

Студентська наукова робота

МОДЕЛЬ ВИБОРУ ОПТИМАЛЬНИХ УМОВ МЕХАНІЧНОЇ ОБРОБКИ  
РІЗАННЯМ ПРИ ТАНГЕНЦІАЛЬНОМУ ТОЧІННІ ЗА ДОПОМОГОЮ  
ГЕНЕТИЧНОГО АЛГОРИТМУ

Шифр (тангенціальне точіння)

## ЗМІСТ

Вступ.....	3
1. Особливості механічної обробки різанням за допомогою тангенціального точіння.....	5
2. Метод вибору оптимальних умов механічної обробки за допомогою генетичного алгоритму.....	12
3. Результати досліджень.....	21
Список літератури.....	24
Додаток А.....	26

## ВСТУП

За останні роки широкого поширення набула автоматизація різних процесів обробки і збірки, включаючи обробку різанням. Від окремих верстатів з ЧПУ і багатоцільових верстатів до гнучких ділянок і верстатних автоматизованих систем, керованих від ЕОМ, такий головний напрям автоматизації сучасного машинобудівного виробництва. Процеси обробки різанням до теперішнього часу зберігають лідируючі позиції в технології машинобудування як найбільш продуктивні, точні, економічні, а у багатьох випадках і екологічні методи формоутворення деталей [1].

Поряд із формуванням характеристик металевих виробів, механічними засобами оброблюються великі об'єми самих різних матеріалів, таких як деревина, поліграфічні матеріали, пластмаси, скло, композиційні матеріали та інші. Ефективність процесів механічної обробки є основою для випуску конкурентоздатної продукції. Підвищення продуктивності, гнучкості, надійності і економічності при забезпеченні якості і точності отримуваних виробів у даний час пов'язано з науково обґрунтованим вибором оптимальних інструментальних матеріалів, геометричних параметрів інструментів і режиму різання, впровадженню систем контролю за станом інструменту і так далі. Досвід сучасного машинобудування показує, що найбільша ефективність обробки матеріалів різанням може бути досягнута при поєднанні і безперервному виконанні декількох послідовних операцій на одному верстаті, що працює в автоматичному режимі. Часто це пов'язано з реалізацією нестационарних процесів різання. Одним з таких процесів є попутне тангенціальне точіння (ПТТ), яке, завдяки його кінематичним і фізичним особливостям, дозволяє поєднати на одному верстаті чорнову і чистову обробку шляхом реалізації роторного принципу її організації, що істотно підвищує продуктивність обробки і зменшує кількість одиниць устаткування. Використання ПТТ є перспективним.

Проте, ми зіткнулися з не великою кількістю рекомендацій з призначення раціональних параметрів обробки в умовах ПТТ –

інструментального матеріалу, геометричних і конструкторських елементів інструменту, параметрів режиму різання. Крім того, досвід експлуатації устаткування, в якому реалізоване ПТТ, свідчить про підвищену інтенсивність відмов ріжучого інструменту. Вирішення проблеми пов'язане з оптимізацією цього нестационарного процесу різання. При оптимізації процесу експлуатації ріжучого інструменту, як правило, доводиться вирішувати суперечливу за своїм змістом задачу багатокритеріального вибору [2]: оптимальні параметри механічної обробки повинні одночасно задовольняти сукупності економічних і технологічних критеріїв, що мають різну природу і призначення. Вибір оптимального режиму різання пов'язаний з ухваленням рішень. Для цього використовують різні методи, у тому числі і евристичний.

На сьогоднішній день можна говорити про стрімкий розвиток евристичних методів. Розроблена величезна кількість алгоритмів функціонування і вчення різних нейронних мереж, ведуться роботи із створення алгоритмів оптимізації їх структури з метою підвищення швидкодії, якості результатів, зменшення часу вивчення. Ці методи знайшли вживання в економіці, медицині, промисловості, багатьох інших галузях науки і техніки, здатні вирішувати практично будь-які завдання, пов'язані з моделюванням, прогнозуванням, оптимізацією [3].

Велика частина досліджень в області генетичних алгоритмів доводиться на завдання машинобудування [4]. Ці алгоритми дають універсальний спосіб вирішення багатьох нестандартних завдань, здатні уловити складні нелінійні залежності, самоудосконалюватися, вивчаючись в процесі використання. Використання побудованих на їх основі систем порівняно ефективно, просто і дешево. Тому, зростаючий інтерес в цій технології привів до значного розширення сфери її використання, появи множини різних підходів, алгоритмів вчення, програмних продуктів, варіантів застосування у машинобудуванні.

## 1. Особливості механічної обробки різанням за допомогою тангенціального точіння

Пошуки нових методів обробки різанням, що дозволяють підвищити продуктивність із забезпеченням вимог до точності і якості оброблюваних поверхонь, є однією із головних задач технології машинобудування. Одним з таких методом обробки є точіння з тангенціальною подачею, яке дає найбільший ефект при обробці «коротких» поверхонь тіл обертання, зокрема деталей типу кілець, дисків, блоків шестерень, фланців, втулок і тому подібне, тобто деталей, що відносяться до класу «диски». Найбільш широке вживання процес отримав в автомобільній промисловості для обробки шийок колінчастих валів, розроблено устаткування для обробки кілець підшипників, зубчастих коліс, шківів і ін. деталей машин [5].

Відомо шість видів тангенціального точіння (рис. 1.1): з прямолінійною подачею (а), з круговою подачею за схемою зовнішнього торкання (б), з круговою подачею за схемою внутрішнього торкання (в). Умовою можливого здобуття безперервної стружки за зворот заготовки, тобто здійснення процесу точіння, а, наприклад, не фрезерування, є таке співвідношення швидкості і подачі, яке більше відношення довжини кола оброблюваної поверхні до довжини контакту різця з деталлю за цикл обробки:

$$V/S = \pi D / l \quad (1.0)$$

Тангенціальне точіння здійснюється при двох рухах: швидкому обертанні заготовки, що визначає швидкість головного руху різання  $D_r$  і повільній тангенціальній подачі інструменту по траєкторії, дотичній до обробленої поверхні. Якщо у момент виходу інструменту з контакту із заготовкою (точка В) напрями векторів швидкостей різання  $\bar{V}$  і подачі  $\bar{V}_s$  збігаються (напрям руху подачі  $D_s$ ) – це точіння називається попутним, якщо ці вектори направлені в протилежні сторони ( $D_s'$ ) – зустрічним (рис. 1.2). При цьому, початком контакту інструменту із заготовкою при зустрічному тангенціальному точінні є точка Е, а точка А – при попутному.

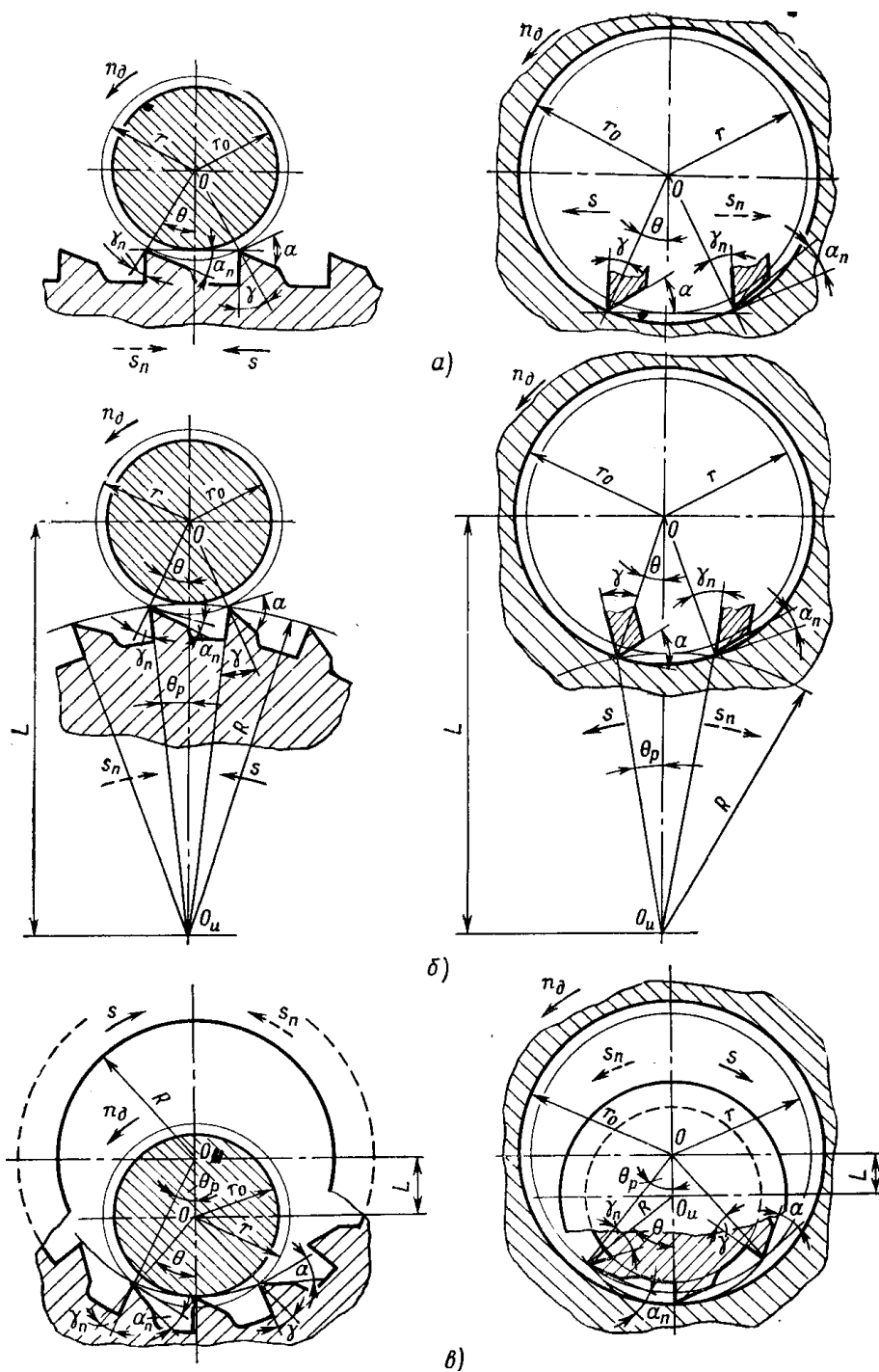


Рисунок 1.1 - Схеми тангенціального точіння зовнішніх і внутрішніх поверхонь[9]:

а) – з прямолінійною подачею; б) – з круговою подачею за схемою зовнішнього торкання; в) – з круговою подачею за схемою внутрішнього торкання ( $S$  - подача при зустрічному точінні,  $S_n$  - при попутному;  $\gamma, \alpha$  - кути різця при зустрічному точінні,  $\gamma_n, \alpha_n$  - при попутному;  $r$  – розмір оброблюваної поверхні,  $r_0$  – обробленої;  $\Theta$  - кут контакту заготівки з різцем,  $\Theta_p$  – кут контакту різця із заготівкою).

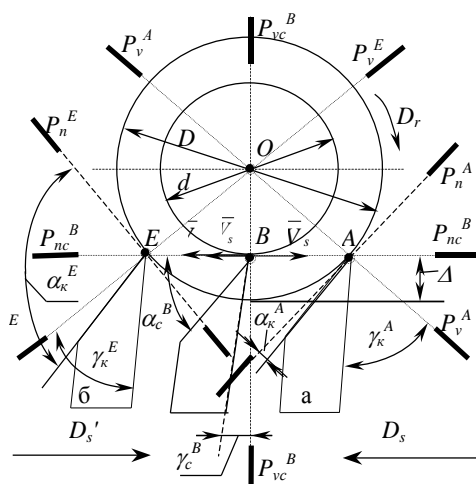


Рисунок 1.2 - Схеми тангенціального точіння [9]:

а) попутного; б) зустрічного;

Для роторного тангенціального точіння валів з подовжньою подачею розробили схеми обробки і конструкції токарних верстатів [6]. Принцип роторного тангенціального точіння за схемою зовнішнього торкання реалізований на вертикальному 12-шпindelному автоматі. Однією з областей використання даного оброблювального устаткування є великосерійне і масове виробництво. Проте дослідний зразок автомата показав відмови різців, які відбувалися унаслідок руйнування їх ріжучої частини у вигляді викрашувань і сколів. Подальші розробки і впровадження у виробництво високопродуктивних верстатів-автоматів роторного типу з тангенціальною схемою різання були припинені. Причиною тому послужила відсутність науково обґрунтованих рекомендацій в організації ефективної роботи ріжучого інструменту на цьому устаткуванні. Аналіз наявної в технічній літературі інформації про проблеми практичної реалізації тангенціального точіння показує, що до справжнього моменту накопичена значна база даних для створення і проектування прогресивних високопродуктивних оброблювальних систем з тангенціальною схемою різання. У реальних виробничих умовах із-за недостатнього вивченого власне процесу тангенціального точіння (ТТ) практично відсутні які-небудь відомості і рекомендації за призначенням оптимальних інструментальних матеріалів, геометричних параметрів інструменту і параметрів режиму

різання [8]. Досвід експлуатації устаткування, в якому реалізоване ПТТ, свідчить про підвищену інтенсивність відмов ріжучого інструменту. Вирішення проблеми розрахунку і призначення необхідного числа інструментів у багатоінструментальній наладці для цього нестационарного процесу різання пов'язане з комп'ютерним моделюванням кінематики формоутворення і зрізання припуску.

На рисунку 1.3 приведена кінематична схема ТТ з прямолінійною попутною подачею. В процесі різання траєкторією відносного руху інструменту є циклоїдна крива. Специфічні особливості ПТТ: короткочасність контакту інструменту із заготовкою не стаціонарність процесу різання за рахунок безперервної трансформації кутів різання і товщини зрізу, наявність ділянок прослизання із значним їх питомою вагою в циклі обробки. Це обумовлено тією обставиною, що в частині циклу «різання» можливі випадки, коли товщина зрізу настільки мала, що різання, як процес відділення стружки, не відбувається [9]. В цьому випадку матеріал заготовки, пластично деформується, як би підминається під задню поверхню. Ці особливості дозволяють розглядати ПТТ з позиції переривистого, короткочасного, періодичного різання, характерного для фрезерування, стругання, проточування і інших процесів.

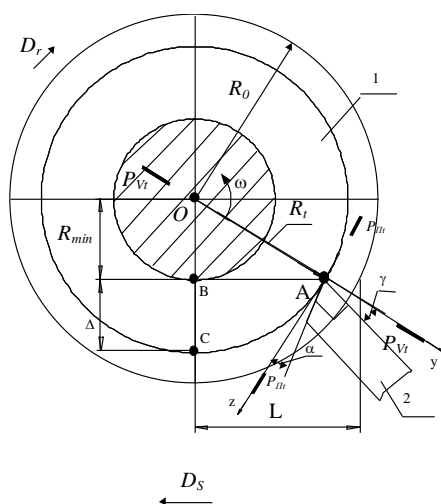


Рисунок 1.3. Кінематична схема процесу тангенціального точіння з попутною прямолінійною подачею [9].

1 – Заготовка; 2- Інструмент.



За час переміщення інструменту (заготовки) в русі подачі ( $D_S$  чи  $D_S'$ ) із заготівки діаметром  $D$  видаляється припуск  $\Delta$  і формується деталь із розміром  $d=D - 2\Delta$ . В процесі різання траєкторія заданої точки ріжучої кромки є циклоїдною кривою, і тому товщина зрізу  $a(\varphi)$  теоретично визначається за формулою:

$$a(\omega) = R_t(\omega - 2\pi) - R_t(\omega), \quad (1.1)$$

де  $R_t(\omega) = \sqrt{R_{\min}^2 + \left(L - \frac{S_0\omega}{2\pi}\right)^2}$  - поточний радіус заготівки в точці А;

$R_0$  - радіус заготівки;

$R_{\min} = R_0 - \Delta$  - радіус заготівки після обробки;

шлях, пройдений вершиною леза за цикл обробки:  $L = \sqrt{R_0^2 - R_{\min}^2}$ ; (1.2)

$\omega$  - кут, відповідний повороту миттєвої основної площини  $P_V$  за час контакту інструменту із заготівкою;

$S_0$  - тангенціальна подача, мм/об. Умовно прийнято, що при  $\omega < 0$   $R_t = R_0$ .

У формулі (1.1) в якості аргументу прийнятий кут повороту  $\omega$  миттєвої основної площини. Проте часто для аналізу залежностей зручно використовувати поточний зворот заготівки  $n_t$ . Тоді зв'язок між  $\varphi$  і  $n_t$  визначається вираженням:

$$n_t = \frac{\omega - \omega_{\text{нпр}}}{2\pi}; \quad (1.3)$$

де  $\omega_{\text{нпр}}$  - поправочний коефіцієнт за формулою:

$$\omega_{\text{нпр}} = \arccos \frac{R_t(\omega - 2\pi)^2 + R_t(\omega)^2 - S_0^2}{2R_t(\omega - 2\pi)R_t(\omega)}; \quad (1.4)$$

Часто поправкою  $\omega_{\text{нпр}}$  можна нехтувати. В цьому випадку формула (1.3) набере вигляду:

$$n_t = \frac{\omega}{2\pi}; \quad (1.5)$$

Кількість зворотів, необхідна для зняття всього припуску дорівнює:

$$n_c = \frac{L}{S_0}; \quad (1.6)$$

При цьому час обробки складає:

$$\tau = \frac{n_c}{n}, \quad (1.7)$$

де  $n$  – частота обертання заготовки.

Максимальна товщина зрізу досягається в кінці першого звороту заготовки ( $\omega_{\max} = 2\pi$ ) і розраховується за формулою:

$$a_{\max} = a(2\pi) = R_0 - \sqrt{R_{\min}^2 + (L - S_0)^2}, \quad (1.8)$$

Кінематичні передній  $\gamma_K(\omega)$  і задній  $\alpha_K(\omega)$  кути при ПТТ розраховуються за формулами:

$$\gamma_K(\omega) = \psi(\omega) - \gamma, \quad \alpha_K(\omega) = \psi(\omega) + \alpha, \quad (1.9)$$

де  $\psi(\omega) = \arccos \frac{R_{\min}}{R_t(\omega)}$  - кут трансформації;  $\gamma, \alpha$  – статичні передній і

задній кути (відповідають значенням кутів леза у момент його виходу з контакту із заготовкою).

Як видно з цієї формули (1.9) для попутного тангенціального точіння характерне зменшення кінематичного переднього кута і збільшення кінематичного заднього кута.

Таким чином, розробка компоновок з тангенціальною схемою різання є перспективним напрямом розвитку сучасних конструкцій вертикальних оброблювальних центрів. Для розширення технологічних можливостей вживання методу ПТТ може бути доповнено рухом, направленим уздовж осі деталі. Завдяки осьовій подачі об'єм металу, що знімається кожним різцем, можна збільшити у декілька разів, при цьому час роботи одним інструментом не змінюється, а максимальна сила різання залишається без зміни. Поєднання

осьової і тангенціальної подач забезпечує високу точність обробки складних поверхонь (наприклад, сферичні і конічні).

Однією з причин низької стійкості інструменту при ПТТ є товщина зрізу, що виникає унаслідок нерівномірної зміни, складових сили різання і геометрії інструменту вібрації.

Істотний вплив на стійкість інструменту в процесі різання надають твердість оброблюваних матеріалів, яка залежно від умов різання може як трохи, так і істотно відрізнитися від твердості матеріалу у вихідному стані і впливати на механізм зношування інструменту. У загальному випадку твердість обробленої поверхні визначається рівнем силової (зміцнення) і теплової (зниження міцності) дії на неї. При тангенціальному точінні істотним чинником, сприяючим збільшенню твердості, є місце, що має, на початку і кінці циклу прослизання, величина якого значною мірою визначається умовами обробки.

В результаті складного впливу кінематичних і фізичних особливостей процесу тангенціального точіння на механізм і закономірності зношування і руйнування інструменту в процесі різання, традиційні методи дослідження, моделювання і прогнозування показників працездатності інструменту, які добре розроблені для інших процесів механічної обробки різанням, для даного методу виявляються мало ефективними. У зв'язку з цим проблема підвищення працездатності ріжучого інструменту в умовах ПТТ є актуальною.

## **2. Метод вибору оптимальних режимів різання за допомогою генетичного алгоритму**

Проблема полягає в тому, що, як правило, виробничі процеси характеризуються величезною різноманітністю динамічно взаємодіючих параметрів і зазвичай дуже складні для створення адекватних аналітичних моделей [10]. До того ж на сучасному етапі розвитку виробництва відбувається постійне ускладнення завдань, що вимагають рішення, що, по-перше, віднімаючи значний час на розробку і перевірку аналітичних моделей, приводить до збільшення загального часу впровадження нових технологічних рішень, по-друге, часто перетворює дану систему на так званий «чорний ящик», тобто систему, для якої відсутня аналітична залежність між вхідними і вихідними параметрами (відгуком) системи. При виборі оптимальних режимів різання вдалим рішенням може стати використання генетичних алгоритмів.

**Генетичний алгоритм (ГА)** — це евристичний алгоритм пошуку, використовуваний для вирішення завдань оптимізації і моделювання шляхом випадкового підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Є різновидом еволюційних обчислень. Відмітною особливістю генетичного алгоритму є акцент на використання оператора «схрещування», який виробляє операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування в живій природі [11].

Генетичні алгоритми в різних формах застосовувалися до багатьох наукових і технічних проблем. Вони використовувалися при створенні інших обчислювальних структур, наприклад, автоматів або мереж сортування. У машинному вченні генетичні алгоритми використовувалися при проектуванні нейронних мереж або управлінні роботами. Вони також застосовувалися при моделюванні розвитку в різних наочних областях,

включаючи біологічні (екологія, імунологія і генетика популяції), соціальні (такі як економіка і політичні системи) і когнітивні системи [12].

Проте, можливе найбільш популярне вживання генетичних алгоритмів – оптимізація багатопараметричних функцій [13]. Багато реальних задач можуть бути сформульовані як пошук оптимального значення, де значення – складна функція, залежна від деяких вхідних параметрів. В деяких випадках, представляє інтерес знайти ті значення параметрів, при яких досягається найкраще точне значення функції. У інших випадках, точний оптимум не потрібний – рішенням може вважатися будь-яке значення, яке краще за деяку задану величину. В цьому випадку, генетичні алгоритми – часто найбільш прийнятний метод для пошуку "хороших" значень. Сила генетичного алгоритму поміщена в його здатності маніпулювати одночасно багатьма параметрами, ця особливість ГА використовувалося в сотнях прикладних програм, включаючи проектування літаків, налаштування параметрів алгоритмів і пошуку стійких станів систем нелінійних диференціальних рівнянь. Ефективність генетичного алгоритму сильно залежить від таких деталей, як метод кодування рішень, оператори, налаштування параметрів, приватний критерій успіху.

**Основний опис.** План основних дій генетичного алгоритму – загальний (рис. 2.1). Є багато речей, які можуть бути реалізовані по-різному в різних завданнях.

З отриманої безлічі рішень («покоління») з врахуванням значення «пристосованості» вибираються рішення (зазвичай кращі особини мають велику вірогідність бути вибраними), до яких застосовуються «генетичні оператори» (в більшості випадків «схрещування» — crossover і «мутація» — mutation), результатом чого є здобуття нових рішень. Для них також обчислюється значення пристосованості, і потім виробляється відбір («селекція») кращих рішень в наступне покоління.

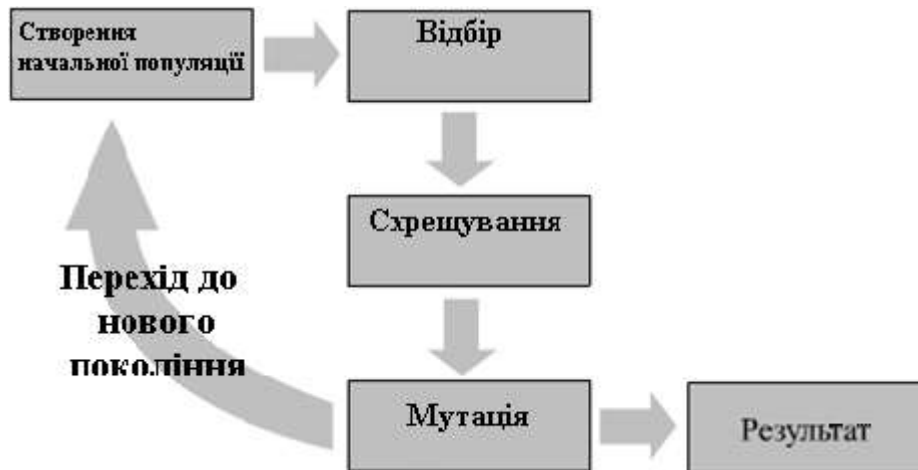


Рисунок 2.1 – Схема роботи генетичного алгоритму

Цей набір дій повторюється ітеративно, так моделюється «еволюційний процес», що продовжується декілька життєвих циклів (поколінь), поки не буде виконаний критерій зупинки алгоритму. Таким критерієм може бути [14]:

- вичерпання числа поколінь, відпущених на еволюцію;
- вичерпання часу, відпущеного на еволюцію;
- знаходження глобального, або субоптимального рішення;

Генетичні алгоритми служать, головним чином, для пошуку рішень в багатовимірних просторах пошуку.

Є два основних параметри ГА - вірогідність кросовера і вірогідність мутації. Існують також деякі інші параметри генетичного алгоритму. Одним з цих параметрів є чисельність населення.

**Кодування значень** – одна з проблем вирішення завдань за допомогою ГА. Вибрати певний спосіб кодування важко, оскільки специфіка завдання накладає свої обмеження і вимоги на спосіб представлення інформації. При цьому немало поважно, що від вибраного способу кодування можуть змінитися генетичні оператори. Для прикладу досить порівняти операторів кросинговера для *цілочисельного* і *дійсного* кодування [15].

*Цілочисельне кодування* [16]. У канонічному генетичному алгоритмі хромосома є бітовим рядком, в якому закодовані параметри рішення

поставленої задачі. На *рис. 2.2* показаний приклад кодування 4-х 10-розрядних параметрів у 40 розрядній хромосомі.



*Рисунок 2.2 – Приклад цілочисельного кодування*

Не дивлячись на те, що кожен параметр закодований в хромосомі цілим числом, йому можуть бути поставлені у відповідність і дійсні числа.

Нижче представлений один з варіантів прямого і зворотного перетворення «цілочисельний ген  $\rightarrow$  дійсне число».

Якщо відомий діапазон, в межах якого лежить значення параметра, то цей діапазон розбивають на  $2^m$  відрізків, де  $m$  – розрядність гена, і кожному відрізку відповідає певне значення гена. При цьому для переведення значень із закодованого значення в дробі застосовують наступні формули:

$$r = (g * (x_{max} - x_{min}) / (2^m - 1)) + x_{min},$$

$$g = ((r - x_{min}) * (2^m - 1)) / (x_{max} - x_{min}),$$

де  $r$  – дійсне (декодоване) значення параметра,  $g$  – цілочисельне (закодоване) значення параметра,  $x_{max}$  та  $x_{min}$  – відповідно максимальне і мінімальне допустиме значення декодованого параметра.

Наприклад, якщо шукане значення параметра лежить у проміжку [1;2], і кожен ген кодується 16 розрядами, то, якщо вміст гена рівний ABCD16 = 4398110, то відповідне дробове значення рівне:

$$r = 43981 * (2 - 1) / (2^{16} - 1) + 1 = 0,6711 + 1 = 1,6711$$

Якщо ж декодоване значення дорівнює 1,3275, то відповідний ген після зворотного перетворення міститиме (з округленням в меншу сторону):

$$g = (1,3275 - 1) * (2^{16} - 1) / (2 - 1) = 21462,7125 \approx 2146210 =$$

$$0101\ 0011\ 1101\ 01102$$

*Дійсне кодування [16].* Зручно кодувати в гені не ціле число, а дійсне. Це дозволяє позбавитися від операцій кодування/декодування,

використовуваних в цілочисельному кодуванні, а також збільшити точність знайденого рішення. Приклад дійсного кодування представлений на рис.2.3.



Рисунок 2.3 – Приклад дійсного кодування

**Селекція (відбір)** необхідна, щоб вибрати більш пристосовані особини для схрещування. Існує безліч варіантів селекції, опишемо найбільш відомі з них [17].

*Селекція «рулетка».* У даному варіанті селекції вірогідність  $i$ -ї особи взяти участь у схрещуванні  $p_i$  пропорційна значенню її пристосованості  $f_i$  і дорівнює:  $p_i = f_i / \sum f_i$

Процес відбору особин для схрещування нагадує гру в «рулетку». Круг рулетки ділиться на сектори, причому площа  $i$ -го сектора пропорційна значенню  $p_i$ . Після цього  $n$  раз «обертається» рулетка, де  $n$  – розмір популяції, і за сектором, на якому зупиняється рулетка, визначається особа, вибрана для схрещування.

*Селекція відсіканням.* При відборі відсіканням після обчислення значень пристосованості для схрещування вибираються  $ln$  кращих особин, де  $l$  – «поріг відсічення»,  $0 < l < 1$ ,  $n$  – розмір популяції. Чим менше значення  $l$ , тим сильніше тиск селекції, тобто менші шанси на виживання у погано пристосованих особин. Як правило, вибирають  $l$  в інтервалі від 0,3 до 0,7.

*Турнірний відбір.* В разі використання турнірного відбору для схрещування, як і при селекції рулетки, відбираються  $n$  особин. Для цього з популяції випадково вибираються  $t$  особин, і сама пристосована з них допускається до схрещування. Турнір формується з  $t$  особин,  $t$  – розмір турніру. Ця операція повторюється  $n$  раз. Чим більше значення  $t$ , тим більше тиск селекції. Варіант турнірного відбору, коли  $t = 2$ , називають бінарним турніром. Типові значення розміру турніру  $t = 2, 3, 4, 5$ .



Слідом за відбором відбувається **кросовер**. Для цілочисельного кодування часто використовують 1-точковий, 2-точковий і однорідний оператори кросоверу. 1-точковий кросовер працює аналогічно операції перехрещення для хромосом при схрещуванні біологічних організмів. Для цього вибирається довільна точка розриву і для створення нащадків виробляється обмін частями батьківських хромосом. Ілюстративний приклад роботи 1-точкового кросовера зображений на рис. 2.4.

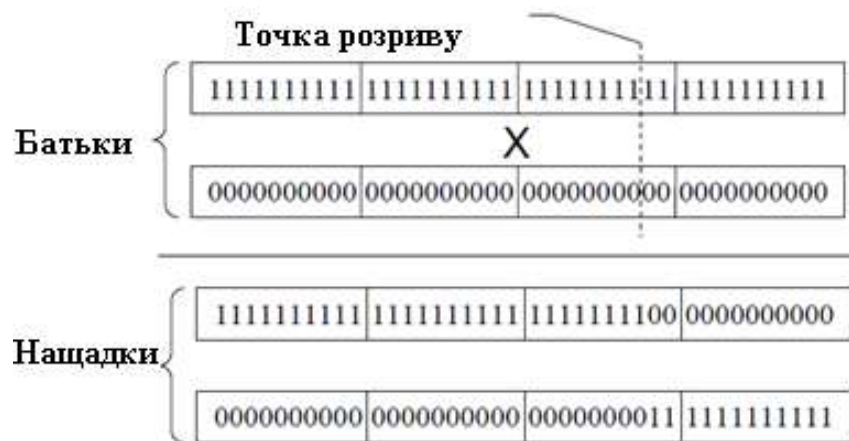


Рисунок 2.4 – Приклад роботи 1-точкового кросовера.

Для оператора 2-точкового кросовера вибираються 2 випадкових точки розриву, після чого для створення нащадків батьківські хромосоми обмінюються ділянками, лежачими між точками розриву.

При використанні однорідного оператора кросовера розряди батьківських хромосом успадковується незалежно один від одного. Для цього визначають вірогідність  $p_0$ , що  $i$ -й розряд хромосоми 1-го батька попаде до першого нащадка, а 2-го батька – до другого нащадка. Вірогідність протилежної події рівна  $(1 - p_0)$ . В більшості випадків вірогідність обох подій однакова, тобто  $p_0 = 0,5$ .

Для дійсного кодування розглянемо 2-точковий, арифметичний і *BLX- $\alpha$*  оператори кросовера.

2-точковий кросовер для дійсного кодування, в цілому, аналогічний 2-точковому кросоверу для цілочисельного кодування. Відмінність полягає в

тому, що точка розриву не може бути вибрана «усередині» гена, а повинна попасти між генами (рис. 2.5).

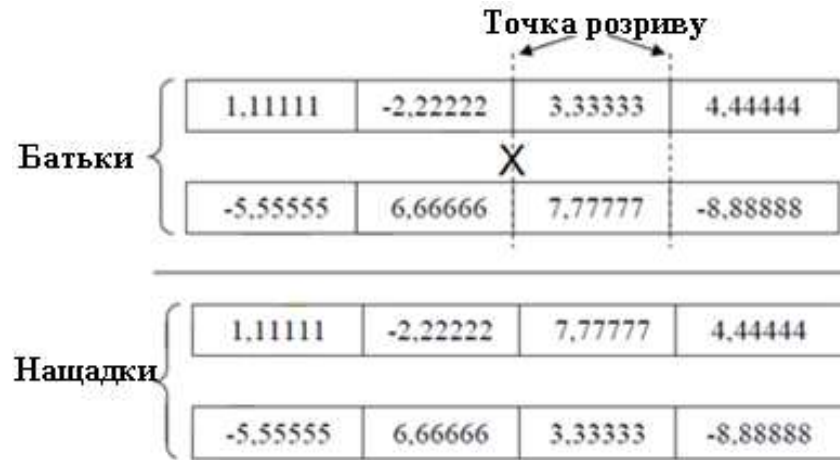


Рисунок 2.5 – Приклад роботи 2-точкового кросовера для дійсного кодування

При використанні арифметичного і  $BLX-\alpha$  операторів обмін інформацією між батьківськими особинами і нащадками виробляється з врахуванням значень генів батьків.

Позначимо  $gk^1$  і  $gk^2$  –  $k$ -і гени батьківських особин,  $1 \leq k \leq N$ ,  $N$  – кількість генів в хромосомі. Нехай також  $hk^1$  і  $hk^2$  –  $k$ -і гени нащадків. Отже, ми бачимо формули для арифметичного кросовера:

$$hk^1 = \lambda c + (1 - \lambda) * gk^2,$$

$$hk^2 = \lambda gk^2 + (1 - \lambda) * gk^1,$$

де  $0 \leq \lambda \leq 1$ .

Якщо використовується  $BLX-\alpha$  кросовер, то значення  $k$ -го гену нащадка вибирається випадковим чином (рівномірний розподіл) з інтервалу:

$$[cmin - \alpha \Delta k, cmax + \alpha \Delta k],$$

де  $\alpha - \text{const}$ ,

$$cmin = \min\{ gk^1, gk^2 \},$$

$$cmax = \max\{ gk^1, gk^2 \},$$

$$\Delta k = cmax - cmin.$$

Зображення інтервалу, використовуваного для  $BLX-\alpha$  кросовера показано на рис. 2.6.

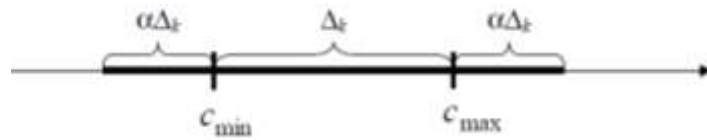


Рисунок 2.6 – Інтервал для BLX-α кросовера

Для появи нової популяції одного кросовера не вистачає, тому відбувається **мутація**. Вона потрібна для внесення випадкових змін до хромосом особин. Це дозволяє «вибиратися» з локальних екстремумів і, тим самим, ефективніше досліджувати простір пошуку. Так само, як і для оператора кросовера, існує вірогідність вживання мутації  $PM$ .

Існують різні оператори мутації залежно від способу представлення генетичної інформації.

*Цілочисельне кодування.* Опишемо оператор бітової мутації. У разі цілочисельного кодування мутація змінює окремі розряди в хромосомі. Для цього кожен розряд інвертується з вірогідністю  $PM$ . Нижче наведений приклад мутації на псевдомові:

```

ДЛЯ КАЖДОЙ k ОСОБИ А ПОПУЛЯЦИИ {
    ДЛЯ КАЖДОГО i РАЗРЯДА В ХРОМОСОМЕ k {
        ЕСЛИ (PM > RANDOM) {
            БИТОВАЯ_МУТАЦИЯ (ОСОБЬ [k], i);
        }
    }
}

```

Внаслідок того, що вживання мутації розігрується стільки раз, скільки розрядів міститься в хромосомі, значення  $PM$  вибирають невеликим, щоб сильно не руйнувати знайдені хороші хромосоми. Один з типових варіантів  $PM = L^{-1}$  де  $L$  – довжина хромосоми в бітах, в цьому випадку кожна хромосома мутує в середньому один раз.

*Дійсне кодування.* Мутація для дійсного кодування змінює вміст кожного гена з вірогідністю  $PM$ . При цьому величина зміни вибирається

випадково в деякому діапазоні  $[-\xi; +\xi]$ , наприклад,  $[-0,5; 0,5]$ , і може мати як рівномірне, так і будь-який інший розподіл, наприклад, нормальний з  $m = 0$ ,  $\sigma = 0,5$ . Таким чином, приклад мутації для дійсного кодування на псевдомові виглядає таким чином (RND – випадкове число, розподілене по заздалегідь визначеному закону):

```

ДЛЯ КАЖДОЙ k ОСОБИ В ПОПУЛЯЦИИ {
    ДЛЯ КАЖДОГО i ГЕНА В ХРОМОСОМЕ k {
        ЕСЛИ (PM > RANDOM) {
            ОСОБЬ [k]. ГЕН [i] = ОСОБЬ [k]. ГЕН [i] + RND;
        }
    }
}

```

Для того, щоб уникнути сильних змін вмісту хромосоми в результаті мутації значення вірогідності  $PM$  ми обрали не великим. Наприклад,  $PM = N^{-1}$ , де  $N$  – кількість генів у хромосомі.

Зрештою, хотілося б сказати, що ми прийшли до виводу про можливе використання генетичних алгоритмів для вибору оптимальних режимів різання. Також, ми переконалися, що ГА мають безліч модифікацій і сильно залежать від параметрів. Частенько, невелика зміна одного з них може привести до несподіваного поліпшення результату. Значить, вживання ГА корисне лише тоді, коли для даного завдання немає відповідного спеціального алгоритму рішення.

### 3. Результати дослідження

*Математична модель.* Мета цієї моделі полягає в мінімізації собівартості продукції, яка включає в себе вартість обробки, вартість заміни інструменту і витратах на інструмент. Формула для розрахунку вартості використовується з роботи Chen і співавт. [19].

Цільова функція - одиниця собівартості продукції UC (\$ / шт), яка може бути представлена наступним чином:

$$UC = C_M + C_R + C_T = k_0 T_M + k_0 \left( t_e \frac{T_M}{t_l} \right) + k_t \frac{T_M}{t_l}, \quad (3.0)$$

де  $C_M$  - собівартість обробки одиниці продукції, \$/шт;

$C_R$  - вартість заміни інструменту на одиницю продукції, \$/шт;

$C_T$  - вартість інструменту на одиницю продукції, \$/шт;

$k_0$  - коефіцієнт прямих витрат на робочу силу і накладні витрати, \$/шт;

$T_M = \frac{2\pi R_0 \sqrt{2R_0 \Delta - \Delta^2}}{S_0 V 1000}$  - час обробки одиниці продукції, хв;

$t_l$  - час роботи інструменту, хв;

$t_e$  - час на заміну інструменту, хв;

$k_t$  - вартість ріжучої твердосплавної пластини, \$/шт;

*Обмеження математичної моделі.* Практичні обмеження для оптимізації сформульовані наступним чином [19]:

— Швидкість різання  $V_{\min} \leq V \leq V_{\max}$

— Тангенціальна подача  $S_{0 \min} \leq S_0 \leq S_{0 \max}$

— Припуск  $\Delta_{\min} \leq \Delta \leq \Delta_{\max}$

— Час роботи інструменту  $t_{l \min} \leq t_l \leq t_{l \max}$

Наступні обмеження моделі отримані шляхом використання функції оклику у вигляді поліному другого степеню [9]:

$$Y = B_0 + B_1 X_1 + B_2 X_2 + B_3 X_3 + B_{12} X_1 X_2 + B_{13} X_1 X_3 + B_{23} X_2 X_3 + B_{11} X_1^2 + B_{22} X_2^2 + B_{33} X_3^2, \quad (3.1)$$

— Сила різання

$$P_z = 1186 - 39V + 445S_0 + 270\Delta + 6VS_0 + 30S_0\Delta + 43V^2 - 25S_0^2 - 2\Delta^2 \leq P_{z \max}$$

$$P_y = 751 - 57V + 89S_0 + 226\Delta + 41VS_0 - 91V\Delta - 133S_0\Delta + 111V^2 + 67S_0^2 + 14\Delta^2 \leq P_{y \max}$$

— Точність форми

$$\delta = 52.582 - 0.92V - 0.312S_0 + 1.623\Delta + 7VS_0 - 3.25V\Delta - 3.5S_0\Delta - 5.6V^2 - 1.9S_0^2 - 1.54\Delta^2 \leq \delta_{\max}$$

— Середню температуру контакту

$$\Theta = 1307 + 326.25V + 127.35S_0 + 61.37\Delta - 125.42V^2 - 57.78S_0^2 - 66.28\Delta^2 \leq \Theta_{\max}$$

Таблиця 3.1

Параметр/ константа	Значення	Параметр/ константа	Значення
$V_{\min}$ , м/хв	86	$V_{\max}$ , м/хв	350
$S_{0 \min}$ , мм/об	0.5	$S_{0 \max}$ , мм/об	1.5
$\Delta_{\min}$ , мм	0.5	$\Delta_{\max}$ , мм	1.5
$t_l \min$ , хв	25	$t_l \max$ , хв	45
$P_z \max$ , Н	2000	$P_y \max$ , Н	1200
$\delta_{\max}$ , мкм	55	$\Theta_{\max}$ , °С	1300
$k_0$ , \$/шт	2	$R_0$ , мм	30
$t_e$ , хв	1.5	$k_t$ , \$/шт	15

*Параметри генетичного алгоритму.* Для реалізації обраної математичної моделі, шляхом тестувань і вивчення літератури [20] [21] [22], були обрані такі параметри генетичного алгоритму:

- розмір популяції 20;
- довжина хромосом (бінарна) 40;
- метод селекції рулетка;
- метод кроссовера двох точковий;
- ймовірність кроссовера 0.9;
- ймовірність мутації 0.01.

*Результати.* Запропонована методика була реалізована у програмі “GA”, виконаній у середовищі об’єктно-орієнтованого програмування DELFY. Спрощена блок схема програми наведена на рисунку 3.1, а текст виконуючої процедури у додатку.

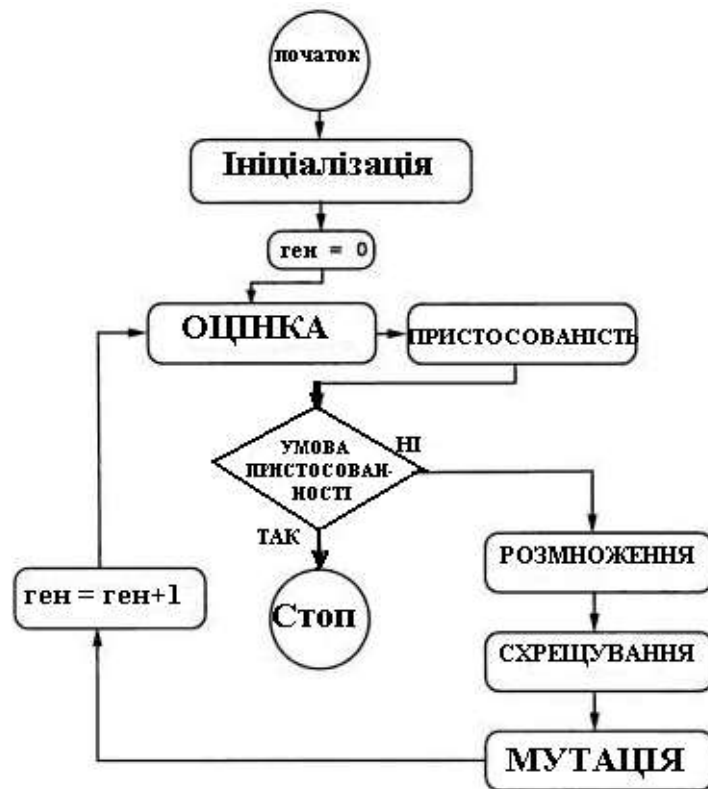


Рисунок 3.1- Блок схема програми “GA”.

Приклад визначення оптимальних режимів різання наведений для обробки заготовки діаметром  $D=60$  мм із сталі ШХ15 твердосплавними пластинами Т15К6. Для заданих умов обробки параметри та константи [9][21], необхідні для обчислень, занесені до таблиці 3.1

Результати обчислень для заданих умов наведені у таблиці 3.2.

Таблиця 3.2 Результати обчислень

V, м/мин	S, мм/об	$\Delta$ , мм	Ціна \$
336.433	1.462	1.069	5.617

**ВИСНОВКИ:** завдяки використанню стохастичного метода пошуку – генетичного алгоритма, вдалося розробити програмне забезпечення для вирішення складного завдання по знаходженню коренів нелінійної функції з обмеженнями у вигляді також нелінійних рівнянь, а саме - визначення режимів обробки тангенційного точіння з попутною подачею виходячи з собівартості обробки одиниці продукції.

## Список літератури:

1. Старков В.К. Физика и оптимизация резания материалов. М.: Машиностроение, 2009. – 640 с.
2. Залога В.А., Приходько В.В., Гребенюк А.Г. Методика исследования сил резания при попутном тангенциальном резании // Вестник Сумского государственного университета. – Сумы: СумГУ. - 1996. – №5. – С.44–49.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - М., 1992. – 118 с.
4. Adlemo A., Andréasson S.-A., Balanced automation in flexible manufacturing systems, International Journal of Studies in Informatics and Control. - Vol. 5, No. 2, June 1996, - P. 179-187.
5. Pakucs Janos. Hazai termelo vallalatok innovacios szolgaltatasai igenyeivel kapcsolatos felmeres // GEPGYARTASTECHNOLOGIA A MUSZAKI FOLYOIARATA, XXXIII. EVFOLYAM (венгр.). -1993. №5-6. -S. 261-268
6. Хмельницкий Ю.В., Залога В.А., Киктенко А.Н. Точность геометрической формы и шероховатость поверхности, обработанной попутным точением // Резание и инструмент: Респ.межвед. науч.-техн. сборник –Х: Вища школа. Изд-во при Харьк. ун-те, 1983. Вып. 30. - 34-38 с.
7. Залога В.А., Хмельницкий Ю.В., Парфененко С.Г. Расширение технологических возможностей многошпиндельных роторных автоматов попутного точения // GEPGYARTASTECHNOLOGIA A MUSZAKI FOLYOIARATA, XXXIV. EVFOLYAM (венгр.). –1994. -№ 7-8.-S. 335-339
8. Каневцев В. М. Исследование технологических возможностей тангенциального точения. Автореф. дис. ... канд. техн. наук:05.03.01. -М: МВТУ им. Баумана, 1954. – 23 с.
9. Дядюра К.А. Повышение работоспособности металлорежущего инструмента при обработке стали ШХ15 в условиях попутного



тангенциального точения: Дис... канд. техн. наук: 05.03.01 / НТУ «ХП». - Харків, 2001. – 248 с.

10. Liao T. W., Chen L. J. Manufacturing Process Modeling and Optimization Based on Multi-Layer Perceptron Network // Journal of Manufacturing Science and Engineering. – 1998. – Vol. 120. – P. 109-119

11. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учебное пособие — 2-е изд.. — М: Физматлит, 2006. – 320 с. — ISBN 5-9221-0510-8.

12. Александров Д. А. Алгоритм муравьиной колонии для задачи о минимальном покрытии. XI междунар. Байкальская школа-семинар Методы оптимизации и их приложения, Труды - т3. Иркутск, 1998. - 17–20 с.

13. Алгоритмы. Методы. Исходники. - <http://algolist.manual.ru/ai/ga/>

14. <http://ru.wikipedia.org>

15. <http://qai.narod.ru/GA/advice.html>

16. [http://qai.narod.ru/Publications/tsoy\\_chapterGA.pdf](http://qai.narod.ru/Publications/tsoy_chapterGA.pdf)

17. <http://www.obitko.com/tutorials/>

18. «Генетические алгоритмы по-русски» - <http://www.chat.ru/~saissa>

19. Chen MC and Su CT (1998) Optimization of machining conditions for turning cylindrical stocks into continuous finished profiles. Int J Prod Res 36(8):2115–2130

20. Dereli T, Filiz IH, Baykasoglu A (2001) Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms. Int J Prod Res 39(15):3303–3328

21. Shanmugham MS, Baskara Reddy SV, Narendran TT (2000) Selection of optimal conditions in multi-pass face milling using a genetic algorithm. Int J Mach Tool Manuf 40:401–414

22. Godfrey C Onwubolu; B V Babu. New optimization techniques in engineering. Berlin [u.a.] : Springer, 2004.

## Додаток А

<pre> unit Unit1;  interface  uses  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;  type  TForm1 = class(TForm)  btn1: TButton;  lbl1: TLabel;  procedure btn1Click(Sender: TObject);  private  { Private declarations }  public  { Public declarations }  end;  var  Form1: TForm1;  implementation  {\$R *.dfm} </pre>	<pre> procedure shuffle(var pop:population); {процедура перемешивания популяции в процессе отбора}  var i,j:integer;  ind0:individual;  begin  for i := popsize downto 2 do begin  j:= random(i-1)+1;  ind0:=pop[i];  pop[i]:=pop[j];  pop[j]:=ind0;  end;  end;  function select_1:integer;  var j1,j2,m:integer;  begin  if (ipick&gt;popsize) then begin  shuffle(oldpop);  ipick:=1  end; </pre>
---	--

```

procedure TForm1.btn1Click(Sender:
TObject);const
maxpop = 100;
maxstring = 30;
dim = 1; {размерность пространства
поиска}
type
allele = boolean;
chromosome =
array[1..maxstring*dim] of allele;
{битовая строка}
fenotype = array[1..dim] of real;
{фенотип = массив вещественных
координат точки в пространстве
поиска}
individual = record
chrom:chromosome;      {генотип
= битовая строка}
V:fenotype;
S:fenotype;
Del:fenotype;          {фенотип =
массив вещественных координат
точки в пространстве поиска}
fitness:real;          {значение

```

```

j1:=ipick;
j2:=ipick+1;
if
(oldpop[j2].fitness<oldpop[j1].fitness)
then m:=j2 else m:=j1;
ipick:=ipick+2;
select_1:=m;
end;
var j:integer;
begin
ipick:=1;
for j:=1 to popsize do begin
intpop[j]:=oldpop[select_1];
end;
oldpop:=intpop;
end;
function mutation (allelevel:allele;
pmutation:real; var
nmutation:integer):allele;
{мутация одного бита в строке
(аллеля) с вероятностью pmutation,
count number of mutations}

```

<pre> целевой функции} end;  population = array[1..maxpop] of individual;  const  Vmax:fenotype=(5.12);  Vmin:fenotype=(-5.12);  Smax:fenotype=(5.12);  Smin:fenotype=(-5.12);  Delmax:fenotype=(5.12);  Delmin:fenotype=(-5.12);  var  oldpop, newpop, inpop :population; {Три непересекающихся популяции ? старая, новая и проме-жуточная}  popsize, lchrom, gen, maxgen :integer; {Глобальные целые переменные}  pcross, pmutation, sumfitness :real; {глобальные вещественные переменные}  nmutation, ncross :integer; {Статистические целые}  avg, max, min :real; </pre>	<pre> var  mutate:boolean;  begin  mutate := flip(pmutation); {Flip the biased coin}  if mutate then begin  nmutation := nmutation + 1;  mutation := not alleleval; {Change bit value}  end else mutation := alleleval; {No change}  end;  procedure crossover(var parent1, parent2, child1, child2:chromosome; flchrom:integer; var ncross, nmutation, jcross:integer; var pcross, pmutation:real);  {Скрещивание 2 родительских строк, результат помещается в 2 строках-потомках}  var  j:integer;  begin  if flip(pcross) then begin </pre>
--	--

<pre> {Статистические вещественные}  {Вероятностные процедуры}  function random_:real;  begin  random_:=random(65535)/(65535-1);  end;  function flip(probability:real):boolean; {подбрасывание монетки ? true если орел}  begin  if probability=1.0 then flip:=true else flip:=(random_&lt;=probability);  end;  {Случайный выбор между low и high}  function rnd(low,high:integer):integer;  var i:integer;  begin  if low &gt;= high then  i := low  else begin  i := trunc( random_ * (high-low+1) + </pre>	<pre> {Выполняется скрещивание с вероятностью pcross}  jcross:=rnd(1,flchrom-1); {Определение точки сечения в диапазоне между 1 и l-1}  ncross:=ncross + 1; {Инкрементирование счетчика скрещиваний}  end else  jcross:=flchrom;  {левая часть обмена , 1 to 1 and 2 to 2}  for j := 1 to jcross do begin  child1[j]:=mutation(parent1[j], pmutation, nmutation);  child2[j]:=mutation(parent2[j], pmutation, nmutation);  end;  {вторая часть обмена, 1 to 2 and 2 to 1}  if jcross&lt;&gt;flchrom then  {пропуск, если точка скрещивания равна flchrom--скрещивание не происходит} </pre>
---	---

<pre> low);  if i &gt; high then i := high;  end;  rnd := i;  end;  {интерфейсные процедуры: decode and objfunc}  function objfunc(V,S,Del:fenotype):real;  begin  objfunc:= (2*(((2*30*Del[1]- Del[1]^2)^0.5*3.1415*2*30)/(S[1]*V[ 1]*1000))+2*(1.5/T)*(((2*30*Del[1]- Del[1]^2)^0.5*3.1415*2*30)/(S[1]*V[ 1]*1000)))+(15/T)*(((2*30*Del[1]- Del[1]^2)^0.5*3.1415*2*30)/(S[1]*V[ 1]*1000)));  end;  procedure decode(chrom:chromosome; lbits:integer; var x:fenotype); {Декодирование строки в массив вещественных координат точки в пространстве поиска - true=1, false=0} </pre>	<pre> for j := jcross+1 to flchrom do begin  child1[j] := mutation(parent2[j], pmutation, nmutation);  child2[j] := mutation(parent1[j], pmutation, nmutation);  end;  end;  {Процедура создания нового поколения: generation}  procedure generation;  {Генерирование нового поколения при помощи отбора, скрещивания и мутации}  {Прим: предполагается, что популяция имеет четный размер}  var  j, mate1, mate2, jcross:integer;  begin  select;  j := 1;  repeat  {выполняются отбор, </pre>
---	---

<pre> var i,j:integer;  f, accum, powerof2:real;  begin  for i:=1 to dim do begin  accum:=0.0; powerof2:=1; f:=1;  for j:=1+lbits*(i-1) to lbits+lbits*(i-1) do begin  if chrom[j] then accum := accum + powerof2;  powerof2 := powerof2 * 2;  f:=f*2;  end;  x[i]:=xmin[i]+(xmax[i]- xmin[i])*accum/(f-1)  end  end;  {Расчет статистических величин: statistics }  procedure statistics(popsizе:integer; var max,avg,min,sumfitness:real; var pop:population); {Расчет статистик популяции }  var j:integer; </pre>	<pre> скрещивание и мутация, пока полностью не сформируется новая популяция ? newpop}      mate1:=j;      {выбор родительской пары}      mate2:=j+1;      {скрещивание и мутация ? мутация вставлена в процедуру скрещивания}          crossover(oldpop[mate1].chrom, oldpop[mate2].chrom, newpop[j].chrom, newpop[j + 1].chrom, lchrom*dim, ncross, nmutation, jcross, pcross, pmutation);          {Декодирование строки и вычисление пригодности}          with newpop[j] do begin              decode(chrom, lchrom,x);              fitness := objfunc(x);          end;          with newpop[j+1] do begin              decode(chrom, lchrom,x);              fitness := objfunc(x); </pre>
---	---

<pre> begin   {Инициализация }   sumfitness := pop[1].fitness;   min := pop[1].fitness;   max := pop[1].fitness;   {Цикл для max, min, sumfitness }   for j := 2 to popsize do with pop[j] do   begin     sumfitness := sumfitness + fitness;     {Накопление суммы значений     функции пригодности}     if fitness&gt;max then max := fitness;     {Новое значение max}     if fitness&lt;min then min := fitness;     {Новое значение min}   end;   avg := sumfitness/popsize;   {Расчет среднего} end; {Процедура инициализации initpop} procedure initpop;   {Инициализация начальной   популяции случайным образом} </pre>	<pre> end;   j := j + 2;   until j&gt;popsize end; begin { Главная программа }   popsize:=20;           {размер   популяции}   lchrom:=20;           {число   битов на один кодируемый параметр}   maxgen:=100;   {максимальное число поколений}   pmutation:=0.01;   {вероятность скрещивания}   pcross:=0.9;   {вероятность мутации}   randomize;   {Инициализация генератора   случайных чисел}   nmutation := 0;  ncross := 0;   {Инициализация счетчиков}   initpop;   statistics (popsize, max, avg, min,   sumfitness, oldpop);   gen:= 0;           {Установка </pre>
--	---



<pre> var j, j1:integer;  begin  for j := 1 to popsize do with oldpop[j] do begin  for j1 := 1 to lchrom*dim do chrom[j1] := flip(0.5);    {Бросок монетки}  decode(chrom,lchrom,x); {Декодирование строки}  fitness := objfunc(x); {Вычисление начальных значений функции пригодности}  end;  end;  {3 генетических оператора: отбора (select), скрещивания (crossover) и мутации (mutation)}  procedure select;    {процедура выбора}  var ipick:integer; </pre>	<pre> счетчика поколений в 0}  repeat {Главный итерационный цикл}  gen:= gen + 1;  generation;  statistics(popsize, max, avg, min, sumfitness, newpop);  oldpop:= newpop;    {переход на новое поколение }  lbl1.caption:=floattostr(min);  until (gen &gt;= maxgen)  end. {End главной программы} </pre>
--	---

