

Застосування генетичного алгоритму для агентного моделювання поведінки об'єктів при виході з палаючої кімнати

Дорогий Я. Ю., к.т.н., доц., ORCID [0000-0003-3848-9852](https://orcid.org/0000-0003-3848-9852)

e-mail argusyk@gmail.com

Цуркан В. В., к.т.н., ORCID [0000-0003-1352-042X](https://orcid.org/0000-0003-1352-042X)

e-mail v.v.tsurkan@gmail.com

Дембіцький А. С., ORCID [0000-0002-5971-576X](https://orcid.org/0000-0002-5971-576X)

e-mail antonde@ukr.net

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського" kpi.ua

Київ, Україна

Реферат—В статті розглянуте питання реалізації генетичного алгоритму для навчання й оптимізації нейронної мережі та її застосування для задач агентного моделювання людської поведінки. Запропоновано систему, що здійснює навчання агентів за допомогою підбору найефективніших стратегій поведінки з наявного набору стратегій із застосуванням генетичного алгоритму. В роботі висвітлено детально розробку одного з таких модулів поведінки агента на основі нейронної мережі, що, будучи навченим, достатньо допомагає агентів орієнтуватися в середовищі. Описано побудову середовища для такого навчання та представлено результати навчання при різних конфігураціях середовища. Головною метою та застосуванням такого підходу є використання навчених агентів для розробки системи моделювання поведінки натовпу під час пожежі.

Бібл. 11, рис. 6, табл. 4.

Ключові слова — нейронні мережі; генетичний алгоритм; навчання без учителя; агентне моделювання; навчання нейронних мереж; геном, симуляція поведінки натовпу.

I. Вступ

Нейронні мережі та різні алгоритми їхнього навчання широко використовуються у розпізнаванні та відтворенні образів [1], тенденцій, послідовностей [2], класифікації даних [3], в моделюванні систем, що імітують агентську поведінку. Використання цього підходу дозволило б створити моделі людей, які перебувають у натовпі, та імітувати поведінку, близьку до реалістичної. Наслідком може бути поглиблення знань про закономірності поведінки у натовпі та організація експериментів з навченими моделями. Такі дослідження потрібні при проектуванні будівель, що вміщують велику кількість людей: концертні зали, виставкові центри, метро, аеропорти [4]. Вони проводяться, зокрема, для забезпечення розміщення екстрених виходів у найнеобхідніших зонах будівлі.

II. АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ І ПОСТАНОВКА ПРОБЛЕМИ

Створення агентних систем моделювання, що імітують повністю або частково реальні системи, в яких поведінка одиничних об'єктів визначає поведінку усєї системи в цілому, є досить популярною проблемою. Першою достатньо відомою з таких систем була математична гра «Життя», запропонована Джоном

Конвеєм [5]. Система складається з кліткового автомату, що має чіткі правила середовища, та агентів у ньому. Територією є розмічена площина, клітки якої можуть знаходитися в одному з двох станів: бути «живими» або «мертвими». На початку гри живі клітини розподіляються рівномірно по середовищу, а наступне покоління розраховується за правилами: в порожній клітці, поруч з якою рівно три живі клітини, зароджується життя; якщо у живій клітині є дві або три живі сусідки, то ця клітина продовжує жити; в іншому випадку, якщо сусідів менше двох або більше трьох, клітина вмирає («від самотності» або «від перенаселеності»). Гра завершується, якщо на полі не залишиться жодної «живої» клітини, конфігурація утворення клітин зациклюється або лишається незмінною, тобто система не може далі розвиватися. Так, з ростом обчислювальних потужностей дослідження поведінки живих організмів за допомогою нейронних мереж набуває популярності, будуються все більші системи для вирішення цих проблем. Одною з таких розробок є «Штучне життя» [6], що імітує поведінку мікроорганізмів та має ціллю дослідження оптимальної поведінки великої кількості агентів у пошуках їжі в замкнутому середовищі. Для досягнення цього застосовано нейронну мережу, що навчається за генетичним алгоритмом. Дослідження показали, що при побудові певної топології мережі та



настройці цілей генетичного алгоритму система досить швидко досягає прийнятних та цікавих результатів. Рухаючись спочатку хаотично, згодом агенти будують ефективну модель поведінки та можуть виживати протягом часу, що в сотні разів перевищує початковий, а при додаванні нових характеристик формують колонії та конкурують між собою. Схожі дослідження проводяться з різними середовищами та можливостями агентів також із застосуванням генетичного алгоритму, оскільки він доцільний для навчання таких систем [7].

Проте розробки, наведені вище, досліджують лише поведінку найпростіших і не вирішують ряд проблем з моделювання та дослідження таких систем, що імітують, наприклад, людську поведінку.

Одними з найбільш відомих розробок з наведеного питання є Pedestrian Dynamics [4] та building EXODUS [8], що створені в якості рішення для агентного моделювання поведінки натовпу. Застосування є комплексним програмним забезпеченням моделювання натовпу. Воно призначене для створення і відтворення великих моделей пішохідного моделювання в складних інфраструктурах. Pedestrian Dynamics також може бути використане для оцінки ефективності та безпеки середовища в кожній фазі життєвого циклу моделювання системи руху агентів від проектування до операції. Програма пропонує швидко опанувати середовище моделювання побудови моделі, що економить час і витрати на розробку. Користувач має виконати кілька кроків, які дозволяють моделювати пішохідні потоки в найскладніших операціях. Є гнучким, надійним і простим у використанні. Широко використовується в багатьох великомасштабних проєктах для імітації потоків натовпу в найбільш критичних умовах інфраструктури, включаючи аеропорти, термінали громадського транспорту та міського планування. Клієнти використовують вищезгадане програмне забезпечення для моделювання для імітації великих логістичних систем і інфраструктур, таких як систем обробки багажу, контейнерних терміналів, залізничних станцій, складальних ліній і футбольних стадіонів. Програмне забезпечення, що здійснює моделювання, дозволяє користувачеві впоратися з плином часу, витратами, ресурсами, надійністю, безпекою і стабільністю. Building EXODUS має схожі характеристики та спеціалізується саме на евакуації натовпів з різних приміщень, що палають.

Звісно, при правильному налаштуванні це програмне забезпечення дозволяє відстежувати потоки людей в різних випадках, в тому числі і евакуації, проте основним його мінусом, на нашу думку, є те, що під час цього процесу не враховується психоемоційний стан натовпу та моделювання не дасть повних результатів у випадку виникнення паніки у натовпі під час вибуху, пожежі чи іншої екстремальної події. При таких подіях не слід очікувати, що натовп буде виходити спокійно та в черзі, як це відбувається в наведених розробках. Тому нами запропоновано методику, яка дозволить навчити агентів утворювати систему, що моделює поведінку панікуючого натовпу.

ТАБЛИЦЯ 1 ПРИКЛАД НАБОРУ СТРАТЕГІЙ АГЕНТА

Тип руху	Виявлення вогню	Почув панічні крики	Побачив натовп перед собою	Реакція на штовхання
Рух до цілі незважаючи ні на що	Рухатися прямо до найближчого головного виходу, незважаючи ні на що	Почати стратегію в виявленні вогню	Обрати інший шлях виходу	Ігнорувати
Перед перпоною змінювати вектор руху на 30°	Рухатися до найближчого виходу, що помітив, почати волати	Ігнорувати крики	Проходити, ігноруючи натовп	Штовхнути у відповідь
Рух на основі стратегії нейронної мережі 1	Рухатися до найближчого виходу, якщо той не зайнявся полум'ям	...	Рішення на основі стратегії нейронної мережі	...
...

III. МЕТА І ЗАВДАННЯ ДОСЛІДЖЕННЯ

Метою дослідження було розробити методику створення системи агентного моделювання поведінки натовпу під час пожежі.

Дослідження спрямоване на розв'язання наступних завдань:

- проаналізувати застосування та побудувати структуру системи, що імітує поведінку натовпу під час пожежі,
- описати кроки, які потрібно здійснити для навчання одиничних об'єктів та роботи системи в цілому,
- описати один з модулів поведінки агента, оснований на штучній нейронній мережі з застосуванням генетичного алгоритму в якості навчального методу.

IV. ОСНОВНА ЧАСТИНА

A. Опис системи моделювання та одиничних об'єктів.

Для моделювання поведінки натовпів найбільш наочним та ефективним, на нашу думку, є підхід повного моделювання та регулювання моделей в реальному часі експериментально. Таким чином, було вирішено створювати систему наступним шляхом. У 3-вимірній моделі великого приміщення з проходами, завадами, стінами та виходами знаходяться об'єкти у вигляді кубів, що стоять або рухаються процедурно навмання. У певний момент в окремих частинах кімнати будуть спостерігатись випадки займання полум'я. Перед кубічними об'єктами стоятиме завдання покинути приміщення, при цьому уникаючи вогню та не створюючи затори на

виходах із приміщення, вони мають це зробити базуючись на даних про попередні події з минулих ітерацій, оскільки поведінка та результат дій об'єктів записуватиметься до бази даних.

Тобто очікується, що спочатку, при перших запусках цього процесу об'єкти будуть нездатні покинути кімнату максимально ефективно для себе та оточуючих, оскільки у них буде недостатньо даних про те, як не уникати вогню, не створювати натовпи та інше, проте на пізніх ітераціях вони зможуть організовуватися в "чергу" та не перешкоджати один одному з метою збереження більшої кількості об'єктів від вогню, або обирати егоїстичну стратегію та намагатимуться врятуватися якомога швидше, не враховуючи інші об'єкти навколо.

Отже, метою навчання об'єктів є за допомогою генетичного алгоритму будувати свою поведінку виходу з палаючого приміщення, властивості та наповнення якого змінюються від ітерації до ітерації.

Агенти матимуть певні стратегії поведінки, що будуть вмикатися за певної події, приклад такого набору показано у табл. 1. При застосуванні генетичного алгоритму буде виведено натовп агентів з різним набором таких стратегій, який сумарно даватиме результати, вказані в функції пристосованості. Тобто отримана система поводитиме себе згідно з потрібними результатами, вказаними оператором. Так, її навчання може бути націленим як на отримання найефективнішої популяції, що залишає будівлю якомога швидше з найменшою кількістю жертв, так і на досягнення інших показників з метою підвищення реалістичності її поведінки. Можливо буде змодельовати певну екстремальну ситуацію, що вже сталася в минулому в ідентичному середовищі та запустити навчання до досягнення схожої конфігурації поведінки агентів. Тоді такі навчені моделі можуть вважатися достатніми для використання в інших середовищах.

В. Розробка стратегії на нейронній мережі.

Найефективнішими стратегіями поведінки можуть виявитися ті, що були не запрограмовані раніше, як-то набір векторів, за яким агент має рухатися до точки, а ті стратегії, в яких рішення приймаються агентом, базуючись на значеннях, що містяться в добре навченій штучній нейронній мережі. Це дозволить йому обирати вектор руху самостійно в залежності від навколишніх умов. Таким чином, вже навчений агент поводитиме себе гнучкіше та реалістичніше, що і є метою даної системи. Такі стратегії поведінки будуть додані в систему моделювання нарівні з запрограмованими та обиратимуться генетичним алгоритмом на основі їх ефективності.

Одною з найбільш необхідних стратегій є огинання вогню та інших перешкод, тому в першу чергу саме цю стратегію було побудовано на нейронній мережі.

Було виявлено, що програмування руху агента під час огинання перепон в даній системі було досить складним та робило дослідження поведінки об'єктів

в екстремальних умовах важчим через те, що навчена заново модель може краще висвітлити поведінку, близьку до людської.

Отже, для огинання перепон та руху до кінцевої точки нейронна мережа повинна мати динамічно оновлювану інформацію про навколишнє середовище, а саме знаходження певних об'єктів навколо агента. Було введено певну кількість датчиків, що повідомляють цю інформацію: 7 променів від агента, що вимірюють відстань до вогню або стіни, якщо та потрапила у промінь, а також напрямок та відстань до виходу. Довжиною променю було обрано 10 м при розмірах агента 1 м, якщо нічого не перетинало промінь, то на вхід подається значення 10, що є максимальним значенням.

Для того щоб зменшити кількість виходів та збільшити точність та зменшити швидкість навчання, вихідними значеннями обрано обертаючи сили агента. Так, сили можуть компенсувати одна одну, даючи змогу агенту рухатися прямо, а також повертати вліво або вправо.

В якості функції активації застосовано біполярну сигмоїду, оскільки виходи можуть набувати від'ємних значень:

$$f(x) = -1 + \frac{2}{1 + e^{-x}}$$

Входи нейронної мережі (рис. 1):

F (front) – відстань до найближчих об'єктів, що потрапили у вектор 0° відносно напрямку руху агента.

Fl (front left) – відстань до найближчих об'єктів, що потрапили у вектор 315° відносно напрямку руху агента.

Fl (front right) – відстань до найближчих об'єктів, що потрапили у вектор 45° відносно напрямку руху агента.

L (left) – відстань до найближчих об'єктів, що потрапили у вектор 270° відносно напрямку руху агента.

R (right) – відстань до найближчих об'єктів, що потрапили у вектор 90° відносно напрямку руху агента.

BR (back right) – відстань до найближчих об'єктів, що потрапили у вектор 135° відносно напрямку руху агента.

BL (back left) – відстань до найближчих об'єктів, що потрапили у вектор 225° відносно напрямку руху агента.

α – кут між вектором front агента та напрямком до кінцевої точки.

d – відстань від агента до кінцевої точки.

Виходи нейронної мережі (рис. 1):

Right rotate force – сила, з якою агент обертається праворуч.

Left rotate force – сила, з якою агент обертається ліворуч.

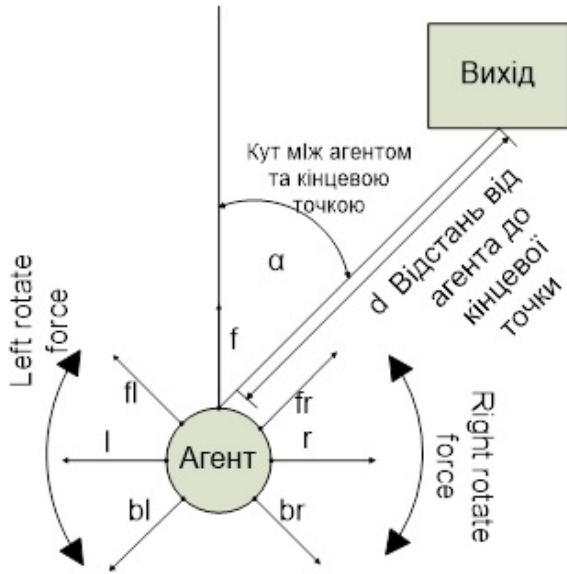


Рис. 1 Входи та виходи агента

В нейронній мережі окрім вхідного та вихідного шарів побудовано один прихований шар, що має 10 прихованих нейронів (рис. 2).

С. Побудова середовища для навчання.

Для навчання агентів було побудовано тривимірну модель кімнати зі стінами та виходами. Для спрощення ілюстрації на рис. 3 вказано схематичне розташування об'єктів у приміщенні. Кожен раз на початку полум'я з'являється в певній точці та розгорається радіально навколо, поки не займе всю кімнату; якщо за цей час агент не дістався виходу, він не має можливості не потрапити у вогонь. Агент має число, що виражає його здоров'я та має початкове значення 100. При потраплянні агента у полум'я число здоров'я знижується, і при досягненні цього значення 0 агент завершує ітерацію навчання.

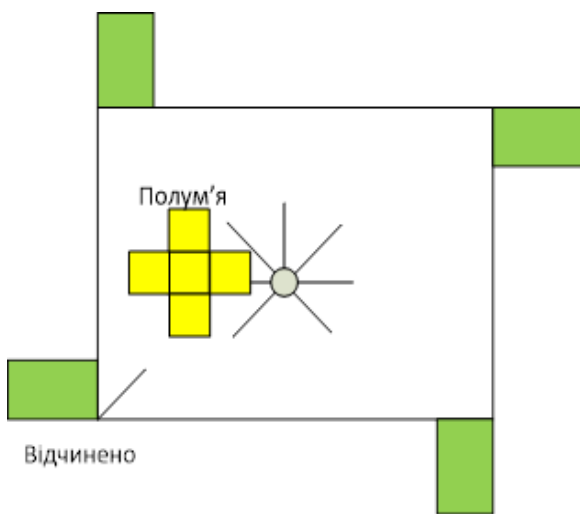


Рис. 3 Варіант розташування об'єктів у тренувальній кімнаті

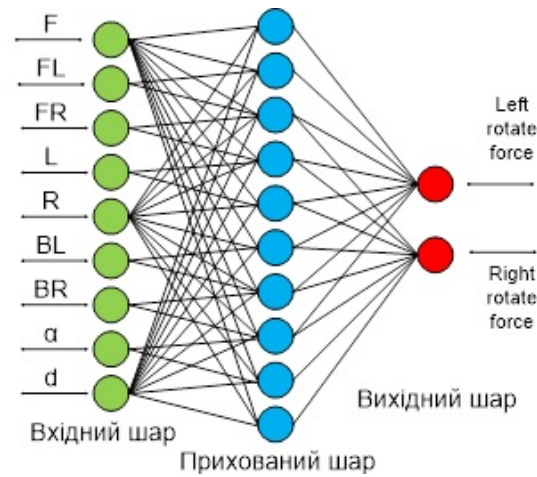


Рис. 2 Архітектура мережі поведінки агента при огинанні вогню

Навчання кожного з геномів поведінки агента проходить таким чином, що всі агенти певного покоління навчання мають одні й ті самі умови кімнати, а саме позицію, обернення по осі Z, місце початку полум'я та вхід, що відкрито. Кожну з навчальних ітерацій агент з'являється в новому місці та записує його для наступних геномів (табл. 2). Такий підхід застосовано для кращого виявлення найприспособанішого геному.

Д. Імплементация алгоритму.

Внутрішньо нейронні мережі представлені у вигляді послідовності числових матриць ваг кожного шару, окрім вхідного. Загалом ці матриці мають вигляд, схожий до того, що описано в [9], додавання та видалення нейронів також відбувається відповідно.

Отже, перелік всіх ваг між вхідними, прихованими та вихідними нейронами є геномом поведінки агента. На рис. 4 вказано, як саме ваги, починаючи з першого вхідного нейрону та закінчуючи останнім вихідним, записуються по порядку.

Таблиця 2 Приклад умов, що має агент для різних ітерацій

№	Позиція агента	Рот. агента по вісі Z	Центр полум'я	Вхід відкрито
1	x1, y1	z1	v1, w1	перший
2	x2, y2	z2	v2, w2	другий
3	x3, y3	z3	v3, w3	четвертий
4	x4, y4	z4	v4, w4	третій
5	x5, y5	z5	v5, w5	другий

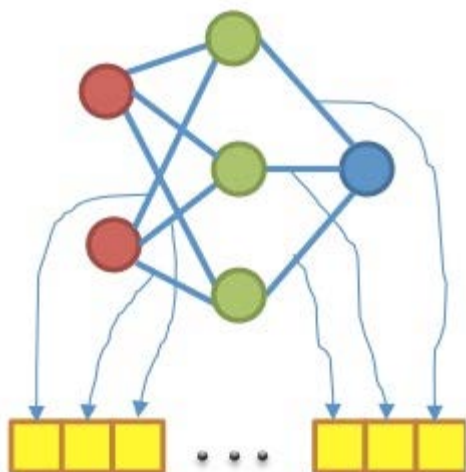


Рис. 4 Вигляд запису геному агента [5]

```

void CrossBreed (Genome g1, Genome g2, refGenome baby1,
refGenome baby2){
int totalWeights = g1.weights.Count;
int crossover = Random.Range(0,totalWeights-1);
baby1 = newGenome (); baby1.ID = genomeID;
baby1.weights = newList<float> ();
for(int i=0; i<totalWeights; i++)
    {baby1.weights.Add (0.0f);}genomeID++;
baby2 = newGenome ();baby2.ID = genomeID;
baby2.weights = newList<float> ();
for (int i=0; i<totalWeights; i++)
    {baby2.weights.Add (0.0f);}genomeID++;
for(int i=0; i< crossover; i++){
    baby1.weights[i] = g1.weights[i];
    baby2.weights[i] = g2.weights[i];}
for(int i = crossover; i< totalWeights; i++){
    baby1.weights[i] = g2.weights[i];
    baby2.weights[i] = g1.weights[i];}
}

```

Script 1 Фрагмент коду, що здійснює мітоз

Найважливішим показником того, що саме цей геном має бути обраним для майбутніх поколінь, є *приспособаність (fitness)* геному [10].

Оскільки головною метою агента є знаходження виходу, а при попаданні у полум'я він зазнає невдачі, то до значення фітнесу додаватиметься 30 при знаходженні виходу, натомість при попаданні в полум'я кожні 200 мс агент разом зі значенням здоров'я втрачатиме 0.5 бала приспособаності, а при встановленні значення здоров'я на 0 втрачатиме ще 5 балів, що дає в середньому штраф до 15 балів приспособаності. Також штраф накладається на агента, якщо він надовго затримується у кімнаті. Починаючи з 5 секунд знаходження в кімнаті агент отримує штраф в 1 бал на секунду. На початку ітерації агент має початковий бал приспособаності 15. При завершенні ітерації отриманий бал додається до сумарного балу приспособаності поточного геному.

При першому запуску у навчанні беруть участь 30 геномів зі значеннями ваг, генерованих навмання. Кожен з них проходить 5 ітерацій навчання, після чого сумарний геном записується до бази даних. Далі всі наступні геноми проходять ті ж ітерації, доки вони

не закінчаться. По завершенню навчання поточного покоління обираються 5 найбільш пристосованих геномів, що породять наступне покоління. Наступні геноми лишаються в популяції та утворюють між собою свої змінені копії шляхом мітозу, як показано на рис. 5. Обидва геноми розділяються на певній мітці та обмінюються частинами генетичного коду, утворюючи при цьому два інші геноми, що наслідуватимуть частини батьківських. На script 1 висвітлено функцію мітозу.

Кожен геном після цього проходить операцію мутації, де кожен його ген, тобто вага зв'язку в нейронній мережі, можуть бути змінені навмання з вірогідністю `MUTATION_RATE` на випадкове значення в діапазоні `MAX_PERBETUATION` [11].

Так утворюються перші 25 геномів наступного покоління. Решта 5 генеруються навмання для підвищення різноманітності у популяції.

Навчання завершиться, коли один і той же геном пройде поріг достатнього балу приспособаності, тобто знайде вихід принаймні 13 разів (4.3 рази за одну епоху) протягом трьох поколінь.

Е. Експериментальні дослідження.

Реалізована програмна система використана для дослідження та тренування моделі агента обходити перешкоди та діставатися до кінцевої точки. Було розглянуто роботу нейронної системи з різною структурою: один запуск відбувався з характеристиками, наведеними вище, в іншій конфігурації системи вихід був лише один та залишався на місці, агент мав лише 4 промені-датчика, а за знаходження більше п'яти секунд на полі надавалася винагорода.

Ф. Дослідження роботи алгоритму.

Було виявлено, що кількість входів та якість тренувального середовища значно впливають на швидкість та ефективність навчання. Було розроблено та побудовано найбільш доречне тренувальне середовище, що дозволяло досягти найбільш точного та універсального результату з навчання агента оминати перепони у просторі.

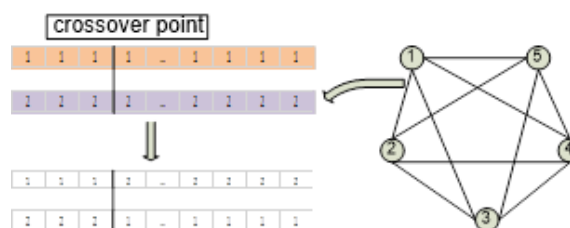


Рис. 5 Мітоз між найприспособанішими геномами

```

void Mutate(Genome genome){
for (int i=0; i<genome.weights.Count; i++) {
    if(RandomClamped() < MUTATION_RATE){
        genome.weights[i] += (RandomClamped()
        ° MAX PERBETUATION);}
}
}

```

Script 2 Фрагмент коду, що здійснює мутацію



Для системи, наведеної вище, геном, що здатен знаходити вхід 13 разів за 3 покоління підряд, з'являвся в середньому на 21-ше покоління. Агенти починали в своїй більшості рухатися завжди у сторону виходу вже починаючи з 6-7-го покоління, при цьому маючи середній рівень пристосованості 130, а максимальний 170. Після того як штраф за знаходження на полі більше 5 секунд замінили на приз за відстань від виходу, тобто чим ближче до виходу знаходився агент, тим більше балів він отримував, фінальний геном формувався вже на 15-му поколінні в середньому.

В свою чергу спрощена система набагато швидше досягала фінальних результатів, близько 11-го покоління вже було сформовано фінальний ген, що міг проходити задану кількість виходів за 3 покоління. Агенти в моделі, що мала лише 4 входи, промені лише за кутами 0° , 315° , 45° та 180° , мали на початку менше середнє значення пристосовності, натомість формували фінальний геном вже до 7-8-го покоління. Проте дані моделі були нездатні поводити себе вірно в інших ситуаціях, окрім конкретної навчальної, в якій вони навчалися. Якщо вихід пересувався або їхня ротація в просторі змінювалася від ітерації до ітерації, було видно, що вони мали помилкову стратегію поведінки та не могли досягти результату. Рис. 6 ілюструє різну поведінку, що показали геноми агентів в різних середовищах. Зміна місця виходів допомагала агентам побудувати чітку залежність між кутом та відстанню до виходу та балом пристосованості. Ті, що знаходилися в незмінній кімнаті, досить швидко почали виходити з кімнати, проте їхні нейронні мережі базувалися лише на входах зліва від агентів у момент початку ітерації та завжди обходили всю кімнату за годинниковою стрілкою, тримаючись при цьому стіни.

З рисунку вище видно, що перша мережа більш гнучка та ефективна через те, що агенти приділяли увагу не тільки оминанню перешкод, а й напрямку виходу. Це відбулося завдяки тому, що його положення також змінювалося.

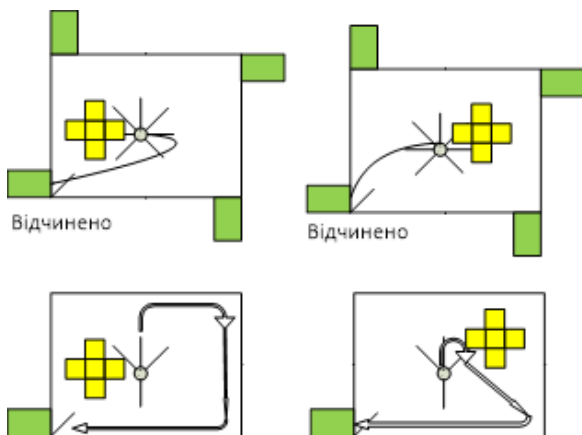


Рис. 6 Маршрут руху найбільш пристосованого агента в різних середовищах

ТАБЛИЦЯ 3 РЕЗУЛЬТАТИ НАВЧАННЯ ПРИ РІЗНИХ КОНФІГУРАЦІЯХ

Конфігурація системи \ № покоління	Покоління форм. геному	Підходить для інших середовищ
4 пр., 1 вих., без штрафу	-	ні
4 пр., 1 вих., штраф	13	ні
9 пр., 1 вих., штраф	9	ні
9 пр., 4 вих., штраф	18	так
9 пр., 4 вих., штраф	15	так

ТАБЛИЦЯ 4 ДОСЯГНЕННЯ ФІНАЛЬНОГО ГЕНОМУ ПРИ РІЗНИХ КОНФІГУРАЦІЯХ

Конфігурація системи	3	6	13	16	19	21
4 промені, 1 вихід, без штрафу за відстань	a-30b-63	a-45b-69	a-44b-68			
4 промені, 1 вихід, штраф за відстань	a-34b-66	a-40b-75	a-53b-95	a-50b-83	a-55b-75	a-57b-81
9 променів, 1 вихід, штраф за відстань	a-40b-72	a-45b-80	a-56b-95	a-59b-95	a-60b-94	a-65b-95
9 променів, 4 виходи, без штрафу за відстань	a-25b-65	a-43b-70	a-51b-75	a-50b-83	a-50b-92	a-50b-95
9 променів, 4 виходи, штраф за відстань	a-23b-67	a-32b-71	a-56b-78	a-62b-94	a-60b-96	a-64b-93

У табл. 3 та табл. 4 вказано середній (а) та найвищій (б) бали пристосованості геномів в різних поколіннях. Значення вказані у відсоток до максимально можливого в даній конфігурації. Як видно з результатів, для досягнення певної цілі важливо мати достатньо повні навчальні фактори середовища. Зрештою було побудовано таку конфігурацію мережі та середовища, при якій агент показував бажану поведінку навіть при зміні середовища.

ВИСНОВКИ

В роботі розглянуті питання розробки системи агентного моделювання поведінки натовпу під час пожежі. Основними метою та завданням даної роботи було створити ефективний підхід до навчання агентів для їхнього використання у подальшій розробці системи, що моделює поведінку натовпу під час екстремальних ситуацій, таких, як пожежа. Зокрема, в ході аналізу застосування та побудови структури системи, що імітує поведінку натовпу під час пожежі, була використана модель великої будівлі. Об'єкти в ній за допомогою генетичного алгоритму навчаються будувати свою поведінку виходу з палаючого приміщення. Описані конкретні кроки, які потрібно здійснити для навчання одиничних об'єктів та роботи системи в цілому. У статті також подано опис одного з модулів поведінки агента, заснованого на штучній нейронній мережі з застосуванням генетичного алгоритму в якості навчального методу.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] J. Tu and R. Honsales, *Printsipyi raspoznavaniya obrazov [Principles of recognition of patterns]*. Moscow: Mir, 1978.
- [2] I. S. Svetunkov and S. H. Svetunkov, *Metodyi sotsialno-ekonomicheskogo prognozirovaniya: Uchebnik dlya vuzov., t. I [Methods of socio-economic prognostication: Textbook for institutions of higher learning., vol. I]*. St. Peterburg: StPbGUEF, 2009.
- [3] S. A. Aivazian, V. M. Bukhshtaber, I. S. Eniukov, and L. D. Meshalkin, *Prikladnaya statistika: Klassifikatsii i snizhenie razmernosti [Applied statistics: Classifications and decline of dimension]*. Moscow: Finansy i statistika, 1989.
- [4] P. Dynamics, "Airports," *INCONTROL Simulation Solutions*. [Online]. Available: <http://www.pedestrian-dynamics.com/crowd-management/airport-crowd-simulation.html>.
- [5] M. Gardner, "The fantastic combinations of John Conway's new solitaire game 'life,'" *Sci. Am.*, vol. 223, no. 4, pp. 120–123, 1970.
- [6] Y. Lahodiuk, "Evolutsiya agentov upravlyaemyih neyronnoy setyu [Evolution of agents guided by a neural network]," *Habrahabr*, 2013. [Online]. Available: <https://habrahabr.ru/post/168067/>.
- [7] foo52ru, *Iskusstvennaya zhizn. Geneticheskiy algoritm v2.0. Pervye probyi [Artificial life. Genetic algorithm of v2.0. First tests]*. 2016. Available: <https://www.youtube.com/watch?v=nIDsMYIbqMM>
- [8] "EXODUS Demo software," *FIRE SAFETY ENGINEERING GROUP*. [Online]. Available: http://fseg.gre.ac.uk/exodus/exodus_registered_visitors.html.
- [9] Ya. Yu. Dorohyi, O. O. Doroha-Ivaniuk, and D. A. Ferens, "Realizatsiia alhorytmu strukturnoi optimizatsii neironnoi merezhi [Implementation of neural network structural optimization algorithm]," *Visnyk NTUU KPI Informatics, Oper. Comput. Sci.*, no. 63, pp. 101–106, 2015.
- [10] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2nd ed. Wiley-Interscience, 2004. ISBN: 978-0471455653
- [11] M. Obitko, "XI. Crossover and Mutation," *Introduction to Genetic Algorithms*, 1998. [Online]. Available: <http://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php>.

Надійшла до редакції 25 червня 2017 року

УДК 004.891.2

Применение генетического алгоритма для агентного моделирования поведения объектов во время выхода из пылающей комнаты

Дорогой Я. Ю., к.т.н., доц., ORCID [0000-0003-3848-9852](https://orcid.org/0000-0003-3848-9852)

e-mail argusyk@gmail.com

Цуркан В. В., к.т.н., ORCID [0000-0003-1352-042X](https://orcid.org/0000-0003-1352-042X)

e-mail v.v.tsurkan@gmail.com

Дембицкий А. С., ORCID [0000-0002-5971-576X](https://orcid.org/0000-0002-5971-576X)

e-mail antonde@ukr.net

Национальный технический университет Украины

"Киевский политехнический институт имени Игоря Сикорского" kpi.ua

Киев, Украина

Реферат—В статье рассмотрен вопрос реализации генетического алгоритма для обучения и оптимизации нейронной сети и ее применения для задач агентного моделирования человеческого поведения. Предложена система, осуществляющая обучение агентов с помощью подбора эффективных стратегий поведения из имеющегося набора стратегий с применением генетического алгоритма. В работе освещена детальная разработка одного из таких модулей поведения агента на основе нейронной сети, который после длительного обучения помогает агенту ориентироваться в среде. Описано построение среды для такого обучения и представлены результаты обучения при различных конфигурациях среды. Главной целью и применением такого подхода является использование обученных агентов для разработки системы моделирования поведения толпы во время пожара.

Библ. 11, рис. 6, табл. 4.

Ключевые слова — нейронные сети; генетический алгоритм; обучение без учителя; агентное моделирование; обучение нейронных сетей; геном, симуляция поведения толпы.



Application of the genetic algorithm for agent-based modeling of the object behavior while leaving the flaming room

Ya. Yu Dorohyi, PhD, Assoc.Prof., ORCID [0000-0003-3848-9852](https://orcid.org/0000-0003-3848-9852)

e-mail argusyk@gmail.com

V. V. Tsurkan, PhD, ORCID [0000-0003-1352-042X](https://orcid.org/0000-0003-1352-042X)

e-mail v.v.tsurkan@gmail.com

A. S. Dembitskyi, ORCID [0000-0002-5971-576X](https://orcid.org/0000-0002-5971-576X)

e-mail antonde@ukr.net

National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute" kpi.ua
Kyiv, Ukraine

Abstract—The article deals with the implementation of the genetic algorithm for training and optimization of the neural network and its application to the tasks related to agent-based modelling of human behaviour. After the analysis of existing agent-based modelling programs, several drawbacks were noticed. The main problem of other systems for crowd modelling was the missing of information about the psychoemotional state of people, who are in crowd. According to the other sources, moods in crowd influence its behaviour the most; therefore, we decided to propose another method of creating more realistic crowd behaviour.

The system that implements training of agents by selecting the most effective strategies of behaviour from the existing set of strategies using the genetic algorithm was proposed. In addition, this article highlights the detailed development of one agent behaviour module based on the neural network, which help the agent to navigate in the environment on condition of being trained enough. Due to created training method, it was mentioned, that training environment affects whole training process, and so several surveys were made at different environment configurations.

As a base for the system, single-layer perceptron was chosen because of its simplicity and small number of connections. Final variant of agent casts seven rays into different vectors from his centre. If any other object interferes those rays, the distance to him comes as input to the neural network. Another two inputs take amount of distance to the exit and angle between agent and exit. Right rotate and left rotate forces of agent are used as outputs for neural network.

Genetic algorithm was used for selecting the proper weights between the neurons. The most important indicator that some genome should be chosen for future generations is the fitness of that genome. The main purpose for the agent is finding the exit and in case of hitting the flame, it will fail. That is why agent will be rewarded for reaching the exit and get penalty for hitting the flame or staying in the room for too long.

During the first launch, 30 initial genomes get involved with the values of weights generated at random. Each of them passes five iterations of training, after which the total gene is written to the database. By the results of current generation training, the five most adapted genomes that will create next generation will be selected. The following genomes remain in the population and form their modified copies by mitosis. Both genomes are divided on a specific point and exchange parts of the genetic code. That process forms two other genomes, which inherit parts of the parents. As for the training environment, there are four different exits from the room but only one of them opens randomly during the iteration.

The implemented software system is used to research and train the agent model bypass the obstacles and get to the endpoint. The work of the neural system with a different structure was considered: one runs with the characteristics given above, in the other system configuration, the exit was only one and stayed in constant place, the agent had only four rays, and for staying in the room for more than five seconds he was rewarded. The development of the environment for such training is described and the results of training are presented for various environmental configurations.

The main goal and mission of such approach implementation is using trained agents to develop a system for crowd behaviour modelling in the building, which was set on fire.

Ref. 11, fig. 6, tabl. 4.

Keywords — neural networks; genetic algorithm; training without a teacher; agent-based modelling; neural networks training; genome; crowd simulation.

