

# 1 Algorytmy heurystyczne

---

**Algorithm 1:** Genetic Algorithm

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*

**Result:** Najlepsze rozwiązanie *best\_solution*

- 1 Zainicjuj populację początkową *population*
  - 2 Oblicz przystosowanie dla każdego osobnika w populacji
  - 3 Zainicjuj *best\_solution* jako najlepszy osobnik z populacji
  - 4 **for**  $i = 1$  **to** *max\_iterations* **do**
    - 5     Wybierz rodziców
    - 6     Krzyżowanie i mutacja w celu stworzenia nowych osobników
    - 7     Ocena nowych osobników
    - 8     Aktualizacja *best\_solution* jeśli znaleziono lepsze rozwiązanie
  - 9 **return** *best\_solution*
- 

---

**Algorithm 2:** Particle Swarm Optimization

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*

**Result:** Najlepsze rozwiązanie *best\_solution*

- 1 Zainicjuj populację i predkości dla *population*
  - 2 Oblicz przystosowanie dla każdego osobnika w populacji
  - 3 Zaktualizuj najlepsze osobiste i globalne rozwiązanie
  - 4 **for**  $i = 1$  **to** *max\_iterations* **do**
    - 5     Dla każdego osobnika aktualizuj predkość i pozycje na podstawie najlepszych rozwiązań
    - 6     Ocena nowych pozycji
    - 7     Aktualizacja najlepszych osobistych i globalnych rozwiązań
  - 8 **return** *best\_solution*
-

---

**Algorithm 3:** Bee Algorithm

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*

**Result:** Najlepsze rozwiązanie *best\_solution*

```
1 Zainicjuj populację pszczoł population
2 Oblicz przystosowanie dla każdego osobnika w populacji
3 Zaktualizuj najlepsze rozwiązanie best_solution
4 for  $i = 1$  to max_iterations do
5     Dla każdego zbieracza z population wykonaj przeszukiwanie lokalne
6     Ocena nowych pozycji
7     Dla każdego obserwatora wybierz i podażaj za najlepszymi źródłami
       na podstawie przystosowania
8     Ocena nowych pozycji po eksploracji
9     Jeśli źródło zostało wyczerpane, zastap je nowymi skautami
       eksplorującymi losowe pozycje
10    Ocena nowych pozycji
11    Aktualizacja best_solution jeśli znaleziono lepsze rozwiązanie
12 return best_solution
```

---

---

**Algorithm 4:** Differential Evolution

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*,  
czynniki  $F$  i  $CR$

**Result:** Najlepsze rozwiązanie *best\_solution*

```
1 Zainicjuj populację population
2 Oblicz przystosowanie dla każdego osobnika
3 for  $i = 1$  to max_iterations do
4     Dla każdego osobnika stwórz mutantów i wykonaj krzyżowanie
5     Ocena nowych rozwiązań
6     Selekcja lepszych rozwiązań do następnej iteracji
7     Aktualizacja best_solution jeśli znaleziono lepsze rozwiązanie
8 return best_solution
```

---

---

**Algorithm 5:** Bat Algorithm

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*

**Result:** Najlepsze rozwiązanie *best\_solution*

- 1 Zainicjuj populację *population* i przypisz każdemu nietoperzowi początkową prędkość i częstotliwość
  - 2 Oblicz przystosowanie dla każdego nietoperza
  - 3 Zaktualizuj najlepsze rozwiązanie
  - 4 **for**  $i = 1$  **to** *max\_iterations* **do**
    - 5     Dla każdego nietoperza aktualizuj częstotliwość, prędkość i pozycje
    - 6     Jeśli losowy puls mniejszy niż próg, wykonaj lokalne przeszukiwanie
    - 7     Ocena nowych pozycji
    - 8     Aktualizacja przystosowania i najlepszego rozwiązania
  - 9 **return** *best\_solution*
- 

---

**Algorithm 6:** Firefly Algorithm

---

**Data:** Lista wejściowa *population*, liczba iteracji *max\_iterations*

**Result:** Najlepsze rozwiązanie *best\_solution*

- 1 Zainicjuj populację *population*
  - 2 Oblicz przystosowanie dla każdego świetlika
  - 3 **for**  $i = 1$  **to** *max\_iterations* **do**
    - 4     Dla każdego świetlika znajdź jaśniejsze świetliki i przesuń się w ich kierunku
    - 5     Ocena nowych pozycji
    - 6     Aktualizacja najlepszych rozwiązań
  - 7 **return** *best\_solution*
-