УДК 519.161

К.т.н., доцент Зорін Ю.М., студент Санжаревський Я.Ю.

Національний технічний університету України
«Київський політехнічний інститут»

ГЕНЕТИЧНИЙ АЛГОРИТМ РОЗФАРБУВАННЯ ГРАФА У МІНІМАЛЬНУ КІЛЬКІСТЬ КОЛЬОРІВ

Abstract

Yuri Zorin, Assoc. Prof., PhD; Jan Sanzharevsky, student
Genetic algorithm for graph minimum coloring problem

The paper presents graph minimum coloring problem solution by means of Genetic algorithm. Proposed approach combines canonic Genetic algorithm with Tabu search metaheuristic for the improvement of solution quality. There were proposed genome coding and genetic operators implementation.

Вступ

Задача розфарбування графа у мінімальну кількість кольорів є комбінаторною задачею, яку відносять до класу NP-складних задач. Загалом, ця задача може бути сформульована як рекурентна задача *k*-розфарбування графа. Тобто фіксується якась початкова (обґрунтована з точки зору можливості розфарбування) кількість кольорів, а потім за деяким алгоритмом виконується розв’язання задачі *k*-розфабрування графа з подальшим зменшенням кількості кольорів.

Нехай є граф

$$G = (V, B) \qquad (1)$$

де V – вершини, B – ребра, причому

$$V = \{v_i \mid \forall i \in \aleph \Rightarrow v_i \in [\aleph \cup \{0\}]\},$$

$$B = \{< v_i, v_j > \mid [(i, j \in \aleph) \wedge (i \neq j)] \Rightarrow v_i, v_j \in V\} \qquad (2)$$

З метою розфарбування графа доповнимо його новою характеристикою

$$G' : G' = (G, C) \equiv (V, B, C) \qquad (3)$$

де C – *множина кольорів*. Задамо функцію розфарбування вершин графа:

$$F_c : \aleph \cup \{0\} \to \aleph \cup \{0\}, F_c(v \in V) = c \in C \qquad (4)$$

Виходячи з того, що кількість кольорів є мінімальною, що тотожно розфабуванню в $k = \chi(G)$ кольорів, де $\chi(G)$ є хроматичним числом графа, введемо обмеження на цільову функцію:

$$card(im(F_c)) = \chi(G) \qquad (5)$$

Тобто будь-яке відображення з точки зору оператора, що задовольняє операторній умові (5) буде точним (тобто в k і лише в k кольорів) розв'язком поставленої задачі.

Постановка задачі

Метою роботи є розробка генетичного алгоритму (ГА) розфарбування графа у мінімум кольорів. Двома загальними проблемами є швидкість знаходження розв'язку та якість отриманого розв'язку. Під якістю розв'язку будемо розуміти наступне відображення:

$$\psi : [\aleph \cup \{0\}]^2 \to [\aleph \cup \{0\}],$$
$$\psi = d_{\Re}(\chi(G), k) \ , \qquad (6)$$

де метрика, використана в (6) є тривіальною метрикою на $\Re$.

Сформулюємо задачу у термінах теорії оптимізації:

$$\psi \to \min,$$
$$opt(\psi) = 0.$$

Враховуючи ці умови, потрібно знайти таке подання геному та спосіб виконання генетичних операторів, щоб результуючий ГА був ефективним.

Кодування геному й обчислення пристосованості

Перше завдання полягає в тому, щоб знайти таке подання, яке дозволяє зменшити час виконання кроссоверу, що обраний для розв'язування поставленої задачі.

Для подання графа скористаємося наступною функцією:

$$MT : \{(G, C)\} \to M_{KxV}(\{0,1,x\}) \qquad (7)$$

тобто кожному розфарбованому графа (3) ставиться у відповідність прямокутна матриця з кількістю рядків, що дорівнює кількості кольорів для розфарбування та кількістю стовпців, що дорівнює кількості вершин графа .

Поряд з (7) будемо розглядати наступну функцію:

$$MO : M_{KxV}(\{0,1,x\}) \to M_{KxV}(\{0,1,x\}) \qquad (8)$$

Множина {0,1,x} є образом наступного відображення:

$$\varphi : K \times V \to \{0,1,x\}, \qquad (9)$$

яке ставить у відповідність індексу двовимірної матриці значення, що формується наступним чином.

Для зручності викладу введемо наступну логічну умову, використовуючи функцію (4) та множину (2):

$$A = \forall c \in K, i \in V \wedge F_c(i) = c \Rightarrow \{v \in V \setminus \{i\} \mid F_c(v) = c \wedge < c, i > \in B\} = \varnothing$$

$$\forall c \in K, i \in V \Rightarrow \varphi(c,i) = \begin{cases} 1, F_c(i) = c \wedge A \\ 0, F_c(i) = c \wedge \overline{A} \\ x, F_c(i) \neq c \end{cases}$$

Оператор *MO* виконує склеювання деяких наборів рядків. Тобто, для того, щоб визначити відображення (8), достатньо визначити компонентне відображення:

$$MO^* : [\{0,1,x\}^{card(V)}]^2 \to \{0,1,x\}^{card(V)}$$

що зводиться до асоціативної комутативної композиції відображень:

$$MO' : \{0,1,x\}^2 \to \{0,1,x\} \qquad (10)$$

Зважуючи на скінченність прообразу відображення (10) і враховуючи також комутативність, задамо його повністю:

$$MO'(a,b) = \begin{cases} 0, a = 0, b \in \{0,x\} \\ 1, a = 1, b \in \{1,x\} \\ x, a = x, b = x \end{cases}$$

причому, помітно, що функція не визначена на наборі (0,1), що свідчить про неможливість склеювання рядків у випадку, коли хоча б двом відповідним вершинам поставлені у відповідність елементи 0 та 1 одночасно.

Відображення (8) допускає зменшення кількості кольорів при відсутності конфліктів. Отже, саме це відображення і є геномом. Причому, первинна таблиця, що задає розфарбування у *card(V)* кольорів, буде незмінною. Доцільно ввести наступну пропозицію.

<u>Пропозиція 1</u>

Усі можливі склеювання рядків можуть бути подані як послідовність номерів цих рядків.

Генетичні оператори

У запропонованому ГА використовується метаевристика Tabu Search (TS) [2]. Множиною сусідів вважається окіл особини, що включає в себе всі комбінації (8), що отримують із заданої за одну транспозицію, за винятком тих, що потрапили в Tabu list.

У роботі застосовано Order-based кроссовер [1], що забезпечує істотне змішування генів батьків. Він генерує деякий вектор цілих попарно різних чисел (вектор *a*). Далі процедура генерування нащадка однакова для обох батьків з точністю до позначень.

1) У одного з батьків за координатами з вектора *a* вибираються значення у вектор *b*.

2) У іншого з батьків за координатами з вектора *b* вибираються значення у вектор *c*.

3) До нащадка потрапляють усі гени першого батька, крім визначених у векторі *c,* що лежать за відповідними координатами у другому батькові.

Мутація виконує єдину довільну транспозицію, що також сприяє процесу змішування.

Висновки

Результати тестування ГА на складних графах (500-1000 вершин) з набору тестів DIMACS [4] показали, що використання TS та кодування (8) істотно поліпшують результати розв’язання задачі за допомогою канонічної моделі ГА. При тестуванні використовувались три схеми селекції: стохастична селекція, селекція за допомогою колеса рулетки, турнірна селекція [3]. Також визначено емпірично кращі параметри ГА: розмір популяції – 80-100, вірогідність мутації – 20-25%, кроссоверу – 80-85% .

Література

1. *Raphael D., Jin-Kao H.* Parallel Problem Solving from Nature. – Springer Berlin / Heidelberg. – 1998. – 845p.
2. *Glover F.* Tabu Search and Uses. – University of Colorado. – 1995. – 84p.
3. *Mitchell M.* An Introduction to Genetic Algorithms. – MIT Press. – 1999. – 158 p.
4. DIMACS Graphs: BenchmarkInstances : http://info.univ-angers/pub/

porumbel/graphs/index.html