

- адаптивний підбір параметрів локальної розгортки та алгоритму розподілу (можливо, за допомогою методів штучного інтелекту – нейронних мереж, систем з нечіткою логікою і т. д.);
- математичне доведення збіжності алгоритмів.

1. Грицуляк Т.О. Колектив автономних мобільних агентів, який самоорганізується за значеннями параметрів, що досліджуються. *Магістерська робота.* – 2003. 2. Поляков В.Г., Перевзев-Орлов В.С. *Электронные системы следящей развертки.* – М.: Энергия, 1968. – 192 с. 3. Варшавский В.И., Поспелов А.Д. *Оркестр играет без дирижера: размышления об эволюции некоторых технических систем и управлении ими.* – М.: Наука, 1984. – 207 с. 4. Cem Unsal. *Self-Organization In Large Populations Of Mobile Robots. Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering,* 1993. 5. Xavier Defago, Akihiko Konagaya. *Circle Formation for Oblivious Anonymous Mobile Robots with No Common Sense of Orientation. Graduate School of Knowledge Science, Japan Advanced Institute of Science and Technology (JAIST),* 2002. 6. Steven Y. Goldsmith and Rush Robinett, III. *Collective Search by Mobile Robots using Alpha-Beta Coordination. Sandia National Laboratories,* 1999. 7. Peter Stone, Manuela Veloso. *Multiagent Systems: A Survey from a Machine Learning Perspective. Computer Science Department, Carnegie Mellon University,* 1997. 8. Michael Wooldridge. *Introduction to MultiAgent Systems.* – John Wiley & Sons, 2002. – 256 с. 9. Gerhard Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* – MIT Press, 2000. – 648 с. 10. Егоров Е.А. *Модели и реализация переговоров в мультиагентных системах. Магистерская диссертация. Санкт-петербургский государственный электротехнический университет ЛЭТИ,* 2001. 11. Джеймс Одделл. *Агенты и сложные системы // Открытые системы.* – 2002. – № 102.

УДК 681.3, 621.3

О.Ю. Бочкарьов, В.А. Голембо, Х.Р. Попадюк  
 Національний університет "Львівська політехніка",  
 кафедра електронних обчислювальних машин

## САМООРГАНІЗАЦІЯ КОЛЕКТИВУ МОБІЛЬНИХ АГЕНТІВ У ПРОСТОРИ: ФОРМУВАННЯ МНОГОКУТНИКА

© Бочкарьов О.Ю., Голембо В.А., Попадюк Х.Р., 2004

**Розглянуто актуальне питання формування правильних геометричних фігур колективом автономних робототехнічних агентів. Запропоновано розподілений алгоритм побудови заданого наказом многокутника за умов відсутності централізованого управління та необмеженого радіуса видимості засобів детектування сусідніх агентів.**

**Actual problem of forming regular patterns by collective of autonomous robotic agents is considered. Distributed algorithm of forming the specified by order polygon under the conditions of no centralized control and unlimited detection range of agents.**

**Вступ.** Тут розглядається одна з основних проблем розробки алгоритмічного забезпечення мобільних багатоагентних систем (колективів робототехнічних агентів). Елементом колективу є агент. Агентом будемо називати комп'ютерну систему, яка розміщується у деякому середовищі і здатна самостійно розв'язувати задачі, які ставить перед нею її розробник [1]. У цьому випадку агентом може бути комп'ютерна система керування мобільного робота (mobile robot), безпілотного літального апарата (unmanned flying vehicle) або автономного підводного апарата (autonomous underwater vehicle). Перед колективом таких робототехнічних агентів ставляться різні завдання,

наприклад, патрулювання деякого району та виявлення об'єктів із заданими характеристиками. Для розв'язання подібних задач є необхідним складне алгоритмічне забезпечення узгоджених колективних дій. Надалі розглядаються багатоагентні системи з такими характеристиками:

- децентралізоване управління (кожний агент самостійно приймає та реалізує рішення),
- однорідність агентів (всі агенти виконують однаковий алгоритм),
- мобільність агентів (агенти здатні рухатись у заданому просторі за власною ініціативою).

Одним з найбільш актуальних завдань розробки алгоритмічного забезпечення багатоагентних систем є розробка алгоритмів самоорганізації колективу мобільних агентів в просторі [2–6] або самовпорядкування, враховуючи гіпотезу Н.М. Цетліна про простоту. Проблема полягає у знаходженні індивідуального уніфікованого алгоритму поведінки окремого агента. Крім того, цей алгоритм має дозволяти колективу агентів управляти розміщенням та переміщенням своїх представників у геометричному просторі середовища, згідно з поставленою перед ним задачею вищого рівня. При цьому розв'язуються такі основні задачі:

- впорядковане розміщення агентів у просторі (ordered placement): рівномірне заповнення обмеженого деякими границями простору агентами колективу; формування агентами правильної ґратки із заданим кроком; формування агентами гнучкої ґратки, форма якої залежить від характеристик середовища (змістовна інтерпретація: розподілені вимірювання, контроль території),
- формування агентами правильних геометричних фігур (geometric pattern formation): лінія (відрізок), коло, квадрат (каре) із заданим кроком міжагентної відстані (змістовна інтерпретація: формування агентами пошукового ланцюжку, створення периметру охорони навколо заданого об'єкта),
- впорядковане переміщення агентів колективу у просторі (motion control): узгоджене групове переміщення колективу, переміщення за схемою лідер-наслідувач (leader-follower), слідування заданій траєкторії руху (path following),
- уникнення зіткнень з іншими агентами (collision avoidance),
- колективне подолання (обминання) перешкод (obstacle avoidance).

Основним критерієм ефективності розв'язання задачі самовпорядкування є час, за який агенти виконують поставлене перед ними завдання (мінімізація кількості часових кроків). Іншим важливим критерієм є кількість просторових кроків, які витрачають агенти на розв'язання задачі (мінімізація витрат енергоресурсу).

Треба зауважити, що середовище, в якому агенти розв'язують поставлені задачі не обов'язково повинне бути фізичним. Це може бути середовище операційної системи, локальна чи глобальна комп'ютерна мережа і т. д.

Тут розв'язується задача формування многокутника колективом мобільних агентів: агенти повинні стати у вершини та на сторони заданого многокутника. Алгоритм розв'язання цієї задачі є одним з базових алгоритмів просторової самоорганізації, на основі якого може бути реалізована колективна поведінка значно більшої складності.

**Постановка задачі.** Під час побудови моделі колективної поведінки важливими є характеристики окремих агентів і їх засобів взаємодії.

#### 1. Характеристики окремого агента:

##### 1.1.

- голономність – агент зображується у вигляді геометричної точки, відбувається абстрагування від швидкісних характеристик агента,
- неголономність – враховуються інерційність руху агента, нездатність до миттєвої зміни напрямку руху, геометричні розміри агента (які, наприклад, можуть заважати детектуванню іншого агента), агенти не можуть знаходитись в одній геометричній точці простору.

## 1.2. Навігаційна система:

### 1.2.1.

- абсолютна система координат
- відносна система координат

### 1.2.1.

- декартові координати
- полярні координати

## 2. Характеристики засобів взаємодії:

### 2.1. Детектування та локалізація сусідів:

- в умовах повної видимості чи в умовах обмеженої видимості,
- вигляд діаграми направленості засобів детектування (повне коло, сектор ( $<180^\circ$ ), сектор ( $>180^\circ$ )),
- засіб детектування одного типу чи комбінація декількох засобів детектування різних типів (тактильні, інфрачервоні, ультразвукові, лазерні, електромагнітні сенсори).

### 2.2. За типом інформаційної взаємодії:

- з обміном інформацією (колективне прийняття рішень, колективне планування),
- без обміну інформацією (лише детектування сусідніх агентів).

У роботі розглядається колектив голономних агентів з абсолютною декартовою системою координат. Агенти мають обмежений радіус видимості засобів детектування інших агентів  $R$  та не можуть спілкуватися з іншими агентами. Крім того, обмежується максимальна швидкість руху  $S$  (розглядається класична модель колісного робота з одним ведучим колесом). Такий колектив, що складається з  $N$  агентів, має сформувати заданий многокутник, у загальному випадку неправильний. Основним критерієм ефективності в нашому випадку є кількість часових кроків  $T$ , яку треба мінімізувати. Середовище, в якому розміщуються агенти, являє собою регулярну двовимірну “чотирикутну” ґратку розміром  $M$ . Початкове розміщення агентів – випадкове, поява агента в кожній точці середовища рівноймовірна. При цьому всім агентам віддається однаковий наказ такого змісту:

- кількість кутів многокутника –  $n$ ;
- координати вершин (у вигляді масиву) –  $V = \{v_i\}$ ,  $i = [1, n]$ , де  $v_i = (x_i, y_i)$ ;
- кількість агентів, що мають стояти на стороні многокутника (у вигляді масиву) –  $E = \{e_j\}$ ,  $j = [1, n]$ .

Приклад наказу:  $n=5$ ;  $V = \{(25, 20), (25, 40), (45, 40), (55, 30), (45, 20)\}$ ;  $E = \{0, 1, 5, 3, 0\}$ .

Отже, треба побудувати п'ятикутник із заданими вершинами. На стороні, що з'єднує першу і другу вершини, має стояти 0 агентів, другу і третю – 1, третю і четверту – 5, четверту і п'яту – 3, п'яту і першу – 0. Відстані між агентами, що стоять на сторонах однакові і залежать від довжини сторони і кількості агентів, що має на ній стояти. Точки, в яких розміщуються ці агенти, наказом не задаються, а обраховуються під час розв'язання задачі.

Цей алгоритм розрахований на те, що многокутник буде будувати певна необхідна кількість агентів ( $N_n$ ), згідно з конкретним наказом – це кількість вершин і сума агентів, що мають стояти на сторонах.

$$N_n = n + \sum_{j=0}^n e_j .$$

Якщо кількість агентів менша, ніж  $N_n$ , то не всі точки многокутника будуть зайняті. Які саме залишаться вільними залежить від початкового розташування агентів. Агенти займуть усі найближчі точки. Якщо кількість агентів більша, ніж необхідно, то зайві агенти будуть продовжувати рух згідно з алгоритмом, шукаючи вільне місце, поки не отримають іншого наказу. Цього можна уникнути, якщо агенти матимуть інформацію про вже проаналізовані точки. У такому випадку після обходу периметра агент зрозуміє, що вільних місць нема і виконає дію згідно із алгоритмом. Наприклад, припинить рух, почне виконувати інший алгоритм і т. д. Це дозволить зберегти енергію, яка була б затрачена на безрезультатні пошуки вільного місця.

**Формування многокутника.** Тут роботі пропонується такий алгоритм побудови многокутника – доки не буде побудовано многокутник кожному агенту повторювати послідовно три кроки:

1. Отримати інформацію про своє розташування та розташування сусідів.
2. Прийняти рішення про напрямок руху.
3. Переміститись у просторі (реалізація рішення).

Цей алгоритм можна подати блок-схемою (рис. 1).

Отримання інформації про своє розташування та розташування сусідів полягає в пошуку найближчої вільної точки в межах радіусу зору.

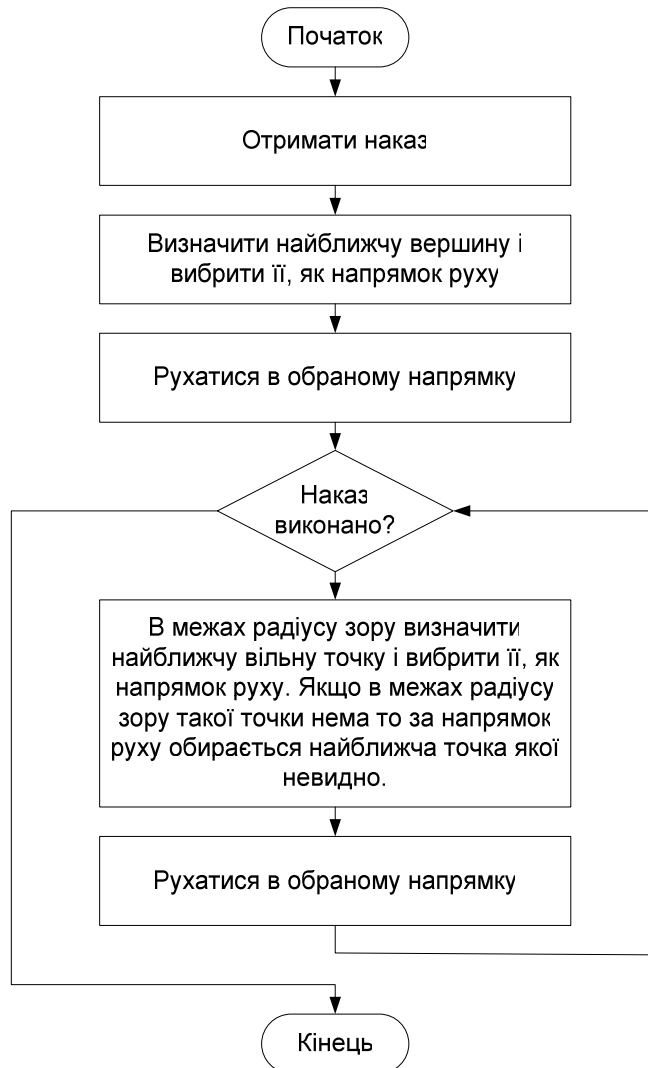


Рис. 1. Блок-схема алгоритму

Прийняття рішення полягає у визначенні точки до якої має прямувати агент (вибір напрямку руху). Координати цієї точки змінюються під час формування многокутника. На блок-схемі показано принцип вибору цієї точки. Треба зазначити, що коли агент визначає найближчу вершину він здійснює аналіз відстаней до всіх вершин. А коли агент шукає найближчу вільну точку, він перевіряє не лише вершини, але й точки розташування на сторонах, координати яких обраховує. Важливо, що перевіряються не всі точки, а точки в межах радіусу видимості в порядку обходу периметра, в напрямку зростання індексів вершин в масиві  $V$ . Якщо в радіусі видимості агента нема вільної точки, то вибирається наступна точка, якої не видно. Вільною вважається точка, яка ще не зайнята іншим агентом. Такий спосіб прийняття рішення простий в реалізації, але не є найкращим з усіх можливих. Агент виконав наказ, якщо він зайняв точку многокутника, вершину чи точку на стороні.

Рух – переміщення агента в просторі, реалізація прийнятого рішення, полягає у зміні координат агента на відповідну швидкість. Якщо, точка куди хоче стати агент, зайнята – рух не здійснюється. Такі простоювання збільшують час побудови многокутника. Їх неможливо передбачити, тому що вони залежать від випадкових величин (початкове розташування агентів) чи величин, які містять випадкові елементи (при обході перешкоди напрямком обходу вибирається випадково, а радіус може змінюватися у визначених межах).

Можна запропонувати такі шляхи вдосконалення алгоритму:

- здійснювати послідовний аналіз точок (зайнята чи вільна) у двох напрямках: в напрямку зростання індексів вершин у заданому масиві координат і в напрямку спадання;
- здійснювати паралельний аналіз усіх точок, що потрапляють в радіус видимості (вершин многокутника та точок на сторонах);
- враховувати біжуче розташування інших агентів;
- враховувати інформацію про невдалі спроби вибору напрямку руху.

Розглядається випадок, коли точка, куди має стати агент, вже зайнята (якщо це не кінцева точка), тоді агент обходить її (обхід перешкоди). При цьому напрямком руху змінюється незначно, що мало впливає на рух агента в цілому.

**Обчислювальні експерименти.** Сплановано та виконано серію обчислювальних експериментів з метою дослідити залежність часу формування многокутника від основних параметрів задачі. План обчислювального експерименту:

1. Задання параметрів експерименту.
2. Повторити задану кількість раз:
  - 2.1. Випадкове розміщення агентів.
  - 2.2. Наказ.
  - 2.3. Для усіх агентів виконати алгоритм.
  - 2.4. Збереження вихідних величин.

До параметрів експерименту належать:

- $N$  – кількість агентів у колективі,
- $M$  – розмір обмеженого простору,
- $S$  – максимальна швидкість агента,
- $n$  – кількість вершин многокутника,
- $(x_c, y_c)$  – координати центра многокутника,
- $e_i$  – кількість агентів, що мають стояти на одній стороні,
- $R$  – радіус видимості засобів детектування сусідніх агентів.

До вихідних величин належать:

- $T$  – час формування многокутника,
- $T_{id}$  – найменший можливий час формування многокутника,
- $T_i$  – час, за який  $i$ -й агент виконує наказ,
- $W$  – показник ефективності роботи алгоритму,
- $L_i$  – відстань, яку проходить  $i$ -й агент від початкового розташування до свого місця в многокутнику,
- $\tau_i$  – середній час простоювання кожного агента, викликаний недосконалістю алгоритму.

Ефективність розроблених алгоритмів визначається так. Розраховується найкращий (ідеальний) розв'язок задачі, з яким порівнюється розв'язок задачі, отриманий за допомогою цього алгоритму. Ідеальний розв'язок знаходимо так: прораховуємо усі можливі комбінації розміщення агентів по точках многокутника. Для кожного такого розташування рахуємо час його досягнення, час за який агенти займуть це розташування, за умови, що вони не шукатимуть кінцеву точку, а відразу будуть іти до неї. Найменший такий час і буде ідеальним розв'язком. Показник ефективності алгоритму  $W$  визначається у вигляді відношення найменшого можливого часу побудови многокутника  $T_{id}$  до часу побудови многокутника цим алгоритмом  $T$

$$W = \frac{T_{id}}{T}.$$

Початкове розташування агентів здійснюється випадковим чином. Тому ефективність розглядається як середнє арифметичне значення ефективностей роботи алгоритму при певній, достатньо великій, кількості повторів алгоритму. Досліджено залежність часу побудови многокутника (результат роботи алгоритму) і ефективності розроблених алгоритмів від розміру та форми середовища, кількості агентів у колективі, характеристик агентів (радіус видимості, максимальна швидкість), кількості вершин многокутника.

**Результати досліджень.** Проаналізуємо, як залежить час побудови многокутника від розташування його в просторі. Розглянемо випадок, коли кожен агент стоїть у вершині многокутника. Коли агенти стоять не лише у вершинах, але й на сторонах – міркування аналогічні. Ефективність алгоритму завжди дорівнює одиниці, коли кожен агент при визначенні найближчої вершини отримує свою. Розіб'ємо увесь простір на  $v$  ділянок таким чином, що  $i$ -та ділянка містить  $i$ -ту вершину  $i$ , попавши в неї, агент знаходиться найближче до  $i$ -ї вершини. Нехай  $S$  – площа середовища,  $S_i$  – площа  $i$ -ї ділянки. Ймовірність, що агент при початковому розташуванні попаде в  $i$ -ту ділянку:

$$P_i = \frac{S_i}{S}.$$

Знайдемо ймовірність події  $B$ : при початковому розташуванні кожен агент потрапить у свою ділянку. Нехай подія  $A$  –  $i$ -й агент потрапив в  $i$ -ту ділянку. Агенти розташовуються незалежно, отже,  $P(A) = P_1 \cdot P_2 \cdot \dots \cdot P_{\text{vertices}}$ . Але для задачі немає значення, який агент в яку ділянку потрапить, оскільки ймовірність потрапити в  $i$ -ту ділянку для усіх агентів однакова, то  $P(B) = \text{vertices!} \cdot P(A)$  – шукана ймовірність.

$$P(B) = v! \cdot P_1 \cdot P_2 \cdot \dots \cdot P_v = v! \cdot \prod_{i=1}^v \frac{S_i}{S}.$$

Отже,  $P(B)$  – це функція від  $v$  аргументів. Інтуїтивно очевидно, що  $P(B)$  досягає максимуму коли усі  $S_i$  рівні. У загальному випадку цього досягнути неможливо, бо многокутник може бути довільним. Але чим ближче до центра середовища розташований заданий многокутник, тим вони рівніші, зростає ймовірність  $P(B)$ , зменшується час побудови многокутника і зростає ефективність алгоритму.

Розглянемо залежність  $t = f(x_c, y_c)$ , де  $x_c, y_c$  – умовний центр многокутника. При  $M = 100$  розмір середовища в умовних одиницях. Колектив агентів будував правильний п'ятикутник.

У результаті обчислювального експерименту одержано ряд залежностей при різних залежностях  $y_c = g(x_c)$  (рис. 2). З графіків видно, що з наближенням  $(x_c, y_c)$  до центра середовища час роботи алгоритму спадає до спільної для усіх залежностей  $y_c = g(x_c)$  точки. Можна помітити симетрію, що свідчить про те, що на ефективність алгоритму впливає не лише відстань від  $(x_c, y_c)$  до центра середовища, але й напрямок зміщення. Це пов'язано з тим, що правильний п'ятикутник осі симетрії.

На основі наведених залежностей можна побудувати графіки залежності ефективності алгоритму від розміщення многокутника –  $W = f(x_c, y_c)$  (рис. 3).

На співвідношення  $S_i$  впливає не лише розташування многокутника та його форма, але й форма середовища. Міркування аналогічні. При однаковій площі ефективність алгоритму буде більша при тій формі середовища, при якій  $S_i$  є якомога рівнішими.

Час роботи алгоритму у загальному випадку дорівнює

$$T_i = \frac{L_i}{S} + \tau_i, T = \max(T_i),$$

де  $\tau$  – час, який не залежить від руху агента і визначається особливостями алгоритму (час простоювань). Передбачити його неможливо, тому що він залежить не лише від кількості агентів у колективі, що будує многокутник, від параметрів агентів та їх співвідношення (швидкість, радіус зору агентів, їх відношення), від відстані між точками в многокутнику; та інших чинників, але й від ряду таких випадкових величин, як розташування агентів, випадковий вибір, який здійснюють агенти. З графіків отриманих експериментально можна побачити, що  $\tau$  майже прямо пропорційне кількості агентів (рис. 4).

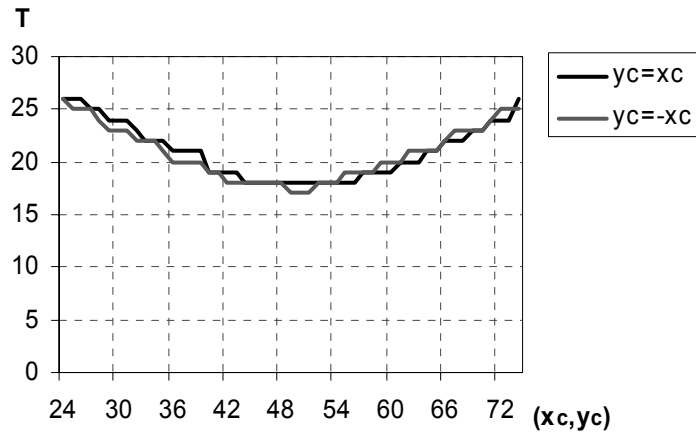


Рис. 2. Залежність часу формування многокутника  $T$  від його розташування в просторі  $(x_c, y_c)$ ,  $N=5$ ,  $v=5$ ,  $M=200$

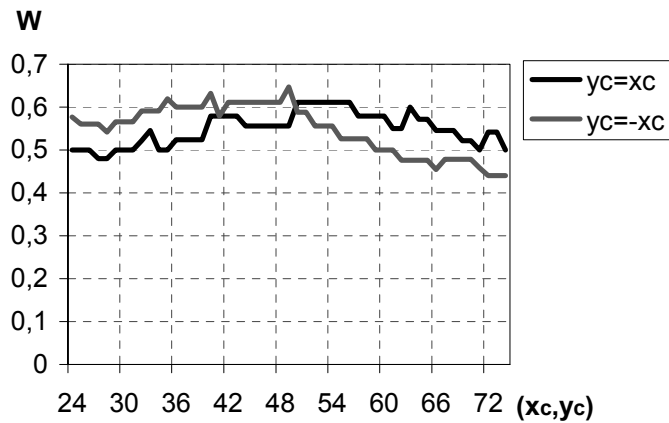


Рис. 3. Залежність показника ефективності алгоритму  $W$  від розташування многокутника в просторі  $(x_c, y_c)$ ,  $N=5$ ,  $v=5$ ,  $M=200$

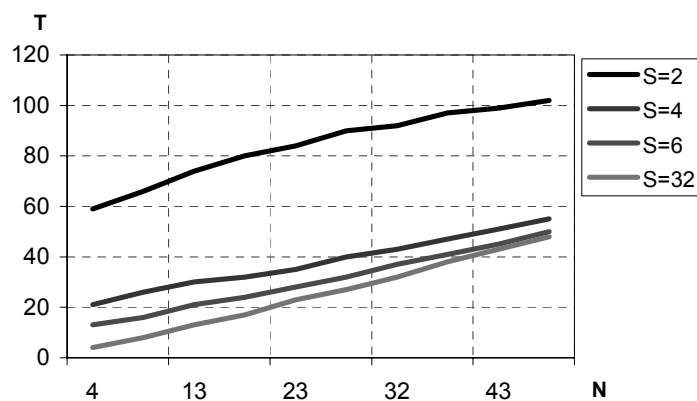


Рис. 4. Залежність часу побудови многокутника  $T$  від кількості агентів у колективі  $N$  для різних швидкостей агента  $S=2, 4, 6, 32$ ,  $M=200$

Відстань  $L$ , яку в середньому проходить кожний агент під час побудови многокутника, в першому наближенні прямо пропорційна розміру середовища:  $L = a \cdot M + b$ , де коефіцієнти  $a$  і  $b$  можуть бути розраховані на основі отриманих експериментальних залежностей (рис. 5).

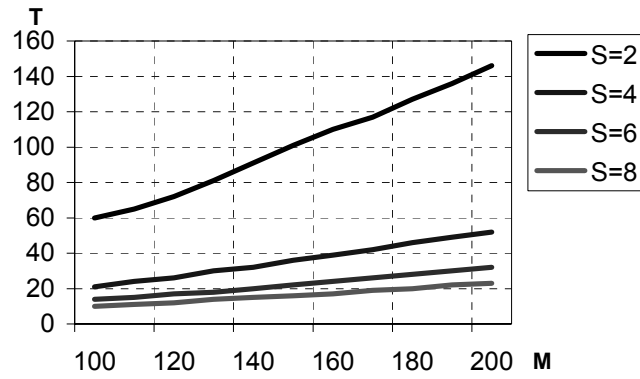


Рис. 5. Залежність часу побудови многокутника  $T$  від розміру простору  $M$  для різних швидкостей агентів  $S=2, 4, 6, 8, N=18$

$S$  – швидкість агента. Час обернено залежить від швидкості. Що ілюструють відповідні графіки (рис. 6).

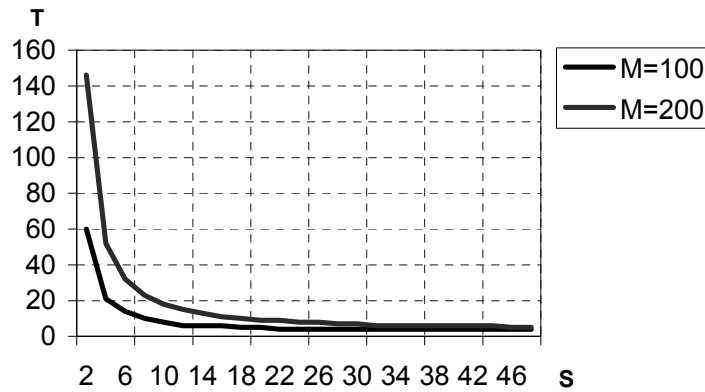


Рис. 6. Залежність часу побудови многокутника  $T$  від швидкості агента  $S$  для різних розмірів простору  $M=100, 200, N=5, v=5$

Від швидкості агентів, розміру середовища, кількості агентів та інших параметрів залежить не лише час побудови многокутника, але й ефективність алгоритму. Для прикладу наведено графіки залежності ефективності від швидкості при різних розмірах середовища (рис. 7). Отже,

$$T = \frac{a \cdot M + b}{S} + \tau.$$

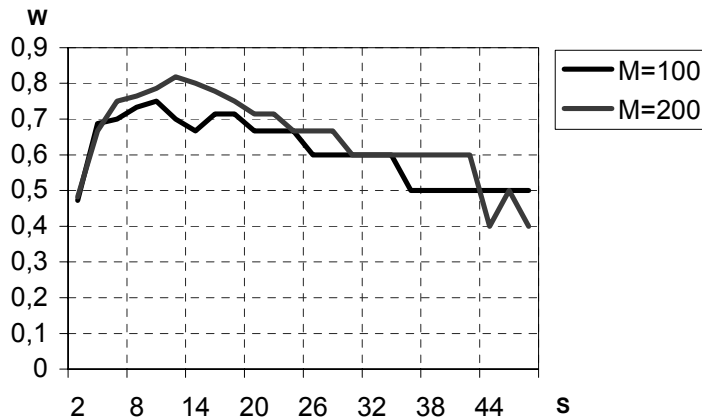


Рис. 7. Залежність показника ефективності алгоритму  $W$  від швидкості агента  $S$  для різних розмірів простору  $M=100, 200, N=5, v=5$



Час побудови многокутника прямо пропорційний розміру середовища  $M$ ; обернено пропорційний максимальній швидкості агентів  $S$  і прямо пропорційний часу простоювань  $t$ .

**Застосування результатів досліджень.** Колектив автономних підводних роботів-освітлювачів самостійно будує многокутник для освітлення майданчика глибинного будівництва. Роботи-охоронці самостійно будують многокутник навколо об'єкта, який вони охороняють. Колектив агентів здійснює пошук, для локалізації території пошуку вони будують многокутник. Побудова многокутника може використовуватися в розподіленій робототехніці [7]: перед стикуванням складові частини повинні зайняти певне розташування, це можуть бути вершини многокутника.

**Висновки.** Розглянуто актуальну проблему розробки алгоритмічного забезпечення децентралізованих багатоагентних систем (колективів агентів). Запропоновано алгоритми для розв'язання задачі побудови многокутника. Проведено ряд обчислювальних експериментів. На основі отриманих даних проаналізовано час роботи алгоритму і ефективність. Наведено графічні залежності, що ілюструють властивості алгоритму. Розглянуто можливість застосування розроблених алгоритмів.

Особливість розв'язаної задачі полягає в тому, що точно відомо місце розташування побудованого многокутника. Це має свої переваги, але обмежує можливість порівняння алгоритму з іншими алгоритмами побудови многокутника (при роботі яких розташування побудованого многокутника не можна передбачити). При постановці задачі введено ряд обмежень – двовимірний простір, відсутність перешкод, голономність агентів. При розв'язанні прикладних задач в реальному середовищі їх не буде. Це дає простір для розвитку розроблених алгоритмів.

1. Woolridge Michael. *An Introduction to MultiAgent Systems*, John Wiley & Sons. – 2002.
2. Parunak H. Van Dyke, Brueckner Sven. *Entropy and Self-Organization in Multi-Agent Systems, Proceedings of the International Conference on Autonomous Agents (Agents 2001)*. – 2001. – P. 124–130.
3. Sugihara Kazuo, Suzuki Ichiro. *Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots*, *Journal of Robotic Systems* 13, 3, March 1996. – P. 127–139.
4. Crombie Duncan. *The Examination and Exploration of Algorithms and Complex Behaviour to Realistically Control Multiple Mobile Robots*, Faculty of Engineering and Information Technology, The Australian National University, 1997.
5. Craig W. Reynolds, *Steering Behaviors For Autonomous Characters*, Sony Computer Entertainment America, presented on Game Developers Conference, February 10, 1999.
6. Cem Ünsal, John S. Bay, *Spatial Self-Organization in Large Populations of Mobile Robots*. – Virginia Polytechnic Institute and State University, 1998.
7. Keller John. *Military scientists investigate networks of tiny robots for counter terrorism and special operations // Military&Aerospace Electronics*. – May 2002. – Vol. 13, № 5. – P. 1, 5.