

УДК 519.7

В. О. Богаєнко, канд. техн. наук

Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

МУЛЬТИАГЕНТНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ВИТЯЖОК У ПРОМИСЛОВИХ ПРИМІЩЕННЯХ

Розглядається задача мінімізації забрудненості в промислових приміщеннях шляхом оптимального розміщення витяжок. Пропонується генетичний алгоритм проведення такої оптимізації і його адаптація до паралельного виконання на системі автономних програмних агентів.

Ключові слова: *оптимізація розміщення, генетичні алгоритми, паралельні алгоритми, мультиагентні системи.*

Вступ. Розглядається задача мінімізації забрудненості в промислових приміщеннях шляхом оптимального розміщення витяжок, розв'язання якої є важливим при проектуванні та забезпеченні санітарних норм у них.

Задача є оберненою щодо моделювання процесу розповсюдження забруднюючих речовин. Математичні моделі, що описують цей процес, мають нелінійність, пов'язану із залежністю поля швидкостей від координат розміщення витяжок, що ускладнює застосування класичних детермінованих методів розв'язування обернених задач.

Широке застосування в таких випадках мають евристичні методи випадкового пошуку, зокрема генетичні алгоритми, особливістю яких є, з одного боку, недетермінованість та незалежність від типу розв'язуваної задачі, а з іншого — складність підбору параметрів. Їх застосуванню до задач оптимізації розміщення присвячені роботи [1—4], зокрема, у роботах [3; 4] розглядаються задачі оптимізації проектування вентиляційних систем.

Пропонується адаптація статичного генетичного алгоритму до задачі що розглядається, у тому числі для паралельних обчислень на кластерних системах та у мультиагентному середовищі.

Постановка задачі. Пряму задачу моделювання дифузійно-конвективного процесу у промисловому приміщенні опишемо наступним рівнянням [5]:

$$\begin{aligned} \frac{\partial \varphi}{\partial t} + \frac{\partial u \varphi}{\partial x} + \frac{\partial v \varphi}{\partial y} + \frac{\partial (w - w_s) \varphi}{\partial z} + \sigma \varphi = \operatorname{div}(\mu \nabla \varphi) + \\ + \sum_{i=1}^N q_i \delta(\vec{x} - \vec{r}_i) - \sum_{j=1}^M \varphi Q_j \delta(\vec{x} - \vec{r}'_j), \end{aligned} \quad (1)$$

$$V = (u, v, w) = \sum_{j=1}^M \frac{Q_j}{4\pi} \operatorname{grad} \frac{1}{\|\vec{x} - \vec{r}'_j\|}, \quad \vec{x} = (x, y, z),$$

де φ — концентрація забруднюючої речовини; u, v, w — компоненти вектору швидкості V повітряного середовища; μ — коефіцієнт дифузії; σ — коефіцієнт, що враховує зміни концентрації забруднюючої речовини з часом; q_i — інтенсивність викиду точкового джерела забруднення; Q_j — потужність витяжки; \vec{r}_i — координати розміщення джерел викидів; \vec{r}'_i — координати розміщення витяжок; $\vec{x} \in \Omega \subset R^3$, Ω — область розв'язання.

До рівняння (1) ставиться умова відсутності дифузійного потоку на площині $z = 0$: $w|_{z=0} = 0$. Для забезпечення виконання даної умови симетрично відносно площини $z = 0$ розміщуються фіктивні витяжки [5]. Тоді

$$V = \sum_{j=1}^M \frac{Q_j}{4\pi} \left(\operatorname{grad} \frac{1}{\|\vec{x} - \vec{r}'_j\|} + \operatorname{grad} \frac{1}{\|\vec{x} - \vec{\bar{r}}'_j\|} \right),$$

де $\vec{r}'_j = (x_j, y_j, z_j)$, $\vec{\bar{r}}'_j = (x_j, y_j, -z_j)$.

На непроникній границі області розв'язання $z = 0$ ставиться також умова непротікання виду $\left. \frac{\partial \varphi}{\partial z} \right|_{z=0} = 0$, а на інших границях області ставляться умови $\varphi|_{\Gamma \setminus \{z=0\}} = 0$, які моделюють втікання в область незабрудненого повітря.

У випадку нульових початкових умов, рівняння (1) буде описувати ситуацію наявності постійно діючих джерел забруднення, якщо ж початкові умови не нульові, модель буде описувати процес ліквідації наслідків аварійного викиду забруднень.

Зауважимо також, що рівняння (1) можна звести до двовимірною випадку усередненням з координатою z .

Обернена задача щодо цього процесу полягає у знаходженні параметрів $\{M, \{\vec{r}'_j, Q_j\}_{j=1}^M\}$, $\vec{r}'_j \in \Omega_1$, при яких значення цільової функції $f(\varphi) = \int_{\Omega_2} \varphi d\Omega$, $t = \tau$ задовольняє умові $f(\varphi) \leq \varepsilon$ за наявності обмежень

на кількість й потужність витяжок виду $M \leq M_{\max}$, $Q_j \leq Q_{\max}$. Тут $\Omega_1 \subset \Omega$ — допустима область розміщення витяжок, а $\Omega_2 \subset \Omega$ — область, у якій мають забезпечуватись санітарні норми щодо забрудненості.

Генетичний алгоритм. Рівняння (1) має нелінійність, пов'язану з залежністю поля швидкостей від правої частині, що ускладнює застосування класичних детермінованих методів (зокрема, [6]) розв'язання обернених задач щодо нього.

Така особливість прямої задачі викликає необхідність використовувати для її розв'язання евристичні методи, зокрема генетичні алгоритми [7; 8], які базуються на моделюванні процесів еволюції біологічних видів.

Статичний генетичний алгоритм [9] є ітераційним і на кожному кроці оперує набором (популяцією) наближень до розв'язку (хромосом). В процесі виконання алгоритму формується послідовність популяцій, при чому хромосоми кожної наступної популяції задовольняють оптимізаційному критерію не гірше ніж їх попередники. Ітерація складається з наступних кроків:

- кросовер — формування нових можливих розв'язків шляхом комбінування двох певним чином відібраних хромосом;
- мутація — випадкова зміна хромосоми популяції;
- селекція — видалення найгірших за значенням цільової функції хромосом з популяції.

Класичні генетичні алгоритми використовують бінарне кодування розв'язків-хромосом й операції кросоверу та мутації визначають, відповідно, як обмін послідовностями бітів та випадкову зміну бітів хромосоми. Оскільки у розглядуваній задачі параметри оптимізації є неперервними, їх бінарне кодування неефективне. Тому будемо представляти кожну хромосому популяції як набір характеристик однієї чи декількох витяжок, що складаються з координат розміщення та потужності, і представлені числами з плаваючою комою.

Зауважимо, що для обчислення значення цільової функції для задачі оптимізації розміщення у випадку конкретного набору параметрів витяжок, необхідно розв'язати пряму задачі щодо, зокрема, але не обов'язково, вищеописаної моделі.

У якості оператора вибору хромосом для кросоверу була використана операція випадкової вибірки з урахуванням значення цільової функції. Ймовірність вибору описується наступною формулою:

$$P(i) = \frac{p(i)}{\sum_i p(i)}, \quad p(i) = \frac{1}{1 + k_{sel} \frac{v(i) - \min_i v(i)}{\max_i v(i) - \min_i v(i)}}$$

де $v(i)$ — значення цільової функції для хромосоми i , k_{sel} — деякий коефіцієнт.

У якості операції кросовера природно була вибрана операція лінійної комбінації з випадковим параметром для даних, що входять до складу хромосоми.

При виконанні операції мутації проводиться випадкова зміна значень компонент хромосоми. Була використана динамічна ймовірність мутації, що залежить від щільності популяції по значенню цільової функції:

$$P_m = \frac{P_m}{1 + k_m \frac{\max v(i) - \min v(i)}{\min v(i)}}$$

де p_m — базова ймовірність мутації, а k_m — деякий коефіцієнт. У цьому випадку, кількість мутацій зростає при ущільненні популяції, що запобігає збіганню алгоритму до локальних оптимумів, які не є розв'язком задачі.

Після проведення операцій кросоверу й мутації проводиться перевірка обмежень на значення параметрів, що оптимізуються, й у випадку, коли обмеження не задовольняються, операція повторюється.

Початкова популяція генерується або випадковим чином, або у вигляді сітки, яка за можливості рівномірно покриває множину допустимих значень параметрів, що оптимізуються. Вибір початкової популяції у вигляді сітки призводить до кращої збіжності алгоритму оскільки розв'язок задачі знаходиться всередині випуклої оболонки початкової множини розв'язків, тоді як при випадковому початковому розподілі він може там не міститись й бути досяжним лише за допомогою операції мутації.

Вибір критерію завершення розрахунків, разом з вибором найкращого набору параметрів, є проблемною частиною евристичних алгоритмів оптимізації.

Був використаний один з класичних критеріїв, згідно якого, розрахунки завершуються тоді, коли різниця мінімального й максимального значень поточної популяції стає меншою за деяке задане значення. В додачу до цього обмежувалась максимальна кількість кроків алгоритму.

Паралельний генетичний алгоритм. Найбільш обчислювально-затратною частиною евристичних алгоритмів оптимізації є обчислення значень функції, що оптимізується. В зв'язку з цим, алгоритм був модифікований таким чином, щоб на кожному кроці його роботи необхідно було обчислювати декілька значень цільової функції для різних наборів параметрів прямої задачі. Якщо кількість обчислень дорівнює кількості наявних процесорних ядер, кожне з них може виконуватись незалежно на окремому ядрі.

За такої схеми розпаралелення, виділяється керуючий вузол, що містить повну реалізацію генетичного алгоритму, й інші вузли, що працюють як сервери віддалених обчислень.

Масштабованість тут обмежена глобальністю операцій обміну даними (від керуючого вузла всім іншим). Така схема є ефективною

для невеликих мереж чи кластерів, проте не для більших паралельних обчислювальних систем.

Мультиагентний генетичний алгоритм. Мультиагентна програма архітектура максимально ефективна для масово паралельної реалізації на розподілених децентралізованих обчислювальних системах, зокрема в системах добровільних обчислень (*volunteer computing*).

Генетичні алгоритми потребують модифікації для виконання їх на системі автономних інтелектуальних агентів [10; 11]. Один з підходів до такої модифікації полягає у припущенні щодо локальності еволюційних процесів.

Збіжність таких «локальних» алгоритмів для мультиагентних систем залежить від повноти графу зв'язків між агентами. Менша кількість цих зв'язків, відповідно й менша кількість обмінів інформацією, призводить до повільнішої збіжності, проте до більшої масштабованості порівняно з вищеописаним паралельним алгоритмом для кластерних систем.

Пропонується алгоритм, що відображає припущення щодо локальності еволюційних процесів наступним чином:

- кожен агент несе в собі одну хромосому;
- агенти організовані у вигляді довільної мережі або тору (кожен агент має чотирьохагентне «оточення»);
- операція кросоверу виконується лише локально — між зв'язаними агентами. При цьому розмір популяції залишається незмінним, лише змінюється хромосома одного з агентів;
- видалення найгірших за значенням цільової функції хромосом не відбувається для забезпечення незмінності розміру популяції.

Для реалізації такого алгоритму пропонується архітектура, що базується на алгоритмах, описаних у [11], в якій агенти організовані у статичну мережу й взаємодіють один з одним за допомогою наступних запитів:

- запит на приєднання до мережі у відповідь на який створюється канал зв'язку та відсилається список всіх агентів, зв'язаних з тим, до якого надходить запит, що необхідно для збільшення кількості зв'язків у мережі. Мережа агентів може містити сервер, на якому зберігаються ідентифікатори (наприклад, *ip*-адреси та *tcp*-порти) всіх агентів. В такому випадку для приєднання до мережі агенту необхідно знати адресу не якогось іншого агента, а сервера. У відповідь на запит на приєднання, сервер реєструє нового агента й повертає йому адресу певної кількості вже зареєстрованих агентів;
- запит на запуск обчислень, який містить параметри прямої задачі й генетичного алгоритму, та ретранслюється агентом по мережі.

Кожен агент при цьому запам'ятовує вузол, що надіслав запит, у якості вузла вищого рівня. Якщо запит надходить до агента, який знаходиться у стані розв'язання задачі, він повертає нульове значення, в протилежному випадку – одиничне. Кожен агент, що відповідає запит, запам'ятовує агентів, що повернули одиничне значення, як «залежних» від нього. Таким чином утворюється покриваюче дерево, коренем якого є деякий керуючий агент, що ініціює обчислення. В певні моменти часу керуючий агент може повторно розсилати по мережі запит на запуск обчислень, що дозволяє залучати до обчислень новоприєднаних агентів та перебудовувати покриваюче дерево у випадку від'єднання агентів;

- запит на обчислення поточного оптимального значення цільової функції та набору параметрів витяжок. При закінченні кроку обчислень кожен агент чекає на локальне оптимальне значення цільової функції від вузлів, пов'язаних з ним в межах дерева, обраховує локальний оптимум з цих значень та власного значення й відсилає його вузлу на вищому рівні. В результаті керуючий агент, який знаходиться в корені дерева, отримує загальне значення цільової функції й набір параметрів витяжок, при якому воно досягається, й може приймати рішення про продовження чи завершення обчислень;
- запит на зупинку обчислень, який відсилається керуючим агентом й ретранслюється іншими агентами;
- запит на отримання значення цільової функції та набір параметрів витяжок, який агент відсилає своїм «сусідам» при виконанні операції кросоверу.

Обчислювальний експеримент. Розв'язувалась задача оптимізації розміщення витяжок для мінімізації наслідків викиду забруднюючих речовин в приміщенні, яке моделювалось у вигляді кубу. У приміщенні були розміщені 5 точкових джерел забруднень загальною потужністю 11 одиниць. Ставилась задача оптимального розміщення до 3-ох витяжок потужністю від 1 до 4 одиниць таким чином, щоб мінімізувати забрудненість всього приміщення. Обмеження на розміщення витяжок не ставились. Пряма задача розв'язувалась методом скінчених різниць на сітці з 1000 комірок.

Обернена задача за такої постановки не має елементарного розв'язку: незважаючи на те, що максимально сумарна потужність витяжок більша за потужність джерел забруднення, максимальна кількість витяжок менша. За такої ситуації не зрозуміло й чи існує такий розв'язок, за якого все забруднення буде прибиратись з приміщення, адже потужності 3-ох витяжок може не вистачити для прибирання забруднення від 5-ти джерел навіть при сумарній більшій потужності.

Параметри витяжок, за яких $f(\varphi) \leq \varepsilon$, $\varepsilon = 10^{-5}$, були знайдені генетичним алгоритмом за 816 кроків при розмірі популяції, що дорівнює 27-ми, й параметрах алгоритму $p_m = 0.0625$, $k_{sel} = k_m = 5.0$, з максимальною зміною значень параметрів при мутації на 0.1.

Розподілення забруднень в приміщенні за відсутності витяжок при нормалізованому часі $t = 1$ зображено на рис.1, а розраховане оптимальне розташування витяжок — на рис. 2.

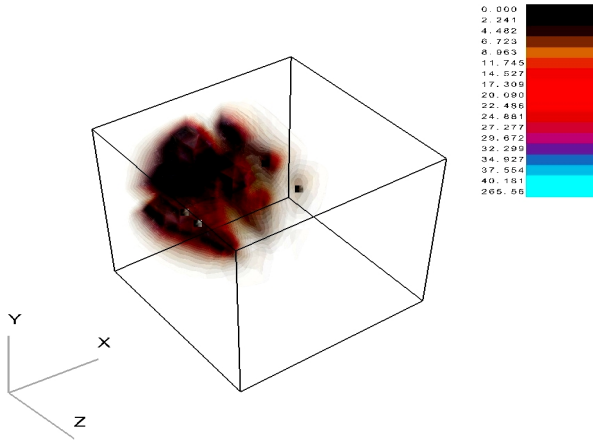


Рис. 1. Розподілення забруднень в приміщенні за відсутності витяжок

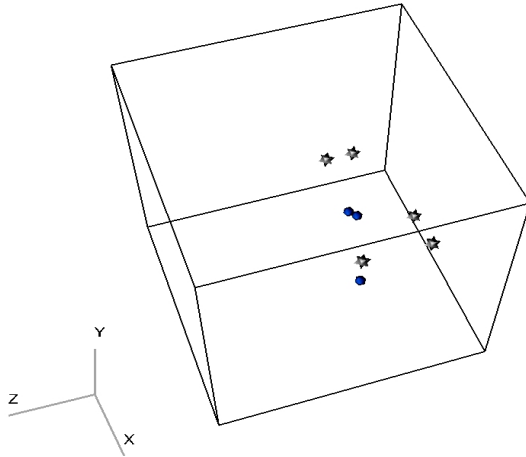


Рис. 2. Оптимальне розташування витяжок (витяжки позначені сферами)

Тестування паралельних алгоритмів. Запропоновані алгоритми були програмно реалізовані та тестувалися на кластері СКІТ-3 Ін-

ституту кібернетики ім. В. М. Глушкова. Результати тестування алгоритму для задачі розмірністю 8000 точок, приміщення з 5-ма джерелами забруднень та кількістю можливих витяжок до 3-х показані у таблиці 1. Популяція при цьому містила 9 хромосом, на кожному кроці проводилась зміна 4-х елементів популяції. Максимальна кількість кроків дорівнювала 300.

Таблиця 1

Результати тестування генетичних алгоритмів

	Генетичний алгоритм		Мультиагентний генетичний алгоритм		
	Випадкова початкова популяція	Початкова популяція у вигляді сітки	Сітковий граф зв'язків		Випадковий граф зв'язків
			Випадкова початкова популяція	Початкова популяція у вигляді сітки	Випадкова початкова популяція
Час роботи послідовної реалізації (хв.)	100	5	116	96	—
Час роботи паралельної реалізації (4 процеси на 1 вузлі) (хв.)	34	—	38	39	—
Час роботи розподіленої реалізації (16 агентів) (хв.)	—	—	—	—	9
Кількість обчислень цільової функції	1218	89	1213	1213	4816

Висновки. Отримані результати показують близьку до лінійної ефективність розпаралелення алгоритмів, при чому найшвидшою з точки зору часу, що витрачається на одне обчислення цільової функції, була мультиагентна розподілена реалізація. Це є наслідком розпаралелення допоміжних операцій, які в інших паралельних реалізаціях виконувались на виділеному процесі-менеджері.

Алгоритмами з сітковим початковим розподілом вдалось отримати кращі розв'язки ніж при використанні випадкового розподілу. Єдиним алгоритмом, при роботі якого критерій збіжності був досягнутий, до того ж досить швидко, був алгоритм саме з сітковим початковим розподілом.

Результати також показали теоретично прогнозоване падіння якості розв'язку при зменшенні кількості зв'язків між агентами (вважаємо, що у звичайному генетичному алгоритмі граф зв'язків повний).

Список використаних джерел:

1. Lijie Cui. Genetic algorithms in probabilistic finite element analysis of geotechnical problems / Cui Lijie, Sheng Daichao // *Computers and Geotechnics*. — 2005. — Vol. 32, Is. 8. — P. 555–563.
2. Well placement optimization using a genetic algorithm with nonlinear constraints. In: *SPE Reservoir Simulation Symposium, SPE 118808* / A. A. Emerick, E. Silva, B. Messer, L. F. Almeida, D. Szwarcman, M.A.C. Pacheco, M.M.B.R. Vellasco. — 2009.
3. Liang Zhou. Optimization of ventilation system design and operation in office environment / Zhou Liang, Haghghat Fariborz // *Methodology, Building and Environment*. — 2009. — Vol. 44, Is. 4. — P. 651–656.
4. Осинцева А. В. Оптимизация регуляторов воздухораспределения в вентиляционной сети подземного рудника / А. В. Осинцева // *Труды XI межрегиональной научно-практической конференции КФ ПетрГУ*. — Апатиты : Изд. КФ ПетрГУ, 2008. — С. 90–91.
5. Численное моделирование распространений загрязнения в окружающей среде / М. З. Згуровский, В. В. Скопецкий, В. К. Хрущ, Н. Н. Беляев. — К. : Наук. думка, 1997. — 368 с.
6. Копець М. М. Задача оптимального керування процесом дифузії / М. М. Копець // *Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки : зб. наук. праць*. — Кам'янець-Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, 2010. — Вип. 3. — С. 94–98.
7. Goldberg D. E. Genetic algorithms in search, optimization and machine learning / D. E. Goldberg. — Reading : MA:Addison-Wesley, 1989.
8. Whitley D. GENITOR II: a distributed genetic algorithm / D. Whitley, T. Starkweather // *J. Expt. Theor. Artif. Intell* 2. — 1990. — P. 189–214.
9. A Multi-Agent Genetic Algorithm for Global Numerical Optimization / ZHONG Weicai, LIU Jing, XUE Mingzhi, JIAO Licheng // *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*. — 2004. — Vol. 34, Is. 2. — P. 1128–1141.
10. Genetic Algorithms in a Multi-Agent System. In *Proceedings IEEE International Joint Symposia on Intelligence and Systems* / Jean-Philippe Vacher, Alain Cardon, Franck Lesage, Thierry Galinho. — Rockville, 1998. — P. 17–26.
11. Attiya H. Distributed Computing: Fundamentals, Simulations, and Advanced Topics. / H. Attiya, J. Welch. // *John Wiley and Sons, Inc.*, 2nd edition. — 2004.

The problem of minimizing pollution in industrial areas by optimal exhauster placement has been considered. Genetic algorithm of such optimization and its adaptation to a parallel implementation on a system of autonomous software agents has been proposed.

Key words: *placement optimization, genetic algorithms, parallel algorithms, multi-agent systems.*

Отримано: 21.03.2013