УДК 519.832

Ю. В. Андрашко (ДВНЗ «Ужгородський нац. ун-т»)

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗУВАННЯ КОНКУРЕНТНОЇ ЗАДАЧІ РОЗМІЩЕННЯ

The competitive placement problems with complete distribution of market where two competitive firms open facilities consecutively for delivering the clients is considered. A hybrid algorithm produced an approximate solution is suggested. Computational results are discussed.

В роботі розглядається конкурентна задача розміщення з повним розподілом ринку, в якій дві фірми послідовно розміщують підприємства для обслуговування клієнтів. Пропонується генетичний алгоритм для знаходження наближеного розв'язку задачі. Наводяться результати чисельних експериментів.

Постановка проблеми. В даній роботі розглядається конкурентна задача розміщення з повним розподілом ринку, на якому присутні дві фірми (Лідер та Конкурент) [5, 7].

Введемо до розгляду змінні:

$$x_i = \begin{cases} 1, & \text{якщо Лідер відкриває підприємство в пункті i;} \\ 0, & \text{якщо Лідер не відкриває підприємство в пункті i;} \end{cases} i \in I;$$

$$y_i = \begin{cases} 1, & \text{якщо Конкурент відкриває підприємство в пункті i;} \\ 0, & \text{якщо Конкурент не відкриває підприємство в пункті i;} \end{cases}, i \in I;$$

$$u_j = \begin{cases} 1, & \text{якщо клієнта обслуговує Лідер;} \\ 0, & \text{якщо клієнта обслуговує Конкурент;} \end{cases} j \in J.$$

Знайти

$$W^1 = \sum_{j \in J} w_j u_j(x, y^*) \to max, \tag{1}$$

(2)

де y^* – розв'язок

$$W^2 = \sum_{j \in J} w_j (1 - u_j(x, y)) \to max. \tag{3}$$

При умовах

$$\sum_{i \in I} f_i x_i \leqslant F^1; \tag{4}$$

$$\sum_{i \in I} f_i y_i \leqslant F^2; \tag{5}$$

$$I_j(x) = \{i \in I | g_{ij} > \max_{l \in I} \{g_{lj} | x_l = 1\}\}, j \in J; \tag{6}$$

$$u_j = 1 - \max_{i \in I_j(x)} y_i; j \in J; \tag{7}$$

$$u \in B^n; x, y \in B^m. \tag{8}$$

Цільова функція задачі визначає сумарний прибуток Лідера (1). Множина допустимих розв'язків описується за допомогою задачі Конкурента (3), (5)-(8). Вектор $x_i, i \in I$ та множини $I_j(x), j \in J$ в задачі Конкурента вважаються фіксованими та відомими.

В роботі [6] наведено алгоритм зведення даної задачі до послідовності однокритеріальних задач булевого програмування. Нагадаємо його.

Набором Конкурента назвемо булевий вектор, що визначає множину пунктів, де Конкурент відкриває підприємства. Допустимим набором назвемо такий набір, що задовольняє умовам (4) та (8). Розглянемо деяку непорожню множину допустимих наборів A. Припустимо, що Конкурент може відкривати підприємства тільки згідно деякого набору із множини A.

Для кожного $y \in A$ побудуємо множину утримання $I_j(y) = \{i \in I | g_{ij} \leqslant \min\limits_{l \in I}\{g_{lj} | x_l = 1\}\}, j \in J$. Ці множини показують пункти розміщення підприємств у яких дозволяють Лідеру зберегти клієнта j, якщо Конкурент відкриє підприємства згідно набору y.

Введемо змінні:

$$v_{jy} = \begin{cases} 1, & \text{якщо клієнта j обслуговує Лідер;} \\ 0, & \text{якщо клієнта j обслуговує Конкурент;} \end{cases} j \in J, y \in A,$$

а Конкурент відкриває підприємства згідно набору y.

Нехай W^1 – сумарний прибуток Лідера.

Розглянемо домоміжну задачу:

Для фіксованої множини A знайти

$$W^1(A) \to max \tag{9}$$

при умовах:

$$\sum_{j \in J} w_j v_{jy} \geqslant W^1(A), y \in A; \tag{10}$$

$$\sum_{i \in I_j(y)} x_i \geqslant v_{jy}, j \in J, y \in A; \tag{11}$$

$$\sum_{i \in I} f_i x_i \leqslant F^1; \tag{12}$$

$$x \in B^m. \tag{13}$$

Зауважимо, що заміна булевих змінних v_{jy} на неперервні $0 \leqslant v_{jy} \leqslant 1$ породжує еквівалентну задачу [6]. Таким чином, в задачі залишиться тільки одна група булевих змінних x_i. Інші змінні можна вважати неперервними.

Розв'язок задачі Лідера для будь-якої фіксованої множини A є верхньою оцінкою оптимального значення цільової функції $\overline{W^1}(A)$ вихідної задачі (1)-(8).

Нехай A – деяка множина допустимих наборів Конкурента. $x(A)$ – оптимальний розв'язок задачі (9)-(13), при $y \in A$. Тоді оптимальний розв'язок задачі Конкурента (3), (5)-(8) визначає нижню оцінку значення цільової функції $\underline{W^1}(A)$ вихідної задачі (1)-(8).

Очевидно, якщо множина A містить всі допустимі набори y, то задачі (9)-(13) та (1)-(8) еквівалентні. Очевидно, що багато з елементів такої множини породжують несуттєві обмеження (10) та (11). Побудуємо еквівалентну задачу, знайшовши таку множину A, елементи якої породжують тільки суттєві обмеження. Одним із способів знаходження такої підмножини є ітеративний процес, що наведено далі.

Алгоритм 1. Розв'язування конкурентної задачі розміщення з повним розподілом ринку:

1) Вибрати початкову множину A_0.

2) Розв'язати задачу (9)-(13), знайти $\overline{W^1}(A_k)$ та $x(A_k)$.

3) Розв'язати задачу (3), (5)-(8), знайти $\underline{W^1}(A_k)$ та $y(A_k)$.

4) Якщо $\overline{W^1}(A_k) = \underline{W^1}(A_k)$, то $x(A_k), y(A_k)$ – розв'язок задачі (1)-(8).

5) Інакше $A_{k+1} = A_k \cup y(A_k)$, $k = k + 1$. Повернутись до кроку 2.

Теорема 1. *Алгоритм 1 скінченний і завершується знаходженням розв'язку задачі (1)-(8).*

Доведення теореми наведене в роботі [6].

На кожній ітерації необхідно знаходити $W^1(A)$, для даної множини A. Ця задача є NP-важкою. На сьогодні ефективні точні методи знаходження розв'язку даної задачі невідомі, тому доцільно застосовувати наближені евристичні алгоритми. Перші кроки в цьому напрямку, навівши деякі схеми часткового перебору, зробив Резенде [3]. Генетичні алгоритми показали свою ефективність при вирішенні багатьох практичних задач дискретної оптимізації. Варто спробувати застосувати їх до даної задачі.

Генетичний алгоритм. Алгоритм розв'язання задач оптимізації, заснований на ідеях спадковості в біологічних популяціях, вперше був запропонований Дж.Холландом в 1975р [2]. Подальший розвиток ця ідея, як і свою назву – генетичний алгоритм, отримала в роботах Гольдберга і Де Йонга [1].

Алгоритм 2. Загальна схема генетичного алгоритму:

1) Побудувати початкову популяцію Π_0.

2) Оператором селекції з популяції вибрати елементи $s_i^1, s_i^2 \in \Pi_i$.

3) Оператором схрещування побудувати новий елемент s_i на основі s_i^1 та s_i^2.

4) Оператором мутації модифікувати s_i, отримати s_i'.

5) Якщо s_i' кращий, ніж найгірший елемент в популяції $s_i^{min} \in \Pi_i$, то оновити популяцію: $\Pi_{i+1} = (\Pi_i \bigcup s_i') \setminus s_i^{min}$.

6) Якщо не виконується критерій зупинки, то перейти до виконання наступної ітерації алгоритму: покласти $i = i + 1$ та повернутись до пункту 2.

7) Вибрати найкращий елемент популяції як розв'язок.

Побудова початкової популяції. Генетичний алгоритм починає роботу з формування початкової популяції – набору допустимих розв'язків задачі. Скористаємось наступним алгоритмом вибору початкових елементів.

Алгоритм 3. Побудова початкової популяції:

1) Знайдемо x^* – розв'язок задачі (14), (15), що полягає в максимізації сумарної привабливості підприємств, відкритих Лідером, коли на ринку відсутній Конкурент:

$$\sum_{j\in J}\max_{i\in I} g_{ij}x_i \to max; \tag{14}$$

$$\sum_{i\in I} f_i x_i \leqslant B^1. \tag{15}$$

2) Застосуємо оператор мутації до x^*.

3) Застосуємо процедуру локального пошуку [8], використовуючи вектор, отриманий на попередньому кроці, як початковий.

4) Знайдений локальний оптимум додамо в популяцію.

5) Якщо не сформовано популяцію потрібного обсягу то повертаємось до 2 кроку.

Оператор мутації. Розглянемо множину $S^1 = \{i | x_i^* = 1, i \in I\}$. Випадковим чином виберемо пару елементів (i_0, i_1) таких, що $i_0 \notin S^1, i_1 \in S^1$. Значення відповідних координат змінимо на протилежні: $x_{i_0} = 1 - x_{i_0}$, $x_{i_1} = 1 - x_{i_1}$.

Оператор селекції будується таким чином, щоб з ненульовою імовірністю будь-який елемент популяції міг бути обраний, як один із батьків. Розглянемо деякі оператори селекції.

При пропорційній селекції два елементи вибираються випадковим чином. Імовірність вибрати елемент s_i рівна $p_i = \frac{W^1(s_i)}{\sum\limits_{s\in\Pi_k} W^1(s)}$.

При турнірній селекції випадковим чином формуються дві підмножини з елементів популяції ($R_k^1 \subset \Pi_k$, $R_k^2 \subset \Pi_k$) і серед елементів кожної із підмножин вибирається один елемент, якому відповідає найбільше значення цільової функції $s^1 = \arg\max\limits_{s\in R_k^1} W^0(s)$ та $s^2 = \arg\max\limits_{s\in R_k^2} W^0(s)$.

Розглянемо оператор селекції "кращий і випадковий". Завжди вибирається кращий елемент популяції $s^1 = \arg\max\limits_{s\in\Pi_k} W^0(s)$. Другий елемент вибирається з решти випадковим чином з рівною імовірністю $p_i = \frac{1}{||\Pi_k||-1}$.

Оператор схрещування застосовується до обраних елементів.

Однорідний оператор схрещуваня будує нащадка s на основі елементів s^1 та s^2 присвоюючи кожній його координаті значення відповідної координати одного з батьків з певною імовірністю (зазвичай рівною $1/2$). Якщо значення первої координати векторів s^1 та s^2 збігаються, то вектор s успадкує це значення.

Жадібний оператор передбачає більш осмислений вибір координати, значення якої покладається рівним одиниці. Як і при однорідному оператору схрещування, якщо значення первої координати векторів s^1 та s^2 збігаються, то

вектор s успадкує це значення. Серед решти вибирається та координата, зміна якої призведе до максимального збільшення цільової функції. По суті, даний оператор знаходить розв'язок вихідної задачі за допомогою жадібного алгоритму на множині координат, що не співпадають у батьків.

Оператор зв'язуючих шляхів будує вектор s наступним чином: будується послідовність допустимих булевих векторів $t^1, t^2, \ldots, t^k$ таких, що $t^1 = s^1; t^k = s^2$, а кожен елемент послідовності задовільняє обмеження (4). Серед елементів цієї послідовності s вибирається таким чином, щоб йому відповідало більше значення цільової функції, ніж сусіднім елементам послідовності. Якщо такий вибір неможливий, береться деякий розв'язок із середини послідовності. Щоб розв'язок успадкував спільні компоненти батьків, слід розглядати тільки найкоротші послідовності. Однак, таких послідовностей експоненціально багато. Серед найкоротших послідовностей прийнято вибирати ту, в якій перехід до сусіднього елементу супроводжується найбільшим збільшенням (або найменшим зменшенням) значення цільової функції, а відстань Хеммінга між сусідніми елементами мінімальна.

Оператор оптимального схрещування полягає в тому, що s знаходиться як оптимальний розв'язок вихідної задачі на грані гіперкуба, якій належать s^1 та s^2. Очевидно, якщо відстань $\rho(s^1, s^2) = n$, то задача оптимального схрещування збігається з вихідною задачею.

До отриманого вектора s застосовується оператор мутації, який змінює значення кількох координат на протилежне.

Результати. Запропонований алгоритм тестувався на прикладах з бібліотеки тестових завдань "Дискретні моделі розміщення" [7]. Розмірність задач $20 \leqslant n = m \leqslant 50$.

Таблиця 1.

Порівняння знайдених розв'язків із відомими оцінками значення цільової функції Лідера

код	нижня оцінка	знайдений результат	верхня оцінка
111	41	50	74
211	41	49	70
311	46	47	73
411	41	48	72
511	48	48	70
611	42	47	67
711	49	51	73
811	42	48	74
911	47	49	68
1011	46	49	70

Верхня і нижня оцінки взяті із робіт Алексеєвої [4]. Результати показують, що алгоритм знаходить розв'язки, значення цільової функції яких близькі до нижньої оцінки. Отже, алгоритм дозволяє знайти досить хороші розв'язки.

Також тестування здійснювалось на власноруч згенерованих прикладах розмірності $n = m = 100$. Для генерування матриці G на одиничному квадра-

ті випадковим чином вибиралися n точок розміщення клієнтів та m точок розміщення пунктів. Величина g_{ij} покладалася рівною величині, оберненій до Евклідової відстані між відповідними точками. Генерувались випадкові величини $1 \leqslant w_j \leqslant 10$, а $f_j = 1, j \in J$ та $F^1 = F^2 = 10$.

Дослідимо залежність знайденого розв'язку від кількості поколінь популяції для різних комбінацій операторів генетичного алгоритму. Для оцінки якості елемента використаємо значення прибутку Лідера. Знайдемо відносне відхилення значення цільової функції між поточним найкращим елементом популяції та найкращим елементом, знайденим при розв'язуванні задачі всіма способами. Далі візьмемо середнє відхилення для задач 111, 211, ..., 1011 із бібліотеки тестових завдань та десяти власноруч згенерованих задач. Результати наведемо на графіку (рис. 1).

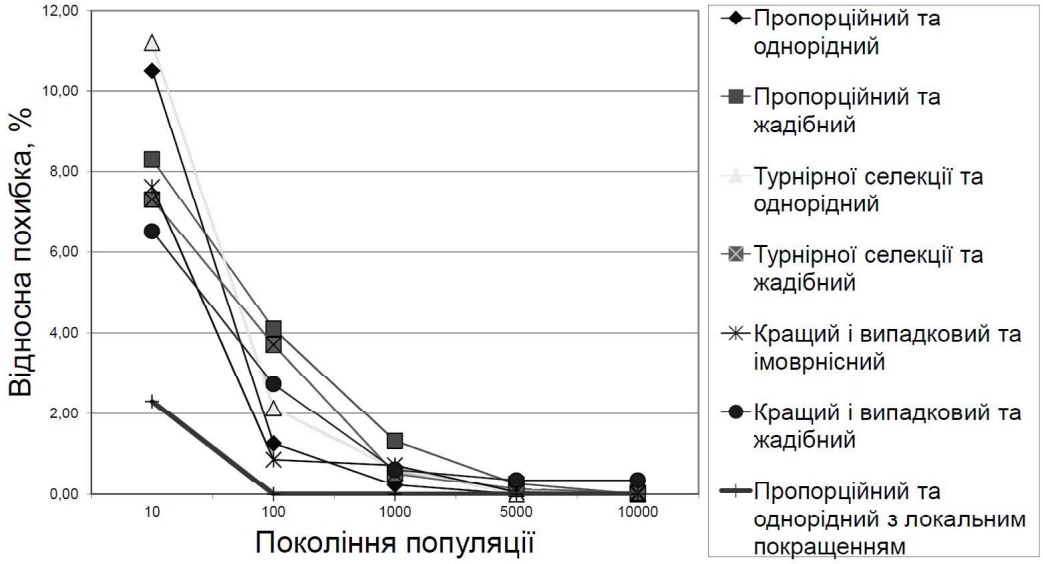

Рис. 1. Порівняння розв'язків конкурентної задачі розміщення знайдених генетичним алгоритмом з різними оперператорами селекції та схрещування.

Результати числових експериментів показали, що найкраще для розв'язування даної задачі підходить генетичний алгоритм із пропорційним оператором селекції та однорідним оператором схрещування. Також близькі результати дають решта варіантів генетичного алгоритму. Тільки генетичний алгоритм із жадібним оператором схрещування та оператором селекції "Кращий та випадковий"не можуть знайти оптимальний розв'язок для деяких задач навіть за 10000 поколінь популяції.

Також розглядався гібридний алгоритм, коли після застосування оператора мутації додано додатковий етап оптимізації за допомогою локального пошуку із заборонами. Хоч це і потребує обчислювальних зусиль, але позитивно впливає на загальний час знаходження розв'язку.

Результати показують значну залежність всіх модифікацій генетичного алгоритму від початкової популяції та їх незначну ефективність без додаткової оптимізації розв'язку, отриманого в результаті схрещування.

1. *Goldberg D.E.* Simple genetic algorithms and the minimal deceptive problem. Genetic Algorithms and Simulated Annealing. Chapter 6. – Los Altos, CA, Morgan Kauffman. – 1987. – P. 74–88.
2. *Holland J. H.* Adaptation in Natural and Artificial Systems. – Ann Arbor: University of Michigan Press, 1975.
3. *Resende M.G.C., Werneck P.F.* A hybrid heuristic for the p-median problem // Journal of Heuristics. – 2004. – V. 10. – P. 59–88.
4. *Алексеева Е.В., Кочетова Н.А.* Верхние и нижние оценки конкурентной задачи о р-мередиане // Методы оптимизации и их приложения. Труды XIV Байкальской международной школы-семинара. – Иркутск, 2008. – С. 563–569.
5. *Андрашко Ю.В., Кузка О.І.* Про деякі конкурентні задачі розміщення // Науковий вісник Ужгородського університету. Серія матем. і інформ./ Редкол.: В.В. Маринець та інші. – Ужгород: Видавництво УжНУ „Говерла“, 2013. – Вип. 24, №1. – С. 5-11.
6. *Андрашко Ю.В.* Зведення конкурентної задачі розміщення до послідовності однокритеріальних задач// Науковий вісник Ужгородського університету. Серія матем. і інформ./ Редкол.: В.В. Маринець та інші. – Ужгород: Видавництво УжНУ „Говерла“, 2013. – Вип. 24, №2. – С. 5-11.
7. *Кочетов Ю.А.* Методы локального поиска для дискретных задач размещения. – Новосибирск: НГТУ, 2009. – 267 с.
8. *Сергиенко И.В., Каспшицкая М. Ф.* Модели и методы решения на ЭВМ комбинаторных задач оптимизации. – К.: Наукова думка, 1981. – 288 с.

Одержано 18.11.2018