

## **KOMPUTEROWY SYSTEM WSPOMAGANIA ALOKACJI OSADZONYCH W ZAKŁADZIE PENITENCJARNYM<sup>1</sup>**

**Sławomir Strzykowski\*, Maciej Krawczak\*\***

\*Microsoft Poland

\*\*Wyższa Szkoła Informatyki Stosowanej i Zarządzania,  
01-447 Warszawa, ul. Nowelska 6

### **Streszczenie**

Artykuł przedstawia algorytm genetyczny GAPAP (ang. Genetic Algorithm of Prisoner's Allocation Problem), który służy do wspomaganie alokowania osób pozbawionych wolności wewnątrz zakładu karnego lub aresztu śledczego. Celem badań było opracowanie autorskiego algorytmu genetycznego GAPAP, jego implementacja komputerowa oraz przetestowanie opracowanego programu komputerowego na rzeczywistym przykładzie zakładu karnego. Wykazano, że zbudowany model alokacji osób pozbawionych wolności, wykorzystujący algorytm genetyczny GAPAP, umożliwia rozmieszczanie osadzonych wewnątrz zakładu penitencjarnego z zachowaniem obowiązujących w tym zakresie reguł i ograniczeń, obniżając jednocześnie stopień średniego niedostosowania populacji. W celu rozwiązania sformułowanego zagadnienia związanej z problematyką alokowania osób pozbawionych wolności wewnątrz zakładu penitencjarnego, została zastosowana teoria algorytmów genetycznych.

Słowa kluczowe: komputerowe systemy wspomaganie decyzji, algorytmy genetyczne

### **1. Opis problemu alokacji osób pozbawionych wolności**

Jednym ze statutowych zadań Służby Więziennej jest konieczność zapewnienia prawidłowego alokowania osób pozbawionych wolności w jednostce penitencjarnej. Alokowanie osadzonych odbywa się poprzez ich fizyczne zakwaterowanie w ramach określonego pawilonu, określonego oddziału i określonej celi mieszkalnej. Prawidłowa realizacja zasad rozmieszczania osób pozbawionych wolności w zakładach karnych i aresztach śledczych jest obecnie niezmiernie trudna. Poniżej wymienione są

---

<sup>1</sup> Artykuł napisany na podstawie rozprawy doktorskiej pierwszego z autorów.

niektóre czynniki, które utrudniają lub wręcz uniemożliwiają zgodne z przepisami rozmieszczanie osadzonych:

- przeludnienie w zakładach karnych i aresztach śledczych,
- bardzo duża liczba realizowanych transportów.

Przepisy wykonawcze dotyczące zasad rozmieszczania osadzonych zawarte zostały opisane w postaci reguł alokacji osadzonych.

### 1.1. Reguły alokacji osób pozbawionych wolności

Reguły alokacji osadzonych to zbiór zasad, określających rozmieszczanie osób pozbawionych wolności w sposób zapewniający prawidłowy tok postępowania karnego, odbywania kary pozbawienia wolności lub stosowania środka przymusu. Ponadto istnieją ograniczenia alokacji osadzonych, które związane są z przeznaczeniem cel mieszkalnych. Wszystkie wymienione poniżej reguły i ograniczenia alokacji muszą być spełnione równocześnie:

REGUŁA 1 (R1): W tej samej celi mogą być alokowani osadzeni posiadający ten sam status prawny. Status prawny określa przynależność osadzonych do kategorii tymczasowo aresztowanych lub skazanych.

REGUŁA 2 (R2): Osadzeni uprzednio karani nie mogą być alokowani wspólnie (w jednej celi) z osadzonymi pierwszy raz karanymi.

REGUŁA 3 (R3): Osadzeni młodociani (poniżej 21 roku życia) nie mogą być alokowani wspólnie (w jednej celi) z osadzonymi dorosłymi.

REGUŁA 4 (R4): Osadzeni uczestniczący w podkulturze przestępczej nie mogą być alokowani wspólnie (w jednej celi) z osadzonymi nie uczestniczącymi w podkulturze przestępczej.

REGUŁA 5 (R5): Osadzeni używający wyrobów tytoniowych nie mogą być alokowani wspólnie (w jednej celi) z osadzonymi nie używającymi wyrobów tytoniowych.

OGRANICZENIE 1 (0.1): Osoby pozbawione wolności, zakwalifikowane do kategorii "niebezpiecznych" muszą bezwzględnie być alokowane w wyznaczonych do tego pomieszczeniach mieszkalnych (celach).

OGRANICZENIE 2 (0.2): Współuczestnicy przestępstwa muszą być alokowani w specjalnie do tego wydzielonych pomieszczeniach mieszkalnych.

OGRANICZENIE 3 (0.3): Osadzeni z problemami zdrowotnymi muszą być alokowani wyłącznie w przeznaczonych do tego pomieszczeniach mieszkalnych.

Rozpatrywane zadanie sprowadza się do uzyskania takiej alokacji osadzonych (otrzymanej w wyniku działania algorytmu GAPAP), która dla każdego pomieszczenia mieszkalnego (celi) w najwyższym stopniu będzie zgodna z wymienionymi powyżej regułami i ograniczeniami.

### 1.2. Założenia dla algorytmu genetycznego GAPAP

Poniżej przedstawiono podstawowe założenia (oraz schemat) dla algorytmu genetycznego GAPAP:

- przyjęto, iż każdy **gen** odpowiada atrybutowi osoby pozbawionej wolności dla chromosomów  $pa(i)$  i  $pc(i)$  lub celi dla chromosomów  $ca(i)$  oraz  $ce(i)$ , gdzie  $i$  oznacza indeks genotypu,
- przyjęto, że zestaw genów stanowi **chromosom**  $pa(j)$  i  $pc(j)$  dla osoby pozbawionej wolności oraz chromosom  $ca(i)$  oraz  $ce(i)$  dla pomieszczenia mieszkalnego, gdzie  $j$  oznacza indeks osobnika, dla określonego  $j$ ,
- przyjęto, że genotyp stanowi zestaw chromosomów  $pa(j)$  i  $pc(j)$  oraz chromosomów  $ca(i)$  oraz  $ce(i)$ , dla określonego  $i$ ,
- przyjęto, że **osobnik** oznacza osobę pozbawioną wolności,
- w algorytmie GAPAP zastosowano strategię losowego przeszukiwania przestrzeni,
- przyjęto, że **populację początkową** stanowi zbiór wszystkich genotypów, składających się z jednej pary chromosomów  $ca(i)$  oraz  $ce(i)$  i odpowiadającej liczbie osadzonych alokowanych w genotypie par chromosomów  $pa(j)$  i  $pc(j)$ ,
- przyjęto, że **parę rodzicielską** stanowi zestaw dwóch genotypów,
- przyjęto, że **funkcją celu** jest minimalizowanie niedostosowania populacji,
- przyjęto, że do rozwiązania postawionego w rozprawie zadania zastosowane zostanie **krzyżowanie jednopunktowe**,
- przyjęto, że w algorytmie GAPAP zastosowana zostanie **mutacja jednopunktowa**,
- przyjęto, że każdy **osobnik** jest **nagradzany** (zmniejszeniem wartości niedostosowania) lub **karany** (zwiększeniem wartości niedostosowania) w sytuacji, gdy w wyniku przeprowadzonej operacji krzyżowania jest alokowany w celi o większej (nagroda) lub mniejszej (kara) zgodności z poszczególnymi ograniczeniami alokacji,
- przyjęto, że każdy **genotyp** jest **karany** (poprzez zwiększenie licznika kar) za niezgodne z regułami alokowanie osobnika w tej celi.

|                                                              |
|--------------------------------------------------------------|
| URUCHOMIENIE APLIKACJI I OKREŚLENIE WARUNKÓW<br>POCZĄTKOWYCH |
| OBLICZENIE NIEDOSTOSOWANIA OSOBNIKÓW                         |
| NAGRADZANIE LUB KARANIE OSOBNIKÓW                            |
| OBLICZANIE ŚREDNIEGO NIEDOPASOWANIA GENOTYPÓW                |
| OBLICZANIE ŚREDNIEGO NIEDOPASOWANIA POPULACJI                |
| ZAPIS WYNIKÓW POCZĄTKOWYCH                                   |
| OBLICZANIE PRAWDOPODOBIENSTWA KRZYŻOWANIA GENOTYPÓW          |
| OBLICZANIE DYSTRYBUANTY DLA GENOTYPÓW                        |
| LOSOWANIE GENOTYPÓW DO KRZYŻOWANIA                           |
| TWORZENIE PAR RODZICIELSKICH                                 |
| LOSOWANIE PUNKTÓW PRZECIĘCIA                                 |
| WYKONANIE KRZYŻOWANIA                                        |
| KARANIE GENOTYPÓW                                            |

|                                                                         |
|-------------------------------------------------------------------------|
| OBLICZANIE ŚREDNIEGO NIEDOSTOSOWANIA OSOBNIKÓW,<br>GENOTYPÓW, POPULACJI |
| ZAPIS WYNIKÓW PO KRZYŻOWANIU                                            |
| OBLICZANIE PRAWDOPODOBIEŃSTWA WYŁOSOWANIA GENOTYPU<br>DO MUTACJI        |
| OBLICZANIE DYSTRYBUANTY GENOTYPÓW DO MUTACJI                            |
| LOSOWANIE GENOTYPÓW DO MUTACJI                                          |
| WYKONANIE MUTACJI                                                       |
| OBLICZENIE ŚREDNIEGO NIEDOSTOSOWANIA OSOBNIKÓW,<br>GENOTYPU, POPULACJI  |
| ZAPIS WYNIKÓW PO MUTACJI                                                |
| SPRAWDZANIE WARUNKU ZATRZYMANIA                                         |

Rysunek 1. Schemat działania algorytmu GAPAP

## 2. Opis algorytmu GAPAP

### Uruchomienie aplikacji i określenie warunków początkowych

Interfejs aplikacji umożliwia wybór następujących warunków początkowych:

- wartości nagrody lub kary  $pp$ ,
- prawdopodobieństwa krzyżowania  $cp$ ,
- włączanie lub wyłączanie operacji mutacji,
- wymaganej liczby iteracji algorytmu  $R$  (określenie warunku zatrzymania).

Ponadto, w interfejsie wyświetlana jest:

- wartość średniego niedostosowania populacji  $pdf$  obliczona w momencie uruchamiania aplikacji,
- wartość średniego niedostosowania minimalnego  $pdf_{it}$ , obliczana w trakcie wykonywania kolejnych faz i iteracji algorytmu,
- typ operacji algorytmu genetycznego (krzyżowanie, mutacja), w którym osiągnięto najniższe średnie niedostosowanie populacji,
- wyniki dla każdej z przeprowadzonych iteracji  $it$ , zawierające wartości średniego niedostosowania populacji odpowiednio dla każdego typu operacji (krzyżowanie, mutacja) oraz procent wymaganych alokacji osadzonych.

Uruchomienie algorytmu genetycznego następuje z interfejsu aplikacji GAPAP. Algorytm wykonuje kolejne iteracje od  $it = 0$  do  $it = R$ , gdzie  $R$  oznacza liczbę iteracji  $it$  zdefiniowaną w warunkach początkowych. Obliczanie niedostosowania osobników następuje poprzez porównanie wartości kolejnych genów  $g(l, i)$ ,  $l = 1, \dots, 5$ ,  $i = 1, \dots, G$  chromosomu  $ca(i)$  z wartościami odpowiednich genów  $h(l, i)$  chromosomu  $pa(i)$ . W przypadku braku zgodności (identyczności) genów, cząstkowe niedo-

stosowanie osobnika  $pfs(i, j)$  wzrasta o cząstkową wartość niedostosowania  $sf$  (przyjęto stałą wartość  $sf = 0,2$ ). Maksymalne cząstkowe niedostosowanie osobnika  $pfs(i, j) = 1$  (5 genów  $\times$  0,2). Minimalne cząstkowe niedostosowanie osobnika  $pfs(i, j) = 0$ . Na podstawie cząstkowego niedostosowania osobników  $pfs(i, j)$  wyznaczane jest średnie niedostosowanie  $pft(i, j)$  tych osobników. Średnie niedostosowanie osobników obliczane jest w każdej iteracji.

### Obliczanie średniego niedostosowania genotypów

Obliczanie średniego niedostosowania genotypów następuje poprzez sumowanie średniego niedostosowania osobników a następnie podzielenie obliczonej wartości maksymalną przez liczbę pozycji w genotypie  $m(i)$ .

$$gfi(i) = \frac{\sum_{po=1}^{m(i)} pft(i, po)}{m(i)} \quad (1)$$

gdzie:

$pft(i, po)$  - średnie niedostosowanie osobnika,

$m(i)$  -liczba pozycji w genotypie.

Średnie niedostosowanie genotypów obliczane jest dla każdej iteracji i wykorzystywane następnie w procesie selekcji.

### Obliczanie średniego niedostosowania populacji

Obliczenie średniego niedostosowania populacji  $pft_{it}$  następuje poprzez sumowanie średniego niedostosowania genotypów, a następnie podzielenie obliczonej wartości przez  $G$ , gdzie  $G$  oznacza liczbę genotypów w populacji. Średnie niedostosowanie populacji obliczane jest dla każdej iteracji i porównywane z najlepszym dotychczas otrzymanym wynikiem.

$$pft_{it} = \frac{\sum_{i=1}^G gfi(i)}{G} \quad (2)$$

gdzie:

$gfi(i)$  -średnie niedostosowanie genotypu,

$G$  -liczba genotypów w populacji.

Obliczone wielkość  $pft_{it}$  zostaje zapisane w bazie danych z typem operacji  $ty_{it} = "S"$ , gdzie "S" oznacza start algorytmu (iteracja zerowa).

### Operacja krzyżowania

Krzyżowanie w opracowanym algorytmie polega na wymianie w określonym punkcie przecięcia (locus) osobników, dla wylosowanej pary genotypów. W momencie wykonywania krzyżowania, osobnicy są oceniani przy użyciu mechanizmu nagradzania lub karania. Krzyżowanie genotypów następuje z określonym w warunkach początkowych prawdopodobieństwem i obejmuje następujące fazy:

- obliczanie prawdopodobieństwa krzyżowania dla poszczególnych genotypów,
- obliczanie dystrybuanty dla poszczególnych genotypów,
- oznaczanie genotypów do krzyżowania -selekcja,
- losowanie genotypów do krzyżowania,
- tworzenie par rodzicielskich z wylosowanych genotypów,
- losowanie punktu przecięcia dla poszczególnych par rodzicielskich,
- wykonanie krzyżowania,
- karanie genotypów.

### Obliczanie prawdopodobieństwa krzyżowania genotypu

Prawdopodobieństwo krzyżowania genotypu  $cp(i)$ ,  $i = 1, \dots, G$ , obliczane jest według następującego wzoru:

$$cp(i) = \frac{gf_i(i)}{pf_{it}} \quad (3)$$

gdzie:

$gf_i(i)$  - średnie niedostosowanie genotypu,

$pf_{it}$  - średnie niedostosowanie populacji.

W wyniku zastosowania wzoru (3), prawdopodobieństwo wylosowania  $i$ -tego genotypu do krzyżowania jest tym większe, im większe jest średnie niedostosowanie tego genotypu.

### Obliczanie dystrybuanty dla genotypów

Obliczanie dystrybuanty ma na celu wyznaczenie wycinków na kole ruletki dla poszczególnych genotypów. Im większy przydzielony wycinek koła ruletki, tym większe prawdopodobieństwo wylosowania genotypu do krzyżowania. Wartość dystrybuanty  $qt(i)$  dla każdego genotypu  $i$  obliczana jest według wzoru:

$$qt(i) = \sum_{i=1}^G cp(i) \quad (4)$$

gdzie:

$G$  -liczba genotypów w populacji,

$cp(i)$  -prawdopodobieństwo wylosowania genotypu do krzyżowania.

### Oznaczanie genotypów do krzyżowania - selekcja

Do procesu selekcji użyto ruletki o wielkości pól zgodnej z wartościami niedostosowania. Proces selekcji polega na wielokrotnym obrocie ruletką i wyborze za każdym razem jednego genotypu do nowej populacji. W celu przeprowadzenia selekcji, w aplikacji GAPAP wywoływana jest procedura generująca zmiennie-pozycyjną liczbę pseudolosową  $rnd$  z zakresu  $[0, 1]$ . Jeżeli  $rnd \leq qt(i)$ , to genotyp zostaje oznaczony do krzyżowania; w przeciwnym przypadku zostaje oznaczony genotyp  $qt(i)$  ( $1 \leq i \leq G$ ), gdzie  $G$  oznacza ilość genotypów w populacji  $i$  dla którego zachodzi

$qt(i - 1) \leq rnd \leq qt(i)$ . Genotypy mogą zostać wylosowane więcej niż jeden raz. Losowanie powtarzane jest  $G$  razy. Oznaczone genotypy posiadają wartość wskaźnika  $rl(i) = 1$ .

#### Losowanie genotypów do krzyżowania

W przedstawionym w pracy rozwiązaniu określono prawdopodobieństwo krzyżowania  $cp(i)$  na poziomie 0,5 lub 0,9. Oznacza to, że spodziewamy się, iż średnio 50% lub 90 % genotypów ulegnie krzyżowaniu. Określenie wysokiego  $cp(i)$  wynika z przyjętego założenia, że działanie algorytmu genetycznego GAPAP musi odzwierciedlać stan faktyczny, gdzie w celu prawidłowej alokacji osadzonych w pomieszczeniach mieszkalnych (celach) możliwe jest przemieszczanie (realokacja) dużej części populacji osadzonych. Jednocześnie przyjęto założenie, że liczba wylosowanych do krzyżowania genotypów musi być parzysta. W przypadku niespełnienia tego warunku, ostatni z wylosowanych genotypów jest oznaczany wskaźnikiem  $rl(i) = 0$  i nie bierze udziału w dalszej selekcji. Losowanie genotypów, wyznaczonych w procesie selekcji za pomocą mechanizmu ruletki polega na:

- wygenerowaniu dla każdego genotypu liczby losowej  $rnd$  z zakresu  $[0,1]$ ,
- jeżeli  $rnd \leq cp(i)$  to genotyp jest oznaczany jako wybrany do krzyżowania (liczba pseudolosowa  $rnd > cp(i)$  wykracza poza określone prawdopodobieństwo krzyżowania).

Wytypowane genotypy są oznaczane wskaźnikiem  $cr(i) = 1, i = 1, \dots, G$ .

#### Tworzenie par rodzicielskich

Tworzenie par rodzicielskich polega na:

- generowaniu liczby losowej  $rnd$  z zakresu  $[1, G]$  dla genotypów takich, dla których  $cr(i) = 1$ ,
- zapisaniu dla wylosowanego genotypu  $i$  jako "ojca" pary rodzicielskiej  $ift(i, f)$ , gdzie  $f$  oznacza indeks pary rodzicielskiej,
- ponownym wygenerowaniu liczby losowej  $rnd$  z zakresu  $[1, G]$  dla genotypów posiadających wskaźnik wylosowane genotypu do krzyżowania  $cr(i) = 1$ ,
- zapisaniu dla wylosowanego genotypu  $i$  jako "matki" pary rodzicielskiej  $imt(i, f)$ .

Wylosowana para stanowi parę rodzicielską  $fpa(ift(i, f), imt(i, f))$ . Proces tworzenia par rodzicielskich powtarzany jest tyle razy, ile występujących w populacji genotypów, dla których wskaźnik wylosowania do krzyżowania  $cr(i) = 1$ . W przypadku, gdy  $ift(i, f) = imt(i, f)$ , para rodzicielska zostaje usunięta (realokacja osobników z genotypu do tego samego genotypu nie ma uzasadnienia). Ponadto przyjęto założenie, że genotypy uprzednio wylosowane jako  $ift(i, f)$  lub  $imt(i, f)$  mogą zostać ponownie wylosowane do kolejnej pary rodzicielskiej.

### Losowanie punktu przecięcia dla par rodzicielskich

Projektując algorytm GAPAP przyjęto, że odcinanie osobników w genotypie następować będzie lewostronnie, tzn.  $fpl(ift(i, f), imt(i, f))$  pierwszych osobników zostanie realokowanych. Losowanie punktu przecięcia  $fpl(ift(i, f), imt(i, f))$  dla poszczególnych par rodzicielskich polega na:

- wygenerowaniu liczby pseudolosowej  $rnd$  z zakresu  $[1, m]$ , gdzie  $m$  oznacza maksymalną liczbę pozycji (miejsc) w obu genotypach,
- oznaczeniu  $fpl(ift(i, f), imt(i, f)) = rnd$  pary rodzicielskiej.

### Wykonanie krzyżowania

Wykonanie krzyżowania dla poszczególnych par rodzicielskich, polega na:

- zamianie  $n$  indeksów osobników wchodzących w skład genotypu "ojca" indeksami genotypów "matki", gdzie  $n$  oznacza punkt przecięcia  $fpl(ift(i, f), imt(i, f))$ ,
- zamianie  $n$  indeksów osobników wchodzących w skład genotypu "matki" indeksami genotypów "ojca",
- oznaczeniu osobników wskaźnikiem  $re(i) = 1$  wykorzystywanym w fazie nagradzania i karania osobników,
- oznaczeniu genotypów wskaźnikiem  $re(i) = 1$  wykorzystywanym w fazie karania genotypów.

### Karanie genotypów

Karanie genotypów to dodatkowa funkcja oceniająca stosowana w sytuacji, gdy alokowany w genotypie osobnik posiada geny chromosomu  $ca$  niezgodne z odpowiednimi genami chromosomu  $pa$  tego genotypu. Karanie genotypów polega na porównaniu wartości genów  $g(l, i)$ ,  $l = 1, \dots, 5$ , chromosomu  $ca(i)$  z wartościami odpowiednich genów  $h(l, i)$  chromosomu  $pa(i)$ , dla wszystkich osobników realokowanych do genotypu. W przypadku braku zgodności (identyczności) wartości genów, suma kar dla genu  $l$  wzrasta o 1. Po obliczeniu kar dla poszczególnych genów genotypu, obliczana jest suma kar  $pet(l)$  dla poszczególnych genów  $l$  wszystkich genotypów według zależności (5).

$$pet(l) = \sum_{l=1}^5 \sum_i^G pes(ca(i, l)) \quad (5)$$

Po wykonaniu operacji krzyżowania:

- przeprowadzane jest ponownie nagradzanie lub karanie osobników,
- obliczane jest ponownie średnie niedostosowanie osobników  $pfi(j)$ , które obliczane jest wyłącznie dla osobników, gdzie wskaźnik przeprowadzonego krzyżowania  $re(j) = 1$ ,
- obliczane jest średnie niedostosowania genotypów  $gfi(i)$ ,  $i = 1, \dots, G$ , po krzyżowaniu,
- obliczane jest średnie niedostosowanie populacji  $pfi_{it}$  po wykonanej operacji krzyżowania,



- obliczone  $pf_{it}$  zostaje zapisane w bazie danych z typem operacji  $ty_{it} = "C"$ , gdzie "C" oznacza wykonanie operacji krzyżowania. W przypadku, gdy w  $it$ -ej iteracji uzyskane średnie niedostosowanie populacji  $pf_{it} < pfd$ , zostaje zapisana struktura osobników.

### Operacja mutacji

Operacja mutacji ma na celu zamianę wartości wylosowanego genu chromosomów  $cc(i)$  genotypów o najwyższej ilości kar. W opracowanym algorytmie przyjęto prawdopodobieństwo mutacji  $mp = 0,1$  oraz  $0,5$ . Oznacza to, że spośród genotypów o najwyższej ilości kar, odpowiednio średnio 10% lub 50% genów może ulec mutacji. W celu wykonania mutacji, algorytm:

- oblicza prawdopodobieństwo mutacji  $mp(l)$ ,  $l = 1, \dots, 5$  każdego z genów, na podstawie wymierzonych dla poszczególnych genów kar, według wzoru:

$$mp(l) = \frac{\sum_i^G pes(ca(i,l))}{pet(l)}, \quad \text{dla } l = 1, \dots, 5 \quad (6)$$

gdzie:

$pes(ca(i, l))$  -kary dla poszczególnych genów, \

$pet(l)$  -liczba wszystkich wymierzonych kar.

- oblicza dystrybuantę  $mq(l)$ , dla  $l = 1, \dots, 5$  dla poszczególnych genów,
- uruchamia mechanizm ruletki, generując liczbę pseudolosową  $rnd$  z zakresu  $[0,1]$ ,
- oznacza wylosowany gen do mutacji wskaźnikiem  $mg(l) = 1$ ,
- dokonuje zamiany wartości wylosowanego genu na przeciwną; operacja wykonywana jest dla określonej prawdopodobieństwem mutacji ilości genotypów, dla których liczba wymierzonych kar dla genu jest większa od zero. Przed zamianą genotypy porządkowane są malejąco, według ilości wymierzonych dla wylosowanego genu kar.

Po wykonaniu operacji mutacji ponownie:

- przeprowadzane jest nagradzanie lub karanie osobników,
- obliczane jest średnie niedostosowanie osobników,
- obliczane jest średnie niedostosowanie genotypów po wykonanej operacji mutacji, • obliczane jest średnie niedostosowanie populacji.

### Zapis wyników po mutacji

Obliczone średnie niedostosowanie populacji  $pf_{it}$  zostaje zapisane w bazie danych z typem operacji  $ty_{it} = "M"$ , gdzie "M" oznacza przeprowadzenie mutacji.

### Zakończenie działania algorytmu GAPAP

Po zakończeniu działania algorytmu, w interfejsie aplikacji wyświetlane są wyniki dla każdej przeprowadzonej iteracji zawierające:

- wartości średniego niedostosowania populacji odrębnie dla operacji krzyżowania i mutacji,
- ilość wykonanych realokacji osobników,
- procent wykonanych realokacji osobników.

Następne iteracje wykonywane są zgodnie z opisanym powyżej schematem. Algorytm genetyczny GAPAP wykonując kolejne kroki, przeszukuje przestrzeń możliwych rozwiązań dążąc poprzez zastosowanie operatorów genetycznych (krzyżowanie i mutacja) oraz heurystyki oceniającej (mechanizm nagradzania lub karania) do osiągnięcia jak najniższego niedostosowania populacji.

### 3. Eksperymenty z użyciem algorytmu GAPAP

W rozprawie przeprowadzono szereg eksperymentów komputerowych z użyciem opracowanego algorytmu genetycznego GAPAP, które miały na celu wykazanie, że algorytm prawidłowo realizuje postawione zadanie alokacji populacji osadzonych. Eksperymenty podzielono na dwie fazy:

- **eksperyment testowy**, który miał na celu wykazać, że GAPAP jest w stanie dokonać alokacji osobników w taki sposób, aby niedostosowanie populacji wynosiło zero dla odpowiednio dobranych danych,
- **eksperyment docelowy**, który dla określonej rzeczywistej populacji miał wykazać, że zmieniając odpowiednio parametry algorytmu, GAPAP jest w stanie osiągnąć niedostosowanie alokacji populacji znacząco niższe od niedostosowania początkowego.

#### 3.1. Założenia dla eksperymentu testowego

Zaprojektowany algorytm genetyczny GAPAP powinien, poprzez zastosowane operacje krzyżowania oraz mutacji, minimalizować funkcję celu, zmniejszając średnie niedostosowanie alokacji populacji osobników. W celu sprawdzenia poprawności działania algorytmu, przeprowadzono niezależne próby dla różnej liczby genotypów, przyjmując następujące założenia:

- wszyscy osobnicy są początkowo poprawnie alokowani - ich struktura jest w 100 % zgodna ze strukturą genotypów (wartości odpowiednich genów chromosomów  $pa$  i  $pc$  są identyczne z wartościami odpowiednich genów chromosomów  $ca$  i  $cc$ ). Początkowe średnie niedostosowanie populacji  $pdf$  wynosi 0 (niedostosowanie zerowe - nie są wymagane żadne realokacje osobników,
- następnie, osobnicy są ręcznie realokowani w taki sposób, aby alokacja miała niedostosowanie niezerowe, co oznacza żeby populacja osobników była w 100 % niezgodna z populacją genotypów (wartości odpowiednich genów chromosomów  $pa$  i  $pc$  nie są identyczne z wartościami odpowiednich genów chromosomów  $ca$  i  $cc$ , wówczas średnie niedostosowanie populacji  $pdf = 1$ ,

- następnie, uruchamiany jest algorytm genetyczny GAPAP. Dla każdego przebiegu algorytmu przyjęto 100 iteracji. W przypadku osiągnięcia niedostosowania zerowego, tzn. wartości wskaźnika  $pf_{it} = 0$ , praca algorytmu jest przerywana i prezentowane są wyniki.

**Celem eksperymentu** było wykazanie, że algorytm w czasie przebiegu dokona takiej realokacji osobników, że zostanie osiągnięte **niedostosowanie zerowe**. Łącznie przeprowadzono 24 próby, przyjmując różne prawdopodobieństwo krzyżowania, różną wartość nagrody lub kary oraz zmienną liczebność populacji osobników i genotypów. Ponadto włączano lub wyłączało operację mutacji dla określonych prób.

W każdej z przeprowadzonych w eksperymencie prób zadanie zostało wykonane. Wyniki wskazują, że:

- największy wpływ na osiągnięcie niedostosowania zerowego ma operacja krzyżowania,
- zastosowanie mechanizmu nagradzania lub karania osobników powoduje szybsze osiągnięcie niedostosowania zerowego.

### 3.2. Założenia dla eksperymentu docelowego

Eksperyment docelowy przewidywał wielokrotne uruchomienie algorytmu genetycznego GAPAP dla populacji osobników i genotypów (zbioru cel) odzwierciedlających populację typowego zakładu karnego na rzeczywistym przykładzie. Jako stałe parametry algorytmu przyjęto:

- rozmiar populacji osobników  $O = 360$ ,
- rozmiar populacji genotypów  $G = 110$ ,
- liczbę powtórzeń uruchomienia algorytmu dla przyjętych warunków początkowych, która wynosi 3.

Jako zmienne parametry algorytmu przyjęto:

- prawdopodobieństwo krzyżowania genotypu,  $cp(i) = \{0,0; 0,5\}$ ,
- prawdopodobieństwo mutacji genu,  $mp(l) = \{0,5; 0,1; 0,0\}$ ,
- wartość nagrody lub kary,  $pp = \{0,9; 0,5; 0,1\}$ ,
- liczba iteracji  $R = \{100, 1000, 5000\}$ .

W ramach przeprowadzonego eksperymentu docelowego, wykonano 4 eksperymenty, oznaczone odpowiednio symbolami GAPAP.1, GAPAP.2, GAPAP.3 oraz GAPAP.4. Przyjęta struktura populacji genotypów oraz osobników odpowiada strukturze pewnej rzeczywistej jednostki penitencjarnej.

### 3.3. Opis eksperymentu GAPAP.1

Eksperyment GAPAP.1 zakładał:

- **sześciokrotne** uruchomienie algorytmu dla prawdopodobieństwa krzyżowania  $cp(i) = 0,9$ ,  $i = 1, \dots, G$ , z podziałem na **trzy próby** dla wartości nagrody lub kary  $pp = 0,5$  oraz **trzy próby** dla wartości nagrody lub kary równej **0**, a następnie

- **sześciokrotne** uruchomienie algorytmu dla prawdopodobieństwa krzyżowania  $cp(i) = 0,5$  z podziałem na trzy próby dla wartości nagrody lub kary równej 0,5 oraz trzy próby dla wartości nagrody lub kary równej 0.

Przeprowadzenie eksperymentu GAPAP.1 pozwala na wysunięcie następujących wniosków:

- **we wszystkich** próbach eksperymentu algorytm genetyczny zrealizował zdefiniowane w rozprawie zadanie, **obniżając** średnie niedostosowanie populacji,
- w seriach eksperymentów, w których zastosowano mechanizm **karania lub nagradzania** genotypów oraz osobników osiągnięto zdecydowanie niższe średnie niedostosowanie populacji. Fakt ten potwierdza przyjęte założenie, że implementacja mechanizmu przypisującego nagrody lub kary będzie miała zasadniczy wpływ na wyniki,
- nie zaobserwowano znaczących różnic w uzyskanych wynikach dla wysokiego (równego 0,9) oraz niższego (równego 0,5) prawdopodobieństwa krzyżowania genotypów. W obu przypadkach uzyskane wyniki były zbliżone,

W tabeli 1 przedstawiono zestawienie najlepszych wyników dla wszystkich prób eksperymentu GAPAP.1.

Tabela 1. Podsumowanie wyników eksperymentu GAPAP 1

| <i>Próba</i>      | <i>pdf</i> | <i>pf<sub>ii</sub></i> | <i>pdf - pf<sub>ii</sub></i> |
|-------------------|------------|------------------------|------------------------------|
| ML.09.05.1        | 0,408227   | 0,040242               | 0,367985                     |
| <b>ML.09.05.2</b> |            | <b>0,027515</b>        | <b>0,380712</b>              |
| ML.09.05.3        |            | 0,030879               | 0,377348                     |
| ML.09.00.1        | 0,507877   | 0,397604               | 0,110273                     |
| ML.09.00.2        |            | 0,398150               | 0,109727                     |
| ML.09.00.3        |            | 0,401698               | 0,106179                     |
| ML.05.05.1        | 0,408227   | 0,038271               | 0,369956                     |
| ML.05.05.2        |            | 0,035000               | 0,373227                     |
| ML.05.05.3        |            | 0,027923               | 0,380304                     |
| ML.05.00.1        | 0,507877   | 0,397393               | 0,110484                     |
| ML.05.00.2        |            | 0,396000               | 0,111877                     |
| ML.05.00.3        |            | 0,40079                | 0,107087                     |

### 3.4. Opis eksperymentu GAPAP.2

Celem eksperymentu było uzyskanie wyników w warunkach, gdzie zarówno **mechanizm nagradzania i karania** oraz **operacja mutacji** są wyłączone. Oznacza

to, że algorytm w kolejnych iteracjach wykonywał tylko i wyłącznie operację krzyżowania, tworząc pary rodzicielskie w oparciu o wyliczoną dystrybucję dla genotypów, która nie została ustalona z uwzględnieniem preferencji (nagród i kar). Eksperyment zakładał przeprowadzenie 3 prób, oznaczonych MH.05.00.1, MH.05.00.2 oraz MH.05.00.3, dla następujących warunków początkowych:

- wyłączony mechanizm nagradzania lub karania (wartość  $pp = 0,0$ ),
- wyłączona operacja mutacji (tylko krzyżowanie),
- prawdopodobieństwo krzyżowania genotypu  $cp(i) = 0,5$ ,
- ilość iteracji  $R = 1000$ .

Eksperyment GAPAP.2 wykazał, że algorytm genetyczny **obniża** wartość średniego niedostosowania populacji. Wyłączenie mechanizmu nagradzania lub karania osobników i genotypów spowodowało, że uzyskane wyniki są gorsze od otrzymanych w eksperymentach z włączonym mechanizmem. W tabeli 2 przedstawiono wartości najniższego średniego niedostosowania populacji  $pf_{it}$  oraz różnicę w stosunku do początkowego średniego niedostosowania dla prób MH.05.00. Wartość średniego niedostosowania populacji  $pf_{it}$  obniżyła się w stosunku do pierwotnego o średnio 0,125847.

*Tabela 2. Podsumowanie eksperymentu GAPAP 2*

| <i>Próba</i>                           | <i>pdf</i> | <i>pf<sub>it</sub></i> | <i>pdf - pf<sub>it</sub></i> |
|----------------------------------------|------------|------------------------|------------------------------|
| MH.05.00.1                             | 0,507877   | 0,390544               | 0,117333                     |
| MH.05.00.2                             |            | 0,380243               | 0,127634                     |
| <b>MH.05.00.3</b>                      |            | <b>0,375303</b>        | <b>0,132574</b>              |
| Średnia poprawa <i>pf<sub>it</sub></i> |            |                        | 0,125847                     |

### 3.5. Opis eksperymentu GAPAP.3

Celem eksperymentu było uzyskanie wyników dla **większej** liczby prób (równej 10), przy założonej liczbie iteracji  $R = 100$  oraz przy wykorzystaniu mechanizmu nagradzania i karania wpływającego w mniejszym stopniu (poprzez ustawienie niskiej wartości  $pp$ ) na działanie algorytmu.

Eksperyment GAPAP.3 zakładał przeprowadzenie dziesięciu prób dla następujących warunków początkowych:

- włączony mechanizm nagradzania i karania (wartość  $pp = 0,1$ ),
- włączona operacja mutacji, gdzie prawdopodobieństwo mutacji  $mp(l) = 0,1$ ,
- prawdopodobieństwo krzyżowania genotypu  $cp(i) = 1$ ,
- ilość iteracji  $R = 100$ .

W tabeli 3 poniżej przedstawiono zestawienie najlepszych wyników uzyskanych w każdej próbie eksperymentu.

Tabela 3. Zestawienie wyników GAPAP 3

| <i>Próba</i> | <i>it</i> | <i>pf<sub>it</sub></i> |
|--------------|-----------|------------------------|
| ML.1.01.1    | 96        | 0,36618                |
| ML.1.01.2    | 34        | 0,391436               |
| ML.1.01.3    | 72        | 0,363682               |
| ML.1.01.4    | 69        | 0,389924               |
| ML.1.01.5    | 86        | 0,375528               |
| ML.1.01.6    | 97        | 0,342955               |
| ML.1.01.7    | 65        | 0,366758               |
| ML.1.01.8    | 34        | 0,388347               |
| ML.1.01.9    | 83        | 0,38327                |
| ML.1.01.10   | 27        | 0,388349               |

### 3.6. Opis eksperymentu GAPAP.4

W eksperymencie GAPAP.4 jako warunek początkowy przyjęto **wyższą** niż w GAPAP.1 oraz GAPAP.2 ilość iteracji (równą 5000) w celu dokonania porównania uzyskanych najlepszych wyników (najniższego niedostosowania  $pf_{it}$ ) z najlepszymi wynikami uzyskanymi w próbach ML.05.05.01, ML.05.05.02, ML.05.05.03 oraz MH.05.05.1, MH.05.05.2 i MH.05.05.3.

GAPAP.4 zakładał przeprowadzenie trzech prób dla następujących warunków początkowych:

- prawdopodobieństwo mutacji genu  $mp(i) = 0,10$ ,
- prawdopodobieństwo krzyżowania genotypu  $cp(i) = 0,50$ ,
- ilość iteracji  $it = 5000$ ,

oraz zmieniającej się wartości nagrody lub kary  $pp$  wynoszącej odpowiednio dla kolejnych prób 0,0, 0,1 oraz 0,5.

Tabela 4. Zestawienie najlepszych wyników eksperymentu GAPAP 4

| <i>Próba</i>              | <i>pdf</i> | <i>pf<sub>it</sub></i> | <i>pdf - pf<sub>it</sub></i> |
|---------------------------|------------|------------------------|------------------------------|
| IT.05.00.1                | 0,49230    | 0,401151               | 0,091149                     |
| IT.05.00.2                | 0,432423   | 0,336151               | 0,096272                     |
| IT.05.00.3                | 0,336939   | 0,110545               | 0,226394                     |
| Średnia poprawa $pf_{it}$ |            |                        | 0,137938                     |

Przeprowadzenie eksperymentu GAPAP.4 miało na celu sprawdzenie, czy efektywność algorytmu (poprzez obniżenie średniego niedostosowania populacji) zostanie zachowana przy większej ilości iteracji, wynoszącej 5000. Wyniki wskazują, że pomimo zwiększenia liczby iteracji nie udało się obniżyć znacząco średniego niedostosowania populacji. W tabeli 4 przedstawiono zestawienie najlepszych wyników

dla eksperymentu GAPAP.4. Z danych wynika, że algorytm uzyskał najlepsze rezultaty dla próby, w której przyjęto wartość nagrody lub kary  $pp = 0,5$ .

### 3.7. Wnioski z eksperymentów GAPAP

Przeprowadzenie 28 prób w 4 eksperymentach docelowych oraz 39 prób w eksperymencie testowym wykazało, że algorytm genetyczny GAPAP realizuje postawione w rozprawie zadanie tj. obniża średnie niedostosowanie populacji. Najlepsze wyniki uzyskano przy średnim (gdzie  $cp(i) = 0,5$ ) oraz dużym (gdzie  $cp(i) = 0,9$ ) prawdopodobieństwie krzyżowania, włączonej operacji mutacji oraz włączonym mechanizmie nagradzania lub karania genotypów i osobników (gdzie wartość nagrody lub kary  $pp = [0,5; 0,9]$ ). W takim przypadku algorytm znacznie obniża średnie niedostosowanie populacji  $pf_{ii}$ .

Algorytm dla przyjętej w eksperymencie GAPAP.4 większej liczby iteracji (5000) nie poprawił znacząco wyników. Wyłączenie operacji mutacji ma wpływ na wyniki; średnie niedostosowanie populacji jest w takim przypadku wyższe niż w przypadku zastosowania mutacji. Stały rozmiar populacji osobników i genotypów w eksperymentach może mieć wpływ na osiągnięte wyniki. W dalszych badaniach należy rozważyć przeprowadzenie

## **4. Wnioski końcowe**

Efektywność opracowanego algorytmu GAPAP powinna być mierzona osiągniętymi wynikami przeprowadzonych eksperymentów. W każdej z poszczególnych serii osiągnięto zamierzony cel, tzn. średnie niedostosowanie populacji było niższe a tym samym alokacja osobników lepsza i lepiej spełniająca istniejące reguły i ograniczenia. Wyniki jednakże jednoznacznie wskazują, że zastosowanie dodatkowej heurystyki oceniającej w postaci mechanizmu nagradzania genotypów i osobników za prawidłową alokację, ma bardzo duży wpływ na osiągnięty poziom niedostosowania populacji. Brak tej dodatkowej funkcji wzmacniającej lub osłabiającej niedostosowanie genotypów lub osobników powoduje, że średnie niedostosowanie populacji było w niewielkim stopniu niższe od początkowego. Na tak silną korelację niedostosowania z mechanizmem nagradzania i karania ma na pewno wpływ rodzaj wybranej metody selekcji - metody ruletki. Wyższe prawdopodobieństwo wylosowania silnie ukaranego genotypu jest w tej metodzie podstawowym kryterium wyboru do krzyżowania. Zwiększenie liczby iteracji do 5000 nie poprawiło wyników eksperymentu. Może to wynikać z faktu, że algorytm natrafia na minimum lokalne, wokół którego algorytm porusza się w kolejnych iteracjach. Przyjęte założenie zwiększania prawdopodobieństwa wylosowania genotypów o jak najwyższym niedostosowaniu powoduje, że liczba tych genotypów systematycznie się zmniejsza (genotypy o niskim niedostosowaniu mają przydzieloną wąską dystrybucję na kole ruletki). Kolejnym czynnikiem, który wpływa na efektywność algorytmu jest "sztywność" reguł i ograniczeń alokacji osadzonych oraz brak możliwości sterowania atrybutami osobników (w przyjętej strukturze osobników nie zachodzą zmiany atrybutów).

W kolejnych rozwinięciach algorytmu należy rozważyć:

- implementację innego mechanizmu mutacji genów, powiązanego z prawdopodobieństwem mutacji (w zaprojektowanym algorytmie wpływ operatora mutacji na osiągnięcie niższego niedostosowania populacji był zdecydowanie niższy niż krzyżowania),
- konieczność przeprowadzenia eksperymentów z bardzo dużymi i małymi populacjami (w rozprawie, rozmiar populacji w docelowym eksperymencie jest stały),
- lepsze sterowanie funkcją oceniającą poprzez wyeliminowanie nadmiernej zbieżności algorytmu,
- zastosowanie innej metody eliminacji rozwiązań niedopuszczalnych,
- wprowadzenie rozmytości do reguł i ograniczeń alokacji,
- zastosowanie innych metod selekcji -np. metody rankingu liniowego lub selekcji turniejowej może w znacznym stopniu wpłynąć na efektywność działania algorytmu, poprzez inne określanie prawdopodobieństwa wylusowania genotypów do krzyżowania i mutacji,
- możliwość zastosowania sieci neuronowych i mechanizmów uczenia do propagacji możliwych rozwiązań,
- zastosowanie logiki rozmytej do obliczania niedostosowania osobników i genotypów (rozmycie "sztywności" obliczeń -"osobnik na pewno jest niedostosowany" poprzez wprowadzenie pojęcia "osobnik być może jest niedostosowany"),
- próbę zbudowania ontologii, umożliwiającej odpowiednią klasyfikację genotypów w procesie selekcji,
- eliminację rozwiązań niedopuszczalnych opartą na innych (niż mechanizm nagrody i kary) funkcjach oceniających,
- zbudowanie systemu hybrydowego, zawierającego wszystkie wymienione powyżej funkcjonalności.

Wydaje się, że niezbędne jest ponadto uogólnienie postawionego problemu w taki sposób, aby zaproponowany model algorytmu genetycznego GAPAP mógł zostać wykorzystany do próby rozwiązań problemów o zbliżonej naturze optymalizacyjnej.

## Bibliografia

- Adeli H., Cheng N.T. (1994) Augmented lagrangian genetic algorithm for structural optimization, *Journal of Aerospace Engineering*, 7, 104-118.
- Antonisse J., Keller K. S. (1987) Genetic Operations on High -Level Knowledge Re-presentations. *Proceedings of the 2nd International Conference on Genetic Algorithms*. Cambridge, MA.
- Arabas J. (2001) Wykłady z algorytmów ewolucyjnych. Warszawa: WNT.
- Banachowski L., Rytter W., Diks K. (1999) *Algorytmy i struktury danych*. Warszawa: WNT.
- Banachowski L., Kreczmar A., Rytter W. (1999) *Analiza algorytmów i struktur danych*. Warszawa: WNT.
- Banzhaf W., Nordin P., Keller R.E., Francone F.D. (1998) *Genetic Programming: An Introduction*. San Mateo: Morgan Kaufmann.



- Baluja S. (1997) Genetic algorithms and explicit search statistics. W: Mozer M. C., Jordan M. L., and Petsche T., eds., *Advances in Neural Information Processing Systems*, 9, pp. 319-325, MIT Press, Cambridge, Massachusetts.
- Baum E., Boneh D., Garrett C. (1995) On genetic algorithms. W: *Proceedings of the Eighth Annual Conference on Computational Learning Theory (COLT-92)*, pp. 230-239, Santa Cruz, California, ACM Press.
- Barricelli N. A. (1957) Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(35-36), 143-182.
- Bartak R. (2006) *On line guide to Constraint Programming*, Pobrano z lokalizacji <http://kti.mff.cuni.cz/~bartak/constraints/>.
- Bullnheimer B., Hart R. F., Strauss Ch. (1997) *Applying the Ant System to the Vehicle Routing Problem*, 2nd Metaheuristics International Conference (MIC-97), Sophia-Antipolis France.
- Bielecki W. (2001) *Informatyzacja zarządzania*, Warszawa, PWE.
- Bowen J., Dozier G. (1995) Solving constraint satisfaction problems using a genetic systematic search hybrid that realizes when to quit. In: Eshelman L. J., ed., *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA 95)*, San Mateo, Morgan Kaufmann.
- Bubnicki Z. (1990) *Wstęp do systemów ekspertowych*, Warszawa, PWN.
- Cholewa W., Pedrycz W. (1987) *Systemy doradcze*, Skrypt Uczelniany Politechniki Śląskiej, (1447), Gliwice.
- Chromiec J., Strzemieczna E. (1994) *Sztuczna inteligencja - metody konstrukcji i analizy systemów eksperckich*, Warszawa, PWN.
- Cormen T. H., Leiserson Ch. E., Rivest R. L. (2002) *Wprowadzenie do algorytmiki*, Warszawa, WNT.
- Cytowski J. (1996) *Algorytmy genetyczne. Podstawy i zastosowania*. Warszawa, Akademicka Oficyna Wydawnicza PLJ.
- Januszewski A. (2001) *Informatyka w przedsiębiorstwie. Systemy i procesy informatyzacji*, Bydgoszcz, Wyższa Szkoła Zarządzania i Finansów.
- Joines J., Houck C. (1994) On the use of non-stationary penalty functions to solve non-linear constrained optimization problems with GAs, *Proceedings of the First IEEE International Conference on Evolutionary Computation*, *IEEE Press*, 579-584.
- Jong K. (1975) An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstract International*, 36(10), (5140B).
- Juels A, Wattenberg M. (1996) Stochastic hill climbing as a baseline method for evaluating genetic algorithms. W: Touretzky D.S., Mozer M.C. and Hasselmo M.E., eds., *Advances in Neural Information Processing Systems*, 8, pp. 430-6, MIT Press, Cambridge, Massachusetts.
- Kacprzak T., Ślot K. (1995) *Sieci neuronowe komórkowe*, Warszawa, PWN.
- Kacprzyk J. (1986). *Zbiory rozmyte w analizie systemowej*, Warszawa, PWN.
- Kacprzyński B. (1974) *Planowanie eksperymentów. Podstawy matematyczne*, Warszawa, WNT.
- Kanoh H. (1997) Solving constraint-satisfaction problems by a genetic algorithm adopting viral infection. *International Journal on Engineering Applications of Artificial Intelligence*, 10(6).
- Kazarlis S., Petridis V. (1998) *Varying fitness functions in genetic algorithms: studying the rate of increase in the dynamic penalty terms*, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, Berlin, Springer Verlag, 211-220.
- Kisielnicki J. (1993) *Informatyczna infrastruktura zarządzania*, Warszawa, PWN.
- Kodeks Karny Wykonawczy*, (Dz. U.z dnia 5 sierpnia 1997 r.), Portal Służby Więziennej. Pobrano z lokalizacji <http://www.sw.gov.pl/download/ustawy/1154694118.pdf>.

- Klein M., Methlie L.B. (1992) *Expert Systems. A Decision Support Approach with Applications in Management and Finance*, Addison-Wesley Publishing Company.
- Koza J. (1990) *Genetic Programming: Aparadigm for Genetically Breeding Population of Computer Programs to Solve Problems*, Stanford University.
- Koza J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, Massachusetts.
- Koza J.R. (1994) *Genetic Programming II-Automatic discovery of reusable programs*, MIT Press, Cambridge, Massachusetts.
- Koza J.R., Bennett F.H., Andre D. and Keane M.A. (1999). *Genetic Programming III-Darwinian invention and problem solving*, Morgan Kaufmann, San Mateo, California.
- Kumar V. (1992) Algorithms for Constraint Satisfaction Problem, A Survey, *AI Magazine*, 13(1), 32-44.
- Larranaga P., Kuijpers C., Murga R, Inza L, Dizdarevic S. (1999) Genetic algorithms for the travelling salesman problem: A review of representations and operators, *Artificial Intelligence Review*, 13, 129-170.
- Le Riche R., Knopf-Lenior C., Haftk R.T. (1995) *A Segregated genetic algorithm for constrained structural optimization*, Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 558-565.
- Mańczak K. (1976) *Technika planowania eksperymentu*, Warszawa, WNT.
- Mańdziuk J. (2000) *Sieci neuronowe typu Hopfielda, Teoria i przykłady zastosowań*, Warszawa, Akademicka Oficyna Wydawnicza EXIT.
- Marciszewski M. (2002) *Rachunek predykatów: słownik, składnia i semantyka*, Warszawa.
- Marciszewski W. (2002) *Wykłady Logika predykatów świat modułów i zbiór relacji*, Collegium Civitas.
- Marciszewski W. (1998) *Sztuczna inteligencja*, Kraków, ZNAK.
- Marriot K., Stuckey P. (1998) *Programming with Constraints: An introduction*, Cambridge, The MIT Press.
- Michalewicz Z., Attia N., *Evolutionary optimization of constrained problems*. Proceedings of the Third Annual Conference on Evolutionary Programming, World Scientific, 98-108.
- Michalewicz Z., Fogel D.B. (2000) *How to solve it: Modern Heuristics*, Berlin, Springer Verlag.
- Michalewicz Z., Janikow C.Z. (1993) *Handling constraints in genetic algorithms*. Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, 151-157.
- Michalewicz Z. (1996) *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Warszawa, WNT.
- Miller G., Todd P., Hedge S. (1989) *Designing neural networks using genetic algorithms*. Proc. Of the 3rd Int. Conference on Genetic Algorithms, San Mateo, Morgan Kaufmann.
- Miller G. (2000) Technological Evolution as Self-Fulfilling Prophecy, W: John Ziman, ed., *Technological Innovation as an Evolutionary Process*, Cambridge, Cambridge University Press.
- Mitchell M. (1996) *An Introduction to Genetic Algorithms*, Cambridge, MIT, Massachusetts.
- Mitchell M., Holland J.H. and Forrest S. (1996) When will a genetic algorithm outperform hill climbing? W: Cowan J., Tesauro G. and Alspector J., eds., *Advances in Neural Information Processing Systems*, 6, MIT Press, Cambridge, Massachusetts.
- Mirkowska G. (2003) *Matematyka dyskretna*, PJWSTK.
- Mockler R.J., Dologite D.G. (1992) *Knowledge-Based Systems. An Introduction to Expert Systems*, Macmillan Publishing Company.

- Monakata T. (1998) *Fundamentals of the New Artificial Intelligence*, New York: Springer Verlag.
- Mordechai B.A. (2005) *Logika matematyczna w informatyce*, Warszawa, WNT.
- Motwani R, Raghavan P. (1995) *Randomized Algorithms*, Cambridge University Press, New York (NY).
- Mulawka J. (1996) *Systemy ekspertowe*, Warszawa, WNT.
- Niederliński A. (2000) *Regulowe systemy ekspertowe*, Gliwice, Pracownia Komputerowa J. Skalmierskiego.
- O'Reilly U.-M., Oppacher F. (1994) *Program search with a hierarchical variable length representation*, Genetic programming, simulated annealing and hill climbing. W: Davidor Y., Schwefel H.P. and Manner R., eds., *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, pp. 397-406, Jerusalem, Israel, Springer-Verlag.
- Parsaye K, Chignel M. (1998) *Expert system for experts*, New York, John Wiley & Sons, Inc.
- Pawlak M. (1999) *Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji*, Warszawa, PWN.
- Polański Z. (1984) *Planowanie doświadczeń w technice*, Warszawa, PWN.
- Radościński E., Zabawa J. (2002) *Interface w hybrydowych inteligentnych systemach wspomaganie decyzji. Symulacja systemów gospodarczych*, Duszniki, 2002.
- Radzikowski W. (1994) *Badania operacyjne w zarządzaniu*, Warszawa, Wydawnictwo UW.
- Rawlins G.J.E. (1991) *Foundations of Genetic Algorithms*, Morgan Kaufmann.
- Rechenberg I. (1973) *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Stuttgart, Frommann-Holzboog.
- Ross K.A., Wright C.H. (1996) *Matematyka dyskretna*, Warszawa, PWN.
- Rutkowska D., Piliński M., Rutkowski L. (1999) *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, Warszawa, PWN.
- Seidler J., Badach A., Molisz W. (1982) *Metody rozwiązywania zadań optymalizacji*, Warszawa, WNT.
- Smith A., Tate D. (1993) Genetic optimization using a penalty function, *Proceedings of the fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, 499-503.
- Smith A.K., Coit D.W. (1997) Constraint handling techniques-penalty functions. *Handbook of Evolutionary Computation*, chapter C 5.2., Oxford University Press and Institute of Physics Publishing.
- Sprague R.H., Carlson K.D. (1982) *Building Effective Decision Support Systems*, Prentice Hall, Englewood Cliffs NJ.
- Statistics HomePage (2006) Pobrano z lokalizacji:  
[http://www.statsoft.pl/textbooklglosfra\\_stat.html?http%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fglosf.html](http://www.statsoft.pl/textbooklglosfra_stat.html?http%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fglosf.html).
- Statystyki\_SW (2006) Portal Służby Więziennej, Pobrano z lokalizacji  
<http://www.sw.gov.pl/images/1158583131.pdf>.
- Strzykowski S. (2003) Komputerowy system ewidencji skazanych i tymczasowo aresztowanych "NOE" jako przykład systemu ekspertowego. W: Hołubiec J., red., *Analiza systemowa w finansach i zarządzaniu, Wybrane problemy*, Tom 5, Warszawa, IBS PAN.
- Strzykowski S. (2004) Algorytm rozmieszczania osób pozbawienia wolności "Avalon" w komputerowym systemie ewidencji skazanych i tymczasowo aresztowanych "NOE". W: Hołubiec J. (red.) *Analiza systemowa w finansach, Wybrane problemy*, Tom 6, Warszawa, IBS PAN.
- Strzykowski S. (2005) Problem alokacji osadzonych: zastosowanie algorytmu genetycznego. W: Studziński J. (red.) *Zastosowania informatyki w nauce, technice i zarządzaniu*. Warszawa, IBS PAN.

- Strzykowski S. (2005) System tworzenia i aktualizacji dostępu do informacji o skazanych elementem wiedzy w zwalczaniu przestępczości. W: Drelichowski L., red., *Studia i Materiały Polskiego Stowarzyszenia Zarządzania Wiedzą*, Tom 4, Bydgoszcz.
- Strzykowski S. (2007) *Prisoner's Allocation Problem*, *Advanced Studies in Contemporary Mathematics* (submitted).
- Strzykowski S. (2007) On some Hybrid Genetic Algorithm, *Control and Cybernetics*, Warsaw.
- Szymanowski T., Świda Z. (1998) *Kodeks Karny Wykonawczy. Komentarz*, Warszawa, Wydawnictwo Liberata.
- Szymczak M. (1996) *Słownik języka polskiego*, Warszawa, PWN.
- Tadeusiewicz R. (1993) *Sieci neuronowe*, Warszawa, Akademicka Oficyna Wydawnicza RM.
- Thro H. (1994) *Sztuczne Życie*, Warszawa, SAMS Publishing.
- Traczyk W., red. (1995) *Problemy sztucznej inteligencji*, Warszawa, PWN.
- Vignaux G.A, Michalewicz Z. (1991) A Genetic Algorithm for the Linear Transportation Problem, *IEEE Transactions on Systems, Man and Cybernetics*, 21(2), 445-452.
- Whitley D. (no date) GENITOR: a different genetic algorithm. W: *Proceedings of the Rocky Mountain conference on Artificial Intelligence*, Denver, Colorado.
- Whitley D. (1989) The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. W: Schaffer J. D., red., *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, 116-121, San Mateo, Morgan Kaufman Publishers.
- Wieczorek K. (2005) *Wprowadzenie do logiki*, Warszawa, Skrypt.
- Wierchoń S. (2001) *Multimodal optimization with artificial immune systems*, *Intelligent Information Systems*.
- Winter P. E., Hickey G., Fletcher H.L. (2004) *Genetyka*, Warszawa, PWN.
- Yeniay G. (2005) Penalty function methods for constrained optimization with genetic algorithms, *Mathematical and Computational Applications*, 10(1).
- Zarządzenie Nr 19/98/ CZSW Ministra Sprawiedliwości z dnia 27 sierpnia 1998 r. Dz. Urz. Min. Sprawiedliwości, Nr 3, poz.13.

## **COMPUTER SYSTEM FOR SUPPORTING THE ALLOCATION OF INMATES IN A PENITENTIARY INSTITUTION**

**Abstract:** The paper describes the genetic algorithm GAPAP (Genetic Algorithm of Prisoner's Allocation Problem) developed for supporting allocation of prisoners in a penitentiary institution. The algorithm fulfils several legal requirements as well as constraints in order to avoid mismatching of prisoners within each cell. The computer implementation of the developed algorithm was tested on data of a real a penitentiary institution. The performed tests show significant improvement compared to actually used practices.

