: 04.10.2024).

7. Dubenko Iu. V., Dyshkant E. E., Obozovskii A. A. Metod avtomaticheskoi reorganizatsii struktury mnogoagentnykh sistem, osnovannyi na primenenii algoritma simulirovannogo zakalivaniia [A method of automatic reorganization of the structure of multi-agent systems based on the application of a simulated hardening algorithm]. *Izvestiia vysshikh uchebnykh zavedenii. Povolzhskii region. Tekhnicheskii nauki*, 2024, no. 1 (69), pp. 5–18.

8. Mahani M. N. *Strategic Structural Reorganization in Multi-agent Systems Inspired by Social Organization Theory: Dissertation, Degree Level - Ph.D., Discipline - Electrical Engineering & Computer Science*. University of Kansas, 2012. 105 p.

9. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*. Cambridge, MIT Press, 2018. 352 p. (Russ. ed.: Sutton R. S., Barto E. Dzh. Obuchenie s podkrepleniem: vvedenie: prakticheskoe rukovodstvo / per. s angl. A. A. Slinkina. M.: ДМК Пресс, 2020. 552 с.).

10. Dan Simon. *Evolutionary optimization algorithms*. Wiley, 2013. 784 p. (Russ. ed.: Den Saimon. Algoritmy evoliutsionnoi optimizatsii / per. s angl. A. V. Logunova. M.: ДМК Пресс, 2020. 1002 с.).

Статья поступила в редакцию 09.12.2024; одобрена после рецензирования 12.02.2025; принята к публикации 24.03.2025  
The article was submitted 09.12.2024; approved after reviewing 12.02.2025; accepted for publication 24.03.2025

## Информация об авторах / Information about the authors

**Юрий Владимирович Дубенко** – кандидат технических наук, доцент; и. о. заведующего кафедрой информатики и вычислительной техники; Кубанский государственный технологический университет; scorpioncool1@yandex.ru

**Iurii V. Dubenko** – Candidate of Technical Sciences, Assistant Professor; Acting Head of the Department of Informatics and Computer Science; Kuban State Technological University; scorpioncool1@yandex.ru

**Евгений Евгеньевич Дышкант** – кандидат технических наук; доцент кафедры внутривзводского электрооборудования и автоматизации; Армавирский механико-технологический институт (филиал) ФГБОУ ВО «Кубанский государственный технологический университет»; ed0802@yandex.ru

**Владислав Анатольевич Демидов** – аспирант кафедры информатики и вычислительной техники; Кубанский государственный технологический университет; demidov\_vladislav96@mail.ru

**Evgenii E. Dyshkant** – Candidate of Technical Sciences; Assistant Professor of the Department of Internal Electrical Equipment and Automation; Armavir Institute of Mechanical and Technological (branch) FSBEI HE “Kuban State Technological University”; ed0802@yandex.ru

**Vladislav A. Demidov** – Postgraduate Student of the Department of Informatics and Computer Science; Kuban State Technological University; demidov\_vladislav96@mail.ru

