



# ОПТИМІЗАЦІЯ ДОСЯГНЕННЯ МЕТИ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ

МАГІСТРАНТ 2 КУРСУ, ГР. ТВ-61М КОСТЕНКО І.П.

КЕРІВНИК: ДОЦ., К.Т.Н. КУБЛІЙ Л.І.

# Загальні положення

Генетичний алгоритм – евристичний алгоритм пошуку, який використовується для знаходження розв'язку задач оптимізації та моделювання шляхом випадкового підбору, комбінування та варіації параметрів з використанням механізмів, аналогічних природному відбору.

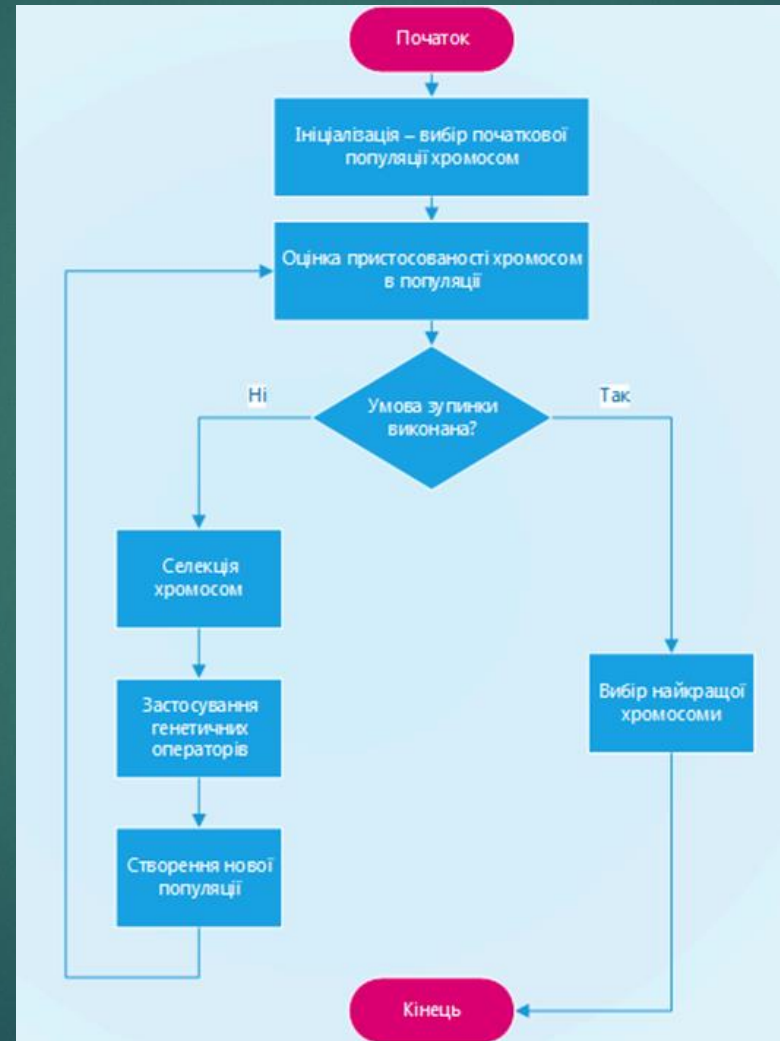
# Загальні положення

Загальними для всіх генетичних алгоритмів та їхніх модифікацій є такі методи природної еволюції:

- ▶ успадкування;
- ▶ мутація;
- ▶ відбір;
- ▶ кросинговер.

Результати оцінюються функцією пристосованості (fitness function), особливим типом цільової функції, порівняльним показником якості, що показує, на скільки придатним є отриманий розв'язок.

# Схема алгоритму



# Обмеження генетичних алгоритмів

Існує ряд обмежень, які не дають ефективно використовувати генетичні алгоритми:

- ▶ складність задання та обчислення функції пристосовності;
- ▶ пропорційне збільшення кількості мутацій відповідно до складності задачі;
- ▶ тенденція збігатися до локального оптимуму;
- ▶ розв'язок придатніший лише порівняно з іншими розв'язками алгоритму.



# Модифікації генетичних алгоритмів

Існує ряд модифікацій генетичних алгоритмів, які можна віднести до двох категорій: методи, акцентовані на роботу з популяцією (методи відбору батьків, методи скорочення популяції), і методи, акцентовані на зміну умов проходження еволюції (методи динамічної зміни популяції, зміни ймовірності мутацій і схрещування, модель островів).

# Можливі рішення

Пропонується для забезпечення якості нащадків використовувати модифікацію (порівняно з базовою схемою генетичного алгоритму) вибору батьківських пар для схрещування на основі турнірного відбору.

Для запобігання передчасної збіжності пропонується використовувати кросинговер SBX.

Оскільки зі збільшенням обсягу задачі збільшується й кількість необхідних розрахунків, доцільною є реалізація системи на базі буферної моделі паралельного генетичного алгоритму.

# Модифікація турнірного відбору

У репродукційну групу включають усіх нащадків та усіх предків.  
Після виконання  $g$  актів кросоверу в групі знаходиться  $4g$  хромосом.  
Серед них відбираються  $N_{\text{pop}}$  кращих для включення в нове покоління.





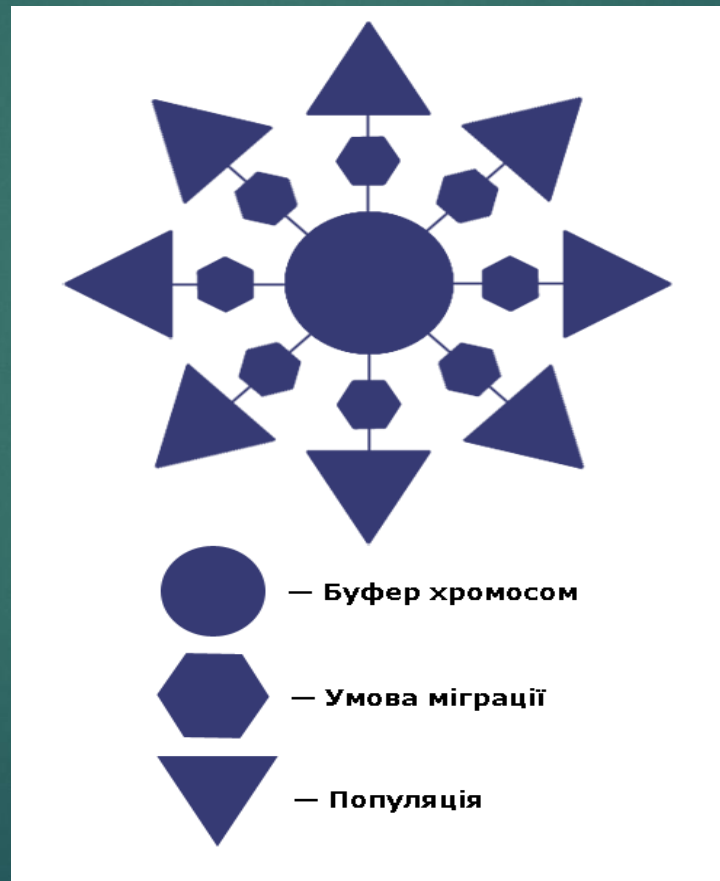
# SVX-кросинговер

Загальна відмінність від двійкового кросинговеру – поняття сили пошуку (search power), це кількісна величина, яка характеризує розподіл вірогідності появи нащадку від двох предків.

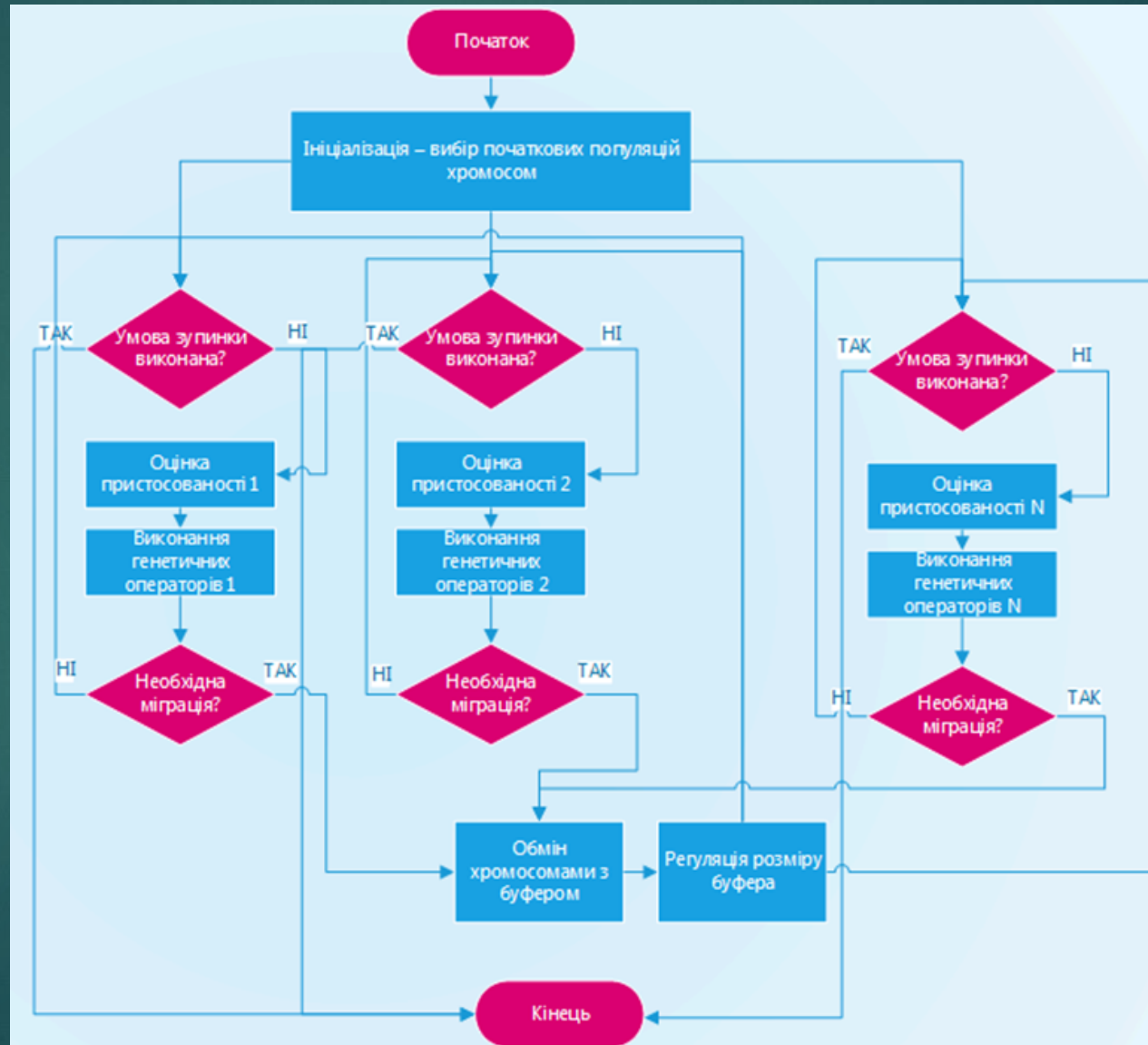
$$P(\beta) = \begin{cases} 0,5(n+1)\beta^n, & \beta \leq 1; \\ 0,5(n+1)\beta^{-(n+2)}, & \beta > 1. \end{cases}$$

Де сила пошуку характеризується розподілом вірогідностей випадкової величини  $\beta$ .

# Нова теоретична модель генетичного алгоритму



# Схема роботи алгоритму



# Основи роботи алгоритму

- ▶ буфер заповнюється популяціями в процесі роботи;
- ▶ кожна популяція еволюціонує окремо;
- ▶ на кожній ітерації перевіряється умова міграції;
- ▶ для кожної популяції можна встановити свої параметри.

# Фактори оцінки показників роботи алгоритму

Для оцінки результатів роботи алгоритму використовувались два показники:

1. Час роботи алгоритму.
2. Оптимальність отриманого рішення.



# Аналіз ефективності



# Використані технології

Для розробки додатку було використано такі технології:

- ▶ мова C#
- ▶ технологія WPF;
- ▶ бібліотека GeneticSharp.

# Набір базових класів для реалізації алгоритму

Для реалізації паралельного генетичного алгоритму зі змінними операторами було розроблено такі класи:

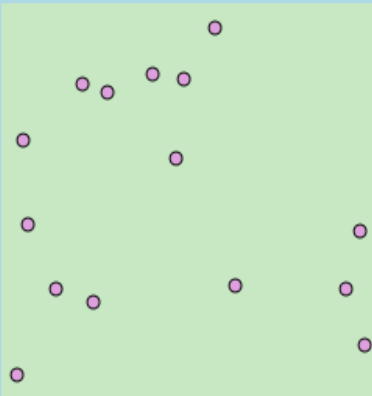
- ▶ `StarGA`, який відповідає за створення алгоритму;
- ▶ `Input`, об'єктом якого є вхідні параметри алгоритму;
- ▶ `ThreadGaParameters`, об'єктом якого є параметри потоків;
- ▶ `ThreadExecutor`, який запускає алгоритми в асинхронному режимі та виконує функції обміну з буфером хромосом;
- ▶ `SimulatedBinaryCrossover` реалізує SBX-кросинговер.

# Работа с программой

Классические алгоритмы | Модификации ГА

Число городов:

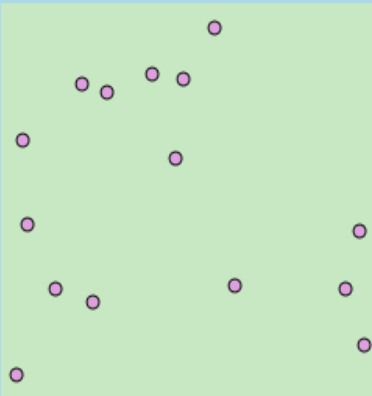
### Классический ГА



Число хромосом =	1000
Элитизм(%) =	5
Вероятность кроссовера(%) =	80
Вероятность мутации(%) =	2
Пауза после(Поколений) =	300

Длина маршрута =  
Время работы =

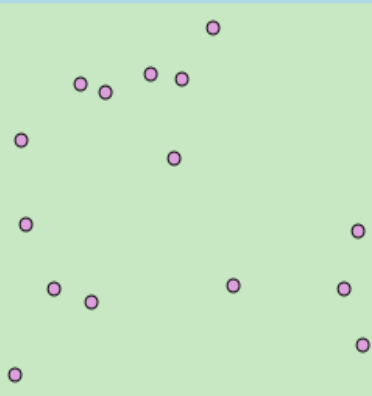
### Параллельный ГА



Число хромосом =	1000
Элитизм(%) =	5
Вероятность кроссовера(%) =	80
Вероятность мутации(%) =	2
Пауза после(Поколений) =	300
Количество потоков =	4

Длина маршрута =  
Время работы =

### Параллельный ГА(Звезда)



Число хромосом =	1000
Поколений до обмена =	10
Остановка после(Поколений) =	100
Вероятность кроссовера =	80
Вероятность мутации =	2

Длина маршрута =  
Время работы =

### Настройки

Поток 1

Размер популяции =

Селекция =

Кроссовер =

Мутация =

Поток 2

Размер популяции =

Селекция =

Кроссовер =

Мутация =

Поток 3

Размер популяции =

Селекция =

Кроссовер =

Мутация =

Поток 4

Размер популяции =

Селекция =

Кроссовер =

Мутация =

# Результаты работы

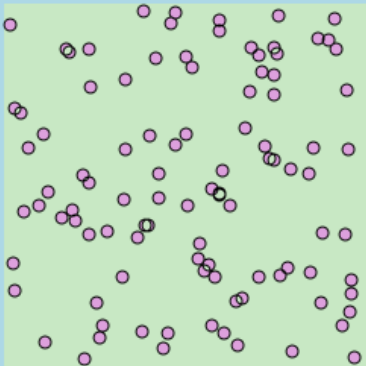
MainWindow

Классические алгоритмы | Модификации ГА

Число городов: 100

Настройки

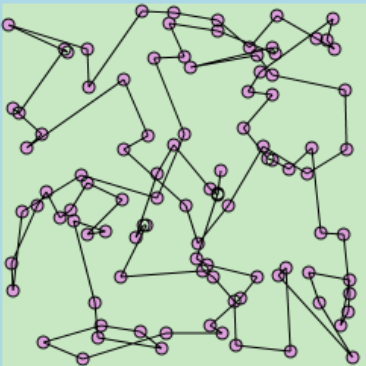
**Классический ГА**



Число хромосом =	1000
Элитизм(%) =	5
Вероятность кроссовера(%) =	80
Вероятность мутации(%) =	2
Пауза после(Поколений) =	300

Длина маршрута =  
Время работы =

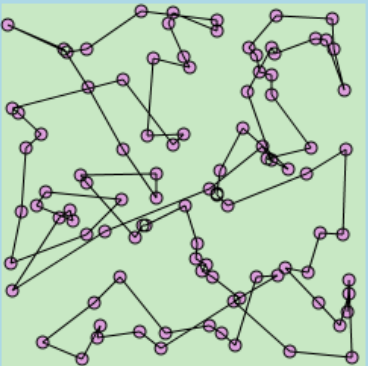
**Параллельный ГА**



Число хромосом =	1000
Элитизм(%) =	5
Вероятность кроссовера(%) =	80
Вероятность мутации(%) =	2
Пауза после(Поколений) =	300
Количество потоков =	4

Длина маршрута = 2982.49  
Время работы = 25705

**Параллельный ГА(Звезда)**



Число хромосом =	1000
Поколений до обмена =	10
Остановка после(Поколений) =	100
Вероятность кроссовера =	80
Вероятность мутации =	2

Длина маршрута = 2803.84  
Время работы = 20083

**Поток 1**

Селекция =

Кроссовер =

Мутация =

**Поток 2**

Селекция =

Кроссовер =

Мутация =

**Поток 3**

Селекция =

Кроссовер =

Мутация =

**Поток 4**

Селекция =

Кроссовер =

Мутация =



Дякую за увагу