

УДК 004.9: 007.5

DOI 10.36910/10.36910/6775-2313-5352-2024-25-21

Фетісенко С. В., Гуменюк Л. О.

Луцький національний технічний університет

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕХАНІЗМІВ ГРУПОВОЇ ПОВЕДІНКИ РОБОТІВ

Робота присвячена дослідженню проблеми ефективної координації роботів для спільної роботи та розробці програмного забезпечення для управління групою роботів на основі механізмів групової поведінки. Розроблено програмне забезпечення для візуалізації результатів розподілу завдань для групи роботів використовуючи алгоритми СВАА та СВВА. Результати представлені для 2D та 3D варіантів. Проведено оцінку впливу параметрів та розміру завдань на групову поведінку роботів.

Ключові слова: мультироботні системи, групова поведінка, моделювання, СВАА, СВВА.

Постановка проблеми. Із запровадженням часткової автоматизації з використанням засобів керування та комп'ютерів із програмованою пам'яттю в 70-х роках, також відомої як третя промислова революція, використання автоматизації у виробничих процесах без втручання людини значно зросло. Одним із прикладів цього є роботи, які виконують запрограмовані послідовності без втручання людини. Тепер, із запровадженням четвертої промислової революції, також відомої як індустрія 4.0, виробничі процеси, які вже використовують комп'ютерні технології, об'єднані в мережу, у якій усі комп'ютери можуть обмінюватися відповідною інформацією.

Прикладом таких мереж є мережі автономних мобільних роботів. Застосування мобільних роботів розширилося від традиційних наземних завдань збору та доставки до повітряних, підводних і навіть космічних застосувань. Експлуатація таких засобів також розширилася від одного робота до систем з кількома роботами. У таких системах роботи повинні постійно спілкуватися з виробничими та складськими зонами, щоб точно знати, коли транспортувати певний матеріал до певного місця на виробництві.

Актуальність проведення дослідження проблеми ефективної координації роботів для спільної роботи обумовлена сучасними викликами у галузі робототехніки та автоматизації. Зростання зацікавленості у використанні груп роботів для вирішення різноманітних завдань вимагає розвитку ефективних механізмів групової поведінки. Групова поведінка роботів має стратегічне значення для підвищення продуктивності та функціональності робототехнічних систем.

Аналіз останніх досліджень. Дослідження кооперативних багатоагентних роботів акцентуються на трьох ключових аспектах: типах агентів (як однорідні, так і неоднорідні), архітектурі керування (реактивна, пошук консенсусу та гібридна) і комунікації (неявна і явна). Вибір і розробка архітектури керування та зв'язку мають визначальне значення для забезпечення ефективної координації між агентами, враховуючи їх збалансовану взаємодію та поведінку [1].

Централізоване і розподілене управління – це два підходи до управління багатьма агентами. У централізованому контролі глобальна інформація використовується для розрахунку шляху, траєкторії та позицій агентів [2].

Інформація може бути передана агентам через відповідне середовище зв'язку, що є перевагою, оскільки агенти можуть отримувати інформацію від центрального процесора. Проте головна проблема централізованого контролю полягає в тому, що зі збільшенням кількості агентів обчислення стають надто великими, оскільки є лише один централізований процесор, що керує всією системою. Велике обчислення, час і енергоспоживання призводять до проблем. Таким чином, для вирішення цього, запропоновано гібридний підхід [2, 3], який збалансовує централізоване та розподілене управління [4].

Розподілений контроль виявився масштабним, адаптивним, гнучким і надійним в області багатьох агентів як у статичному, так і в динамічному середовищі. У розподіленому управлінні головна проблема полягає в ефективному та надійному розподілі завдань між неоднорідними агентами, що є складнішим порівняно з однорідними агентами.

Обмежений діапазон сприйняття та низька пропускну здатність є фізичними обмеженнями в розподіленому підході. З обмеженою локальною інформацією агент не може ефективно контролювати поведінку групи.

Оскільки мультиагентні роботи повинні взаємодіяти та спілкуватися для ефективної співпраці, проблема досягнення консенсусу у вигляді узгодження між агентами стала важливим об'єктом дослідження протягом останніх років.

Управління формуванням є ключовим аспектом для координації та контролю групи мультиагентних роботів [5].

При розробці алгоритмів для формування мобільних роботів важливо враховувати численні аспекти, такі як стабільність структури, управління різними конфігураціями, забезпечення безпеки та врахування невизначеностей у розташуванні [5]. Додатково вказується на аспекти, пов'язані з генерацією структури, її перетворенням, визначенням та відслідковуванням, а також розподілом обов'язків у процесі створення.

Існує три основні стратегії управління формуванням, запропоновані дослідниками [5], такі як: на основі поведінки, віртуальна структура, лідер-підлеглий. Кожен метод управління формуванням має свої переваги та недоліки.

Метою роботи є розробка програмної системи для управління групою роботів на основі механізмів групової поведінки.

Викладення основного матеріалу. Для програмної реалізації алгоритму аукціону на основі консенсусу (СВАА) та алгоритму пакету на основі консенсусу (СВВА) для розподілу завдань було обрано мову програмування Python.

За основу взято процедурний підхід до програмування, який передбачає розбиття коду на окремі функції. Цей підхід сприяє більшій модульності та організованості коду, що робить його зрозумілішим.

Алгоритм аукціону на основі консенсусу (СВАА) з графічним виводом. Алгоритм Cooperative Bargaining Auction Algorithm (СВАА) для розподілу завдань між роботами включає в себе кілька елементів, таких як ініціалізація списку роботів за допомогою СВАА агентів, створення повністю з'єднаної мережі та ітеративний процес аукціонування та узгодження.

Розглянемо алгоритм поетапно.

Перший елемент – ініціалізація СВАА агента включає початкову позицію агента та кількості завдань. Також створюються переліки в яких зберігається інформація про те, які завдання були призначені роботам, а також про виграшні ставки, які пов'язані з цими завданнями. Визначаємо початковий бал для кожного завдання на основі евклідової відстані.

Другий елемент – вибір завдання агентом на основі локальної інформації. Якщо агент ще не взявся за жодне завдання, він обирає завдання з більшим балом, ніж відповідна виграшна ставка. Після вибору завдання він позначає його як виконане та оновлює виграшну ставку для цього завдання.

Третій елемент – оновлення призначень завдань. Цей елемент оновлює призначення завдань на основі інформації про виграшні стави від сусідів, перевіряє можливе випередження та визначає переможця за оновленим списком виграшних ставок. Якщо агент не є переможцем, він звільняє призначення завдання. Після цього повертається інформація про те, чи збігся агент з попереднім станом призначення завдань.

Четвертий елемент – відправлення повідомлень. Повертає інформацію виграшних ставок агента.

Основна частина алгоритму – це аукціонування та узгодження розподілу завдань між роботами. Приклади роботи розподілу завдань з використанням алгоритму аукціону на основі консенсусу (СВАА) з графічним виводом у 2D варіанті для чотирьох роботів наведено на рисунку 1.

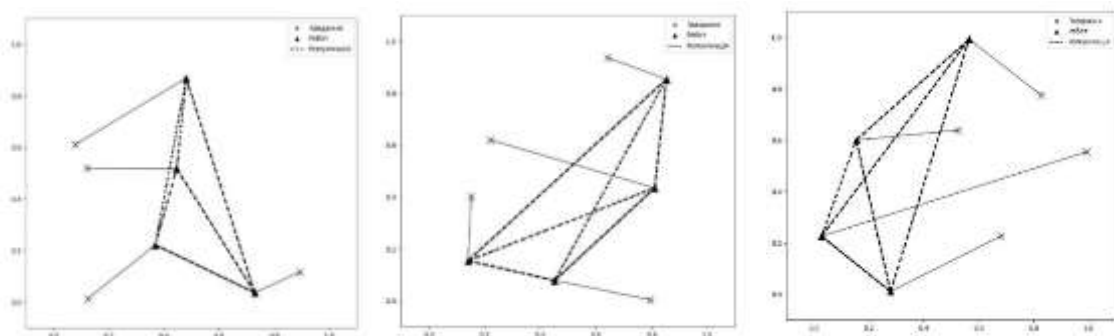


Рисунок 1 – Приклади роботи розподілу завдань з використанням алгоритму аукціону на основі консенсусу (СВАА) у 2D варіанті для чотирьох роботів

Основна різниця між 2D та 3D варіантами полягає у вимірах простору та кількості координат для кожного завдання та агента.

Приклади роботи розподілу завдань з використанням алгоритму аукціону на основі консенсусу з графічним виводом у 3D варіанті для чотирьох та шести роботів наведено на рисунку 2.

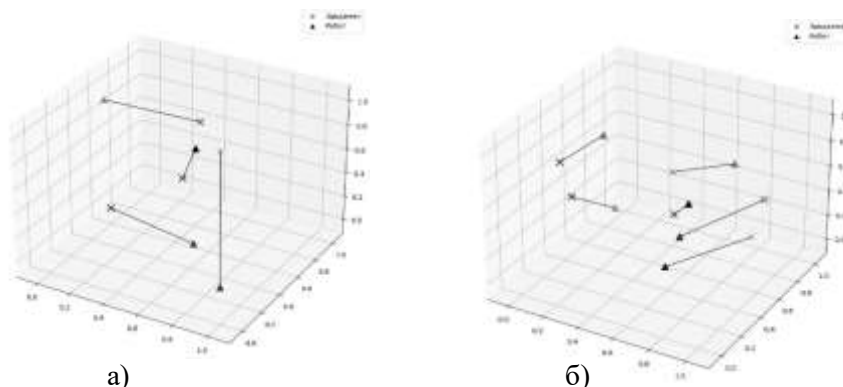


Рисунок 2 – Приклади роботи розподілу завдань з використанням алгоритму аукціону на основі консенсусу (CBAA) у 3D варіанті: а) чотири роботи; б) шість роботів

Алгоритм пакета на основі консенсусу (CBBA) з графічним виводом. Алгоритм Cooperative Bundle-Based Multi-Robot Task Allocation (CBBA) призначений для ефективного розподілу завдань між роботами у мультиагентних системах. Він включає такі основні етапи.

Перший етап – ініціалізація агента. На цьому етапі кожен агент у системі ініціалізується з унікальним ідентифікатором, швидкістю, кількістю завдань, які він може виконати, кількістю агентів загалом у системі та розміром бандлу (групи) завдань, який агент може взяти на себе одночасно. Під час ініціалізації формуються структури даних, такі як масиви з виграшними ставками, масиви для позначення виконаних та призначених завдань, що допомагають агентам в подальших обчисленнях та координації. Цей етап дозволяє створити початковий стан для кожного агента у системі, з урахуванням його характеристик та параметрів, необхідних для ефективного виконання завдань.

Другий етап – вибір завдань. На цьому етапі кожен агент вибирає завдання на основі локальної інформації та власної стратегії. Якщо агент ще не взявся за жодне завдання, він обирає те, що має більший бал, ніж відповідна виграшна ставка. Для кожного завдання агент має певний бал, який може бути визначений, наприклад, на основі евклідової відстані до завдання чи інших параметрів. Після вибору завдання агент позначає його як виконане та оновлює виграшну ставку для цього завдання. Цей етап допомагає агентам ефективно вибирати завдання для виконання з урахуванням їхньої власної локальної інформації та стратегій.

Третій етап – оновлення призначень завдань. На даному етапі агент оновлює призначення завдань на основі інформації, яку він отримав від інших агентів. Ця інформація включає в себе виграшні ставки, що були представлені сусідніми агентами за певні завдання. Агент перевіряє можливі випередження та визначає переможця за оновленим списком виграшних ставок. Якщо агент не є переможцем, він звільняє призначення завдання та повертає його в пул доступних завдань. Цей етап дозволяє агентам адаптуватися до змін в системі та узгоджувати свої дії з іншими учасниками з метою досягнення оптимального розв'язку.

Четвертий етап – відправлення повідомлень. На цьому етапі агент відправляє інформацію про свої виграшні ставки та виконані завдання сусіднім агентам. Ця інформація може включати в себе такі дані, як виграшні ставки за кожне завдання, список виконаних завдань та інші важливі параметри. Взаємодія та обмін інформацією між агентами є важливим елементом алгоритму CBBA і допомагає координувати їхні дії з метою досягнення оптимального розподілу завдань.

П'ятий етап – аукціонування та узгодження розподілу завдань між роботами. На цьому етапі запускається цикл, в якому відбувається процес аукціонування та узгодження. Кожен агент пропонує свою ціну за виконання певного завдання. Потім вони конкурують між собою, щоб визначити, хто отримає право виконати це завдання. Після визначення переможця, завдання призначається цьому роботу, і процес узгодження завдань продовжується з наступними завданнями та роботами. Інформація про оновлені призначення та виграші виводиться для аналізу. Після того, як всі роботи збіглися у розподілі завдань, алгоритм виходить з циклу.

Шостий етап – вивід результату. На завершальному етапі виводиться результат розподілу завдань між роботами.

Приклади роботи розподілу завдань з використанням алгоритму пакету на основі консенсусу (CBBA) для дванадцяти завдань з графічним виводом у 2D варіанті для чотирьох роботів наведено на рисунку 3.

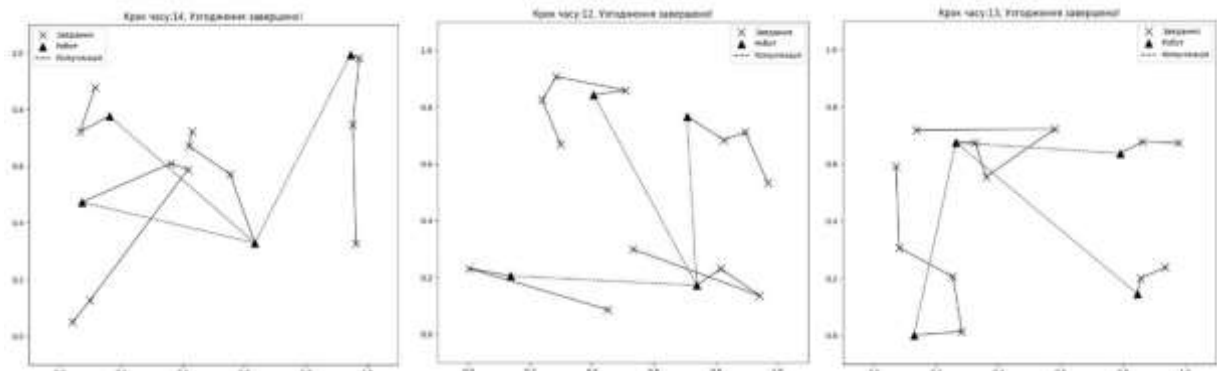


Рисунок 3 – Приклади роботи розподілу завдань з використанням алгоритму пакету на основі консенсусу (CBBA) у 2D варіанті для чотирьох роботів

Основна різниця між 2D та 3D полягає у кількості координат для кожного завдання та агента.

Приклади роботи розподілу завдань з використанням алгоритму пакету на основі консенсусу (CBBA) з графічним виводом у 3D варіанті для чотирьох роботів і шести, восьми, десяти та дванадцяти завдань наведено на рисунку 4.

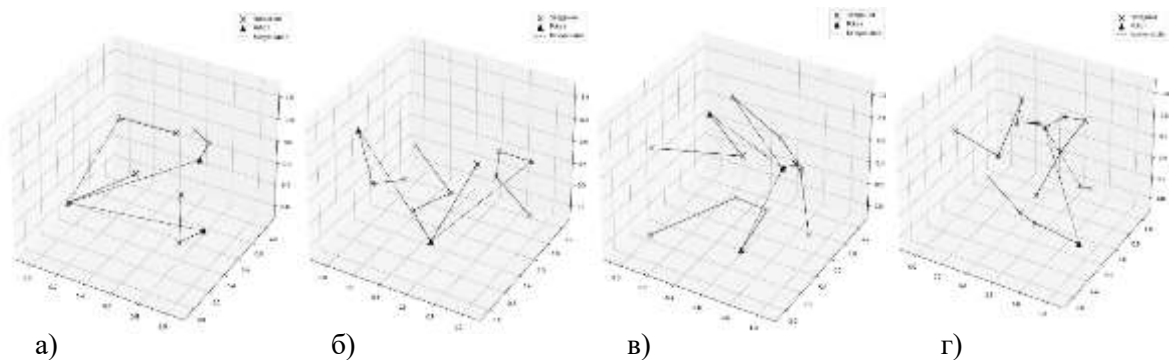


Рисунок 4 – Приклади роботи розподілу завдань з використанням алгоритму пакету на основі консенсусу (CBBA) з графічним виводом у 3D варіанті для чотирьох роботів і а) шести завдань, б) восьми завдань, в) десяти завдань, г) дванадцяти завдань

Розглянуто вплив розміру завдань та параметрів на групову поведінку роботів за допомогою алгоритмів СВАА (Consensus-Based Bundle Algorithm) та СВБА (Consensus-Based Bundle Algorithm). Такий аналіз є важливим для оптимізації роботи алгоритмів та покращення їхньої ефективності в реальних умовах застосування.

Виконано програмну реалізацію алгоритму СВАА з різною кількістю роботів і завдань, коли кількість роботів дорівнює 50, 100, 150, 200, а кількість завдань дорівнює кількості роботів.

Визначено, що алгоритм СВАА має тенденцію збільшувати час виконання при збільшенні обсягу оброблюваних даних. Такий ріст може бути пов'язаний зі збільшенням обчислювальної складності алгоритму або збільшенням кількості операцій, які потрібно виконати для обробки кожного завдання.

Отримані дані описано за допомогою квадратичної регресії. Отримана регресія добре описує збільшення часу виконання алгоритму СВАА від кількості завдань і може використовуватись для прогнозування часу роботи алгоритму.

Виконано програмну реалізацію алгоритму СВБА для п'яти роботів з різною кількістю завдань, кількість завдань змінювали від 10 до 50 з кроком 10.

З форми отриманої залежності видно, що час розподілу збільшується зі зростанням кількості завдань. Зростання часу розподілу прискорюється, особливо при збільшенні кількості

завдань після певного значення (після 30 або 35). З огляду на це, припускаємо, що форма розподілу носить експоненціальний характер.

Отримане рівняння гарно описує те, як час роботи алгоритму СВВА змінюється зі збільшенням кількості завдань. Його можна використовувати для прогнозування часу роботи алгоритму залежно від кількості завдань.

Висновки. Таким чином, запропоновано програмну реалізацію двох алгоритмів на основі консенсусу: аукціону на основі консенсусу (СВАА) та алгоритму пакету на основі консенсусу (СВВА) для ефективного розподілу завдань у групі роботів (агентів). Результати роботи програм продемонстрували, що обидва алгоритми можуть бути успішно використані для розподілу завдань у децентралізованих системах з великою кількістю роботів (агентів). Проаналізовано вплив кількості завдань на час їх розподілу між роботами згідно алгоритмів СВАА і СВВА. Відзначено, що кількість завдань впливає на час роботи алгоритмів СВАА та СВВА, який збільшується зі збільшенням завдань. Зазначено, що для опису цього часу в алгоритмі СВАА підходить квадратична регресія, а для СВВА добре описує цю залежність експоненціальний розподіл.

Збільшення кількості завдань суттєво збільшує середній час виконання, що може вказувати на ускладнення пошуку компромісу між роботами. Проте кожна ітерація алгоритму СВВА призводить до зменшення кількості нерозподілених завдань, що свідчить про покращення координації та скорочення часу вирішення конфліктів, а отже, часу на кожну наступну ітерацію.

Отримані результати підтвердили ефективність розробленого алгоритму та його здатність адаптуватися до різних умов та вимог, що робить його потенційно корисним для широкого кола застосувань у реальних умовах.

Інформаційні джерела

1. Parker L-E. Heterogeneous Multi Robot Cooperation. URL: https://www.researchgate.net/publication/34328142_Heterogeneous_multi-robot_cooperation (дата звернення 20.05.2024).
2. Brunete A., Hernando M., Gambao E., Torres J.. A behavior based control architecture for heterogeneous modular, multi configurable, chained micro robots. URL: <https://www.sciencedirect.com/science/article/abs/pii/S092188901200173X> (дата звернення 20.05.2024).
3. Simonin O., Grunder O. A cooperative multi robot architecture for moving a paralyzed robot. URL: https://www.researchgate.net/publication/228795153_A_cooperative_multi-robot_architecture_for_moving_a_paralyzed_robot (дата звернення 20.05.2024).
4. Franco M-L., Sanchez E., Alanis A. Franco C., Daniel N. Decentralized control for stabilization of nonlinear multi agent systems using neural inverse optimal control. URL: https://www.researchgate.net/publication/279517678_Decentralized_Control_for_Stabilization_of_Nonlinear_Multi-Agent_Systems_Using_Neural_Inverse_Optimal_Control (дата звернення 20.05.2024).
5. Chen Y., Wang Z. Formation control: A review and a new consideration. URL: https://www.researchgate.net/publication/224623514_Formation_control_A_review_and_a_new_consideration (дата звернення 20.05.2024).

Fetisenko S., Gumeniuk L.

Lutsk National Technical University, Lutsk, Ukraine

SOFTWARE IMPLEMENTATION OF MECHANISMS OF GROUP BEHAVIOR OF ROBOTS

The article is devoted to the study of the problem of effective coordination of robots for joint work and the development of software for managing a group of robots based on group behavior mechanisms. The software for visualizing the results of task distribution for a group of robots using CBAA and CBBA algorithms was developed. The results are presented for 2D and 3D variants. The influence of parameters and task size on the group behavior of robots is evaluated.

Keywords: multi-robot systems, group behavior, modeling, CBAA, CBBA.