

КВАНТОВИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ ВИЩИХ ПОРЯДКІВ ДЛЯ 0–1 ЗАДАЧІ ПАКУВАННЯ РЮКЗАКА

В.М. ТКАЧУК, О.М. ТКАЧУК

Анотація. Для підвищення ефективності роботи квантового генетичного алгоритму (QGA) запропоновано в поданні квантової хромосоми перейти до квантових реєстрів вищих порядків. Таке подання дозволяє використати такий потужний механізм квантових обчислень, як заплутаність квантових станів. Для реалізації алгоритму використано адаптивний оператор квантового гейту та запропоновано технологію відновлення квантової хромосоми для розв'язання комбінаторних задач з обмеженнями. Досліджено вплив розміру квантового реєстра на ефективність роботи алгоритму. Переваги запропонованого підходу порівняно із традиційною реалізацією QGA проілюстровано на прикладі 0–1 задачі пакування рюкзака великої розмірності та різного рівня кореляції вхідних даних.

Ключові слова: квантовий генетичний алгоритм, 0–1 задача пакування рюкзака, оператор квантового гейту, кубіт, квантовий реєстр, заплутаність квантових станів.

ВСТУП

Класичний генетичний алгоритм є стохастичним алгоритмом оптимізації, робота якого ґрунтується на принципах еволюції біологічних систем [1]. Для пошуку оптимального значення використовується випадковим чином згенерована множина можливих розв'язків – популяція, за допомогою якої досліджується область пошуку. Алгоритм використовує кращі з них для їх відтворення та поступового ітераційного наближення до оптимуму в ході еволюції популяції в часі [2].

Квантовий генетичний алгоритм (QGA) є відносно новим еволюційним алгоритмом, який ґрунтується на поєднанні принципів квантових обчислень, таких як квантові біти, квантові гейти, суперпозиція станів і технології класичних генетичних алгоритмів [3,4]. Він проілюстрував свою ефективність для широкого класу задач оптимізації та пошуку, які потребують наближених, по можливості максимально близьких до оптимуму, розв'язків [4–8]. Порівняно з класичним підходом QGA забезпечує кращий баланс між глобальним пошуком і технологією локальної збіжності [4, 5].

Алгоритм QGA працює також із множиною потенційних розв'язків, але ймовірнісний механізм роботи квантових генетичних операторів забезпечує глобальний пошук за швидкої локальної збіжності та невеликого розміру популяції. Замість бінарного подання популяції QGA використовує її квантово-механічне, імовірнісне подання, що також забезпечує збереження різноманітності популяції в ході еволюції.

Згідно із працею [3] мінімальною одиницею інформації у квантових обчисленнях є кубіт $|q\rangle$ — квантова система, що може перебувати в двох основних станах: $|0\rangle$ та $|1\rangle$. Відповідно до принципу суперпозиції він може перебувати у довільному стані, що є лінійною комбінацією базових:

$$|q\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle. \quad (1)$$

Інформація, що міститься в амплітудах α_1 та α_2 , власне є квантовою частиною інформації. Фактично α_1^2 та α_2^2 — імовірності знаходження кубіта у станах $|0\rangle$ та $|1\rangle$ відповідно, тому вони повинні задовольняти умову нормування:

$$\alpha_1^2 + \alpha_2^2 = 1.$$

Якщо скористатися матричним поданням квантових станів

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

то стан кубіта (1) можна задати як

$$|q\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}.$$

Для реалізації QGA [4, 5] зазвичай заплутаність квантових станів не використовується, тому хромосома являє собою впорядкований набір незалежних кубітів. Так, якщо довжина $N = 16$ кубітів, її можна подати у вигляді

$$|q_1\rangle |q_2\rangle |q_3\rangle |q_4\rangle |q_5\rangle |q_6\rangle |q_7\rangle |q_8\rangle |q_9\rangle |q_{10}\rangle |q_{11}\rangle |q_{12}\rangle |q_{13}\rangle |q_{14}\rangle |q_{15}\rangle |q_{16}\rangle$$

Для задачі 0–1 пакування рюкзака довжина хромосоми N визначається загальною кількістю предметів, які можна помістити в нього.

ЗАСТОСУВАННЯ QGA ДО ЗАДАЧ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ

Загальна схема роботи QGA аналогічна класичному генетичному алгоритму. Принципова відмінність виявляється у способі подання потенційних розв'язків, використовуваних квантових генетичних операторах, особливостях їх реалізації з урахуванням подання хромосоми як системи кубітів, принципу суперпозиції та заплутаності квантових станів.

Алгоритм QGA можна реалізувати у такому вигляді:

Квантовий генетичний алгоритм

- 1 $t \leftarrow 0$
- 2 ініціалізація $Q(t)$
- 3 квантове вимірвання $Q(t)$ та перехід до $P(t)$
- 4 оцінка пристосованості $P(t)$

```

5   відновлення  $P(t)$ 
6   відновлення  $Q(t)$ 
7   знаходження  $b$  – найкращого розв'язку в  $P(t)$ 
8   while (умова завершення еволюції)
9        $t \leftarrow t+1$ 
10  оновлення  $Q(t)$  за допомогою квантового гейту
11  квантове вимірювання  $Q(t)$  та перехід до  $P(t)$ 
12  оцінка пристосованості  $P(t)$ 
13  відновлення  $P(t)$ 
14  відновлення  $Q(t)$ 
15  знаходження  $b$  – найкращого розв'язку в  $P(t)$ 
16  end while

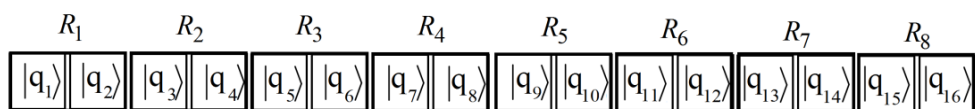
```

Тут $Q(t)$ — квантова популяція розв'язків на момент часу t ; $P(t)$ — класичне (бінарне) подання популяції, отримане в результаті квантового вимірювання.

МЕТА ДОСЛІДЖЕННЯ

Мета дослідження — включення в роботу QGA такого потужного механізму квантових обчислень, яким є заплутаність квантових станів. Можливості реалізації цього алгоритму частково проаналізовано у праці [6], де проілюстровано ефективність такого підходу до задач комбінаторної оптимізації. Загальну ідею його реалізації в задачах функціональної оптимізації детально розглянуто у праці [7].

Принципова відмінність у реалізації QGA вищих порядків полягає у потребі під час реалізації квантових операторів урахувати заплутаність станів квантових регістрів. Якщо кожен два кубіти хромосоми попарно перебувають у заплутаному стані, то її можна подати у такому вигляді:



Тут R_i — квантовий регістр, що складається із двох кубітів ($r = 2$), які перебувають у заплутаному стані. Кількість можливих станів такого регістра

$$n = 2^r = 2^2 = 4.$$

У результаті квантового вимірювання він може перейти в один із чотирьох можливих основних станів:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle.$$

У загальному випадку така система перебуває в стані, що є лінійною комбінацією базових:

$$|q\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle.$$

Тут $|\alpha_1^2\rangle$, $|\alpha_2^2\rangle$, $|\alpha_3^2\rangle$ та $|\alpha_4^2\rangle$ — імовірності перебування квантового регістра у відповідному стані. У разі переходу до $r=2$ розмір матриці M , потрібний для подання квантової хромосоми, залишається незмінним. Так, для подання одного кубіта засобами класичного комп'ютера потрібні два елементи матриці, тому

$$M = 2 \cdot N = \frac{N}{2} \cdot 2^r = \frac{N}{2} \cdot 4 = 32.$$

У разі переходу до квантових регістрів вищих порядків, тобто, якщо $r > 2$,

$$M = \frac{N}{r} \cdot 2^r > 2 \cdot N.$$

Це означає, що в цьому випадку для подання квантової хромосоми необхідно додатково збільшити розмір матриці, потрібний для подання як однієї особини, так і популяції в цілому. Так, якщо $r = 4$, квантова хромосома складатиметься із чотирьох регістрів (R_1, R_2, R_3, R_4):

R_1				R_2				R_3				R_4			
$ \alpha_1\rangle$	$ \alpha_2\rangle$	$ \alpha_3\rangle$	$ \alpha_4\rangle$	$ \alpha_5\rangle$	$ \alpha_6\rangle$	$ \alpha_7\rangle$	$ \alpha_8\rangle$	$ \alpha_9\rangle$	$ \alpha_{10}\rangle$	$ \alpha_{11}\rangle$	$ \alpha_{12}\rangle$	$ \alpha_{13}\rangle$	$ \alpha_{14}\rangle$	$ \alpha_{15}\rangle$	$ \alpha_{16}\rangle$

Кількість основних станів регістра дорівнює $2^4 = 16$:

$$|0000\rangle, |0001\rangle, |0010\rangle, \dots, |1111\rangle.$$

Згідно з принципом суперпозиції (1) стан одного регістра можна подати як

$$|q\rangle = \alpha_1|0000\rangle + \alpha_2|0001\rangle + \alpha_3|0010\rangle + \dots + \alpha_{16}|1111\rangle.$$

Розмір матриці, потрібний для подання однієї такої квантової хромосоми,

$$M = \frac{N}{r} \cdot 2^r = \frac{16}{4} \cdot 2^4 = 64 > 2 \cdot N = 32.$$

Для реалізації QGA вищого порядку для подання особини популяції зручно використати структуру, що складається з $k = N/r$ квантових регістрів відповідно до такої схеми:

R_1	R_2	R_3	...	R_i	...	R_k
α_1^1	α_1^2	α_1^3	...	α_1^i	...	α_1^k
α_2^1	α_2^2	α_2^3	...	α_2^i	...	α_2^k
\vdots	\vdots	\vdots	...	\vdots	...	\vdots
$\alpha_{2^r}^1$	$\alpha_{2^r}^2$	$\alpha_{2^r}^3$...	$\alpha_{2^r}^i$...	$\alpha_{2^r}^k$

Тут $\{\alpha_1^i, \alpha_2^i, \alpha_3^i, \dots, \alpha_{2^r}^i\}$ визначає стан одного квантового регістра R_i , а множина $\{R_1, R_2, R_3, \dots, R_k\}$ формує одну особину популяції.

ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВОЇ КВАНТОВОЇ ПОПУЛЯЦІЇ

Початковий стан як кубіта, так і квантового регістра в цілому не містить інформації про задачу та алгоритм її розв’язання, тому найпростішим способом його ініціалізації є рівність усіх амплітуд між собою [3,4]. Це означає, що на початку еволюції кожен квантовий регістр розміру r перебуває в стані:

$$|q\rangle = \frac{1}{\sqrt{2^r}}|00\dots00\rangle + \frac{1}{\sqrt{2^r}}|00\dots01\rangle + \dots + \frac{1}{\sqrt{2^r}}|11\dots10\rangle + \frac{1}{\sqrt{2^r}}|11\dots11\rangle.$$

УМОВА ЗАВЕРШЕННЯ ЕВОЛЮЦІЇ

Як умовою завершення еволюційного процесу можна скористатися одним із таких критеріїв:

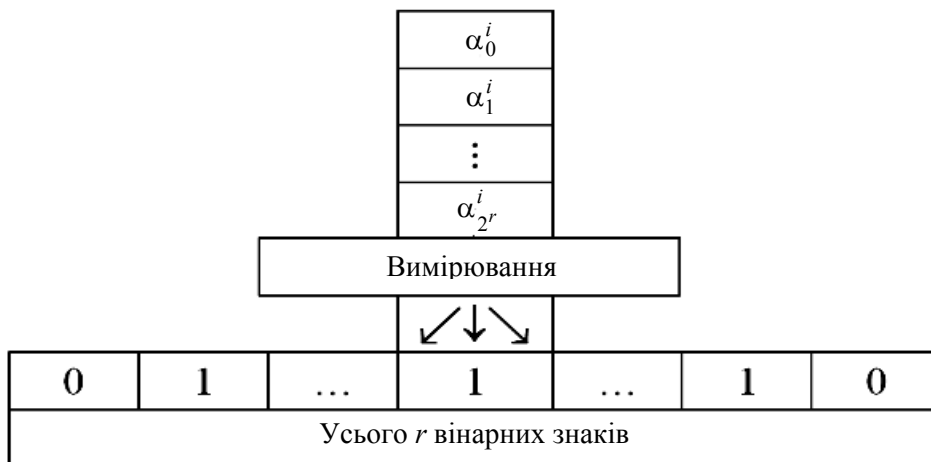
- кількість звернень до функції пристосованості. Це 5000 звернень до функції пристосованості [4–7]. Ця умова використовується для порівняння ефективності різного типу еволюційних алгоритмів;
- загальний час роботи. Дозволяє порівняти ефективність однотипних алгоритмів, що відрізняються тими чи іншими особливостями реалізації;
- усереднене значення ймовірностей квантових станів популяції [9].

Для подання хромосоми з використанням квантових регістрів критерій можна задати у такий спосіб:

$$c_{avr}^i = \frac{1}{k} \sum_{i=1}^k |1 - 2 \cdot \alpha_i^b|^2 ; \quad C_{avr} = \frac{1}{s} \sum_{i=1}^s c_{avr}^i > \gamma,$$

де стан b визначається класичним значенням відповідного квантового регістра, отриманого в результаті квантового вимірювання; s — потужність популяції. Так, якщо $\gamma = 0.99$, у середньому 99% імовірностей квантових станів майже збігаються до значення одного з основних станів регістра.

Операція вимірювання (етапи 3 та 11 роботи QGA) стану квантового регістра реалізовано згідно із запропонованим у праці [6, 7] підходом і додаткового пояснення не потребує. Схематично процес вимірювання стану регістра R_i та перехід до класичного його подання можна зобразити таким чином:



ОПЕРАТОР КВАНТОВОГО ГЕЙТУ

Для реалізації обертання квантових станів для визначення кута та напрямку повороту традиційно використовується таблиця пошуку [4,5], що принципово обмежує універсальність роботи QGA. Фіксоване значення кута, яке використовується при цьому, також негативно впливає на швидкість збіжності, тому QGA інколи реалізують як адаптивний процес зміни його величини у процесі еволюції [5]. Більш ефективним, як показали проведені моделювання, може бути адаптивний характер роботи самого оператора квантового гейту.

Роботу оператора можна поділити на два етапи. На першому етапі збільшується амплітуда ймовірності вибраного стану квантового регістра R_i :

$$(\alpha_i^b)' = \sqrt{[\alpha_i^b]^2 + \mu(1 - [\alpha_i^b]^2)}, \quad (2)$$

де стан b визначається класичним поданням квантового регістра R_i найкращої особини популяції, що була отримана на попередній ітерації за часом. Функціональна залежність (2) забезпечує також той факт, що α_i^b не може перевищувати 1. Параметр μ визначає кут повороту квантових станів, підбирається за результатами модельних досліджень і перебуває в межах $[0,1]$.

На другому етапі необхідно зменшити амплітуди ймовірностей інших станів квантового регістра R_i для виконання умови нормування. Таким чином, у кожному новому поколінні забезпечується збільшення ймовірності того, що в результаті спостереження генеруються класичні особини, більш схожі на найкращу.

У цілому алгоритм роботи оператора до квантової хромосоми, що складається із $k = N/r$ регістрів, можна реалізувати таким чином:

Оператор квантового гейту

```

1  for  $i \in \{1, \dots, N/r\}$  do
2       $s \leftarrow 0$ 
3      for  $j \in \{0, \dots, 2^r - 1\}$  do
4           $s \leftarrow s + b_{i \cdot 2^r + j} \cdot 2^j$ 
5      end for
6       $M \leftarrow 1 - [\alpha_i^s]^2$ 
7       $\alpha_i^s \leftarrow \sqrt{[\alpha_i^s]^2 + \mu(1 - [\alpha_i^s]^2)}$ 
8       $M \leftarrow \sqrt{(1 - [\alpha_i^s]^2) / M}$ 
9      for  $j \in \{0, \dots, 2^r - 1\}$  do
10         if  $j \neq s$  then
11              $\alpha_i^j \leftarrow M \cdot \alpha_i^j$ 
12         end if

```

```

13     end for
14 end for

```

Динаміку поведінки пристосованості найкращої особини популяції залежно від величини μ ілюструє рис. 1. З огляду на ймовірнісний характер роботи алгоритму надалі всі значення наведено як результат усереднення за 100 запусками QGA. Як випливає із наведених даних, розмір квантового реєстра не впливає на величину оптимального значення параметра μ . Під час проведення експерименту за оптимальне значення можна взяти $\mu \approx 0,09$.

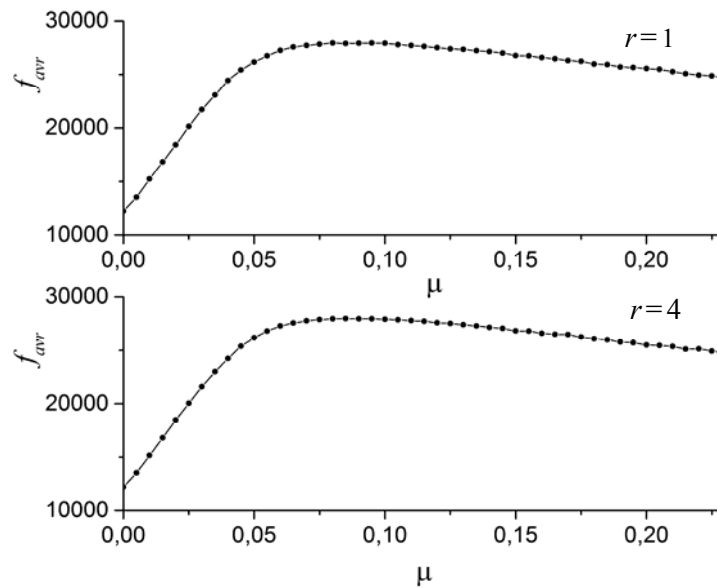


Рис. 1. Вплив параметра μ на середню пристосованість найкращої особини f_{avr} за різних розмірів квантового реєстра для $N = 500$ некорельованих вхідних даних

ЗАДАЧА ПАКУВАННЯ РЮКЗАКА

Проблема 0–1 пакування рюкзака є NP — повною задачею дискретної комбінаторної оптимізації, яка традиційно використовується для тестування пошукових алгоритмів. Вона зводиться до знаходження для скінченної множини N речей такого бінарного вектора заповнення $X = \{x_1, x_2, x_3, \dots, x_N\}$, який забезпечує максимальне значення функції

$$f(X) = \sum_{i=1}^N p_i x_i$$

і задовольняє обмеження

$$\sum_{i=1}^N \omega_i x_i \leq C, \tag{3}$$

де p_i — вартість i -го предмета; ω_i — маса i -го предмета; C — максимально можлива маса рюкзака.

У процесі ініціалізації популяції чи її еволюції завжди є ймовірність отримати ряд «поганих» особин, для яких не виконується обмеження (3) і які необхідно «відремонтувати». Процедура «ремонт» може бути зведена до видалення випадковим чином елементів з рюкзака доти, доки не виконається обмеження (3). Після відновлення функція пристосованості набуде вигляду

$$f(X') = \sum_{i=1}^N p_i x'_i,$$

де $X' = \{x'_1, x'_2, x'_3, \dots, x'_N\}$ — відновлена версія бінарного вектора X .

Процедура відновлення в QGA принципово інша, ніж в класичному генетичному алгоритмі, оскільки потребує коригування і ймовірностей квантових станів. На першому, традиційному етапі, якщо рюкзак надто важкий, елементи видаляються з нього доти, доки не буде задоволено обмеження (3). Якщо рюкзак надто легкий, то в рюкзак випадковим чином додаються речі, поки це дозволяє обмеження маси.

Перший етап роботи оператора можна подати у вигляді такого алгоритму:

«Ремонт» класичної хромосоми $P(t)$

```

// видалення випадковим чином речей з рюкзака
1  переповнення_рюкзака ← false
2      if  $\left( \sum_{i=1}^N \omega_i x_i > C \right)$  then
3          переповнення_рюкзака ← true
4      end if
5      while (переповнення_рюкзака ← true)
6           $k \leftarrow \text{random} \{1 \dots N\}$  — вибір випадковим чи-
ном предмета з рюкзака
7           $x_k \leftarrow 0$  — видалення предмета з рюкзака
8          if  $\left( \sum_{i=1}^N \omega_i x_i \leq C \right)$  then
9              переповнення_рюкзака ← false
10             end if
11         end while
// додавання випадковим чином речей до рюкзака
12     for  $i \in \{1, \dots, D\}$ 
13          $k \leftarrow \text{random} \{1 \dots N\}$  — вибір випадковим чи-
ном предмета для рюкзака
14         if  $\left( \sum_{i=1}^N \omega_i x_i + \omega_k > C \right)$  or  $(x_k = 1)$  then
15             continue
16         end if
17          $x_k = 1$ 
18     end for

```


Тут $D = 0,02dN$ — параметр роботи оператора відновлення. Його величина визначає кількість спроб випадковим чином додавання предметів до рюкзака. Згідно з проведеними дослідженнями за оптимальне можна взяти значення $d = 4 \dots 6$; воно забезпечує ефективність процесу відновлення за незначного, у межах 15%, зростання часу роботи алгоритму в цілому (рис. 2). Такий алгоритм «ремонт» забезпечує різноманітність популяції та ефективний глобальний пошук за множиною всіх допустимих розв'язків.

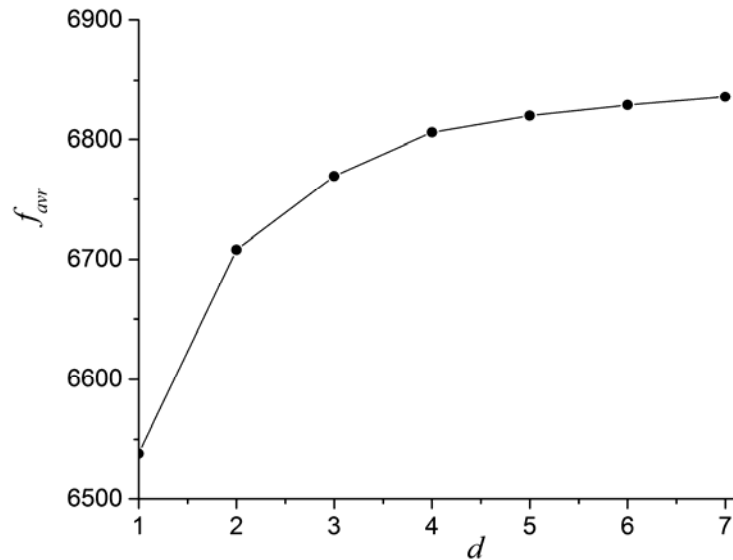


Рис. 2. Вплив параметра відновлення d на ефективність роботи алгоритму, якщо $N = 500$, сильнокорельованих вхідних даних ($r = 2$)

На другому етапі необхідно відкоригувати амплітуди ймовірностей квантової хромосоми відповідно до нової, відновленої класичної хромосоми X' . Процес відновлення реалізовано як оператор квантового гейту, напрямку кута повороту, у якому визначається значенням відновленої класичної хромосоми. Такий алгоритм забезпечує збереження генетичної інформації, яку нагромадила особина в процесі всієї своєї попередньої еволюції:

Відновлення квантової хромосоми

```

1   for  $i \in \{1, \dots, N/r\}$  do
2        $s \leftarrow 0$ 
3       for  $j \in \{0, \dots, 2^r - 1\}$  do
4            $s \leftarrow s + x_{i \cdot 2^r + j} \cdot 2^j$ 
5       end for
6        $M \leftarrow 1 - [\alpha_i^s]^2$ 
7        $\alpha_i^s \leftarrow \sqrt{[\alpha_i^s]^2 + \beta(1 - [\alpha_i^s]^2)}$ 
8        $M \leftarrow \sqrt{(1 - [\alpha_i^s]^2) / M}$ 
9       for  $j \in \{0, \dots, 2^r - 1\}$  do

```

```

10         if    $j \neq s$  then
11              $\alpha_i^j \leftarrow M\alpha_i^j$ 
12         end if
13     end for
14 end for

```

Тут $\beta \in [0, 1]$ — параметр відновлення. Його вплив на ефективність роботи QGA ілюструє рис. 3. Така поведінка є типовою, не залежить від рівня кореляції вхідних даних та розміру квантового регістра. Як випливає з рис. 3, оптимальним для подальших досліджень можна вважати $\beta \approx 0.15$.

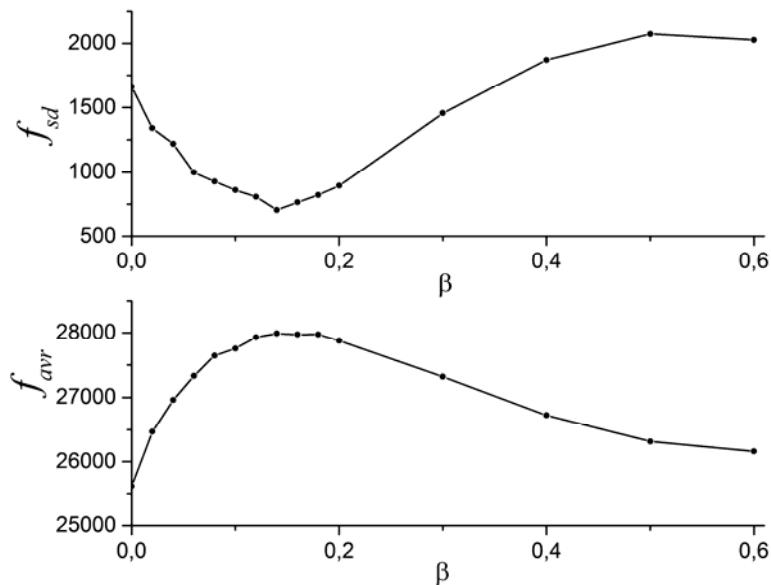


Рис. 3. Вплив β на середню пристосованість найкращої особини f_{avr} та середньоквадратичне відхилення f_{sd} для $N = 500$ некорельованих вхідних даних, якщо $r = 3$

РЕЗУЛЬТАТИ ЧИСЛОВОГО ЕКСПЕРИМЕНТУ

Квантовий генетичний алгоритм вищого порядку реалізовано на C++, а моделювання виконано на процесорі Intel Celeron CPU G1840 2,80 GHz, 4,0 Гб оперативної пам'яті. Для ілюстрації ефективності роботи алгоритму розглянуто ряд задач комбінаторної оптимізації великої розмірності та різного рівня кореляції вхідних даних (рис. 4). Задачі малої розмірності не розглядаються, оскільки за умови $N \leq 100$ отримуються майже точні результати, що не дозволяє оцінити вплив розміру квантового регістра на ефективність запропонованого алгоритму. Розмір популяції $s = 10$, що забезпечує 5000 звернень до функції пристосованості за часу еволюції $t = 500$.

Вплив розміру квантового регістра на середній час роботи QGA показано на рис. 5. Якщо $r = 1$, маємо випадок класичного квантового генетичного алгоритму і власне відносно нього виконано порівняння ефективності переходу до QGA вищого порядку. Середній час роботи t_{avr} залежить тіль-

ки від розмірів області пошуку, яка для задачі пакування рюкзака визначається лише довжиною квантової хромосоми N (множиною речей).

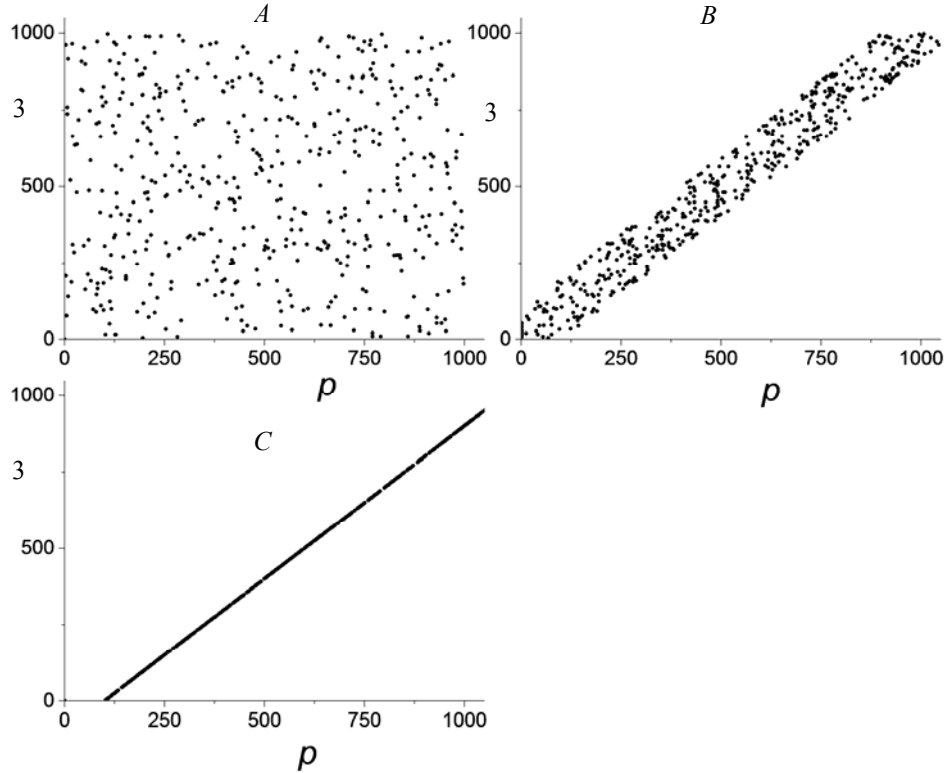


Рис. 4. Розподіл вартості предметів ω як функція їх маси ρ для різного рівня кореляції вхідних даних: A — некорельовані дані; B — слабкорельовані дані; C — сильнокорельовані дані

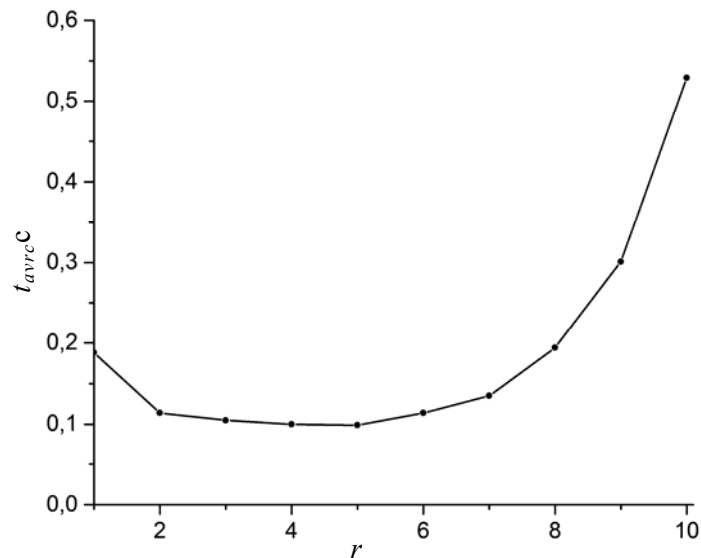


Рис. 5. Середній час роботи t_{avr} як функція розміру квантового регістра r для $N = 500$ некорельованих вхідних даних

Як видно з рис. 6, якщо як критерій зупинки використати середній час роботи QGA першого порядку ($r = 1$), то найефективнішими можна вважати розміри квантового регістра 3 та 6 кубітів.

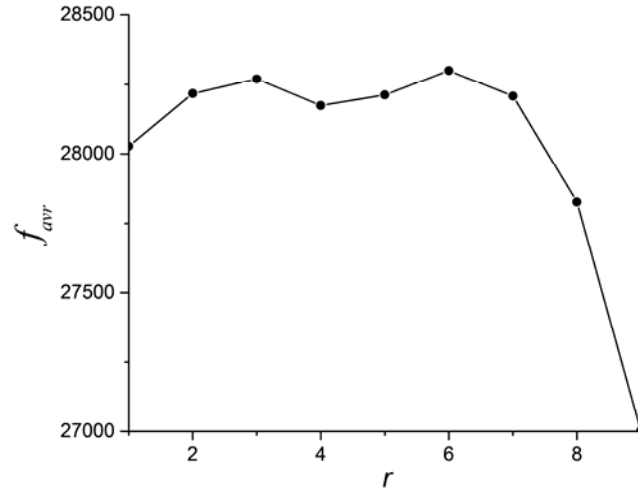


Рис. 6. Вплив розміру квантового регістра r на величину f_{avr} для $N = 500$ некорельованих вхідних даних

Згенерована на початку роботи QGA популяція ($t = 0$) включає широкий спектр усіх можливих розв'язків (рис. 7), які однак достатньо далекі від оптимального значення. Це означає, що робота QGA починається з випадкового пошуку. На початкових етапах еволюції в межах часу до $t = 100 \dots 150$ основним механізмом пошуку є глобальний пошук, зумовлений імовірнісним характером роботи оператора квантового вимірювання. Розподіл розв'язків поступово зсувається до оптимального значення за майже незмінної величини середньоквадратичного відхилення (різноманітності розв'язків).

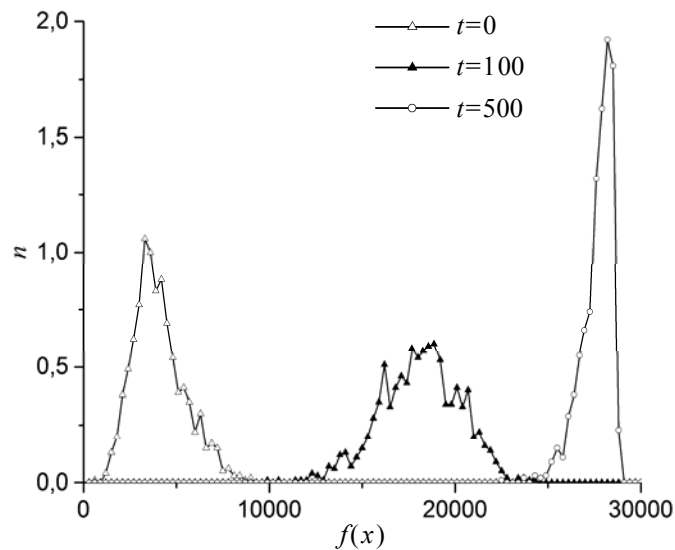


Рис. 7. Усереднений розподіл розв'язків по області пошуку залежно від часу еволюції популяції t

На кінцевих етапах еволюції ($t = 500$) характер розподілу знає принципів змін, що означає включення механізму локального пошуку за рахунок оператора квантового гейту. Детальніше механізм роботи QGA на різних етапах еволюції проаналізовано у праці [7].

Статистичні результати моделювання — середнє значення найкращої особини популяції f_{avr} , середньоквадратичне відхилення f_{sd} , найкращий та найгірший результати, середній час роботи t_{avr} при розв'язанні тестових задач великої розмірності — наведено в табл. 1–9. Розрахунки виконано за розмірів квантового регістра в межах 1–7 кубітів, бо власне за таких значень час роботи не перевищує час роботи традиційного QGA.

Таблиця 1. Результати оптимізації, якщо $N = 200$ некорельованих даних (точне значення — 11238)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,066	10903	651,9	11238	8872
2	0,042	10848	708,7	11238	9020
3	0,036	11137	131,4	11238	10630
4	0,034	10861	674,7	11238	8767
5	0,035	10825	708,7	11238	8989
6	0,040	11135	133,2	11238	10693
7	0,048	11119	157,3	11238	10509

Таблиця 2. Результати оптимізації, якщо $N = 500$ некорельованих даних (точне значення — 28857)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,175	27984	730,4	28623	25207
2	0,116	27987	741,3	28703	22020
3	0,098	28048	396,1	28834	27244
4	0,095	27974	735,4	28834	22926
5	0,097	27936	747,9	28834	22751
6	0,109	28041	416,3	28801	26746
7	0,136	28023	398,1	28790	26988

Таблиця 3. Результати оптимізації, якщо $N = 1000$ некорельованих даних (точне значення — 54503)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,372	49802	1243,3	52610	46101
2	0,233	49785	1224,3	52511	46127
3	0,203	50330	971,9	53482	47600
4	0,197	49882	1270,6	52598	46414
5	0,202	49780	1310,0	52338	46737
6	0,233	50311	982,6	52203	47434
7	0,281	50319	922,1	52348	47871

Таблиця 4. Результати оптимізації, якщо $N = 200$ слабокорельованих даних (точне значення — 1634)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,068	1542,1	31,3	1634	1486
2	0,044	1544,0	33,5	1627	1478
3	0,039	1608,3	16,6	1624	1544
4	0,036	1542,3	33,0	1633	1497
5	0,038	1543,5	32,5	1634	1495
6	0,044	1608,0	17,2	1634	1567
7	0,054	1608,1	17,6	1634	1541

Таблиця 5. Результати оптимізації, якщо $N = 1000$ слабокорельованих даних (точне значення — 4566)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,194	4309,6	152,5	4493	3899
2	0,127	4313,5	128,3	4517	3933
3	0,101	4373,0	61,9	4506	4179
4	0,098	4302,1	157,4	4514	3953
5	0,100	4301,6	155,0	4489	3950
6	0,114	4371,8	62,9	4508	4176
7	0,141	4376,6	63,2	4512	4130

Таблиця 6. Результати оптимізації, якщо $N = 1000$ слабокорельованих даних (точне значення — 9052)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,368	8260,6	139,8	8582	7918
2	0,239	8267,9	139,0	8621	7830
3	0,203	8309,1	129,2	8697	7945
4	0,197	8261,3	142,9	8604	7917
5	0,202	8253,1	146,8	8579	7973
6	0,227	8311,7	129,5	8679	7991
7	0,300	8309,7	138,4	8668	7818

Таблиця 7. Результати оптимізації, якщо $N = 200$ сильнокорельованих даних (точне значення — 2697)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,066	2580,3	86,5	2697	2397
2	0,043	2579,7	84,0	2697	2494
3	0,037	2661,6	47,1	2697	2594
4	0,035	2578,5	85,4	2697	2493
5	0,035	2574,1	84,2	2697	2491
6	0,041	2659,9	48,5	2697	2593
7	0,050	2663,4	46,7	2697	2594

Таблиця 8. Результати оптимізації, якщо $N = 500$ сильнокорельованих даних (точне значення — 7117)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,184	6791,2	175,2	7017	6409
2	0,115	6784,3	177,9	7017	6315
3	0,099	6898,9	76,3	7017	6716
4	0,097	6788,9	178,2	7017	6317
5	0,098	6787,6	172,8	7017	6417
6	0,112	6896,5	73,4	7017	6717
7	0,142	6891,2	75,3	7017	6714

Таблиця 9. Результати оптимізації, якщо $N = 1000$ сильнокорельованих даних (точне значення — 14390)

r	t_{avr}, c	f_{avr}	f_{sd}	Найкраще значення	Найгірше значення
1	0,361	13532,0	157,0	13887	13074
2	0,234	13528,2	157,2	13884	13089
3	0,203	13577,8	155,3	13890	13186
4	0,197	13534,2	156,6	13883	13088
5	0,204	13529,7	161,2	13874	13162
6	0,228	13567,0	156,2	13890	13190
7	0,287	13564,0	155,7	13889	13190

ВИСНОВКИ

У роботі запропоновано новий квантовий генетичний алгоритм вищого порядку та проілюстровано ефективність його роботи на прикладі задач 0–1 пакування рюкзака великої розмірності та з різним рівнем кореляції даних. Для подання квантової хромосоми застосовано квантові реєстри вищих порядків ($r > 1$), використання яких дозволяє включити в роботу такий механізм інтенсифікації квантових обчислень, як заплутаність квантових станів. Такий перехід зменшує час роботи алгоритму майже в два рази, точність пошуку розв'язку залишається незмінною, а за деяких розмірів реєстра навіть зростає.

Використання адаптивного оператора квантового гейту та оператора відновлення квантової хромосоми забезпечує ефективне поєднання глобального та локального пошуку алгоритму. Запропонований оператор відновлення квантової хромосоми, робота якого реалізована аналогічно до принципів роботи квантового гейту, забезпечує збереження генетичної інформації, що набула особина в ході всієї своєї попередньої еволюції.

Ефективність роботи запропонованого QGA вищого порядку оцінено порівняно із традиційним QGA ($r = 1$). Порівняння з іншими алгоритмами розв'язку задач оптимізації не виконувалося, бо, як впливає з багатьох публікацій [6, 10, 11], його можна вважати одним з найефективніших.

Експериментальні результати 0–1 задачі пакування рюкзака ілюструють можливості застосування QGA до задач великої розмірності за розмірів популяції, що не перевищують 10 особин. За співвідношенням швидкодія/ефективність оптимальним можна вважати розмір квантового регістра $r = 3$ і $r = 6$.

ЛІТЕРАТУРА

1. *Holland J.H.* Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence / J.H. Holland. — London: Bradford book edition, 1994. — 211 p.
2. *Simon D.* Evolutionary Optimization Algorithms: Biologically Inspired and Population-Based Approaches to Computer Intelligence / D. Simon. — John Wiley & Sons, 2013. — 742 p.
3. *Narayanan A.* Quantum-inspired genetic algorithms / A. Narayanan, M. Moore // Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'96), Nagoya, Japan, 1996. — P.61–66.
4. *Han K.–H.* Genetic quantum algorithm and its application to combinatorial optimization problem / K.–H. Han, J.–H. Kim // Proc.Congress on Evolutionary Computation. — Vol. 2. — La Jolla, CA, July 2000. — P. 1354–1360.
5. *Wang H.* The improvement of quantum genetic algorithm and its application on function optimization / H. Wang, J. Liu, J. Zhi, C. Fu // Math. Probl. Eng. — 2013. — P.1–10.
6. *Nowotniak R.* Higher-Order Quantum-Inspired Genetic Algorithms / R. Nowotniak, J. Kucharski // Federated Conference on Annals of Computer Science and Information Systems, 2014. — P. 465–470.
7. *Tkachuk V.* Quantum Genetic Algorithm Based on Qutrits and Its Application / V. Tkachuk // Mathematical Problems in Engineering. — Vol. 2018, Article ID 8614073, 8 p.
8. *Zhang G.* Quantum-inspired evolutionary algorithms: a survey and empirical study / G. Zhang // Journal of Heuristics. — 2010. — P. 1–49.
9. *Han K.* Quantum-inspired evolutionary algorithms with a new termination criterion, h-epsilon gate, and two-phase scheme / K. Han, J. Kim // IEEE Trans. Evol. Comput. — 2014. — 8 (2). — P. 156–169.
10. *Iimura I.* Integer-Type Gene-Coding Method Based on Quantum Bit Representation in Quantum-Inspired Evolutionary Algorithm: Application to Integer Knapsack Problem / I. Iimura // Journal of Signal Processing. — 2012. — Vol. 16, N 6. — P. 495–502.
11. *Takata T.* Performance Analysis of Quantum-Inspired Evolutionary Algorithm / T. Takata, T. Isokawa, A. Saitoh et al // SCIS & ISIS 2010, Dec. 8-12, Okayama Convention Center, Okayama, Japan.

Надійшла 10.07.2018