



Науковий вісник НЛТУ України
Scientific Bulletin of UNFU

<http://nv.nltu.edu.ua>

<https://doi.org/10.15421/40270636>

Article received 13.09.2017 р.

Article accepted 28.09.2017 р.

УДК 004.[056+832]

ISSN 1994-7836 (print)

ISSN 2519-2477 (online)

@ ✉ Correspondence author

V. M. Teslyuk

vasyl.teslyuk@gmail.com

В. М. Теслюк, А. Я. Лозинський, Т. В. Теслюк

Національний університет "Львівська політехніка", м. Львів, Україна

ЗАСТОСУВАННЯ МУЛЬТИАГЕНТНОГО ПІДХОДУ В ПРОЦЕСІ РЕАЛІЗАЦІЇ ІГРОВОЇ ПРОГРАМИ "BACTERIAL WAR"

З огляду на інтенсивний розвиток мережевих інформаційних технологій, актуальним є розроблення методів та засобів розподіленого прийняття рішення, які ґрунтуються на кількох інтелектуальних агентах. Проаналізовано імітаційні моделі для ігрових систем, які використовують мультиагентний підхід прийняття рішень. Використання мультиагентного підходу для ігрових систем розпочалося в далекі 1960-ті роки і з кожним роком складність багатоагентних систем в ігрових та імітаційних системах збільшувалася. Збільшення використання мультиагентних систем сприяло створенню багатьох технологій та стандартів, які спрощують розробку мультиагентних систем. Описано розроблену ігрову програму "Bacterial War" та імітаційну модель, яка симулює комунікацію бактерій в навколишньому середовищі. Розроблена програма використовує основні принципи мультиагентних систем, що забезпечує високу надійність та швидкість системи. У процесі розроблення програми використано гомогенну модель комунікації агентів, що забезпечило високий рівень реагування агентів на зовнішні чинники. Пошук цілей відбувається за допомогою алгоритму пошуку мінімального шляху A*. Надсилання інформації між агентами відбувається за допомогою серверної частини, що забезпечує високу швидкість програми. Розроблену програму написано за допомогою мови програмування Java та допоміжної бібліотеки LibGDX, що забезпечує використання системи на кількох платформах.

Ключові слова: імітаційні системи; прийняття рішень; кооперативність; гомогенні агенти; колонія.

Вступ. Підвищення продуктивності сучасних обчислювальних систем, розвиток інформаційних мережевих технологій, вдосконалення нових мультимедій та засобів паралельного і розподіленого програмування – дали новий поштовх у дослідженні колективної поведінки автоматів (агентів), розподіленого штучного інтелекту, децентралізованого керування та прийняття рішень (Fudenberg, & Levine, 1998). Системи з децентралізованим керуванням будують на основі автономних, адаптивних до середовища агентів (активних елементів), які досягають системних цілей шляхом локального обміну інформацією та координації дій (Lytvyn, et al., 2017).

Загалом агент – це певний складник системи, що реагує на середовище через давачі або інші засоби збору інформації та діє на це середовище через актуатори та ефектори, контролери та інші системи, за допомогою яких агент так чи інакше впливає на навколишнє середовище (Lozynskyi, et al., 2017a; 2017b).

Багатоагентні системи – це системи, що використовують для розв'язання таких проблем, які складно або неможливо вирішити за допомогою одного агента чи монолітної системи (Lytvyn, et al., 2017).

1. Огляд наявних імітаційних моделей

Мультиагентні системи широко використовують у багатьох сферах нашого життя: логістиці, графіці, нав-

чанні, комп'ютерних іграх, імітаційних моделях та інших сферах. Широка сфера застосувань мультиагентних систем сприяла створенню нових засобів розроблення цих систем. Одними з найпопулярніших є: Net-Logo, VisoalBots, MASON, Repast, Jade (Lozynskyi, et al., 2017a).

Існує безліч симуляторів, які побудовані на основі мультиагентних систем (Kravets, 2012; 2017). Для прикладу, викладач Каліфорнійського університету Chapel Hill розробив імітаційну програму під назвою "Density Sensitive Crowd Simulation", яка ілюструє поведінку натовпу (Goldberg, 1989). Ця система призначена для визначення найоптимальніших пішохідних траєкторій. Density Sensitive Crowd Simulation використовують під час моделювання стадіонів, для того, щоб створити найоптимальніші плани евакуації людей, зменшити скупченість людей та ін. Також Chappel Hill розробив імітаційну модель "Directing Crowd Simulating Usin Navigation Field", яка призначена для дослідження поведінки натовпу під час переходу вулиць (Russel & Norvig, 1995). За допомогою цієї імітаційної моделі розробляють транспортні розгалуження (Lozynskyi, et al., 2017b).

Однією з перших ігор, яка побудована на основі мультиагентних систем, вважають гру "Pacman" (Ni,

Інформація про авторів:

Теслюк Василь Миколайович, д-р техн. наук, професор кафедри ICT. **Email:** vasyt.teslyuk@gmail.com

Лозинський Андрій Ярославович, аспірант кафедри САПР. **Email:** lozynskuj.andrij@gmail.com

Теслюк Тарас Васильович, аспірант кафедри АСУ. **Email:** taras.teslyuk@gmail.com

Цитування за ДСТУ: Теслюк В. М., Лозинський А. Я., Теслюк Т. В. Застосування мультиагентного підходу в процесі реалізації ігрової програми "Bacterial war". Науковий вісник НЛТУ України. 2017. Вип. 27(6). С. 178–181.

Citation APA: Teslyuk, V. M., Lozynskyi, A. Ya., & Teslyuk, T. V. (2017). Application of Multi-Agent Approach in the Process of Implementation of the Game Program "Bacterial War". *Scientific Bulletin of UNFU*, 27(6), 178–181.

<https://doi.org/10.15421/40270636>

Heydt, & Mili, 2002). Мета гри полягає у збиранні їжі в лабіринті, при цьому потрібно уникати зустрічі з ворогами. Одним з основних недоліків цієї гри є примітивність дій агентів (ворогів), оскільки комунікацію агентів не реалізовано (Weiss, & Sen, 1996; Wooldridge, 2002). Однією з найновіших ігрових симуляторів, яка використовує принципи мультиагентних систем – є серія ігор "Fifa". "Fifa" – це імітація футбольної гри, де гравців комунікують між собою і передають м'яч один одному, ціллю гри є забити м'яч у ворота суперника. Порівняно із грою "Pacman", складність мультиагентних алгоритмів в "Fifa" набагато вища, що забезпечила високий рівень комунікації агентів, можливість зміни рівнів складності та багатьох інших додаткових властивостей імітаційної системи (Ni, Heydt, & Mili, 2002). Хоча і в імітаційній моделі, яку використовують у "Fifa", є недоліки, одним з таких – є вузька сфера прийняття рішень агентів, що робить агентів примітивними та прогнозованими.

2.1. Алгоритм роботи агентів в ігровій програмі "Bacterial war"

Агент – це примітивний складник системи, що складається зі свого процесора, давачів і т. ін. У розробленій програмі агентами є кожна бактерія, яка першим кроком шукає противника і наступним надсилає отриману інформацію на базу даних системи (Stone & Velloso, 2000; Rehtanz, 2003; Jung, 2002).

У роботі реалізовано кілька варіантів агентів, кожен із яких має свій час реакції та внутрішній алгоритм для аналізу ситуації. На рис. 1 зображено загальну схему алгоритму функціонування агентів (Krishna, & Ramesh, 1998; Liu, et al., 1998).

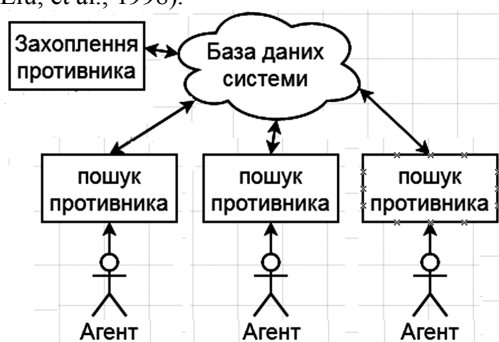


Рис. 1. Схема алгоритму функціонування агентів

Агент відповідає за вибір напрямку руху груп бактерій від заданої колонії. В один момент часу, колонія може вибирати не більше одного напрямку руху (Liu, et al., 1998; Wang, Li, & Chen, 2003). У разі втрати або захоплення колонії, агент змінює своє підпорядкування гравцю. Час реакції визначає, як часто агент здатний включитися, аналізувати ситуацію та вибирати напрямки. Існує три основні підходи до реалізації дій, які реалізовані в агентах:

- 1) Аналіз і пошук суперників, вибір суперника. Проектування маршруту переміщення до суперника, якщо він є досяжний (вироблення стратегії захоплення).
- 2) Аналіз і пошук агентів, вибір агента, якому потрібна допомога. Проектування маршруту переміщення до агента, якщо він є досяжний.
- 3) Аналіз проєктованих маршрутів і корегування рішення.

Аналіз і пошук противника відбувається за допомогою основних ідей алгоритму A*. Алгоритм A* спершу відвідає ті вершини, які ймовірно ведуть до найкоротшого шляху. Аби розпізнати такі вершини, кожній відо-

мій вершині x зіставляється значення $f(x)$, яке дорівнює довжині найкоротшого шляху від початкової вершини до кінцевої, який пролягає через обрану вершину. Вершини з найменшим значенням f обираються в першу чергу.

Функція $f(x)$ для вершини x визначається так:

$$f(x) = g(x) + h(x), \quad (1)$$

де: $g(x)$ – функція, значення якої дорівнює вартості шляху від початкової вершини до x , $h(x)$ – евристична функція, яка оцінює вартість шляху від вершини x до кінцевої.

2.2. Спосіб комунікації в мультиагентній системі

Розроблена програма "Bacterial war" використовує гомогенний принцип комунікації агентів (Goldberg, 1989). Цей клас кооперативних МАС найкраще підходить для агентів, які реалізують однакову мету. На рис. 2 зображено схему функціонування гомогенної мультиагентної системи із властивістю комунікації між агентами.

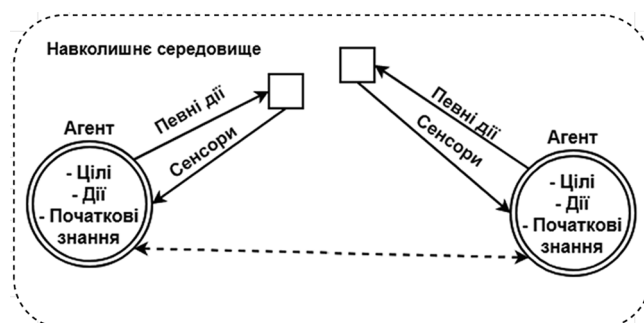


Рис. 2. Схема гомогенної мультиагентної системи із властивістю комунікації (Goldberg, 1989)

Агенти гомогенних МАС, з властивістю комунікації, можуть надсилати сигнали напряму агенту, або зв'язуватися з агентом через серверну частину для покращення синхронізації дій у системі з великою кількістю агентів (Fudenberg, & Levine, 1998). Гомогенний принцип комунікації характеризується високою швидкістю. Висока швидкість забезпечується фільтруванням надісланих сигналів. Агенти, які виконують однакові завдання, мають певний спектр даних, який вони можуть надсилати іншому агенту, що не можна сказати про гетерогенну МАС, де агенти виконують різні завдання й агенти надсилають дані про стан виконаного завдання іншим агентам, які не пристосовані до інших завдань (Fudenberg, & Levine, 1998).

Гомогенний принцип комунікації належить до кооперативних МАС, а це означає, що агенти до виконання завдання знають про присутність інших агентів у системі та навколишньому середовищі (Ni, Heydt, & Mili, 2002). На рис. 3 зображено спосіб комунікації агентів за допомогою серверної частини.

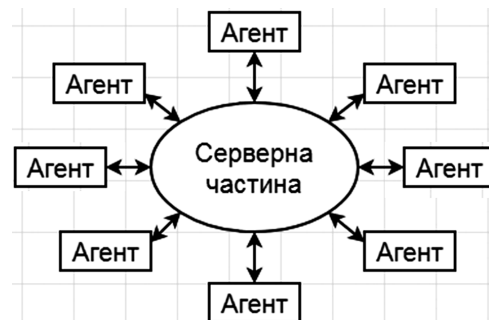


Рис. 3. Схема комунікації агентів

Комунікація агентів відбувається за допомогою серверної частини, а не безпосередньо надсиланням інформації агенту. Цей спосіб комунікації забезпечує високу швидкість, але надійність системи погіршується, оскільки важливу роль відіграє серверна частина. При виході з ладу сервера – вся система перестане працювати.

3.1. Структура програми Bacterial War

Розроблену програму написано за допомогою мови Java в середовищі розробки програмного забезпечення Eclipse. Для написання програми використано бібліотеку LibGDX, яку застосовують для розроблення ігор під кілька платформ. Загалом програма складається з двох основних частин:

- системи вибору і налаштування гри. Ця частина містить головне меню, меню вибору рівнів, вікно про програму, налаштування ігрового процесу, звуку та швидкості.
- системи контролю ігрового поля. Ця частина містить в себе проєктування ігрового поля, контролю ігрової механіки, виконання команд від користувача, управління агентами згідно із їхніми алгоритмами роботи.

3.2. Алгоритм роботи програмної системи

У процесі роботи ігрової програми використано такий побудований алгоритм:

Крок 1. Після вибору рівня гри, відбувається ініціалізація ігрового поля, початкове відображення та ініціалізація агентів.

Крок 2. Після команди про початок гри, запускається ігровий контролер. Він відповідає за управління ігровою механікою, відрахування тиків та внутрішнього часу, відображення ігрового поля та запуск агентів.

Крок 3. На кожному із тиків перевіряється стан кожної із колоній та група бактерій і агенти.

Крок 4. Якщо колонія має змодельований маршрут до інших колоній, то формується та надсилається група бактерій по заданій дорозі.

Крок 5. Якщо група бактерій долітає до колонії супротивника, то вона завдає шкоди колонії. В іншому разі вона підсилює власну колонію.

Крок 6. Якщо агент є вільний, то він аналізує ситуацію: оцінює своїх супротивників та сусідів і приймає рішення про прокладання дороги для покращення ігрової ситуації (виробляє стратегію).

Крок 7. Після завершення тiku відбувається перевірка балансу сил та перевірка умови на завершення гри.

Крок 8. Промальовування розташування колоній, доріг та груп бактерій на ігровому екрані.

Крок 9. Якщо гра не завершена, перехід на крок 3.

3.3. Особливості реалізації системи

На практиці доволі складно досліджувати мультиагентні системи, оскільки передбачити дії агентів практично неможливо, оскільки і неможливо передбачити способи комунікації між агентами. Для більш реального дослідження систем, потрібно використовувати імітаційні моделі, які напряму будуть відображати поведінку агентів у системі.

У роботі побудовано імітаційну модель у вигляді стратегічної ігрової програми, яка містить: бактерії, які складають колонії, що шукають найоптимальніші шляхи для захоплення всіх вершин (планет) й ігрового поля.

Кожен із ботів аналізує ситуацію на полі і синтезує своє рішення, яке було б вигідним для захоплення наступної вершини, тобто яку стратегію треба побудувати для подальшої колонізації ігрового поля та як зміцнити своє

перебування на вже завойованих вершинах. На рис. 4 зображено головне меню імітаційної гри Bacterial War.



Рис. 4. Головне меню імітаційної гри

Окрім цього, в роботі реалізовано інтерактивну співпрацю між ворожими колоніями, які можуть об'єднуватися задля боротьби над сильнішим противником або підтримку атаки в моменти спільного нападу.

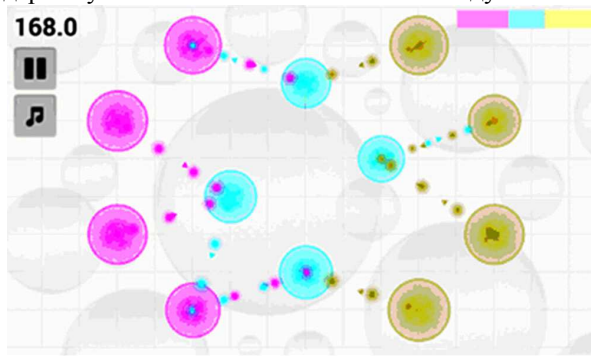


Рис. 5. Приклад функціонування агентів

На рис. 5 зображено приклад функціонування агентів в ігровій імітаційній системі. Цей приклад відображає систему із трьома колоніями різних кольорів. Кожна з колоній – це система, наділена елементами штучного інтелекту, метою якого є захопити все поле. Захоплення поля відбувається шляхом надсилання на колонії противника своїх агентів для збільшення впливу своїх бактерій у заданій колонії.

Висновок

Розроблено імітаційну модель, яка базується на мультиагентному підході прийняття рішень. Комунікацію агентів реалізовано на основі гомогенного принципу MAC. Надсилання повідомлень між агентами відбувається за допомогою серверної частини.

Розроблено систему на основі імітаційної моделі, яка імітує дії бактерій в навколишньому середовищі. Пошук мінімального шляху бактерій відбувається за допомогою алгоритму A*.

Перелік використаних джерел

- Fudenberg, D., & Levine, D. K. (1998). *The Theory of Learning in Games*. MIT Press, 428 p.
- Goldberg, D. E. (1989). Genetic Algorithms in Search. *Optimization and Machine Learning*, Addison-Vesley, 235–241.
- Jung, J., Liu, C. C., Tanimoto, S. L., & Vittal, V. (2002). Adaptation in Load Shedding under Vulnerable Operating Conditions. *IEEE Transactions on Power Systems*, 17(4), 1199–1205. <https://doi.org/10.1109/TPWRS.2002.805023>
- Kravets, P. (2012). Mulyahentna ihrova model pryiniattia rishen z kolovanyamy stratehiamy. *Kompiuterni nauky ta informatsiini tekhnologii: Visnyk NU "Lvivska politekhnika"*, 744, 159–168. [in Ukrainian].

- Kravets, P. (2017). Ihrova model tsentralizovanoho pryiniattia rishen v multyahentnii systemi. Intelektualni systemy pryiniattia rishen i problemy obchysluvalnoho intelektu: tezy Mizhnar. nauk. konf., (pp. 202–163). Kherson: Zaliznyi Port. [in Ukrainian].
- Krishna, V., & Ramesh, V. C. (1998). Intelligent Agents for Negotiations in Market Games, Part 1: Models. *IEEE Transactions on Power Systems*, 13(3), 1103–1108. <https://doi.org/10.1109/59.709106>
- Liu, C. C., Jung, J., Heydt, G. T., Vittal, V., & Phadke, A. G. (2000). The strategic power infrastructure defense (SPID) system: A conceptual design. *IEEE Control Systems Magazine*, 20(4), 40–52. <https://doi.org/10.1109/37.856178>
- Lozynskiy, A. Ya., Tesliuk, V. M., Zelinskyi, A. Ya., & Narushynska, O. O. (2017a). Zastosuvannya multyahentnoho pidkhodu dlia pokrashchennia parametriv synkhronizatsii v ihrovykh systemakh. *Modeliuvannya ta informatsiini tekhnologii*, 79, 183–190. Kyiv: Instytut modeliuvannya v enerhetytsi im. H. Ye. Pukhova NAN Ukrainy. [in Ukrainian].
- Lozynskiy, A. Ya., Tesliuk, V. M., Zelinskyi, A. Ya., & Narushynska, O. O. (2017b). Vdoskonalennia modeli komunikatsii ahentiv vmultyahentnykh systemakh. *Modeliuvannya ta informatsiini tekhnologii*, 78, 190–195. Kyiv: Instytut modeliuvannya v enerhetytsi im. H. Ye. Pukhova NAN Ukrainy. [in Ukrainian].
- Lytvyn, V., Vysotska, V., Pukach, P., & Vovk, D. (2017). Method of functioning of intelligent agents, designed to solve action planning problems based on ontological approach. *Eastern-European Journal of Enterprise Technologies*, 3(2), 11–17. <https://doi.org/10.15587/1729-4061.2017.103630>
- Ni, H., Heydt, G. T., & Mili, L. (2002). Power System Stability Agents Using Robust Wide Area Control. *IEEE Transactions on Power Systems*, 17(4), 1123–1131. <https://doi.org/10.1109/TPWRS.2002.805016>
- Rehtanz, C. (Ed.). (2003). *Autonomous systems and intelligent agents in power system control and operation*. Springer, 320 p.
- Russel, S., & Norvig, P. (1995). *Artificial intelligence – A modern approach*. Prentice Hall, 520 p.
- Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 345–383. <https://doi.org/10.1023/A:1008942012299>
- Sycara, K. P. (1998). Multiagent systems. *AI Magazine*, 19(2), 79–92. <https://doi.org/10.1609/aimag.v19i2.1370>
- Wang, H. F., Li, H., & Chen, H. (2003). Coordinated Secondary Voltage Control to Eliminate Voltage Violations in Power Systems Contingencies. *IEEE Transactions on Power Systems*, 18(2), 588–595. <https://doi.org/10.1109/TPWRS.2003.810896>
- Weiss, G., & Sen, S. (Eds.). (1996). *Adaptation and Learning in Multiagent Systems*. Springer Verlag, Berlin, 384 p.
- Wooldridge, M. (2002). *An Introduction to Multiagent Systems*. John Wiley & Sons (Chichester, England), 288 p.

В. М. Теслюк, А. Я. Лозинський, Т. В. Теслюк

Національний університет "Львівська політехніка", г. Львів, Україна

ПРИМЕНЕНИЕ МУЛЬТИАГЕНТНОГО ПОДХОДА В ПРОЦЕССЕ РЕАЛИЗАЦИИ ИГРОВОЙ ПРОГРАММЫ "BACTERIAL WAR"

В связи с интенсивным развитием сетевых информационных технологий, актуальным является разработка методов и средств распределенного принятия решения, основанных на нескольких интеллектуальных агентах. Проведен анализ существующих имитационных моделей для игровых систем, использующих многоагентный подход принятия решений. Использование мультиагентного подхода для игровых систем началось в далекие 1960-е годы и с каждым годом сложность многоагентных систем в игровых и имитационных системах увеличивалась. Увеличение использования мультиагентных систем способствовало созданию многих технологий и стандартов, которые упрощают разработку мультиагентных систем. Описаны разработанная игровая программа "Bacterial War" и имитационная модель, которая симулирует коммуникацию бактерий в окружающей среде. Разработанная программа использует основные принципы мультиагентных систем, обеспечивает высокую надежность и быстродействие системы. В процессе разработки программы использована гомогенная модель коммуникации агентов, что обеспечило высокий уровень реагирования агентов на внешние факторы. Поиск целей происходит с помощью алгоритма поиска минимального пути A*. Отправка информации между агентами происходит с помощью серверной части, обеспечивает высокое быстродействие программы. Разработанная программа написана с помощью языка программирования Java и вспомогательной библиотеки LibGDX, что обеспечивает использование системы на нескольких платформах.

Ключевые слова: имитационные системы; принятие решений; кооперативность; гомогенные агенты; колония.

V. M. Teslyuk, A. Ya. Lozynskiy, T. V. Teslyuk

Lviv Polytechnic National University, Lviv, Ukraine

APPLICATION OF MULTI-AGENT APPROACH IN THE PROCESS OF IMPLEMENTATION OF THE GAME PROGRAM "BACTERIAL WAR"

Due to the intensive development of networked information technologies, it is important to develop methods and tools for distributed decision-making that are based on several intelligent agents. In the beginning of the article, the author described several simulation models that are based on the multi-agent approach. The described simulation models are used to build stadiums and develop a transport system. The described simulation model shows the actions of the crowd under various circumstances. Furthermore, the author described the first gaming systems that use multi-agent technology and compared them with modern gaming systems. In the main part of the article the author describes the developed game program "Bacterial War" and an imitation model that simulates the communication of bacteria in the environment. The developed program uses the basic principles of multi-agent systems, which ensures high reliability and speed of the system. In the next part of the article, the author describes the algorithm used by system agents to perform the goal. The used algorithm uses the basic idea of finding the minimal path A*. With this algorithm agents seek the purpose and neutralize it. In the process of developing the program, a homogeneous agent communication model was used, which provided a high level of reacting agents to external factors. The transfer of information between agents takes place through the server part, which ensures high performance of the program. This signaling technology provides an improved level of communication of agents, but the reliability of the system depends on the server. If the server fails, the entire system will stop working. At the end of the article, the author describes the technology of developing a gaming system. The developed system is written using Java programming language. In order to ensure the use of the program on many platforms, the author used the LibGDX auxiliary library. The result of the article is the developed game program and simulation model, which visualizes the actions of bacteria in the environment and ways to neutralize bacteria by other bacteria.

Keywords: imitation systems; decision-making; co-operative; homogeneous agents; colony.