

```

GeneracjaPopulacji[wielkoscPopulacji_, iloscWymiarow_, ograniczenia_] :=
Module[{λ = wielkoscPopulacji, wym = iloscWymiarow, populacja},
moduł
    populacja = Table[Table[RandomReal[{ograniczenia[[j, 1]], ograniczenia[[j, 2]]}],
tabela tabela losowa liczba rzeczywista
        {j, 1, wym}], {i, 1, λ}];
    Return[populacja]
zwróć

```

```

Reprodukcja[q_, populacjaBazowa_,
    funkcjaOceny_, wielkoscPopulacjiBazowej_, iloscWymiarow_] :=
Module[{zmienne, λ = wielkoscPopulacjiBazowej, populacjaTymczasowa},
moduł
    turniej, indeksNajlepszegoOsobnika, wartosciTurnieju},
    Clear[zmienne, wartosciTurnieju, turniej, indeksNajlepszegoOsobnika];
wyczyść
    populacjaTymczasowa = {};
    zmienne = Table[Symbol[StringJoin["x", ToString[i]]], {i, 1, iloscWymiarow}];
tabela symbol połącz ciągi znaków przemień na ciąg znaków
    Do[turniej = Table[populacjaBazowa[RandomInteger[{1, λ}]]], {i, 1, q}];
rób tabela losowa liczba całkowita
    wartosciTurnieju = Table[funkcjaOceny, {i, 1, q}];
tabela
    Do[Do[wartosciTurnieju[[k]] =
rób rób
        wartosciTurnieju[[k]] /. {zmienne[[i]] → turniej[[k, i]]}
        , {i, 1, iloscWymiarow}];
        , {k, 1, q}];
    indeksNajlepszegoOsobnika = Position[wartosciTurnieju, Min[wartosciTurnieju]];
pozycja minimum
    AppendTo[populacjaTymczasowa, turniej[[indeksNajlepszegoOsobnika[[1, 1]]]]];
dołącz na końcu do wartości zmiennej
    , {j, 1, λ}];
    Return[populacjaTymczasowa]
zwróć

```

```

Krzyzowanie[osobnikRodzicielski1_, osobnikRodzicielski2_, iloscWymiarow_] :=
Module[{wym = iloscWymiarow, osobnikPotomny1, osobnikPotomny2, punktPrzeciecia},
moduł
    punktPrzeciecia = RandomInteger[{1, wym}];
losowa liczba całkowita
    osobnikPotomny1 = Flatten[{Part[osobnikRodzicielski1, 1 ;; punktPrzeciecia],
splaszcz część
        Part[osobnikRodzicielski2, punktPrzeciecia + 1 ;;]}];
część
    osobnikPotomny2 = Flatten[{Part[osobnikRodzicielski2, 1 ;; punktPrzeciecia],
splaszcz część
        Part[osobnikRodzicielski1, punktPrzeciecia + 1 ;;]}];
część
    Return[{osobnikPotomny1, osobnikPotomny2}]
zwróć

```

```

KrzyzowanieUsredniajace[osobnikRodzicielski1_, osobnikRodzicielski2_] :=
Module[{osobnikPotomny1, osobnikPotomny2, wspolczynnik},
  |moduł
  wspolczynnik = RandomReal[];
  |losowa liczba rzeczywista
  osobnikPotomny1 =
    osobnikRodzicielski1 + wspolczynnik * (osobnikRodzicielski2 - osobnikRodzicielski1);
  osobnikPotomny2 = osobnikRodzicielski2 + osobnikRodzicielski1 - osobnikPotomny1;
  Return[{osobnikPotomny1, osobnikPotomny2}]
  |zwróć

MutacjaZParametrem[mutowanyOsobnik_, parametr_, iloscWymiarow_] :=
Module[{σ = parametr, ξ, x = mutowanyOsobnik},
  |moduł
  ξ = RandomVariate[NormalDistribution[0, σ], iloscWymiarow];
  |próba losowa |rozkład normalny
  x += ξ;
  Return[x];
  |zwróć

MutacjaDeterministyczna[mutowanyOsobnik_,
  iloscWymiarow_, numerGeneracji_, maksymalnaIloscGeneracji_] :=
Module[{ξ, σ, t = numerGeneracji, T = maksymalnaIloscGeneracji, x = mutowanyOsobnik},
  |moduł
  σ = 1 - 0.9 * t / T;
  ξ = RandomVariate[NormalDistribution[0, σ], iloscWymiarow];
  |próba losowa |rozkład normalny
  x += ξ;
  Return[x];
  |zwróć

MutacjaAdaptacyjna[mutowanyOsobnik_,
  iloscWymiarow_, numerGeneracji_, czestoscSukcesow_] :=
Module[{ξ, σ = 0.5, x = mutowanyOsobnik, n = 10, p = czestoscSukcesow, c = 0.82},
  |moduł
  If[numerGeneracji % 10 == 0, If[p > 1 / 5, σ = σ / c, σ = σ * c]];
  |operator warunkowy |operator warunkowy
  ξ = RandomVariate[NormalDistribution[0, σ], iloscWymiarow];
  |próba losowa |rozkład normalny
  x += ξ;
  Return[x];
  |zwróć

```

```

MutacjaSamoadaptująca[mutowanyOsobnik_, iloscWymiarow_] :=
Module[{ $\tau = 1 / \text{Sqrt}[\text{iloscWymiarow}]$ , x = mutowanyOsobnik,  $\sigma$ ,  $\xi$ },
   $\sigma = x[\text{iloscWymiarow}]$ ;
   $\sigma *= E^{\text{RandomVariate}[\text{NormalDistribution}[0, \tau]]}$ ;
  x[[iloscWymiarow]] =  $\sigma$ ;
   $\xi = \text{RandomVariate}[\text{NormalDistribution}[0, \sigma], \text{iloscWymiarow} - 1]$ ;
  AppendTo[ $\xi$ , 0];
  x +=  $\xi$ ;
  Return[x]

ZnalezienieNajlepszegoOsobnika[populacja_,
  funkcjaCelu_, iloscWymiarow_, wielkoscPopulacji_] :=
Module[{zmienne, wartosciFunkcji,  $\lambda = \text{wielkoscPopulacji}$ , indeksNajlepszegoOsobnika},
  zmienne = Table[Symbol[StringJoin["x", ToString[i]]], {i, 1, iloscWymiarow}];
  wartosciFunkcji = Table[funkcjaCelu, {i, 1,  $\lambda$ };
  Do[Do[
    wartosciFunkcji[[k]] = wartosciFunkcji[[k]] /. {zmienne[[i]]  $\rightarrow$  populacja[[k, i]]}
    , {i, 1, iloscWymiarow}];
    , {k, 1,  $\lambda$ };
  indeksNajlepszegoOsobnika =
    Position[wartosciFunkcji, Min[wartosciFunkcji]][[1, 1]];
  Return[{populacja[indeksNajlepszegoOsobnika]},
    wartosciFunkcji[indeksNajlepszegoOsobnika]]];
]

```

```

AGzParametrem[licznoscPopulacji_, funkcjaCelu_,
  q_, iloscWymiarow_, ograniczenia_, prawdopodobienstwoMutacji_,
  prawdopodobienstwoKryzowania_, przewidywaneOptimum_, parametr_] :=
Module[{λ = licznoscPopulacji, f = funkcjaCelu, populacjaBazowa,
  _moduł
  populacjaPotomna, pm = prawdopodobienstwoMutacji,
  pk = prawdopodobienstwoKryzowania, populacjaTymczasowa, najlepszyOsobnik,
  numerGeneracji, osobnikiPotomne, numeryOsobnikowRodzicielskich},
  numerGeneracji = 1;
  populacjaBazowa = GeneracjaPopulacji[λ, iloscWymiarow, ograniczenia];
  najlepszyOsobnik =
    ZnajdzenieNajlepszegoOsobnika[populacjaBazowa, f, iloscWymiarow, λ];
  While[Abs[najlepszyOsobnik[[2]] - przewidywaneOptimum] > 10-4,
    _podciąg _wartość bezwzględna
    populacjaTymczasowa = Reprodukacja[q, populacjaBazowa, f, λ, iloscWymiarow];
    populacjaPotomna = {};
    Do[numeryOsobnikowRodzicielskich = RandomInteger[{1, λ}, 2];
      _rób _losowa liczba całkowita
      osobnikiPotomne = KrzyzowanieUsredniajace[
        populacjaTymczasowa[numeryOsobnikowRodzicielskich[[1]]],
        populacjaTymczasowa[numeryOsobnikowRodzicielskich[[1]]]];
      AppendTo[populacjaPotomna, osobnikiPotomne[[1]]];
      _dołącz na końcu do wartości zmiennej
      AppendTo[populacjaPotomna, osobnikiPotomne[[2]]];
      _dołącz na końcu do wartości zmiennej
    ], {i, 1, Floor[λ]};
    _podłoga
    Do[If[RandomReal[] ≤ pm, populacjaPotomna[[i]] =
      _rób _losowa liczba rzeczywista
      MutacjaZParametrem[populacjaPotomna[[i]], parametr, iloscWymiarow]
    ], {i, 1, λ}];
    populacjaBazowa = populacjaPotomna;
    najlepszyOsobnik =
      ZnajdzenieNajlepszegoOsobnika[populacjaBazowa, f, iloscWymiarow, λ];
    numerGeneracji++;
  ];
  Print[numerGeneracji];
  _drukuj
  Return[najlepszyOsobnik]
  _zwróć

```

(\*TESTY\*)

```
pop = GeneracjaPopulacji[500, 2, {{-5, 5}, {-5, 5}}];
```

```
ZnajdzenieNajlepszegoOsobnika[pop, x12 + x22, 2, 500]
```

```
{{-0.38597, 0.0608789}, 0.152679}
```

```
Reprodukacja[50, pop, x12 + x22, 500, 2];
```

```
AGzParametrem[500, x12 + x22, 10, 2, {{-10, 10}, {-10, 10}}, 0.4, 0, 0, 1]
```

92

```
{{0.00339844, -0.00754992}, 0.0000685507}
```