

УДК 681.3

О. П. Ротштейн, д. т. н., проф.; С. Д. Штовба, к. т. н., доц.;
С. Б. Дубіненко к. т. н., доц.; О. М. Козачко асп.

ЕВРИСТИЧНА ОПТИМІЗАЦІЯ РОЗСТАНОВКИ КОНТРОЛЬНИХ ТОЧОК В ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ ПРИ БАГАТОВИМІРНОМУ ПРОСТОРІ ТИПІВ ДЕФЕКТІВ

Вступ

Під час проектування технологічного процесу необхідно вміти давати відповідь на такі важливі питання: «Чи забезпечує технологічний процес необхідні рівні надійності функціонування?» та «Які зміни в процес необхідно внести, щоб забезпечити необхідні чи максимально можливі рівні надійності?». Моделі аналізу надійності технологічних процесів розроблені в роботах [1, 2]. В цих роботах технологічний процес представляється як послідовність основних (робочих) та допоміжних (контрольних та доробчих) операцій, виконання яких призводить до одержання вихідного продукту праці. В [2] запропоновані алгоритми оптимального синтезу, що забезпечують необхідні або мінімально можливі рівні ймовірності наявності дефектів у вихідному продукті праці та трудомісткості технологічного процесу.

Використання алгоритмів оптимізації [2] обмежуються бінарним простором типів дефектів: є/немає. В багатьох реальних задачах використання бінарної концепції врахування дефектів є недоцільним, оскільки для різних типів дефектів різняться ймовірності їх внесення, розпізнавання та усунення, так само як і витрати часу на ці процедури. Напри-

клад, для процесу виробництва друкованих плат ймовірність виправлення дефектів травлення, пов'язаних з неправильним орієнтуванням фотошаблону, майже завжди дорівнює нулю, а ймовірність виправлення дефекту, пов'язаного з неповною очисткою плати від відпрацьованого фоторезисту є близькою до одиниці. В зв'язку з цим виникає зацікавленість в розробленні теорії проектування технологічних процесів з урахуванням дефектів різних типів, тобто в багатовимірному просторі типів дефектів.

1. Багатовимірний простір типів дефектів

Ідея моделювання надійності дискретних процесів в багатовимірному просторі типів дефектів запропоновані в [2]. Технологічний процес можна уявити як потік води, в якому одночасно «пливуть» забруднення-дефектів різних типів. Наприклад, для процесу виробництва радіоапаратури такими дефектами можуть бути неякісна пайка, неправильна орієнтація мікросхеми, розрив друкованого провідника тощо. Дефекти вносять під час виконання основних технологічних операцій, а виявляють та усувають під час виконання допоміжних операцій контролю та доробки. Позначимо через m кількість можливих типів дефектів. Тоді стан предмету праці можна характеризувати вектором $Z = (z_1, z_2, \dots, z_m)$ в m -вимірному просторі типів дефектів, де координата z_j вказує на наявність ($z_j = 1$) або відсутність ($z_j = 0$) дефектів j -го типу, $j = \overline{1, m}$.

2. Матричні моделі прогнозування надійності технологічних процесів

Відповідно до [2], ймовірнісні характеристики якості виконання основних та допоміжних технологічних операцій представимо у вигляді матриць. Таке матричне представлення забезпечує компактність моделей прогнозування надійності технологічних процесів у багатовимірному просторі типів дефектів.

Ймовірнісні характеристики якості виконання робочої операції A представимо у вигляді такої матриці розміром $(m+1) \times (m+1)$ [2]

$$\mathbf{P}_A = \begin{pmatrix} p^1 & p_1^0 & \dots & p_m^0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

де p_j^0 — ймовірності внесення дефекту j -го типу під час виконання робочої операції A ; p^1 — ймовірність правильного виконання робочої операції A без внесення будь-яких дефектів в предмет праці, тобто ймовірність того, що вектор стану Z предмету праці після виконання робочої операції A залишиться без змін. Зауважимо, що $p^1 + \sum_{j=1}^m p_j^0 = 1$.

Ймовірнісні характеристики якості виконання доробки V представимо у вигляді такої матриці [2]

$$\mathbf{P}_V = \begin{pmatrix} 1 & 0 & \dots & 0 \\ v_1^1 & v_1^0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_m^1 & 0 & \dots & v_m^0 \end{pmatrix},$$

де $v_j^1(v_j^0)$ — ймовірність усунення (не усунення) дефектів j -го типу, причому $v_j^1 + v_j^0 = 1$, $j = \overline{1, m}$.

Наведена матриця відповідає випадку, коли в процесі доробки переходи з одних типів дефектів на інші не дозволяються і додаткові дефекти не вносяться.

Ймовірнісні характеристики якості виконання контролю ω представимо у вигляді двох матриць, які задають ймовірності помилок першого та другого родів [2]

$$\mathbf{K}_w^1 = \begin{pmatrix} k^{11} & 0 & \dots & 0 \\ 0 & k_1^{01} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_m^{01} \end{pmatrix}, \quad \mathbf{K}_w^0 = \begin{pmatrix} k^{10} & 0 & \dots & 0 \\ 0 & k_1^{00} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_m^{00} \end{pmatrix},$$

де $k^{11}(k^{10})$ — ймовірність того, що під час контролю відсутність дефектів ідентифіковано правильно (неправильно); $k_j^{01}(k_j^{00})$ — ймовірність пропуску (виявлення) дефектів j -го типу, причому $k_j^{01} + k_j^{00} = 1$, $j = \overline{1, m}$.

З наведених матриць випливає, що здатності дефектовиявлення (дефектопропуску) контрольних операцій можуть бути різними для різних типів дефектів. При цьому припускається, що в процесі контролю відсутні помилки типу $i-j$, тобто явища віднесення дефекту i -го типу до дефекту j -го типу.

Витрати на виконання робочої операції A , контролю ω і доробки V позначимо через C_A , C_ω та C_V , відповідно.

Моделювання надійності дискретних процесів здійснюється з використанням типових технологічних структур. Під типовою технологічною структурою розуміється комбінація основних та допоміжних операцій, для якої отримані математичні моделі, що дозволяють замінити її єдиною робочою операцією з еквівалентними ймовірно-вартісними характеристиками [2]. Задача моделювання надійності функціонування технологічного процесу зводиться до ітераційної процедури виявлення та укрупнення типових структур, доки не залишиться одна робоча операція. Еквівалентні характеристики цієї операції і будуть шуканими характеристиками якості виконання всього технологічного процесу. Для моделювання надійності технологічних процесів без зворотних зв'язків достатньо дві типові структури: «послідовна» і «робота-контроль-доробка». Для алгоритмічного опису типових структур будемо використовувати мову алгоритмічних алгебр В. М. Глушкова [3, 4].

Структура «послідовна» $B = A_1 \dot{A}_2$ полягає у виконанні робочих операцій A_1 і A_2 в порядку запису. Еквівалентні характеристики цієї типової структури розраховуються за формулами

$$\mathbf{P}_B = \mathbf{P}_{A_1} \cdot \mathbf{P}_{A_2}; \quad (1)$$

$$C_B = C_{A_1} + C_{A_2}. \quad (2)$$

Структура «робота-контроль-доробка» $C = A (E \vee V)$ полягає у виконанні робочої операції A , контролю ω та доробки V при виявленні дефектів, і тотожного оператора E в протилежному випадку. Тотожний оператор E не змінює стан системи, він може бути інтерпретований як фіксація результатів контролю. Еквівалентні характеристики типової структури «робота-контроль-доробка» розраховуються за формулами:

$$\mathbf{P}_C = \mathbf{P}_A \cdot \mathbf{K}_w^1 + \mathbf{P}_A \cdot \mathbf{K}_w^0 \cdot \mathbf{P}_V; \quad (3)$$

$$C = (C_A + C_\omega) \cdot \mathbf{I} + \mathbf{P}_A \cdot \mathbf{K}_w^0 \cdot C_v, \quad (4)$$

де \mathbf{I} — одинична матриця.

3. Постановки задач оптимізації надійності в багатовимірному просторі типів дефектів

Існують різні можливості управління надійністю технологічних процесів: вибір способів реалізації технологічних операцій, розстановка контрольних точок, вибір кратностей та часток контрольних процедур тощо. В статті пропонуються алгоритми розв'язання однієї з найважливіших в практичному плані задач оптимізації надійності технологічних процесів — розстановки контрольних точок.

Позначимо через $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ вектор контролів, де значення координати x_i вказують на наявність ($x_i = 1$) або відсутність ($x_i = 0$) контролю після i -ї потенційно контрольованої робочої операції ($i = \overline{1, n}$); n — кількість потенційно контрольованих

операцій в заданому технологічному процесі.

В загальному вигляді задачу оптимізації розстановки контрольних точок можна сформулювати як задачу мінімізації середніх сумарних витрат [2]

$$C_{\Sigma} = C(\mathbf{X}) + C_e(\mathbf{X}) \rightarrow \min, \quad (5)$$

де $C(\mathbf{X})$ – витрати на виробництво продукту праці технологічним процесом, який задано вектором \mathbf{X} ; $C_e(\mathbf{X})$ – витрати на експлуатацію продукту праці, що вироблений технологічним процесом, який заданий вектором \mathbf{X} .

Відомо [2], що середні витрати на експлуатацію пов'язані з ймовірностями наявності дефектів у вихідному продукті праці

$$C_e(\mathbf{X}) = f(p_1^0(\mathbf{X}), p_2^0(\mathbf{X}), \dots, p_m^0(\mathbf{X})), \quad (6)$$

де $p_j^0(\mathbf{X})$ – ймовірність дефектів j -го типу на виході технологічного процесу, який заданий вектором \mathbf{X} ($j = \overline{1, m}$); f – деяка нелінійна функція.

Для практичних задач ідентифікувати залежність (6) дуже важко, а інколи і неможливо. В цих випадках можна використовувати лінійну апроксимацію і представити залежність (6) як функцію ризику. Тоді задача (5) перетвориться до вигляду

$$C_{\Sigma} = C(\mathbf{X}) + (p_1^0(\mathbf{X}), p_1^0(\mathbf{X}), \dots, p_1^0(\mathbf{X})) \cdot (R_1, R_2, \dots, R_m)^T \rightarrow \min, \quad (7)$$

де R_j – витрати на експлуатацію, що пов'язанні з наявністю дефектів j -го типу в вихідному продукті праці ($j = \overline{1, m}$).

Якщо відсутні достовірні дані про значення параметрів функції ризику (що часто буває на етапі проектування технологічних процесів), то задачу мінімізації (7) зручно поставити як задачу умовної оптимізації [5]:

– пряма постановка задачі

$$\left. \begin{aligned} C(\mathbf{X}) &\rightarrow \min, \\ p^1(\mathbf{X}) &\geq P_{\text{доп}}, \\ p_j^0(\mathbf{X}) &\leq P_j^{\text{max}}, \quad j = \overline{1, m}, \end{aligned} \right\} \quad (8)$$

де $p^1(\mathbf{X})$ – ймовірність бездефектного виконання технологічного процесу, який задано вектором \mathbf{X} ; $P_{\text{доп}}$ – мінімально допустима ймовірність бездефектного виконання технологічного процесу; $P_j^{\text{max}}(\mathbf{X})$ – максимально допустима ймовірність наявності дефектів j -го типу на виході технологічного процесу;

– зворотна постановка задачі

$$\left. \begin{aligned} p^1(\mathbf{X}) &\rightarrow \max, \\ C(\mathbf{X}) &\leq C_{\text{доп}}, \\ p_j^0(\mathbf{X}) &\leq P_j^{\text{max}}, \quad j = \overline{1, m} \end{aligned} \right\} \quad (9)$$

де $C_{\text{доп}}$ – максимально допустимі витрати на виконання технологічного процесу.

4. Алгоритми оптимізації

Поставлені задачі (8) і (9) відносяться до бульових задач нелінійного програмування. Задача (8) може інтерпретуватися як нелінійний аналог задачі про рюкзак. В цьому випадку, аналогом рюкзаку є технологічний процес, предметів – контрольно-добробчі процедури, об'єму рюкзака – максимально допустима трудомісткість процесу, а критерію корисності – ймовірність виконання технологічного процесу без будь-яких дефектів. Як відомо [6], лінійна задача про рюкзак є NP-складною, для якої не існує поліноміального алгоритму розв'язання. Тому, для задач (8)–(9) не існує алгоритму знаходження оптимальних розв'язків за поліноміальний час. Це обумовлює пошук евристик та метаевристик, використання яких забезпечувало б швидке знаходження якісних, тобто близьких до

оптимальних розв'язків. Нижче пропонується два алгоритма розв'язання задач (8)–(9): перший алгоритм відноситься до класу «пожадливих» (greedy) [8], а другий — до генетичних (genetic) [8].

4.1. Пожадливий алгоритм оптимізації

В теорії оптимального резервування надійності пожадливі алгоритми отримали назву градієнтних [6]. В основу пожадливих алгоритмів покладена ітераційна процедура зміни значення однієї керованої змінної, що призводить до найкращого локального покращення критерію ефективності.

Як критерій ефективності контрольно-добробчої процедури пропонується використовувати такий вираз

$$\gamma = [\mathbf{B}_i]^{-1} \cdot [\mathbf{B}_i - \mathbf{P}_{A_i}] \cdot [\mathbf{I} \cdot C_b]^{-1}, \quad (10)$$

де $\mathbf{B}_i = \mathbf{P}_{A_i}(\mathbf{K}_{\omega_i}^1 + \mathbf{K}_{\omega_i}^0 \cdot \mathbf{P}_{V_i})$ — ймовірність переходу на добробку, $C_b = C_{\omega_i} + \frac{1}{m} \sum_{i=1}^m (p^1 k^{10} + p_i^0 k_i^{00}) C_v$ — середні витрати, які пов'язані з встановленням контролю після i -ої потенційно контрольованої робочої операції.

Вираз (10) є аналогом градієнта контролю в алгоритмах оптимізації з бінарною концепцією врахування дефектів [2]. Пожадливі алгоритми розв'язання задач (8)–(9), що пропонуються нижче, є узагальненням градієнтних алгоритмів розстановки контрольних точок, які розроблені в монографії [2].

Алгоритм розв'язання задачі (8)

1⁰. Встановити всі контрольні операції: $\mathbf{X}^{(0)} = (x_1^0, x_2^0, \dots, x_n^0)$, $x_j^0 := 1$, $i = \overline{1, n}$. Розрахувати ймовірність бездефектного виконання технологічного процесу $p^1(\mathbf{X}^{(0)})$.

2⁰. Перевірити умову « $p^1(\mathbf{X}^{(0)}) < P_{\text{доп}}?$ », якщо «так», то задача не має розв'язків, інакше перейти до кроку 3⁰.

3⁰. Встановити лічильник ітерацій $N:=1$. Зняти всі контрольні операції $\mathbf{X}^{(N)} = (x_1^N, x_2^N, \dots, x_n^N)$, $x_j^N := 0$, $i = \overline{1, n}$.

4⁰. Розрахувати ймовірність бездефектного виконання технологічного процесу $p^1(\mathbf{X}^{(N)})$ та ймовірності появи дефектів $p_j^0(\mathbf{X})$, $j = \overline{1, m}$.

5⁰. Перевірити умову « $p^1(\mathbf{X}^{(N)}) < P_{\text{доп}}$ та $p_j^0(\mathbf{X}) > P_j^{\text{max}}$, $j = \overline{1, m}?$ », якщо «так», то перейти до кроку 6⁰, інакше — до кроку 9⁰.

6⁰. Обчислити градієнт контролю γ_i за формулою (10) для всіх i таких, що $x_i^N := 0$.

7⁰. Визначити координату l , при якій градієнт максимальний: $\gamma_l = \max_i \gamma_i$.

8⁰. Встановити $x_l^N := 1$, $N := N + 1$ та перейти до кроку 4⁰.

9⁰. Сформуувати таку множину індексів $L = (i \mid x_i^N = 1, i = \overline{1, n})$, для кожного елемента якої виконуються умови задачі (8) зі зняттям i -ї контрольної процедури $x_i^N := 0$. Якщо $L = \emptyset$, то перейти до кроку 13⁰.

10⁰. Зняти i -ту контрольну процедуру $x_i^N := 0$ та розрахувати вартість $C_i(\mathbf{X}^{(N)})$ виконання процесу для всіх $i \in L$.

11⁰. Знайти координату k , для якої виконується умова

$$C_k(\mathbf{X}^{(N)}) = \max_i C_i(\mathbf{X}^{(N)}).$$

12⁰. Встановити $x_k^N := 0$ в векторі \mathbf{X}^N .

13⁰. Кінець.

Алгоритм розв'язання задачі (9)

1⁰. Зняти всі контрольні операції $\mathbf{X}^{(0)} = (x_1^0, x_2^0, \dots, x_n^0)$, $x_j^0 := 0, j = \overline{1, n}$. Обчислити вартість виконання технологічного процесу $C(\mathbf{X}^{(0)})$.

2⁰. Перевірити умову « $C(\mathbf{X}^{(0)}) > C_{\text{доп}} ?$ », якщо «так», то задача не має розв'язків, інакше перейти до кроку 3⁰.

3⁰. Обчислити градієнт контролю γ_i за формулою (10) для всіх i таких, що $x_i^N = 0$.

4⁰. Визначити координату l , для якої градієнт є максимальним: $\gamma_l = \max_i \gamma_i$.

5⁰. Встановити $x_l^N := 1, N := N + 1$ та перейти до кроку 6⁰.

6⁰. Обчислити ймовірність появи кожного типу дефекту $p_j^0(\mathbf{X}), j = \overline{1, m}$ і вартість виконання технологічного процесу $C(\mathbf{X}^N)$.

7⁰. Перевірити умову « $C(\mathbf{X}^{(N)}) < C_{\text{доп}}$ та $p_j^0(\mathbf{X}) > P_j^{\text{max}}, j = \overline{1, m} ?$ », якщо «так» то перейти до кроку 3⁰, інакше – до кроку 8⁰.

8⁰. Сформувати таку множину індексів $L = \{i \mid x_i^N = 0, i = \overline{1, n}\}$, для кожного елемента якої виконуються умови задачі (9) зі встановленням i -ої контрольної процедури $x_i^N := 1$. Якщо $L = \emptyset$, то перейти до кроку 12⁰.

9⁰. Встановити i -ту контрольну процедуру $x_i^N := 1$ та розрахувати ймовірність правильного виконання процесу $p_i^1(\mathbf{X}^{(N)})$ для всіх $i \in L$.

10⁰. Знайти координату k , для якої виконується умова:

$$p_k^1(\mathbf{X}^{(N)}) = \max_i p_i^1(\mathbf{X}^{(N)});$$

11⁰. Встановити $x_k^N := 1$ у векторі \mathbf{X}^N ;

12⁰. Кінець.

Наведені алгоритми мають дві ітераційні процедури. В першій процедурі послідовно встановлюються контролю з максимальними градієнтами, поки виконуються обмеження. В другій процедурі здійснюється відкат на один крок для подальшого покращення знайденого розв'язку.

4.2. Генетичний алгоритм оптимізації

Генетичні алгоритми представляють собою стохастичний метод оптимізації, оснований на імітації процесів природної еволюції [8]. Типовий генетичний алгоритм складається з виконання такої послідовності кроків:

1⁰. Провести генетичне кодування.

2⁰. Випадковим чином згенерувати початкову популяцію, що містить *pop_size* хромосом.

3⁰. Випадковим чином обрати *cros_num* пар хромосом і виконати схрещування.

4⁰. Випадковим чином обрати *mut_num* хромосом і провести мутації.

5⁰. Сформувати нову популяцію в результаті селекції.

6⁰. Виконати кроки 3⁰ – 5⁰ *epoch_num* раз.

7⁰. Декодувати хромосоми і зафіксувати найкраще рішення.

Використання типового генетичного алгоритму для розв'язання конкретних задач вимагає визначення способів реалізації генетичних операцій кодування (декодування), схрещування, мутації і селекції, а також ініціалізації параметрів алгоритму: *pop_size*; *cros_num*; *mut_num* і *epoch_num*.

Генетичне кодування

Варіант технологічного процесу, представимо хромосомою $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$. Гени a_i ($i = \overline{1, n}$) хромосоми \mathbf{A} можуть приймати тільки два значення: 1 – робоча операція з Вісник ВПІ, 2004, № 1

контролем; 0 – без контролю. На рис. 1 показано приклад генетичного кодування для задачі розстановки контрольних точок з 10 потенційно контрольованими робочими операціями. У цьому прикладі контрольні точки встановлені після 2-ї, 5-ї, 7-ї і 10-ї робочих операцій.

0	1	0	0	1	0	1	0	0	1
1	2	3	4	5	6	7	8	9	10

Рис. 1. Приклад генетичного кодування

Операції схрещування і мутації

Будемо використовувати типові операції схрещування і мутації [8]. Операція схрещування полягає у обміні генів між двома випадково обраними хромосомами відносно деякого гена. В результаті операції схрещування отримуються два нащадки, які успадковують частини генів від обох батьків.

Операція мутації полягає у інверсії значенн випадково обраного гена.

Фітнес-функція

Розстановки контрольних точок є задачею умовної оптимізації, тому розв'язок повинен задовольняти усім обмеженням. Для врахування обмежень будемо використовувати ідеї методу штрафних функцій: якщо закодований хромосомою \mathbf{A} розв'язок не задовольняє хоча б одному з обмежень, тоді значення фітнес-функції необхідно погіршити. Комп'ютерні експерименти по розв'язанню генетичними алгоритмами задач дискретної оптимізації [7] свідчать, що добрі результати забезпечує використання так званої адаптивної штрафної функції. Ідея цієї функції полягає у використанні середньої суми відносних штрафних коефіцієнтів – відхилень обмежень, для яких не виконуються умови задачі. Для задач (8) – (9) будемо використовувати таку штрафну функцію:

$$S(\mathbf{A}) = \begin{cases} 1, & \text{якщо } \mathbf{A} \text{ допустимий розв'язок;} \\ 1 - \frac{1}{m+1} \left(\Delta d(\mathbf{A}) + \sum_{i=1}^m \left(\frac{\Delta b_i(\mathbf{A})}{\Delta b_i^{\max}} \right)^\alpha \right), & \text{якщо } \mathbf{A} \text{ недопустимий розв'язок,} \end{cases} \quad (11)$$

де $\Delta b_i(\mathbf{A}) = \max\{0, p_i^0(\mathbf{A}) - P_i^{\max}\}$, $i = \overline{1, m}$ – відхилення між ймовірністю появи і максимально допустимою ймовірністю i -го типу дефекту. Якщо відхилення від'ємне, то $\Delta b_i(\mathbf{A}) = 0$; $\Delta d(\mathbf{A}) = \max\{0, C(\mathbf{A}) - C_{\text{доп}}\}$ – відхилення між вартістю правильного виконання технологічного процесу і максимально допустимою вартістю для задачі (8), і $\Delta d(\mathbf{A}) = \max\{0, P_{\text{доп}} - p^1(\mathbf{A})\}$ – відхилення між мінімально допустимою ймовірністю та ймовірністю правильного виконання технологічного процесу для задачі (9); $\Delta b_i^{\max} = \max\{\Delta b_i(\mathbf{A}) | \mathbf{A} \in T\}$, $i = \overline{1, m}$ – максимальне відхилення для i -го обмеження в поточній популяції (варіантів технологічного процесу) T , $i = \overline{1, m+1}$; α – параметр, що визначає жорсткість штрафної функції.

На кожній ітерації генетичного алгоритму штрафна функція (11) дозволяє адаптивно корегувати недопустимі розв'язки через зменшення фітнес-функцій при відхиленні будь-якого обмеження.

Як фітнес-функцію хромосоми \mathbf{A} будемо використовувати:

$$F(\mathbf{A}) = \frac{1}{C(\mathbf{A})(1 - S(\mathbf{A}))} \quad (12)$$

для задачі (8) та

$$F(\mathbf{A}) = P(\mathbf{A}) S(\mathbf{A}) \quad (13)$$

для задачі (9).

Запропоновані фітнес-функції мають дві особливості, що забезпечують просту реалізацію

процедури селекції, а саме: 1) значення фітнес-функції завжди є додатними; 2) чим кращий розв'язок задачі, тим більше значення має фітнес-функція відповідної хромосоми.

Селекція

Будемо використовувати селекцію з елітизмом [8]. Навідміну від традиційної селекції за принципом «колеса рулетки» вона не дозволяє втратити найкращу хромосому на будь-якій ітерації генетичного алгоритму. Селекції з елітизмом відповідає така послідовність кроків.

1⁰. Включити в нову популяцію хромосому, що має максимальну фітнес-функцію.

2⁰. Сформувати множину хромосом, яка задовільняє всім обмеженням (8), (9) і серед цих хромосом обрати хромосому, яка має максимальну фітнес-функцію.

3⁰. Включати інші хромосоми з використанням «колеса рулетки».

5. Комп'ютерні експерименти

В цьому розділі наводяться результати комп'ютерних експериментів по розв'язанню задач розстановки контрольних точок з використанням запропонованих пожадливого та генетичного алгоритмів. Експерименти проведені в семивимірному просторі типів дефектів ($m = 7$) для технологічних процесів різної розмірності з кількістю потенційно контрольованих робочих операцій від 20 до 110. Умови тестових задач наведені на сайті www.ksu.vstu.vinnica.ua/shtovba/vysnik.

Параметри генетичного алгоритму для різних тестових задач показані в табл. 1.

Таблиця 1

Параметри генетичного алгоритму

параметр \ n	20	30	40	50	60	70	80	90	100	110
<i>Pop_size</i> ;	20	30	50	50	55	60	75	80	85	90
<i>Cros_num</i> ;	20	25	30	35	40	45	50	55	60	65
<i>Mut_num</i>	35	40	45	55	60	65	75	80	85	90
<i>Epoch_num</i>	100	100	100	100	100	100	100	100	100	100
<i>a</i>	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Результати застосування пожадливого та генетичного алгоритмів для вирішення оптимізаційних задач (8) та (9) показані в табл. 2 та 3. Видно, що різниця між розв'язками задач, що отримані різними методами, є не дуже суттєвою, але все ж генетичні алгоритми забезпечують знаходження кращих варіантів технологічних процесів.

Таблиця 2

Ймовірність бездефектного виконання технологічного процесу для знайдених розв'язків задачі (8)

метод \ n	20	30	40	50	60	70	80	90	100	110
пожадливий алгоритм	0,825	0,721	0,656	0,553	0,503	0,462	0,389	0,352	0,311	0,275
генетичний алгоритм	0,836	0,732	0,667	0,563	0,514	0,473	0,400	0,363	0,323	0,291

Таблиця 3

Вартість виконання технологічного процесу для знайдених розв'язків задачі (9)

метод \ n	20	30	40	50	60	70	80	90	100	110
пожадливий алгоритм	80,04	125,2	156,24	213,36	249,99	286,05	355,94	390,18	431,15	480,94
генетичний алгоритм	78,47	123,2	152,17	210,04	246,89	283,10	348,46	390,99	427,56	475,78

На рис. 1 та рис. 2 показані залежності середнього часу оптимізації різними методами від розмірності задач (8) та (9), відповідно. Для задач невеликої розмірності, з числом потенційно контрольованих робочих операцій до 50, простий пожадливий алгоритм оптимізації є швидшим в порівнянні з генетичним. Для задач великої розмірності, коли кількість потенційно контрольованих робочих операцій перевищує 70, генетичні алгоритми стають ефективнішими за пожадливі. Зазначимо, що обчислювальна ефективність генетичних алгоритмів збільшується зі зростанням розмірності оптимізаційних задач.

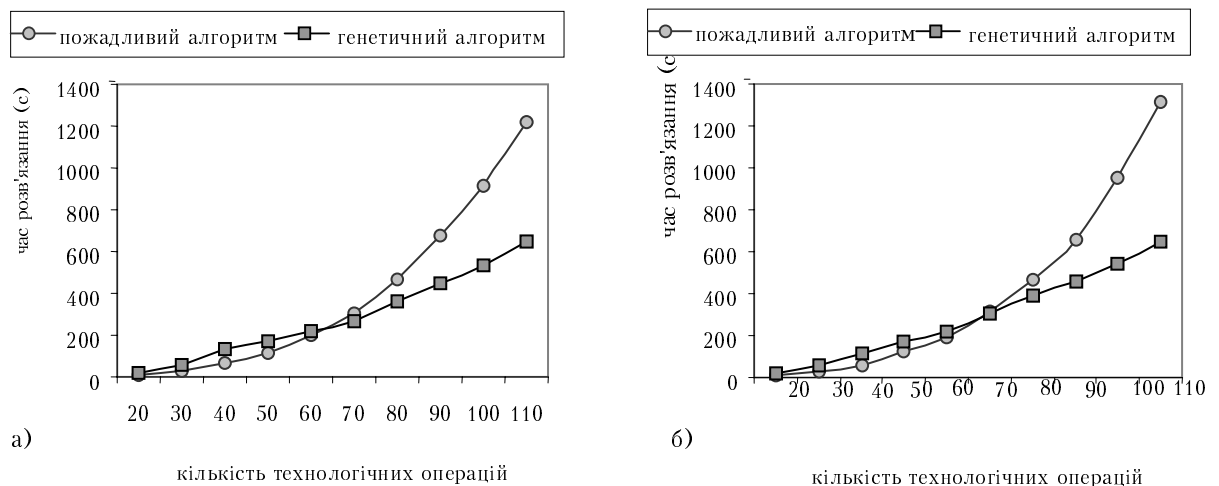


Рис. 2. Залежність часу оптимізації технологічних процесів від кількості операцій:
а) – для задачі (8); б) – для задачі (9)

Висновки

В статті запропоновані пожадливий та генетичний алгоритми оптимізації розстановки контрольних точок в технологічному процесі. Новизною алгоритмів є те, що на відміну відомого підходу [2], який ґрунтується на бінарній концепції врахування дефектів, запропоновані алгоритми дозволяють оптимізувати технологічні процеси, в яких дефекти розрізняються за типами. Проведені комп'ютерні експерименти показують, що застосування генетичних алгоритмів забезпечує знаходження кращих варіантів технологічних процесів в порівнянні з пожадливими алгоритмами оптимізації. Також встановлено, що вирішення оптимізаційних задач невеликої розмірності швидше здійснюють пожадливі алгоритми. З ростом розмірності задач кращу обчислювальну ефективність демонструють генетичні алгоритми оптимізації. Ідеї, що закладені в запропоновані алгоритми, є привабливими для розроблення методів розв'язання інших задач надійної оптимізації технологічних процесів в багатовимірному просторі типів дефектів, наприклад, оптимізації кратності контролів, знаходження оптимальних часток контролів, вибору оптимальних реалізацій основних та допоміжних операцій технологічного процесу.

СПИСОК ЛІТЕРАТУРИ

1. Дружинин Г. В. Надежность автоматизированных производственных систем. — М.: Энергоатомиздат, 1986. — 480 с.
2. Ротштейн А. П., Кузнецов П. Д. Проектирование бездефектных человеко-машинных технологий. К.: Техніка, 1992. — 180 с.
3. Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. Алгебра. Языки. Программирование. Киев: Наука, 1982. — 270 с.
4. Ющенко К. Л., Суржко С. В., Цейтлін Г. О., Шевченко А. І. Алгоритмічні мови: Навчальний посібник. — К: ІЗМН, 1997. — 480 с.
5. Ротштейн А. П., Штовба С. Д., Козачко А. Н. Оптимизация расстановки контрольных точек в многомерном технологическом процессе на базе генетических алгоритмов. Праці V Міжнародної конференції з індуктивного моделювання, Львів, Україна. Том 2. — С. 103—109.
6. Michael R. Garey, David S. Johnson. Computers and Intractability. A Guide to the Theory of NP-Completeness. — W. H. Freeman and company. — 2000. — 338 p.
7. Ушаков И. А. Методы решения простейших задач надежности при наличии ограничений. — М.: Сов. Радио, 1969. — 162 с.
8. Gen, M., Cheng R. Genetic Algorithms and Engineering design. — John Wiley&Sons. — 1997. — 352 p.

Рекомендована кафедрою комп'ютерних систем управління

Надійшла до редакції 22.05.03
Рекомендована до опублікування 11.10.03

Ротштейн Олександр Петрович — професор, кафедри інформаційного менеджменту.
Штовба Сергій Дмитрович — докторант, **Дубіненко Світлана Борисівна** — доцент,
Козачко Олексій Миколайович — аспірант.

Кафедра комп'ютерних систем управління, Вінницький національний технічний університет